JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Preface

We present in this issue a miniseries on ad hoc wireless networks. It has been inspired by the discrepancy between the amount of effort (be it academic research or industrial push) expended for the promotion of ad hoc wireless networking concepts and the minuscule degree of their materialization in our lives. As academics, we are used to the fact that only a small fraction of our work finds its way into applications. After all, the role of academic research is not only to immediately bring about tangible products, but also to enlarge the intellectual base of hypothetical solutions constituting the advanced framework for education and professional development of ourselves. In other words, we could not create even as little as we do in the way of real-life substance, if we did not contribute to the Platonic world of pure concepts.

Yet practical impact of research counts too, and in our own area, i.e., telecommunication, we often feel short-changed more than other disciplines. Especially when one looks at the multitude of algorithms and protocol improvements being published every month, accompanied by performance studies demonstrating their superiority over old solutions, the disappointment from the confrontation with reality must be painful. Shouldn't the industry people come along and do things "the right way"?

Wireless ad hoc networking is a particularly bitter example. This is because its practical side is virtually nonexistent, despite the apparent demand on the one hand (if only from the sensing industry), and the proliferation of ideas on the other. In the area of inexpensive sensor networks, where the commercial pull is especially strong, the solutions the industry has to offer fail to catch on, as if they are missing something important. As for wireless Internet, access points have stolen the show completely leaving no room for the ad hoc paradigm. How many wireless ad hoc networks have you seen in existence? Isn't this number in sharp contrast with the size of their bibliography?

The papers of our collection attempt to explain the practical failure of wireless ad hoc networks and point out ways towards improvement. Criticizing is always an easy task – however, in our opinion what makes all these papers worthwhile is their constructive collective message: they identify specific and solvable problems (Bouckaert *et al.*, Kosek *et al.*), find interesting niches for ad hoc networking in areas dominated by access points (Gierłowski *et al.*), and demonstrate comprehensive cost-effective solutions verified by commercial deployments (Gburzyński and Olesiński). One paper (Malhotra *et al.*) suggests novel sensing applications where wireless ad hoc meshes may be truly essential, thus providing a rationale to break the monopoly of fixed-infrastructure systems.

We would like to thank all the authors for their valuable contributions to this unique set, especially that, as it usually happens, the notice was short and the timing could have been friendlier. We very much appreciate their dedication and promptness in delivering the manuscripts, which made our work so much easier.

Paweł Gburzyński
Józef Woźniak
Guest Editors

Paper

Making ad hoc networking a reality: problems and solutions

Stefan Bouckaert, Dries Naudts, Ingrid Moerman, and Piet Demeester

Abstract—Despite the fact that many electronic devices are equipped with wireless interfaces and numerous publications on wireless ad hoc and mesh networking exist, these networks are seldom used in everyday life. A possible explanation is the fact that only few of the numerous theoretically promising proposals lead to practical solutions on real systems. Currently, wireless network design is mostly approached from a purely theoretical angle. In this paper, common theoretical assumptions are challenged and disproven, and key problems that are faced when putting theory to practice are determined by experiment. We show how these problems can be mitigated, and motivate why a heterogeneous hierarchical wireless mesh architecture, and a multidisciplinary research approach can help in making wireless ad hoc networking a reality.

Keywords*— *wireless ad hoc, wireless mesh, experiments, interference, implementation, hierarchical heterogeneous architecture.

1. Introduction

Ad hoc networks have been the subject of international research for over thirty years [1]. Since the initial work on the packet radio network (PRNet) in 1972 [2], computer networks have evolved from small-scale initiatives connecting a few geographically separated sites, into a worldwide broadband communication network. Research interest in wireless packet radio networks initially came from the military. Since the mid-1980s, lots of civil applications for wireless ad hoc networks have been studied, such as emergency communication for public services or communication in disaster areas. In the late 1990s, wireless enabled hardware became cheap and omnipresent, and with the foundation of the MANET (mobile ad hoc networks) Working Group [3], the Internet Engineering Task Force (IETF) started standardization efforts for routing protocols supporting mobile wireless networks.

Countless publications exist covering numerous research topics such as wireless ad hoc, mesh, or sensor networks, studying aspects at all layers of the open system interconnection (OSI) stack. In spite of all these research efforts, wireless ad hoc networks are rarely used in everyday life. How can this contrast be explained, even though lots of scenarios exist that could benefit from ad hoc technology?

In [4], the authors answer this question by suggesting that most ad hoc network design focuses on military or specialized civilian applications, making the solutions impractical for everyday life. This is an important observation, however, this paper addresses other, perhaps more fundamental problems. We feel that a lot of research gets stuck in a crucial phase of development: while there are a massive number of initiatives to design wireless ad hoc solutions, few ideas are implemented on actual systems. Unfortunately, as promising as some ideas may be, they do not always lead to good or practical solutions. Using a purely theoretical top-down approach where implementation is the last step in designing wireless network protocols, architectural decisions are often made which, after months of research, turn out to be impossible to realize because of unforeseen implementation problems.

Wireless research focused on single-interface homogeneous ad hoc networks for a long time. Recently, an evolution in wireless networking research is observed, where researchers start focusing on multiple interface nodes in mesh topologies. Additionally, several aspects of cross-layer research, such as power control, are gaining popularity. While there is a growing awareness within the research community that simulation of protocols might not be sufficient in order to validate the stable operation of wireless networking protocols [5, 6], a lot of assumptions are still made while studying old and new topics in wireless research.

In Section 2 of this paper, we will verify the validity of common assumptions by experiments, and formulate lessons learned during the evaluation of several experimental set-ups and real life implementations using IEEE 802.11 hardware. While some findings may be trivial to people who are familiar with the physical layer of wireless networks, we believe that it is important to show wireless protocol designers the incorrectness of many assumptions, as they can be a major contributor to the slow adoption of ad hoc technology. We are aware of the fact that some problems are vendor specific and that, strictly spoken, it is not the task of people performing research at the higher layers of the OSI stack to actually solve hardware problems. However, we feel that a careful choice of research methodology and network architecture can severely reduce the observed problems. In Section 3, we discuss how heterogeneous hierarchical architectures can help the successful realization of ad hoc networks, and how wireless protocol research can benefit from a close cooperation between research groups working at the physical layer and research groups working at higher network layers.

2. Observations and experiments

When collecting data from test set-ups using any wireless driver, one should always keep in mind that the driver could be causing errors at a node. We have tried, to the extent possible, to exclude driver-caused errors from the observations below. The observations follow from several real-life experiments and test-beds, using a broad range of hardware. A first test-bed consists of 18 Linux nodes, equipped with D-Link AG520 wireless a/b/g peripherial component interconnect (PCI) cards and external antennas. In addition, a lot of research on mesh networks at our research lab is done using modified 4G meshcubes with up to 4 wireless mini-PCI a/b/g cards per node. Other tests are performed using Linksys WRT54GL wireless routers with modified firmware. All devices can be powered using batteries, allowing to test real mobility.

2.1. Single frequency, single hop networking

Assumption. *In an isolated situation, an IEEE 802.11 wireless link between two nodes will be of better quality if transmission power is increased.*

Observation. Consider a test set-up from Fig. 1, where three identical network nodes, each equipped with a single wireless network interface are stacked on top of each other in a rack. The external antennas are positioned in a triangle, the antennas separated about 1.5 m. Although at some times data can be sent at the theoretical speed limit, results are very unpredictable. Even if a link seems to be stable for a certain period of time, data rates can drop below one third of the stable rate, seemingly without a reason. In addition, links are not always symmetrical: changing the direction of the traffic flow can result in degraded throughput.

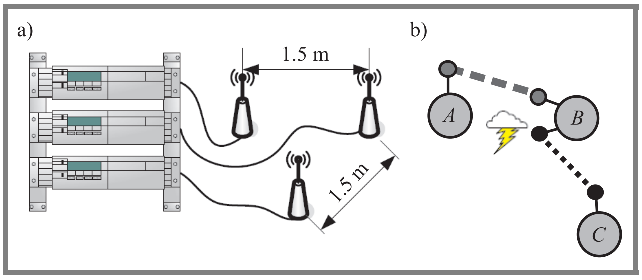


Fig. 1. (a) Test set-up using rack mount devices; (b) simple two-hop test. Node *B* has two wireless interfaces.

In this specific setup, one solution seems to solve most of these problems. *Reducing* the transmit power of the wireless cards results in highly increased stability. In this case, a transmit power of 10 mW gives the best results.

Experiment. When putting single interface meshcubes to test in an RF-shielded box [7], the same observation is made in a controlled environment: reducing power when sender and transmitter are at close distances, increases networking quality.

Placing the antennas relatively close to each other, as in this case, might seem artificial – there are however several situations imaginable where two single interface nodes are placed at comparable distances: e.g., a user can carry a wireless-enabled personal digital assistant (PDA) and a laptop, or, during meetings, there is a high laptop density.

2.2. Multiple frequencies, multihop networking

Assumption. *When configuring the different wireless interfaces of a multi-interface integrated node to theoretically non-overlapping channels, these different links will not interfere. Capacity of a wireless network can be increased dramatically by adding multiple interfaces.*

Observation. Reading through literature, lots of innovative solutions involving wireless networking can be found, and numerous protocols are designed to support systems with multiple network interfaces [8, 9]. Many of these solutions are based on the assumption that multiple theoretically non-interfering channels can be operated simultaneously at a node's different interfaces. This seems obvious when considering theory and simulations. Unfortunately, when these solutions are deployed in a test-bed, it turns out that this assumption is not necessarily true.

When a two-hop path is created using a traditional single-interface approach, the wireless medium has to be shared between two links. The maximum reachable throughput of the total path thus roughly halves because of a single additional hop. Adding a second wireless interface to the middle node and choosing orthogonal frequencies for the first and second link (cf. Fig. 1b) solves this problem, in theory, at the cost of adding a single extra interface. In practice, different results can be observed: even when two interfaces are available at the middle node and a two-hop path is constructed using two theoretically non-overlapping frequencies, the throughput does not rise considerably.

Still, in theory, devices supporting the IEEE 802.11b and IEEE 802.11g (802.11a) standard should be able to use, depending on the region, three (twelve) fully non-overlapping frequencies. Recently, other researchers described the same effects and concluded that wireless nodes with multiple interfaces can suffer severely from self-generated interference between the different interfaces [10, 11].

Experiment. Measurements done at our lab (cf. Fig. 2) show that these effects of self-generated interference can be severely reduced by limiting the transmit power used at the different interfaces and physically separating the antennas of a node. On the other hand, these results also reveal the sad truth that – at least when using off the shelf hardware – a single wireless interface using high transmit power can severely degrade the performance of the other interfaces and surrounding nodes, even when they are set to operate on non-overlapping channels.

When using integrated IEEE 802.11a devices such as the meshcubes in a two-hop test, interference problems are still observed, even when lowering transmit power. The first and second link can be set up separately on orthogonal channels with a goodput of over 30 Mbit/s. However, if both links are used at the same time in a multihop configuration, the goodput drops to about 16 Mbit/s even though the processor of the middle node can handle the packet forwarding.

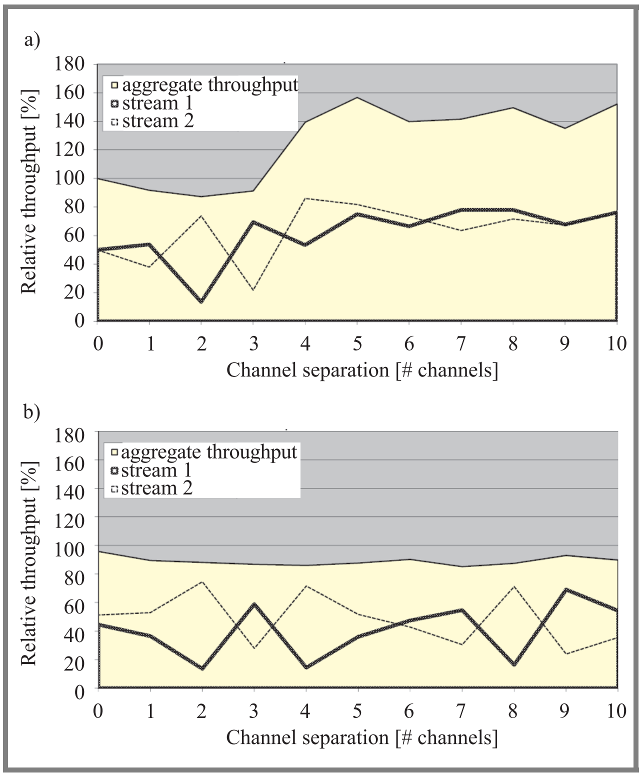


Fig. 2. Throughput measurement. Using two distant groups of two, 1.5 m separated, ethernet-linked WRT54GL routers – representing two nodes with two interfaces each, two connections are set up in parallel. The *Y* axis shows the throughput, relative to the maximum throughput of a single, non-interfered stream. Output power (a) 1 mW and (b) 100 mW.

We have found that in this case, the problem is essentially due to the fact that the meshcubes are densely integrated: it's not only the fact that two mini-PCI cards are located right on top of each other that causes problems, but especially the fact that the distance between the antennas is too small. This is not surprising: the antennas in the integrated devices are located very close to each other. At a frequency of 5 GHz, the wavelength used for transmission is about 6 cm, thus in our integrated devices, up to 4 antennas are separated by less than a single wavelength, resulting in a very unpredictable system. Increasing the distance between two antennas to about 15 cm alleviates this intra-node interference and goodput rises considerably. In [10], the authors conclude that these interference effects can be solved by providing an antenna separation of 35 dB. However, this separation is hard or even impossible to obtain in mobile devices with a small form factor.

Figure 3 shows additional measurements using WRT54GL routers at a transmission power of 1 mW, quantifying the effect of antenna distance on interference between two non-overlapping IEEE 802.11g channels: channel 1 and 11. In scenario of Fig. 3a, two separate user datagram protocol (UDP) streams (starting at 30 Mbit/s, decreasing the bit rate until packet loss is minimal) with a UDP packet size of 1470 bytes are set up, sequential at first, then simultaneous. The graphs show the resultant sum of the average throughput observed at the receiving interfaces. The individual throughput is not shown on the graph, as bandwidth is equally divided between the two flows. The figure shows that the sum of throughputs is reduced by 16.8% when the flows are set up simultaneously with a distance *d* of 1 m between the antennas. At a distance of 5 cm, throughput is reduced by over 45% compared to non simultaneous transmission. The sum of throughputs reduces both in the sequential and parallel tests when the antenna distance is decreased.

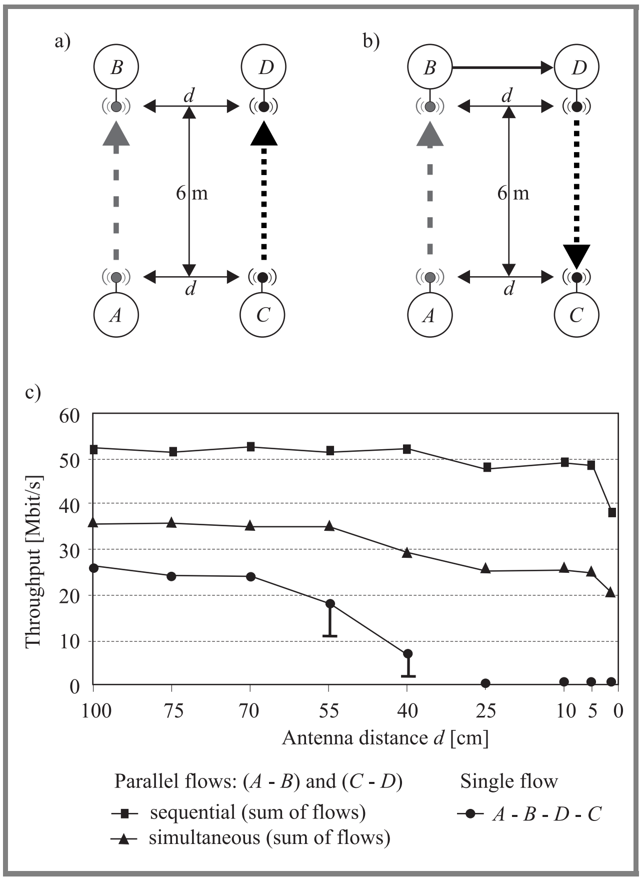


Fig. 3. Two scenarios quantifying the effect of the antenna distance on throughput: (a) parallel flows; (b) single flow. Link *A-B* operates on channel 1, link *C-D* on channel 11; (c) measurements at $d = 1$ cm, 5 cm, $10\text{ cm} + k \cdot 15$ cm, $k = 0 \ldots 6$. Transmission power = 1 mW.

In scenario of Fig. 3b, a single UDP stream with the same characteristics is set up. The test packets are now transferred over wire between interface *B* and *D*, recreating a typical multihop situation where every receiving interface has a sending interface nearby. Looking at the graph with the triangles, a first observation is that the end-to-end throughput is higher than half the aggregate throughput from scenario of Fig. 3a, for antenna distances *d* larger than 55 cm: the single flow is transferred more efficiently than two traffic flows originating separately. However, performance drops faster with the decrease of antenna distance. It is also observed that unlike in the previous situation, throughput is very unstable when the antennas are moved more closely to each other: moving the antennas slightly can result in a serious performance drop or increase. For this reason, the graph with the triangles shows the maximum values that were obtained from a large amounts of tests. Minimum throughput results are shown with error bars. When the distance between antennas drops below about 40 cm, communication at reasonable throughputs is no longer possible. It is clear that the consequences of interference between neighboring interfaces with nearby antennas are worse in scenario of Fig. 3b than in scenario of Fig. 3a. The receiving interfaces are disturbed by a signal of the nearby transmitter, even though it is sending on a non-overlapping channel, rendering successful reception of the UDP test packets impossible. Note that in both situations, IEEE 802.11 acknowledgement (ACK) messages are sent in the opposite direction of the UDP flows. However, these small packets suffer less from the interference of neighboring interfaces.

This observation raises questions about using multiple interfaces in integrated devices: even if perfect algorithms for using multiple interfaces on devices can be thought of and simulated, it is very likely that the result will not be as expected when deploying them in real integrated systems. We also believe that a test-bed with "full size" computers and wireless PCI cards with external antennas, can not fully represent an integrated end-user device with multiple interfaces.

2.3. Power adaptation

From previous paragraphs, we have learned, that changing power levels can lead to a more reliable link, but increasing transmit power does not necessarily increase communication quality. Furthermore, it was shown that theoretically non-interfering channels will interfere when using off-the-shelf hardware at mid to high transmit power, and that integrated systems with multiple interfaces will suffer from self-generated interference if there is no adequate antenna separation. Consequently, a single device transmitting at a relatively high output power may render all surrounding communication virtually impossible.

Lowering transmission power is often considered as a measure of freedom, in order to decrease interference and thus increase the number of possible simultaneous transmissions in a certain area, or to increase the lifetime of battery powered devices [12]. Our experience shows that when using today's IEEE 802.11 hardware as a base for multi-interface nodes, power adaptation is not a measure of freedom but rather a necessity in order to guarantee network operation. End-user hardware can (and probably will) improve in quality over time, however, we predict that it is very unlikely that in the near future palm-size systems will be able to fully take advantage of using multiple interfaces at relatively high output powers, if the interfaces are tuned to neighboring channels. As frequency regulations only allow a small part of the spectrum to be used for unlicensed civilian WLAN communication, it is not easy to provide the required channel separation. When using only two interfaces, putting one interface in the 2.4 GHz range and the other one in the 5 GHz might turn out effective with some types of hardware, but it is a mere palliative as this solution can currently not be scaled.

2.4. Hardware issues

Assumption. *If an algorithm works on the IEEE 802.11 hardware of vendor A, it will work on the IEEE 802.11 hardware of vendor B.*

A recurring problem faced during tests with hardware from different vendors, is that changing an interface to some specific channels can result in a wireless link of bad quality. As communicating on those channels is possible using hardware from a different vendor, and bad channels stay bad when replacing hardware with an identical spare, the problem is most likely hardware related.

In general, there is a big difference in stability and performance between hardware from different vendors. In Fig. 4,

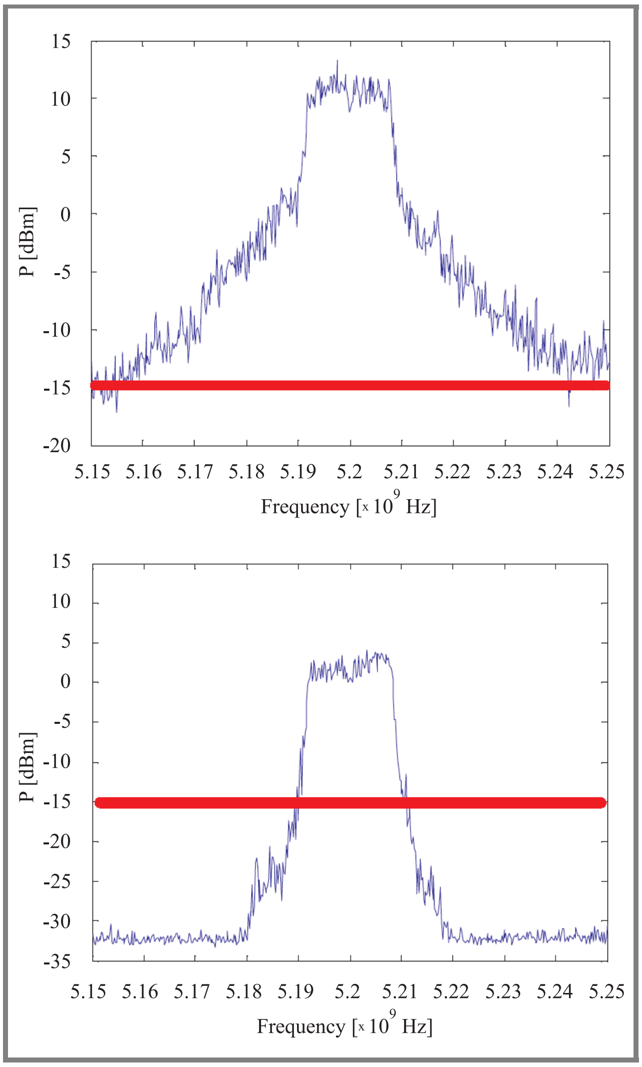


Fig. 4. Spectrum measurements of two different IEEE 802.11a mini-PCI cards operating at same transmission power (15 dBm), both using channel 40.

the spectrum of two mini-PCI cards from different vendors, operated at the same power level and frequency are shown. The figure clearly shows that the second card has a better spectral purity than the first card. Consequently, when building a test-bed with hardware of the first type, test results will be much more pessimistic than test results with hardware from the second vendor, especially when operating at multiple theoretically non-interfering frequencies.

In another test set-up, the maximum throughput of a single non-interfered UDP stream between two identical retail IEEE 802.11 mini-PCI network interfaces, installed at two identical network nodes was measured. The network nodes were separated by 20 m, with a wall in the middle. The same measurements were repeated several times, each time replacing the mini-PCI adapters with hardware from a different vendor. During all tests, the multiband Atheros driver for wifi [13] was used. The performance difference between cards of different vendors can clearly be seen from Fig. 5. Note that the adapters are all about equally expensive.

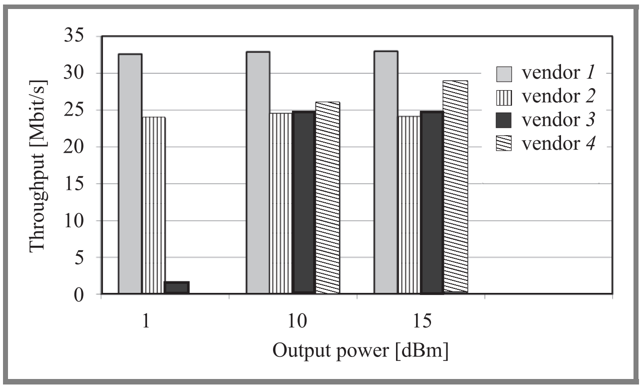


Fig. 5. Throughput measurements on a single unidirectional non-interfered IEEE 802.11a link, using hardware from different vendors, at three discrete power settings.

Although these problems can be solved by replacing faulty hardware with hardware from a different vendor, ad hoc networking protocols will only become successful if a large group of end users can use them instantly without problems, regardless of their choice of vendor. Unfortunately, in a traditional ad hoc network where nodes can join freely, it is wrong to assume that all nodes will react identically to a specific algorithm's action. For example, if a certain algorithm would reduce transmission power of a node to 1 dBm, the algorithm might operate as expected on hardware from vendor *1* or *2*, but fail to work on hardware from vendor *3* or *4*. If the quality of the hardware cannot be guaranteed, control loops should be provided within algorithms to verify whether a certain action has the desired effect. These effects are very hard to model in a simulator and can only be discovered by putting algorithms to test on real-life test-beds.

2.5. A lab environment is not a real environment

Assumption*. If it works in simulation, it will work on a test-bed. If it works on a test-bed, it will work in real life.*

From previous paragraphs, it is clear that designing algorithms and protocols for wireless systems should preferably not be done solely by considering theory or simulations. Creating a wireless test environment in a laboratory is not an easy task. Not only does it require a lot of – sometimes costly – hardware and space, it is also time consuming. On the other hand, creating these test environments and implementing developed algorithms on actual hardware forces the researcher to develop a system close to reality.

However, there is always a risk that the testing environment itself will lose its value as "real test case", as over time wireless systems could be tuned – unintentionally – to work great in the testing environment only. This way, a solution that evaluates positive in a testing environment can at the same time be useless when deployed in an uncontrolled real-life situation. This is a frustrating experience that was witnessed before at our lab: a demonstrator to transmit video over a self-forming and self-recovering multihop mesh/relay network was developed. After the demonstrator had proved to be working perfectly in the lab environment – even when moving the battery fed relay stations through the building – it was taken to a large hangar. Surprisingly, even with relatively short distances between the relaying hardware, and with line of sight communication, link breaks occurred frequently and maximum throughput was low. In this case, the set-up probably suffered from the absence of the waveguide effect described in [14]: there are circumstances where a wireless signal does not degrade as fast when using devices indoor, compared to using them in an open space.

This example, amongst others, shows that a system should not be declared stable based on a single test environment, and certainly not based on simulations. We believe that wireless ad hoc networking protocols and systems will only be used in everyday life if their use is not limited to a specific scenario or environment. However, today, a lot of algorithms are evaluated only in simulators using a very specific test scenario and very simplified propagation models, which are not valid in real-life environment, in particular indoor environments.

3. Solving issues

In the previous section, it was shown how hardware issues hamper successful realization of ad hoc networks. Subsection 3.1 points out which architectural choices can help to reduce the observed hardware problems, and, more specifically, explains why a heterogeneous hierarchical architecture avoids several of the described problems. Finally, Subsection 3.2 discusses how the inheritance of layered network design still impedes the development of algorithms for wireless networks today, and how a multidisciplinary research approach can help in developing more robust solutions.

3.1. A heterogeneous hierarchical architecture

Section 2 listed many problems that can be observed while bringing wireless ad hoc and mesh networking algorithms to real systems. Some of these problems are vendor related and can be solved by replacing defective hardware. One might argue that it is not the task of networking algorithms to account for these problems. However, it is inherent to the nature of wireless ad hoc networks that low-quality nodes will sooner or later join the network. Wireless systems will always be more unreliable than their wired counterparts, and therefore, algorithms must be able to detect anomalies and react accordingly.

Firstly, because of interference and hardware related issues, the choice for a specific channel has an impact on the wireless link quality. Problems occur at various layers of the protocol stack when wireless links break due to changing channel conditions or failing hardware.

Secondly, transmission power should be chosen wisely: neither too low nor too high. While in a static set-up, transmission power can be set manually by trial and error, there is need for automatic tuning in dynamic environments. Cross-layer protocols might provide a way to implement these control loops.

Thirdly, it was shown that small devices with multiple interfaces suffer from self-generated interference. In order to overcome this problem we should focus our research on an architecture which takes this fact into consideration. An algorithm which presupposes a complete separation between multiple interfaces at end-user nodes will most likely never be able to achieve its claimed results when used in real systems.

In a heterogeneous architecture, devices have distinct capabilities and technologies. In a hierarchical architecture, different nodes can belong to different logical groups, for example, backbone nodes and clients. Heterogeneous hierarchical architectures (Fig. 6) have been described in the past, however, we believe that their true potential has not been discovered yet. In [15], the authors describe a (hybrid) wireless mesh architecture. In a wireless mesh network (WMN), two types of nodes are distinguished: mesh routers and mesh clients. Mesh routers hold superior properties concerning processing power, interfaces, available power and memory, enabling them to perform more complex functions. In addition, they have limited mobility compared to the clients, resulting in a wireless mesh backbone. Mesh routers can be added or removed at any time and act as gateways to other networks such as the Internet. In a *hybrid* WMN, mesh clients can connect to the backbone network either directly, or by using a multi hop path through other clients.

Some benefits of heterogeneous hierarchical networks have been described in the past, such as an increase in coverage, or the (theoretical) ease of set-up. However, we believe that there are more reasons why hierarchical heterogeneous architectures can help to realize robust wireless networks, and that a conscientious choice of networking architecture can help certain assumptions that are invalid for homogeneous wireless networks become valid.

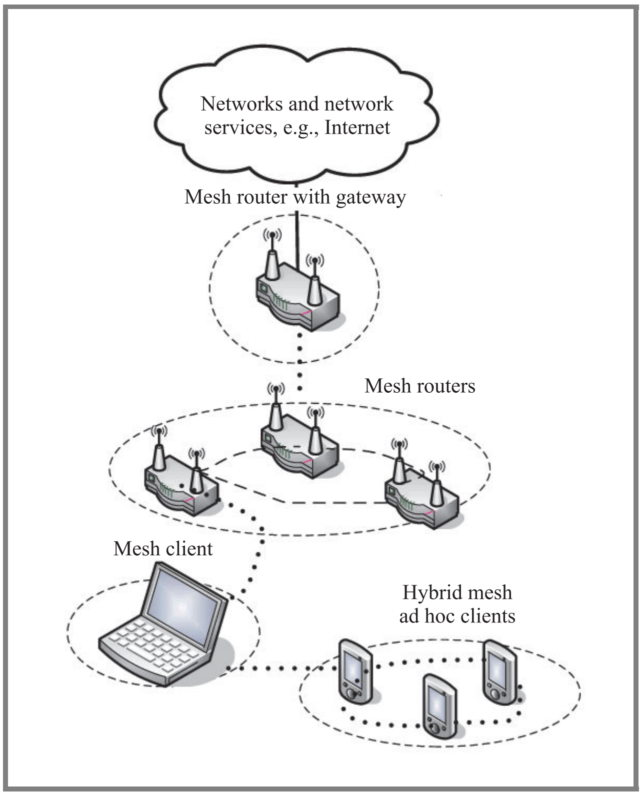

Networks and network services, e.g., Internet
Mesh router with gateway
Mesh routers
Mesh client
Hybrid mesh ad hoc clients

Fig. 6. Heterogeneous hierarchical mesh network, i.e., a hybrid wireless mesh ad hoc network. Clients connect either directly or through another client to a mesh backbone.

Most small and mobile end user devices such as PDAs or smartphones will probably only have a single (high speed) wireless network interface, using the unlicensed bands, enabled at a time, as adding interfaces is suboptimal due to the described interference problems and other limitations such as power and cost. On the other hand, the mesh routers in the backbone can and should have multiple interfaces: they can be bigger in size and antennas can adequately be separated, thereby reducing the interference problems. Additionally, they have an "unlimited" power supply as they are most likely connected to a host system with plenty of power such as a building or a truck.

Faulty hardware may be used within a cooperative wireless network, resulting in decreased performance and satisfaction for the end-users. In a traditional ad hoc network, even if one user invests in high-quality hardware, he can still experience bad performance if the person he is connecting through uses faulty hardware. In a heterogeneous architecture, end users can, e.g., connect to a mesh backbone which is constructed with hardware of better quality. The nodes which are higher in the hierarchy can be more expensive, as less nodes of higher hierarchy are needed.

In a hierarchical network architecture, operators can invest in high quality wireless backbone nodes. In addition to increasing the number of interfaces, more expensive network nodes could also use alternative technologies such as WiMAX. By using more interfaces or better technologies at higher hierarchical layers, wireless networks become more scalable, as throughput and transmission range can increase with every hierarchical level. Ideally, the technologies which are used at a higher layer can operate in other (licensed) frequency ranges, reducing the interference even more.

Changing protocols or interface configuration on all end user devices, constructed by different vendors, is harder to achieve than making changes to a smaller group of devices at a higher hierarchical layer, all the more since it is more likely that wireless devices at a higher hierarchical layer are controlled by a single administrator. For example, multi-interface wireless IEEE 802.11a backbone routers with (proprietary) cross-layer optimizations can easily provide wireless coverage within a building, or at a fair or festival. By adding an extra wireless interface to every backbone router, configured as an IEEE 802.11g access point, end users are able to connect to this backbone with hardware from any vendor, without compromising the quality of the backbone network. If a user has the right hardware and chooses to function as a relaying node and extend the network, he can do this by voluntarily installing the required protocols.

3.2. The need for cooperation

For years, researchers strictly followed a layered approach when designing networking protocols. When adopting a layered network design, different network layers can be optimized separately, and researchers optimizing a certain layer do not need to know the implementation details or exact operation of the adjoining layers.

As a logical consequence of a layered network design, most network research groups historically specialized in either (one of the) upper layers of the network stack, or in the physical layer. The same can be said about wireless research groups in particular, where people developing upper layer protocols, mostly have to rely on simulators to model the behavior of the physical layer. In order to decrease complexity and to ensure a reasonable simulation time, the physical models of these network simulators typically make abstraction of several complex electromagnetic phenomena.

In fixed networks, the layered approach has most certainly contributed to the Internet as we know it today. Inspired by this success, traditional MANET protocol research also followed the layered paradigm. More recently, driven by the continuous search for increased reliability and performance, several authors pointed out how wireless networks can benefit from exchanging parameters between different network layers [16, 17], and researchers started exploring the use of multi-interface nodes and modified physical layers.

The conceptual exploration of new wireless research fields by people originally involved in higher layers only has accelerated so fast, that it is sometimes forgotten that the physical layer models of most of the popular network simulators, such as ns-2 [18], were never designed to model the complex effects that come with these new research topics. For example, the same network simulators are now used to, among other things, simulate the use of multi-interface network nodes or cross-layer parameter exchange. Even if research groups working at the physical layer are aware of the extremely complex interference behavior of multiple antennas placed in each others vicinity, this wisdom seldom makes it to people designing upper layer protocols and architectures, because of the historical separation between research groups.

In our view, and from our experience, a lot of problems which are faced when implementing ad hoc and mesh network protocols cannot be explained solely by studying the upper network layers. The historical layered approach, allowed physical layer and upper layer research groups to work separately. However, as the border between physical layer and upper layers becomes faint at a conceptual level, the need to start or tighten interdisciplinary cooperation between lower and upper layer research groups has never been higher.

4. Conclusion

In this paper, assumptions that are commonly made when researching wireless ad hoc networking protocols were challenged. It was shown that, whether installing a single interface or multiple wireless interfaces at a node, real-life performance is always worse than can be expected from theoretical models or simulations. We raised questions about the usefulness of embedding multiple interfaces of the same type in a palm-size device, and argued that, in contrast to what is believed in many research papers, adjusting transmission power is not a measure of freedom but a necessity. We described how a choice of hardware affects the efficiency of algorithms, and how this influences the stability of wireless networks.

In order to test the robustness of algorithms, testing on one or multiple test-beds is a necessity. However, one must keep in mind that positive test-bed results do not always imply a stable system under all circumstances. Next, we argued that a heterogeneous hierarchical wireless mesh network architecture can help solving the observed problems, by reducing the need for miniaturization and providing incentives for network operators and businesses. Finally, we argued that an effective cooperation between research groups working at physical layer and working at higher protocol layers might help in avoiding many of the described problems. We believe that, if protocols are developed closer to reality, and more realistic architectural choices are made at the start of a design process, the usability of ad hoc and mesh networks can drastically be improved.

Acknowledgements

Stefan Bouckaert would like to thank the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) for funding this research through a grant. This research is also partly funded through the IBBT-Cisco Eye-Sense project.

References

[1] C. E. Perkins, *Ad Hoc Networking.* Boston: Addison-Wesley Longman, 2001.

[2] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet ratio technology", *Proc. IEEE*, vol. 66, no. 11, pp. 1468–1496, 1978.

[3] MANET Working Group, http://www.ietf.org/html.charters/manet-charter.html

[4] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks", *IEEE Commun. Mag.*, pp. 123–131, March 2005.

[5] C. F. Tschudin, P. Gunningberg, H. Lundgren, and E. Nordström, "Lessons from experimental manet research", *Ad Hoc Netw.*, vol. 3, no. 2, pp. 221–233, 2005.

[6] P. De, A. Raniwala, S. Sharma, and T. Chiueh, "MiNT: a miniaturized network testbed for mobile wireless research", in *INFOCOM 2005, 24th Ann. Joint Conf. IEEE ComSoc., Proc. IEEE*, Miami, USA, 2005, vol. 4, pp. 2731–2742.

[7] Qosmotec. Air interface simulator, http://www.qosmotec.com/

[8] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks", in *Proc. 6th ACM Int. Symp. Mob. Ad Hoc Netw. Comp.*, Urbana-Champaign, USA, 2005, pp. 68–77.

[9] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks", in *Wirel. Commun. Netw. Conf. 2005*, New Orleans, USA, 2005, vol. 4, pp. 2051–2056.

[10] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy, "Experimenting with a multi-radio mesh networking testbed", in *WiNMee 1st Worksh. Wirel. Netw. Measur.*, Riva Del Garda, Italy, 2005.

[11] W. Vandenberghe, K. Lamont, I. Moerman, and P. Demeester, "Connection management over an Ethernet based wireless mesh network", in *WRECOM Wirel. Rur. Emerg. Commun. Conf.*, Rome, Italy, 2007.

[12] E.-S. Jung and N. H. Vaidya, "Power aware routing using power control in ad hoc networks", Tech. Rep., CSL, University of Illinois, Urbana, Febr. 2005.

[13] Multiband Atheros driver for wifi, http://madwifi.org/

[14] D. Lu and D. Rutledge, "Investigation of indoor radio channels from 2.4 GHz to 24 GHz", in *IEEE Anten. Propag. Soc. Int. Symp.*, Columbus, USA, 2003, pp. 134–137.

[15] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey", *Comput. Netw.*, vol. 47, no. 4, pp. 445–487, 2005.

[16] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges", *Ad Hoc Netw.*, vol. 1, no. 1, pp. 13–64, 2003.

[17] C. Verikoukis, L. Alonso, and T. Giamalis, "Cross-layer optimization for wireless systems: a European research key challenge", *Glob. Commun. Newslett.*, pp. 1–3, July 2005.

[18] NS2. The network simulator, http://www.isi.edu/nsnam/ns

Stefan Bouckaert received the Masters degree in electrotechnical engineering from the University of Ghent, Gent, Belgium, in 2005. In August 2005, he joined the INTEC Broadband Communications Networks Group at the same university, where he is currently working as a doctoral researcher. His Ph.D. research includes the design and implementation of a cross-layer framework for wireless mesh networks. His main research interests are in wireless mesh networks and cross-layer optimized dynamic channel selection protocols.

e-mail: Stefan.Bouckaert@intec.ugent.be
IBCN – Department of Information Technology
Ghent University
Gaston Crommenlaan 8 bus 201 – B-9050 Gent, Belgium

Dries Naudts received the Masters degree in computer science from the University of Ghent, Gent, Belgium, in 2001. In April 2005, he joined the INTEC Broadband Communication Networks Group, where he is currently working as a researcher. He is currently involved in the research on broadband mobile and wireless communication. His main research interest are in ad hoc and mesh networking and focuses on the development of routing protocols and monitoring algorithms.

e-mail: Dries.Naudts@intec.ugent.be
IBCN – Department of Information Technology
Ghent University
Gaston Crommenlaan 8 bus 201 – B-9050 Gent, Belgium

Ingrid Moerman received the Masters degree in electrotechnical engineering and the Ph.D. degree from the Ghent University, Gent, Belgium, in 1987 and 1992, respectively. Since 2000, she is a part-time Professor at Ghent University. In 2006 she also joined IBBT, where she is coordinating several interdisciplinary research projects. She is currently involved in the research and education on broadband mobile and wireless communication networks. Her main research interests are: wireless broadband networks for fast moving users, mobile ad hoc networks, interaction between ad hoc and infrastructure networks, personal networks, body area networks, sensor and control networks, wireless mesh networks, fixed

mobile convergence, protocol boosting on wireless links and QoS support.
e-mail: Ingrid.Moerman@intec.ugent.be
IBCN – Department of Information Technology
Ghent University
Gaston Crommenlaan 8 bus 201 – B-9050 Gent, Belgium

Piet Demeester received the Masters degree in electro-technical engineering and the Ph.D. degree from the Ghent University, Gent, Belgium, in 1984 and 1988, respectively. In 1992 he started a new research activity on broadband communication networks resulting in the IBCN-Group (INTEC Broadband Communications Network Research Group). Since 1993 he became Professor at the Ghent University where he is responsible for the research and education on communication networks. The research activities cover various communication networks (IP, ATM, SDH, WDM, access, active, mobile), including network planning, network and service management, telecom software, internetworking, network protocols for QoS support, etc. He is author of more than 300 publications in the area of network design, optimization and management. He is member of the editorial board of several international journals and has been member of several technical program committees (ECOC, OFC, DRCN, ICCCN, IZS).
e-mail: Piet.Demeester@intec.ugent.be
IBCN – Department of Information Technology
Ghent University
Gaston Crommenlaan 8 bus 201 – B-9050 Gent, Belgium

Paper

Practical analysis of IEEE 802.11b/g cards in multirate ad hoc mode

Katarzyna Kosek, Szymon Szott, Marek Natkaniec, and Andrzej R. Pach

Abstract—In multirate ad hoc networks, mobile stations usually adapt their transmission rates to the channel conditions. This paper investigates the behavior of IEEE 802.11b/g cards in a multirate ad hoc environment. The theoretical upper bound estimation of the throughput in multirate ad hoc networks is derived. The measurement scenarios and obtained results are presented. For result validation the theoretical and experimental values are compared. The achieved results, presented in the form of figures, show that cards manufactured by independent vendors perform differently. Therefore, choosing the optimum configuration, according to the user's requirements, is possible.

Keywords*— *ad hoc, IEEE 802.11b/g cards, measurements, multirate.

1. Introduction

Wireless networks based on the IEEE 802.11 family of standards have become widespread in recent years. Even though access points are being deployed both at home and public places, it is the ad hoc mode of 802.11 which is expected to become increasingly popular in the near future. One of the features of 802.11 devices, which can significantly increase their performance, is the use of adaptive multirate transmission schemes.

All four currently used IEEE standards support multirate, i.e., 802.11 [1], 802.11a [2], 802.11b [3], and 802.11g [4]. Each of them allows different speeds in the uplink and downlink directions depending on current physical conditions of the radio channel.

The theoretical performance of multirate capable devices has been measured extensively but practical results vary. This is not only due to different test-beds and radio conditions, but also because of vendor implementations.

A good example of this problem can be found in [5], where several IEEE 802.11 cards from different vendors were analyzed. The stress was on medium access control (MAC) implementations and hardware delays. Two meaningful conclusions appeared. First of all, it was shown that a notable unfairness in rate selection was present among different commercial cards and, furthermore, that the unfairness is a result of different hardware/firmware implementations. It is expected that a similar situation will be observed in multirate IEEE 802.11b/g ad hoc environments.

The aim of this work is to show the differences in performance and interoperability of multirate IEEE 802.11b/g cards of the following vendors: Linksys, Lucent and Proxim. All cards were operating in ad hoc mode. One server was sending file transfer protocol (FTP) traffic to two clients.

The rest of the paper is organized as follows. The state of the art is presented in Section 2. A mathematical model for calculating transmission rates in IEEE 802.11 is described in Section 3. The measurement scenarios and results are shown in Sections 4 and 5, respectively. Section 6 gives a validation of the achieved results. Section 7 closes the paper summarizing the main conclusions.

2. State of the art

The IEEE 802.11 family of standards does not provide any method of automatic rate selection in the presence of multirate capable devices. Because of this, there are many possible schemes of choosing the appropriate rate and it is up to the card vendors to decide which one to use.

The cooperation of cards of different standards is possible because the preamble and header of each frame is sent with the basic rate – understandable by all cards. Only the payload can be sent at higher rates (cf. Table 1). This is especially important for IEEE 802.11b and IEEE 802.11g cooperation.

It must also be noted that transmission rates are not linear. Therefore, e.g., an 11 Mbit/s link with a delivery ratio of just above 50% always outperforms a 5 Mbit/s link.

Table 1
Comparison of preamble, header, and payload rates

Mode (lp/sp: long/short preamble)	Physical layer convergence procedure (PLCP)		Payload [Mbit/s]
	preamble [Mbit/s]	header [Mbit/s]	
802.11	1	1	1 or 2
802.11b lp	1	1	1, 2, 5.5 or 11
802.11b sp	1	2	2, 5.5 or 11
802.11g lp	1	1	1, 2, 5.5, 6, 9, 11, 12, 18, 22 (optional), 24, 33 (optional), 36, 48 or 54
802.11g sp	1	2	2, 5.5, 6, 9, 11, 12, 18, 22 (optional), 24, 33 (optional), 36, 48 or 54

Multirate algorithms can be based on statistics. The auto rate fallback (ARF) [6] protocol is perhaps the first multirate algorithm developed and one of the most commonly used. To determine the channel quality, ARF utilizes link layer acknowledgement (ACK) frames (i.e., the frame error rate – FER). After a given number of consecutive ACKs have been received, the transmission rate is increased. The loss of a similar number of ACKs causes the node to decrease the transmission rate. The main advantage of ARF is that it is simple to implement and does not interfere with the IEEE 802.11 standards. However, it is slow to adapt to channel conditions. It tries to change the rate even for stable links, and can mistake collisions for channel losses.

Most popular WLAN cards currently use the Atheros chipset which (under Linux) can be configured with the innovative MadWiFi driver. This driver implements three different rate adaptation algorithms: Onoe [7], adaptive multi rate retry (AMRR) [8], and SampleRate [9]. Onoe, the default algorithm, is based on ARF and looks for the highest bitrate that has a loss rate less than 50%.

A binary exponential backoff scheme enables AMRR to work well for high latency systems. SampleRate uses aggressive probe packets to estimate the optimum transmission rate.

A different approach to multirate selection is presented by signal-to-noise ratio (SNR)-based algorithms such as receiver based auto rate (RBAR) [10]. In this solution, the receiver measures the SNR value of the received request to send (RTS) and uses the clear to send (CTS) frame to inform the sender of the desired rate. This allows for very fast adaptability, but requires changes in the IEEE 802.11 standard and the constant use of the RTS/CTS mechanism.

A very efficient approach seems to be the opportunistic auto rate (OAR) protocol [11]. It utilizes the coherence times of good channel conditions to send high-rate multiframe bursts. This is similar to the transmission opportunity (TXOP) feature of IEEE 802.11e [12]. OAR has low overhead and can increase fairness in the network. However, it also requires changes to the IEEE 802.11 standard.

Despite many theoretical analyses of IEEE 802.11 performance, not much study has been done to measure interoperability performance between cards belonging to different vendors. A recent analysis can be found in [5] (closely related to the work done in [13]). The authors measure the performance of six IEEE 802.11b cards (in infrastructure mode) to determine whether they adhere to standards. Their main conclusion is that most of the unfairness between commercial cards is due to the hardware/firmware implementations, rather than channel properties. Furthermore, they state that cards belonging to the same vendor exhibit better fairness.

Garoppo *et al.* have presented an interesting comparison between analytical, simulation and experimental results for two IEEE 802.11b cards from different vendors [14]. Their results show high correlation between the modeled, simulated and measured values. However, they also notice a meaningful difference in the performance of the two cards in an infrastructure network.

Performance measurements of the saturation throughput[1] of five different IEEE 802.11b access points (APs) can be found in [15]. The upper bound of the AP throughput was considered. The three major observations are as follows. Firstly, an increase in the load offered to the AP's Ethernet interface does not always result in throughput increase. Secondly, for several APs, if the offered load exceeded their bridging capabilities they reduced their downlink throughput. Finally, better performance in certain directions was observed. The overall conclusion was that meaningful differences in the maximum saturation throughput exist for APs from different vendors.

3. Mathematical model

The mathematical model derived in this section is based on work presented in [16] and [17]. The aim of this model is to obtain the theoretical upper bound estimation of the throughput in a multirate ad hoc environment.

We consider a situation in which station i starts its transmission of a data (DATA) frame of length l to station j at time t. The basic assumptions are that data frames are of equal length, there are no hidden stations and all data frame transmissions are independent. Furthermore, the MAC performance is only evaluated, pure DATA/ACK mode is assumed and all currently transmitting/receiving stations remain stationary.

Let us assume the following notation: A is the set of all stations in a base station system (BSS), N is the total number of stations in A, l is the length of the data frame, l_{ACK} is the length of the ACK frame (all measured in normalized time units). Other parameters are as follows: β is the propagation delay, S is the overall system throughput, T_S (or T_C) is expected time interval between periods when the channel is idle for a distributed inter-frame space (DIFS) period, within which at least one successful (or collided) transmission took place.

A successful transmission must fulfill the following three conditions. Firstly, the sender and the receiver stations are not hidden from each other. Secondly, no other station being within the range of the receiver starts its transmission within the time period $\left[t-\beta, t+\beta\right]$. Finally, no other station being within the range of the sender receives any successful frame within the time period $\left[t-\beta, t+\beta\right]$.

Once a channel is sensed idle for a DIFS interval, the time needed for the data frame destined to station j to be generated at station i is assumed to be exponentially distributed with a rate λ or $G(i,j)$ (equivalent terms). As a consequence, the total rate for a common channel in a single BSS is $N(N-1)\lambda$ or $G=\sum\limits_{i,j} G(i,j)$.

[1] We define throughput as the ratio of the data transmitted in the link layer (including frame headers) to the time needed to deliver the traffic from one node to another.

A simple observation shows that:

$$T_C \geq \frac{1}{G} + l + DIFS + \beta\,, \tag{1}$$

$$T_S = \frac{1}{G} + l + SIFS + l_{ACK} + DIFS + \beta\,, \tag{2}$$

where $\frac{1}{G}$ is the expected time until the beginning of a transmission of the first frame after the channel was sensed idle for DIFS, and SIFS is short inter-frame space.

Let us denote by $p_s(i,j|m,n)$, where $m,n \in A$, the probability of a successful data frame transmission from station i to station j under the condition that, after a DIFS interval, a data frame transmission between stations m and n occurs. As a result, the effective lower bound estimation of the expected number of successful transmissions for a Poisson process can be given as follows:

$$p_s(i,j) = \mathrm{e}^{-N(N-1)\lambda\beta}\,. \tag{3}$$

The probability that station i starts its transmission to station j before the end of an idle period is $\frac{G(i,j)}{G}$. The probability that station i starts its transmission to station j before β (after the idle period was interrupted by a transmission between stations m and n) is given by $\frac{G(m,n)}{G}\left(1-\mathrm{e}^{-\beta G(i,j)}\right)$.

Let us denote by $S(i,j)$ the throughput between stations i and j and, because $l_{ACK} + SIFS \ll l$, let us assume that $T_C \approx T_S$. As a consequence we get:

$$S(i,j) \approx \frac{p_s(i,j)\dfrac{G(i,j)}{G} + p_s(i,j)\left(1-\mathrm{e}^{-\beta G(i,j)}\right)\sum\limits_{(m,n)}\dfrac{G(m,n)}{G}}{T_S}$$

$$= \frac{p_s(i,j)\dfrac{1}{N(N-1)} + p_s(i,j)\left(1-\mathrm{e}^{-\beta\lambda}\right)\dfrac{N(N-1)-1}{N(N-1)}}{\dfrac{1}{G} + l + SIFS + l_{ACK} + DIFS + \beta}\,. \tag{4}$$

Denoting the overall upper bound on the system throughput as $S = \sum\limits_{(i,j)\in A} S(i,j)$ we get:

$$S = N(N-1)\frac{p_s(i,j)\dfrac{1}{N(N-1)} + p_s(i,j)\left(1-\mathrm{e}^{-\beta\lambda}\right)\dfrac{N(N-1)-1}{N(N-1)}}{\dfrac{1}{G} + l + SIFS + l_{ACK} + DIFS + \beta}$$

$$= \frac{p_s(i,j) + p_s(i,j)\left(1-\mathrm{e}^{-\beta\lambda}\right)N(N-1) - 1}{\dfrac{1}{G} + l + SIFS + l_{ACK} + DIFS + \beta}\,. \tag{5}$$

4. Measurement scenarios

The measurements of the performance and interoperability of 802.11b/g wireless cards from different vendors were carried out in usual office conditions. The tested cards were: Linksys WPC-11, Lucent Silver PC24E, and Proxim 8480-WD. All cards worked in ad hoc mode. Their output power was set to 30 mW. The card vendors do not provide information on the type of multirate algorithms used.

In the considered scenario, the test-bed consisted of three homogenous stations (Fig. 1): one FTP server (station C) and two clients (stations A and B). Both clients, when connected to the server, began downloading a 1 GB file what allowed to capture more than 50 thousand FTP frames transmitted from the server to the clients.

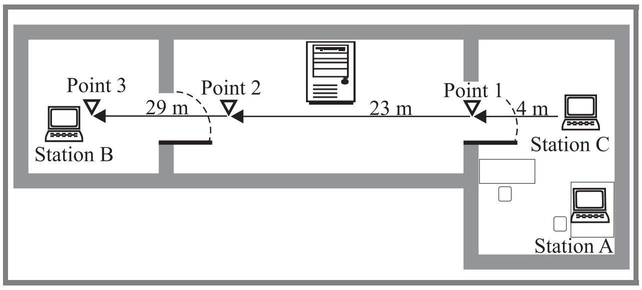


Fig. 1. Test-bed.

Station B was mobile. It increased its distance from the server. Station A was stationary. All measurements were performed in three different points marked in Fig. 1 by triangles. The aim of the experiment was to determine, whether the increasing distance of station B would impact the multirate capabilities of station C, i.e., whether the transmission from the server to station A would be influenced.

All possible sources of interference in the 2.4 GHz and 5 GHz bands (e.g., access points or Bluetooth devices) were eliminated for all experiments.

5. Measurement results

From all the acquired results, we have decided to present six case scenarios, which serve as an illustration for certain important findings. It is important to keep in mind that since the clients were downloading data from the FTP server, the vast majority of the analyzed data are the DATA frames sent by the server and the ACK frames it received in return. Therefore, the results show how the server behaved (in terms of rate selection) when simultaneously communicating with the two clients.

The first, the second and the third scenarios are presented in Figs. 2–4, respectively. They show the percentage of DATA and ACK frames received/sent by the clients from/to the server during the whole experiment. As can be seen in the figures, three different measurement points are considered. In all of the three scenarios the stations were communicating with the use of the IEEE 802.11b standard. In terms of actual bytes sent, the overall share of the ACK frames is of course extremely small compared to the DATA frames.

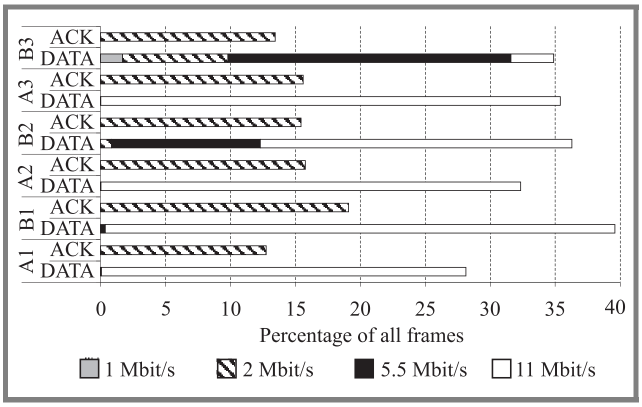


Fig. 2. Multirate performance of two Linksys cards (A and B) at three measurement points (1, 2, and 3): transmission speed versus percentage of frames sent (at a given measurement point). The server was using a Lucent card.

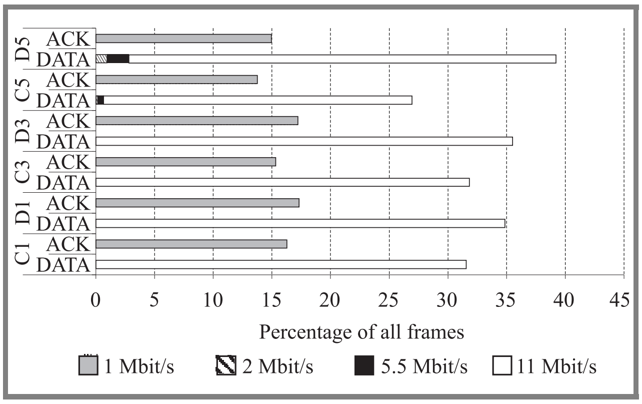


Fig. 3. Multirate performance of a Cisco 350 (C) and a DLink (D) card at three measurement points (1, 3, and 5): transmission speed versus percentage of frames sent (at a given measurement point). The server was using a Cisco 350 card.

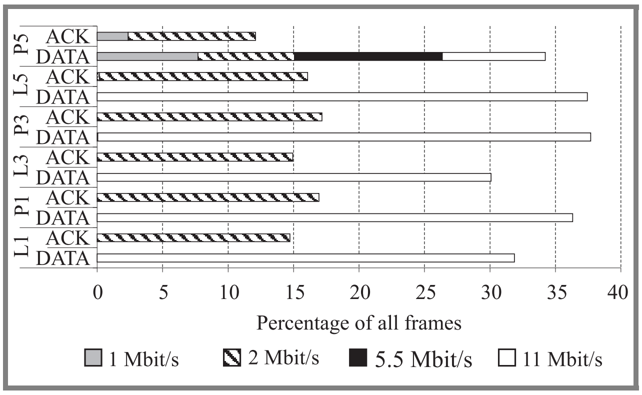


Fig. 4. Multirate performance of a Proxim (P) and a Linksys (L) card at three measurement points (1, 3, and 5): transmission speed versus percentage of frames sent (at a given measurement point). The server was using a 3Com card.

The first scenario consisted of a Lucent server and two Linksys clients (stationary – A, moving – B, see Fig. 2). The second scenario consisted of a Cisco 350 server and two clients (Cisco 350 – stationary, DLink – moving, see Fig. 3). The third scenario consisted of a 3Com server and two clients (Linksys – stationary, and Proxim – moving, see Fig. 4). In all cases, practically all the time, the servers were sending their DATA frames to stationary clients at a constant rate of 11 Mbit/s (independently of the measurement point). Whenever their transmission rates dropped, they dropped to 5.5 Mbit/s. Such a situation did not happen often, i.e., almost all frames were sent at the highest possible rate. For the moving stations, however, the servers' transmission rates dropped the further the clients were away from the servers. The worst performance of a moving station was observed in the first case, slightly better for the third case and the best for the second case. Therefore, it can be concluded that at long distances it is hard for a Linksys client card to communicate with a Lucent server card. Additionally, the Proxim client can communicate with the 3Com card, though, at long distances its transmission speed drops. The most satisfying conclusion is that a DLink client can communicate flawlessly with a Cisco 350 server even at long distances. In the view of ACKs for both stationary and moving clients, in the second scenario the cards were sending ACKs with a lower transmission rate (i.e., 1 Mbit/s) than in the first (i.e., 2 Mbit/s) and the third scenario (i.e., generally 2 Mbit/s but also 1 Mbit/s at long distances).

In the second set of measurement scenarios (fourth to sixth) the cards were operating in the IEEE 802.11g standard, which allows for a wide range of transmission rates (up to 54 Mbit/s). These scenarios proved to be more complex in terms of the data rates used.

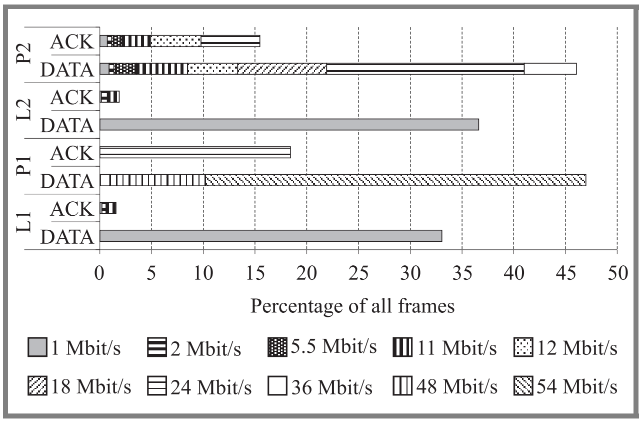


Fig. 5. Multirate performance of a Linksys (L) and Proxim (P) card at two measurement points (1 and 2): transmission speed versus percentage of frames sent (at a given measurement point). The server was using a Proxim card.

In the fourth scenario, the server used a Proxim card, the stationary client – a Linksys card, and the moving client – a Proxim card as well. The results are shown in Fig. 5. The first observation from the presented figure is that the Proxim card present at the server was using the basic rate (1 Mbit/s) to send its DATA frames to the Linksys client. This occurred despite the fact that the Linksys card was returning ACK frames in multiple rates (up to 11 Mbit/s).

The reason for this is most likely vendor incompatibility. On the other hand, the Proxim client established a high speed link with the Proxim server. Both the DATA and ACK frames were able to utilize the potential of multiple transmission rates. At the first measurement point, the majority of DATA frames were sent with the highest available speed (54 Mbit/s), whereas all the ACK frames were sent at 24 Mbit/s. In the third measurement point up to 8 different rates were used (depending on radio conditions). The fact that the Linksys card was transmitting at 1 Mbit/s means that it was underusing the channel and, therefore, degrading overall network performance. This is an example of how vendor incompatibility can lead to unfairness in the shared radio channel.

The next scenario (Fig. 6) had a similar configuration as the previous one. The only difference was that the stationary client was using a Lucent card. This time the cooperation between different cards was somewhat better. The server was sending data at 11 Mbit/s to the Lucent client. The client was responding with ACK frames sent in multiple rates up to 11 Mbit/s. This result is better than in the previous scenario where only 1 Mbit/s was achieved. However, the communication between Proxim cards was better because they made use of the full range of possible rates.

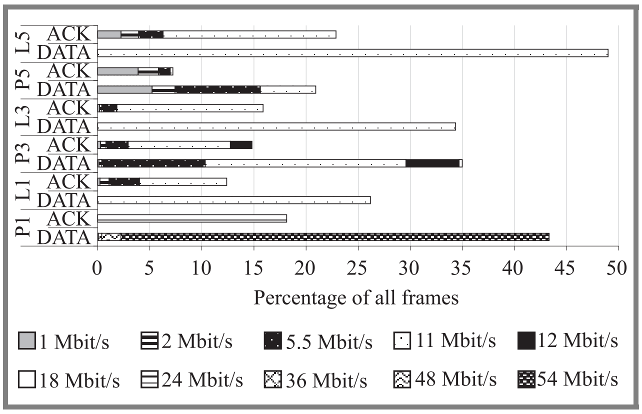


Fig. 6. Multirate performance of a Lucent (L) and a Proxim (P) card at three measurement points (1, 3, and 5): transmission speed versus percentage of frames sent (at a given measurement point). The server was using a Proxim card.

The behavior described above was present in the final scenario (Fig. 7) in which the stationary client's card was a Cisco card. The only difference was that the Proxim server managed to send not only 11 Mbit/s frames but also a small number of 12 Mbit/s frames to the Cisco client. The stationary client responded with ACK frames in multiple rates (up to 11 Mbit/s).

The main conclusion from these three scenarios operating in the IEEE 802.11g standard is that the Proxim card had severe problems with establishing a high rate connection with cards from other vendors. No two cases were the same, with the maximum rate used being 1, 11 and 12 Mbit/s. However, all of the non-Proxim clients sent ACK frames at a rate of 11 Mbit/s (or less) and the Proxim client used a maximum of 24 Mbit/s. The exact reason of this behavior is of course unknown as we do not know how the cards chose to adapt their rate.

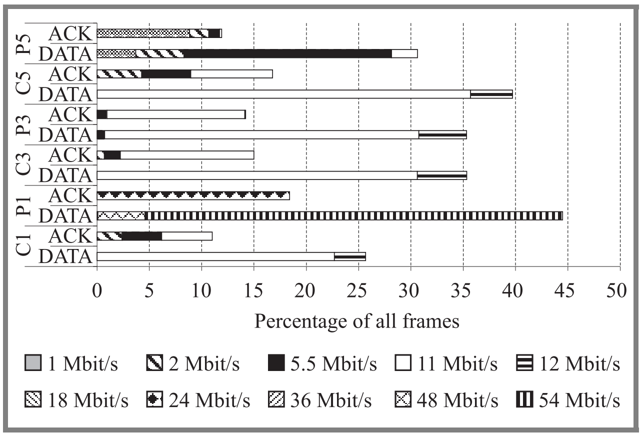


Fig. 7. Multirate performance of a Cisco (C) and a Proxim (P) card at three measurement points (1, 3, and 5): transmission speed versus percentage of frames sent (at a given measurement point). The server was using a Proxim card.

Comparing the scenarios operating in the 802.11b and 802.11g standards, we can see that in the first ones the ACK frames were sent at basic rates of either 1 Mbit/s or 2 Mbit/s. However, in the second set of scenarios multiple rates were used. Based on these measurements it seems that in the IEEE 802.11b standard ACK frames are transmitted at a rate no larger than 2 Mbit/s, whereas in 802.11g much higher rates can be used (up to 24 Mbit/s). Furthermore, we can see that the rate of the mobile station does not impact the established rate of the stationary one. This means that near and far stations can coexist with multiple rates.

6. Result validation

In order to evaluate the obtained link layer throughput, we have compared two scenarios (first and fourth) with theoretical values derived from the analytical model presented in Section 3. This comparison is presented in Table 2.

In order to take into account the use of multiple rates by the station, the theoretical value of the system throughput was calculated for each available rate and then summed up using a weighted average (based on bytes transmitted at a given rate). The DATA frame length l was taken as the weighted average of all transmitted DATA frames.

For the first scenario (Lucent server, Linksys clients), the measured results quite closely resemble the theoretical calculations. In this scenario, not many transmission rates were used and we believe this is the reason why the results are similar. This is also a further validation of our mathematical model.

In the fourth scenario, however, the number of rates used was much larger and the difference between the theoretical and the measured values is quite significant. This is because our model did not take into account the procedures needed to change the rate and the impact of lost frames. This is why the measured values were much lower than the theoretical ones.

Table 2
Comparison of theoretical and achieved throughput

Point	FTP server	Receiving station	Throughput [Mbit/s]		Difference [%]
			theoretical	measured	
1	Lucent	Linksys A	4.99	5.16	3.4
	Lucent	Linksys B	4.96	5.03	1.4
2	Lucent	Linksys A	4.97	4.72	0.75
	Lucent	Linksys B	4.15	3.80	8.4
3	Lucent	Linksys A	5.01	4.11	17.96
	Lucent	Linksys B	2.35	2.35	0
1	Proxim	Linksys	0.51	0.004	99.2
	Proxim	Proxim	21.64	3.66	83.1
2	Proxim	Linksys	0.51	0.005	99.0
	Proxim	Proxim	9.13	1.95	78.6

7. Conclusions

The behavior of IEEE 802.11b/g cards in multirate ad hoc environments has been presented in this paper. Certain popular and widely available WLAN cards from different vendors were tested in terms of throughput and interoperability. Both the measurements and analytical results were compared. The obtained results show, that the performance of a WLAN card highly depends on its manufacturer. Some cards turned out to be significantly worse than others, because they implement the multirate functionality differently. Furthermore, the authors are convinced that the sensitivity of the cards also had a significant impact on the correct reception of packets.

Therefore, to achieve high performance, it is crucial to implement an appropriate algorithm which can choose the best transmission rate. If the rate is chosen too high, the frame error rate increases which leads to more retransmissions and, as a consequence, network performance decreases. If the card is not able to quickly adapt to varying radio channel conditions or if it chooses a rate which is too low, the degradation of network performance will also occur. Thus, high adaptability with the utilization of short periods of good conditions seems to be a good solution.

The following general conclusions can be formulated. First of all, the obtained results show the inefficiency of multirate algorithms used in commercial cards. Secondly, it can be observed that cards of the same model from one vendor cooperate much better. If the number of used rates grows significantly (which is possible for IEEE 802.11g), the achieved throughput drastically decreases. This is because cards spend time adjusting to the channel conditions by trying to find the appropriate rate. Finally, perhaps the rate used to send the ACK frames should suggest to the sender of the DATA frames which rate to choose.

The differences between the ideal, theoretical and measured results (as exemplified in the fourth scenario) can be 1000-fold. Therefore, there is a strong need to develop new, efficient multirate algorithms. Most importantly, adequate agreements between different vendors are required to improve the cooperation of WLAN devices, especially since multirate IEEE 802.11b/g combo cards dominate the market.

When buying a WLAN card it is important to take into account the transmission rates but also other parameters (e.g., sensitivity, output power and laboratory tests). Therefore, to facilitate the user's final choice Table 3 was prepared and presented. It contains the comparison of subjective card compatibility. ProximG card is the winner since it reaches the best compatibility results. D-Link and Cisco cards are a little poorer. The Lucent and Linksys cards seem to be the worst choice considering the compatibility aspect.

Table 3
Comparison of subjective card compatibility

Cards	3Com	Proxim G	Lucent	Cisco 350	Average
Proxim G	2	1.75	—	—	1.88
Lucent G	1.5	1	—	—	1.25
Linksys	1	0	2	—	1.00
DLink	—	1	2	2	1.67
Cisco	2	1	—	2	1.67

Future research should provide more information about the problem of multirate adaptation and card behavior. The proposed mathematical model should be revised to be in line with experimental results. Furthermore, it is important to continue studying the problem of achieving multirate compatibility between cards belonging to different vendors.

Acknowledgment

This work has been carried out under the Polish Ministry of Science and Higher Education grant no. N N517 4391 33.

References

[1] "IEEE Standards for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements" – Part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999).

[2] "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements" – Part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", Amendment 1: "High-speed Physical Layer in the 5 GHz band", IEEE 802.11a-1999 (8802-11:1999/Amd 1:2000(E)).

[3] Supplement to 802.11-1999, "Wireless LAN MAC and PHY Specifications: Higher-speed Physical Layer (PHY) Extension in the 2.4 GHz Band", IEEE 802.11b-1999.

[4] "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements" – Part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" – Amendment 4: "Further Higher-speed Physical Layer Extension in the 2.4 GHz Band", IEEE 802.11g-2003.

[5] D. Stefano, G. Terrazzino, L. Scalia, I. Tinnirello, G. Bianchi, and C. Giaconia, "An experimental test-bed and methodology for characterizing IEEE 802.11 network cards," in *Proc. Int. Symp. World Wirel. Mob. Multimed. Netw*., Niagara-Falls, USA, 2006.

[6] B. Awerbuch, D. Holmer, and H. Rubens, "High throughput route selection in multi-rate ad hoc wireless networks", Tech. Rep., Johns Hopkins University, Computer Science Department, 2003.

[7] Madwifi: "Multiband Atheros driver for WiFi", Sept. 2005, http://madwifi.sourceforge.net/

[8] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach", INRIA Res. Rep., no. 5208, 2004.

[9] J. Bicket, "Bit-rate selection in wireless networks", Master's thesis. Cambridge: MIT, 2005.

[10] G. Holland, N. H. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks", in *Proc. ACM Int. Conf. Mob. Comp. Netw. MobiCom 2001*, Rome, Italy, 2001.

[11] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks", in *Proc. ACM MobiCom 2002 Conf.*, Atlanta, USA, 2002.

[12] "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements" – Part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications" – Amendment 8: "Medium Access Control (MAC) Quality of Service Enhancements", IEEE 802.11e-2005.

[13] D. Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi, and C. Giaconia, "On the fidelity of IEEE 802.11 commercial cards", in *Proc. First Int. Conf. Wirel. Intern.*, Budapest, Hungary, 2005.

[14] R. G. Garroppo, S. Giordano, and S. Lucetti, "IEEE 802.11 b performance evaluation: convergence of theoretical, simulation and experimental results", in *11th Int. Telecommun. Netw. Strat. Plann. Symp., NETWORKS 2004*, Vienna, Austria, 2004.

[15] E. Pelletta and H. Velayos, "Performance measurements of the saturation throughput in IEEE 802.11 access points", in *Proc. Third Int. Symp. Model. Opt. Mob. Ad Hoc, Wirel. Netw.*, Riva Del Garda, Italy, 2005.

[16] J. B. Lee and A. Elftheriadis, "A performance analysis for the IEEE 802.11 wireless LAN MAC protocol", *IEEE Trans. Netw.*, March 1998, http://comet.columbia.edu/∼campbell/andrew/publications/alex.htm

[17] M. Natkaniec, "A performance analysis of the IEEE 802.11 DCF protocol", in *PGTS 2002 Conf.*, Gdańsk, Poland, 2002.

Katarzyna Kosek received her M.Sc. degree in telecommunications from the AGH University of Science and Technology, Kraków, Poland, in 2006. Now, she works towards her Ph.D. degree at the Department of Telecommunications, AGH UST. Her general research interests are focused on wireless networking. The major topics include wireless LANs (especially ad hoc networks) and quality of service provisioning within these networks. She is a reviewer of international journals and conferences. She is/was involved in several European projects: DAIDALOS II, CONTENT, CARMEN as well as grants supported by the Polish Ministry of Science and Higher Education. She has also co-authored a number of research papers.

e-mail: kosek@kt.agh.edu.pl
Department of Telecommunications
AGH University of Science and Technology
A. Mickiewicza st 30
30-059 Kraków, Poland

Szymon Szott received his M.Sc. degree in telecommunications from the AGH University of Science and Technology, Kraków, Poland, in 2006. Currently he works towards his Ph.D. degree at the Department of Telecommunications, AGH UST. His professional interests are related to wireless networks (in particular: QoS provisioning and security of ad hoc networks). He is a reviewer of international journals and conferences. He is/was involved in several European projects: DAIDALOS II, CONTENT, CARMEN as well as grants supported by the Polish Ministry of Science and Higher Education. He is the co-author of several research papers.

e-mail: szott@kt.agh.edu.pl
Department of Telecommunications
AGH University of Science and Technology
A. Mickiewicza st 30
30-059 Kraków, Poland

Marek Natkaniec received the M.Sc. and Ph.D. degrees in telecommunications from the Faculty of Electrical Engineering, AGH University of Science and Technology, Kraków, Poland, in 1997 and 2002, respectively. In 1997 he joined AGH UST. Now, he works as an Professor Assistant at the Department of Telecommunications. He is responsible for lectures, projects and laboratories of the computer networks. His general research interests are in wireless networks. Particular topics include wireless LANs, designing of protocols, formal specification of communication protocols, modeling and performance evaluation of communication networks, quality of service, multimedia services provisioning and cooperation of networks. He is a reviewer to international journals and conferences. He has/had actively participated in European projects: MOCOMTEL, PRO-ACCESS, DAIDALOS I, DAIDALOS II, CONTENT, CARMEN as well as grants supported by the Polish Ministry of Science and Higher Education. He co-authored four books and over 50 research papers.

e-mail: natkanie@kt.agh.edu.pl
Department of Telecommunications
AGH University of Science and Technology
A. Mickiewicza st 30
30-059 Kraków, Poland

Andrzej R. Pach received the M.Sc. degree in electrical engineering and the Ph.D. degree in telecommunications from the University of Science and Technology, Kraków, Poland, in 1976 and 1979, respectively, and the Ph.D.Hab. in telecommunications and computer networks from the Warsaw University of Technology in 1989. In 1979, he joined the Department of Telecommunications at the University of Science and Technology, where he is currently a Professor and Chair. He spent his sabbatical leaves at CNET, France, and University of Catania, Italy. He is the Vice-President of the Foundation for Progress in Telecommunications and serves as a Chairman of the IEEE Communications Society Chapter. He has been a consultant to governmental institutions and telecom operators in modern telecommunication networks. His research interests include design and performance evaluation of broadband networks, especially quality of service and network performance of access networks and wireless LANs. He has/had actively participated in COST, COPERNICUS, ACTS, ESPRIT and IST European programs. He is an author of 150 publications including 5 books. He serves as a technical editor to “IEEE Communications Magazine” and editor-in-chief to “Digital Communications – Technologies and Services”. He has also been appointed as an expert in Information Society Technologies by the European Commission.

e-mail: pach@kt.agh.edu.pl
Department of Telecommunications
AGH University of Science and Technology
A. Mickiewicza st 30
30-059 Kraków, Poland

Paper

A hybrid-mesh solution for coverage issues in WiMAX metropolitan area networks

Krzysztof Gierłowski, Józef Woźniak, and Krzysztof Nowicki

Abstract—The new WiMAX technology offers several advantages over the currently available (GSM or UMTS-based) solutions. It is a cost effective, evolving, and robust technology providing quality of service guarantees, high reliability, wide coverage and non-line-of-sight (NLOS) transmission capabilities. All these features make it particularly suitable for densely populated urban environments. In this paper we discuss the design and implementation difficulties concerning network coverage discovered in a test-bed implementation of WiMAX. We point out the presence of unexpected "white spots" in the coverage, which are not inherently characteristic of the WiMAX concept. As a possible remedy to this significant drawback of the otherwise very promising technology, we consider reconfigurable mesh organization of WiMAX base stations. We also suggest directions for further development of this kind of network operation, partly based on our practical experience. Despite the clear advantages of the mesh mode in WiMAX networks, its development is currently at an early stage, due to the high complexity of the necessary mechanisms. In this situation, we propose an original, much simpler solution: the so-called support-mesh mode.

Keywords— IEEE 802.16, WiMAX, coverage, mesh, measurements.

1. Introduction

The new worldwide interoperability for microwave access (WiMAX) technology, based on the IEEE 802.16 standard [1], offers reliable, fast and quality of service (QoS) aware transmission over significant distances. It provides both line-of-sight (LOS) and no-line-of-sight (NLOS) solutions. The LOS solution allows transmissions at rates over 70 Mbit/s over distances up to 50 km (or even more), as long as the antennas of both devices have a direct (not shaded) view of each other. The second solution provides connectivity using reflected signals when the path between the antennas is obstructed. In such a case the range is limited to about 5 km only. The technology supports different modulation and coding schemes coupled with adaptive adjustment of transmission parameters in order to maximize the stable coverage area. Other strong advantages of WiMAX systems include high security, reliability and integrated QoS support, which together allow operators to guarantee their users a contracted level of network services.

The most popular WiMAX system architecture follows a point-to-multipoint (PtMP) data communications model with a coordinating base station (BS) and participating client terminals (subscriber stations – SSs). The standard also specifies the foundation for a mesh mode whereby peer stations participate in self-organizing the network structure and/or its connectivity. All these characteristics make WiMAX an economically appealing solution. Especially, the NLOS capability makes WiMAX the key technology for urban environments, making it possible to cover a large area with a relatively small number of BSs working in PtMP mode. However, at the same time, its complexity significantly complicates system design, particularly in terms of coverage prediction and client station capabilities at a given location, with respect to the available throughput and specific QoS parameters. These difficulties pose a need for specialized software tools that would help system designers assess the effective coverage area and accurately estimate transmission parameters within it.

We shall start from a discussion of the basic theoretical models for wireless network design, from the viewpoint of their possible application to NLOS-capable WiMAX systems. Next, we shall describe measurements conducted in our test-bed installation of WiMAX, which uncovered the white spots effect, a phenomenon very disadvantageous to nomadic and mobile users. This so far undocumented drawback of WiMAX strongly undermines the popular highly favorable opinion regarding the performance of its NLOS mechanisms and can significantly raise the cost of network deployment. As this kind of deficiency is difficult to accept, steps must be taken to resolve the problem. To this end, we shall describe a self-organizing ad hoc WiMAX-based mesh architecture as a solution for the coverage issues and propose mechanisms necessary for the mesh mode to effectively counter the white spots effect for nomadic and mobile networks.

Unfortunately, the mesh mode of WiMAX is currently at an early stage of development, and the requisite mechanisms necessary for it to operate appear numerous and complicated. For these reasons the idea of a WiMAX mesh network receives at best a limited support from hardware manufacturers, and we will probably wait for a long time to see it in a fully functional standardized form. In this situation, we propose an alternative and original solution combining the classical WiMAX point to multipoint mode with the mesh architecture. This hybrid solution is relatively simple, easy to incorporate into current hardware, and it can coexist seamlessly with the standard unmodified equipment.

2. Theoretical coverage models

In all types of wireless systems, including WiMAX, prediction of their coverage area is a very challenging task, especially when we want to mark out the coverage with a practically relevant accuracy. Thus, there is a need for methods to verify provisional results obtained through theoretical calculations. Two basic approaches are popular with the current design practices. The first one requires a test-bed installation and depends entirely on empirical measurements. The second one relies on software tools able to estimate the system coverage with the use of one of the available propagation models. As the first method is rather time consuming and costly, software tools are widely used to support coverage calculation for wireless systems, such as short-range local area systems (WLANs) or more complex wide area networks, consisting of multiple BSs (WMANs, WWANs).

Two basic types of propagation models are employed in wireless systems design [2, 3]:

- Empirical (or statistical) models based on a stochastic analysis of a series of measurements conducted in the area of interest. They are relatively easy to built but not very sensitive to the environmental geometry.

- Site-specific (or deterministic) models, which are far more accurate and need no signal measurements. However, they require a huge amount of data concerning environment geometry, terrain profile, etc., and pose high computational demands.

Owing to the fact that WiMAX systems are intended to provide effective coverage in highly urbanized environments, we are mostly interested in deterministic models, as they can produce sufficiently accurate results for such areas.

Of course, there is always a theoretical possibility to calculate the exact propagation characteristics by solving sets of Maxwell's equations. However, this method would require complex data and very high computational power, causing it to be very inefficient. Therefore, current software tools, based on deterministic propagation models, usually employ simplified simulations: mostly ray-tracing or ray-launching techniques based on uniform geometrical theory of diffraction (UTD) [2]. Such an approach brings about a significant simplification in calculations, making the model an efficient design tool, albeit with a loss of accuracy.

The coverage characteristics of WiMAX differ significantly from those associated with other wireless network technologies employed in similar environments, mainly due to the NLOS capability of WiMAX (see Fig. 1). Thus, a dedicated software model is required to give accurate results [4].

Regardless of the theoretical models employed and their accuracy, experience in wireless systems design and implementation suggests a necessity for empirical measurements in order to confirm that the system design and theoretically obtained parameters are correct [5]. In accordance with good design practice, we implemented a test-bed installation of WiMAX with one base station and conducted extensive measurements and tests of its coverage and transmission parameters [6].

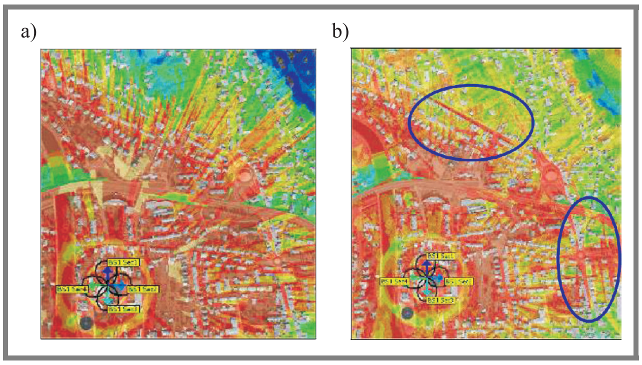


Fig. 1. Results of WiMAX BS coverage simulation: general model (a) and specialized WiMAX coverage model (b).

3. Test-bed installation and example measurements

Both the modeling procedures and tests carried out by hardware manufacturers indicate that the WiMAX technology is indeed very well suited for metropolitan environments and generally offers good coverage, even in highly urbanized areas [7]. To verify these statements and prove the accuracy of the available software design tools, as well as to gather practical design experience, we prepared a test-bed installation consisting of a single WiMAX BS located at Gdańsk University of Technology. We employed a BreezeMAX micro base station [8] provided by the Alvarion company using 3.5 GHz licensed frequency band. We also developed a dedicated software package consisting of a number of control and monitoring tools. These tools communicate with the BS, client terminals, GPS receivers and are able to automate the experiments to a significant degree. They also handle a real-time, initial data analysis to help the measurement teams optimize their work. We have been monitoring long-term operation parameters of the WiMAX installation, with the use of an simple network management protocol (SNMP)-based monitoring system developed especially for this purpose. It allows us to collect and present over 200 parameters concerning BS, SSs and the provided services.

One of our main points of interest was the coverage of WiMAX services in a densely populated metropolitan environment. We carried out a variety of tests including:

- measurements of the BS signal strength in physical layer;

- modulation and coding profile usage as a function of signal quality;

– efficiency of transmission in the medium access control layer (BER, PER);

– quality of service contract adherence for transport layer services.

The tests were performed with the assistance of a hardware spectrum analyzer equipped with an omnidirectional antenna, BreezeMAX PRO BS, SS subscriber stations (PRO and Si models) [8] and transmission performance counters of the base station. In the case of LOS, using the equipment mentioned above, we could expect a reliable communication up to 30 km, and 5 km in the majority of cases related to NLOS scenarios [6].

Such general conclusions sound promising. However, we also made quite unexpected observations. It turned out that in the case of NLOS communication the network did not cover entirely the tested area. We were able to identify some points, dubbed "while spots", that were not covered by our BS. At the same time, we also detected many locations at which the measured coverage (signal parameters) differed significantly from the theoretical estimates. In some places the coverage was a result of repeatedly reflected signals or signals reflected by various objects either impossible or difficult to map, like trees, billboards, trains, trucks, etc. Other places exhibited lack of coverage, even though the obstacles between the BS and the client terminal were relatively minor (Fig. 2). Our measurements also showed that even a very small displacement (20 m horizontal and/or 3 m vertical) of a client station can result in a dramatic degradation of the transmission parameters, from the best possible modulation and coding profile (QAM64 3/4) to the complete loss of connectivity. This effect makes WiMAX system design a very difficult task, requiring empirical measurements to validate the project.

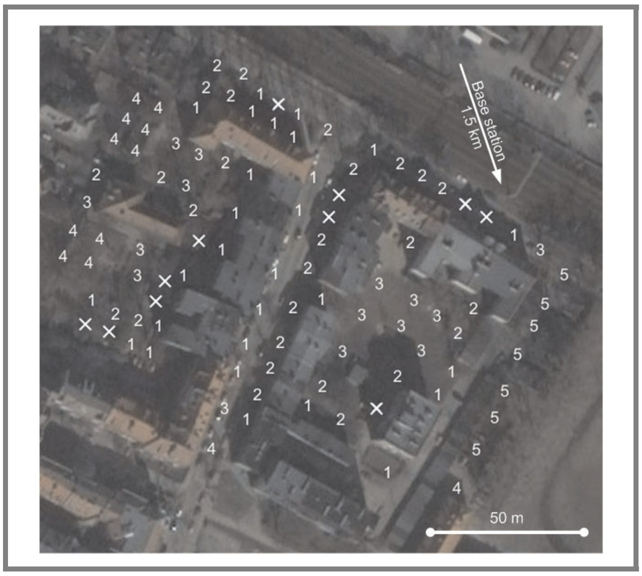


Fig. 2. WiMAX coverage hole effect: measured coverage of the WiMAX base station in 3.5 GHz band. Stable modulation profile chosen by client terminal: × – no communication; 1 – binary phase-shift key (BPSK); 2 – quaternary phase-shift key (QPSK); 3 – quadrature amplitude modulation 16 (QAM16); 4 – QAM64 1/2; 5 – QAM64 3/4.

The described effect of white spots demonstrates difficulties faced by attempts to precisely predict the coverage and QoS parameters of any non-trivial real system. If a service provider is interested in a complete and continuous coverage of an area, they may have to accept and absorb a higher system deployment costs. Also, there is no efficient way to validate the coverage without explicit and detailed measurements.

The problem becomes even more serious for a mobile operator, because mobile terminals can loose and regain connectivity as they move. Such effects can be especially laborious in WiMAX, where each network entry operation is complicated and consumes a significant amount of network and terminal resources (bandwidth, BS processing power, battery).

Summarizing the measurement results, we can state that while the NLOS capability of WiMAX indeed makes the technology fit for highly urbanized areas, it is not without disadvantages and requires a careful design and troublesome practical validation.

4. Network design considerations

The WiMAX poses considerably more complex network design problems than other technologies of lesser coverage. For example, a wireless fidelity (WiFi) [9] wireless local area network can be deployed and verified with a relatively simple, authoritative, and complete set of test measurements. Its typical range (50–300 m) makes such an approach possible. In the case of WiMAX, where the range is measured in kilometers, this solution is not viable, as it is practically impossible to compile a full, empirical coverage map by inspecting all relevant points within the system's range. In a dense metropolitan area with WiMAX NLOS capability, we would have to measure an extremely thick layout of points. Also, as we pointed out in the previous section, we would need a resolution of under 20 m horizontally. Moreover, restricting the measurements to 2-dimensional would not work, because there are significant variations of the effective signal strength related to the vertical placement of the client station, particularly prominent near ground level.

Computational propagation models can help us highlight potential trouble spots and hint at relevant measurement points. They offer a great support during the design process. Regretfully, their application can be costly, because they usually require detailed 3-dimensional digital maps, which may be expensive or even unavailable for the area of interest [10]. Furthermore, commercial products, based on ray-tracing and ray-launching models [2], are not equipped to detect coverage anomalies as small as the described white spots. Our research shows that in order to detect them we must employ a very high resolution of modeling, often higher than the popular resolution of 3-dimensional maps [3]. At such resolution, the simplifications inherent in (and shared by) those models no longer work in our favor. This leads to the need for much higher computational power and longer modeling time and still does not guarantee the detection of all anomalies.

Whether we are able to detect all coverage holes or not, the actual number of BSs needed to provide consistent coverage of the area is higher than inferred from the theoretical modeling. Also, in many places such coverage holes may be impossible to eliminate without a large and economically impractical number of BSs.

There are at least two basic approaches that could be proposed as possible solutions to this problem:

– heterogeneous approach: use a combination of different connectivity technologies;

– homogeneous approach: base the network on a single wireless technology, i.e., WiMAX.

Currently there is a strong trend towards creation of heterogeneous systems, where users can choose from a variety of connectivity technologies [11]. The emerging IEEE 802.21 standard [12] is devoted to a seamless handover between networks of the same or different types. In this case, the best connection (ABC strategy – always best connected) is automatically selected at a given user location and the handover is performed without losing quality of service, if possible [13]. This approach takes into account several different wireless technologies and we will not consider it here. We will focus on the homogenous approach, limited to the WiMAX technology, considering WiMAX mesh architecture as a possible solution.

5. The WiMAX-mesh mode

In the mesh mode of WiMAX, there is no prominent BS, but SSs communicate directly with their neighbors forming a dynamic, self-organizing, multi-hop network. This way, a client station need not be in range of one of the relatively few BSs, but it is sufficient for it to be in range of any other participating client station. The number of those devices is usually much higher than the number of BSs (Fig. 3). Moreover, with correctly designed control protocols and effective methods of joining the network by new stations, its available capacity can be increased (instead of going down) as new clients join the pool.

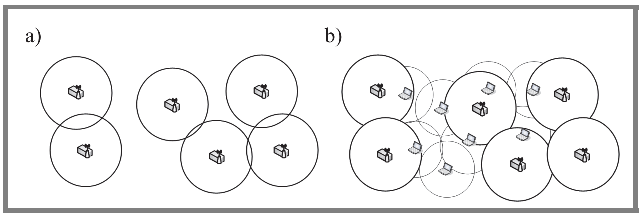


Fig. 3. Coverage comparison of (a) PtMP and (b) mesh mode with the same number of operator provided base stations/nodes.

Unfortunately, this architecture requires much more advanced support mechanisms than a simple PtMP setup, where a single entity (BS) sees and controls all the network activity. In the case of wireless ad hoc mesh architecture, these mechanisms (medium access control, security, etc.) have to be significantly extended and able to operate in a distributed environment. Also the following new mechanisms, not required in a PtMP setup, are necessary:

- Topology control: selects logical network node neighborhood based on its physical neighborhood. Running in all network nodes, it is responsible for overall network topology and a large number of derived characteristics (path lengths, bandwidth available, network capacity, transmission delay, error rates, etc.).

- Route discovery: the set of mechanisms employed to find a route through network nodes to any required destination within and outside the wireless mesh. In WiMAX, it should be able to provide paths with specific QoS guarantees.

- Data forwarding: responsible for retransmitting received traffic addressed to remote nodes, in accordance with QoS guarantees and based on the routing information obtained from the discovery mechanisms.

Most studies on mesh networks, including test implementations, exploit short-range wireless technologies (WLANs or sensor networks) to ensure wide area coverage and high reliability. We claim that, also in the case of a wireless metropolitan network (WMAN) of a much higher intended range, WiMAX-based ad hoc mesh architecture can provide the required functionalities and become practical and economically viable [14].

Due to the relatively high complexity, the WiMAX-mesh mode is not yet specified in the IEEE 802.16 standard. The lack of detailed specification gives us the opportunity to incorporate into the created standard new mechanisms, which will make IEEE 802.16 especially attractive for metropolitan environments, being its prime areas of deployment. Our research related to IEEE 802.16 and measurements in the test-bed shows that mesh architecture based on WiMAX metropolitan area network is likely to solve the coverage-hole problems, as long as a sufficient number of client stations will participate in the network. It will tend to provide a much better terrain coverage than a typical PtMP WiMAX installation. Also the infrastructure cost will be significantly lower.

With the currently observable trends regarding the user demand and manufacturer support for the technology, one can safely predict that the density of client stations will pose no problem in a typical metropolitan environment. It is also likely that the price of a mesh-capable subscriber station will be similar to that of a classical PtMP WiMAX terminal.

The WiMAX-based ad hoc mesh network can provide much better terrain coverage and its development costs are significantly lower than in the corresponding coverage scenario supported only by standard BSs operating in PtMP mode. Of course, there is still a need for a number of operator-provided network nodes as a foundation of the network. Moreover, mesh architecture can provide high reliability due to the high number of redundant network devices, wireless links, and paths to most destinations. It will also scale well, because any participating node brings in additional resources to the network pool.

While the white spots problem can be solved by a sufficiently dense network of mesh nodes and their redundant links, the network control mechanisms need to be able to deal with the rapidly changing mesh topology. In a static mesh configuration (as in a static PtMP scenario), the problem is not serious, as in most cases the placement of network nodes can be pre-optimized. However, with mobile mesh nodes, even a small movement can lead to unpredictable breakdowns and reappearance of inter-node links.

Mobility can significantly reduce the efficiency of network mechanisms, e.g., leading to frequent activation of the discovery mechanisms of the underlying ad hoc routing protocol, which will flood the network with control traffic. Also QoS guarantees are very difficult to maintain in such an environment, as fast and frequent, short-term connectivity losses can occur.

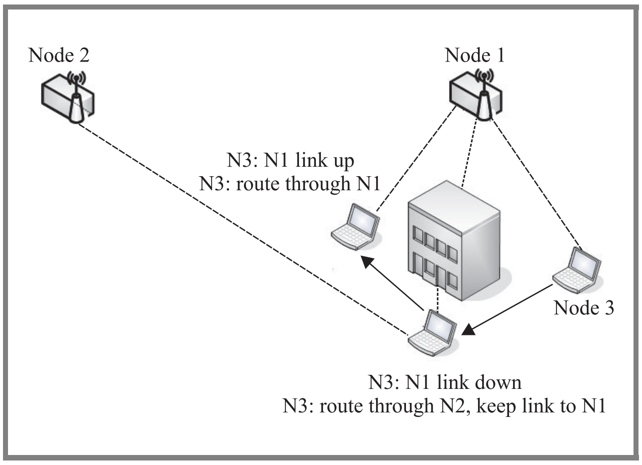


Fig. 4. Short-period link losses and a possible solution with redundant path routing and coverage hole aware topology control.

Fortunately the massive redundancy present in a sufficiently dense mesh network can be harnessed to offset the effect without losing transmission reliability and QoS guarantees. Furthermore, our observations and measurements tell us that signal losses and the corresponding link breakdowns caused by the coverage hole effect are mostly short term events and as such can be efficiently countered with properly designed network control mechanisms. Our work leads us to believe that efficient topology control, taking into account the possibility of short-time disappearance of network links, coupled with redundant path routing and stability-aware routing metric, can solve the described problem (Fig. 4). We are currently working on a simulation model of WiMAX-based self-organizing mesh network resistant to the topology stability issues described above.

6. Support-mesh mode

The main problem with the solution sketched above is the fact that the specification of full WiMAX mesh mode is far from completion. Moreover its development does not receive the attention and support necessary to obtain a practicable solution in the predictable future. Because of this situation we would like to propose an alternative solution, which we call WiMAX support-mesh mode (SMM). It is a hybrid solution utilizing both the standard PtMP architecture with a base station and a self-organizing mesh to support the standard network architecture to eliminate the coverage problems. Our main concern has been to keep the necessary modifications to the existing systems at a minimum. Unfortunately, this goal is in contradiction with the postulate of system efficiency, because the PtMP environment of WiMAX is strictly controlled by the BS. Thus, without a modification of those functions, it is very difficult to arrive at an efficient solution. In this situation we considered two approaches:

- **Variant 1**. A completely transparent solution that requires no alternations to BS hardware or software, only slight modifications in SS functionality.
- **Variant 2**. Requires a slight modification of both BS and SS software, but promises higher system efficiency and robustness.

Both variants allow for seamless coexistence of standard and modified subscriber stations.

In most circumstances the operation of an installation equipped with the support-mesh mode does not differ from that of a classic WiMAX PtMP system. Only in the case of low quality link or connectivity loss between SS and BS the new functionality comes into play (Fig. 5).

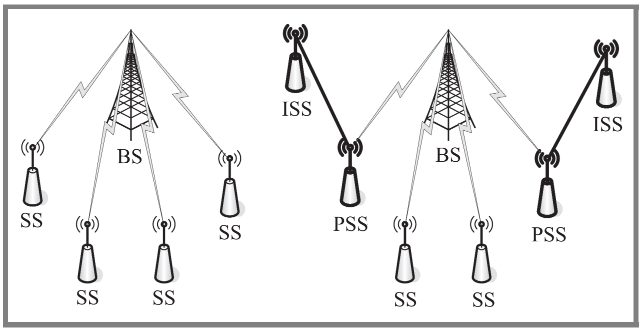


Fig. 5. WiMAX support-mesh mode usage.

In such a case, a support-mesh mode enabled subscriber station (SMM-SS) can connect to another SMM-SS instead of BS and use it as a proxy to maintain its presence in the WiMAX PtMP system. The SMM-SS acting as a proxy (PSS) is then responsible for providing communication between the BS and its connected SMM-SSs (indirectly connected SS – ISS). It is even possible to create multiple layers of proxies in the situation when the PSS loses connectivity to the BS and becomes an ISS itself, without abandoning its PSS role (Fig. 6). Such a multilayer network layout is not particularly efficient and should be avoided by ISSs, which should try to find an alternative (directly connected) PSS, but it is nonetheless admissible and can be useful with low-bandwidth, high-reliability applications.

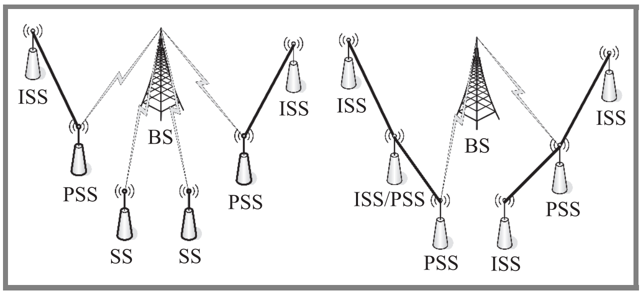


Fig. 6. Multilayer proxying example.

For the system to function effectively, it is recommended that the stations (most importantly PSSs) use omnidirectional antennas. The field of coverage achieved with directional antennas would be very limited and the advantage of employing such PSSs would be questionable. This requirement is currently in conflict with the large volume of WiMAX hardware available on the market, but the situation in likely to change, as a strong trend towards omnidirectional antennas is being noticed.

Proxy-capable subscriber stations work in the same frequency channel as their main BS, and need to perform all operations in their allocated (by BS) transmission times. That includes:

- their own traffic to/from BS;
- receiving transmissions form ISS and retransmitting them to BS;
- receiving transmissions from BS and retransmitting them to ISS;
- maintenance of their own proxy-WiMAX cell.

The PSSs will advertise their capabilities by emulating BS frame structure inside their allocated transmission times, which will allow prospective ISSs to detect them, connect to them, and expedite their traffic. This task may seem highly hardware intensive, as the PSS will need to conduct network maintenance tasks similar to that of the BS. However, there are many simplifications that can be made, taking into account the small expected number of ISSs and their narrow range. Advanced physical transmission control, QoS, and the network control mechanisms can be radically simplified or, in some cases, removed. If only SMM-capable SSs are allowed to connect to proxy stations (and unmodified SSs are prohibited even from connecting to PSSs), the simplifications can be even greater. The exact extent of simplification is currently a subject of our research.

Connecting ISSs can choose their PSSs according to a number of criteria and, possibly, use multiple simultaneous links to several PSSs to enable dynamic soft handover.

6.1. Support-mesh mode – variant 1

This variant is fully compatible with existing unmodified WiMAX systems. Because of this assumption, the PSS makes all requests to the BS, to accommodate its own needs as well as those of the ISSs served by it.

Because the ISSs are not visible to the BS, the PSS is expected to:

- authenticate the connecting ISSs and grant them resources using its own authentication and access control mechanisms;
- obtain bandwidth from the BS, necessary to: service its own traffic, communicate with its connected ISSs, and retransmit the ISS traffic to and from the BS;
- handle the PSS-ISS communication and correctly retransmit unidirectional traffic between the ISS and the BS.

Detailed aspects of the authentication and access control mechanisms are beyond the scope of this paper. Many solutions (strictly local and centralized or distributed) can be employed to address these issues.

The remaining problem is the support for PSS-ISS communication within the constraints of the strictly controlled WiMAX PtMP environment. The WiMAX downlink phase is exclusively controlled by the BS, according to the traffic contracts of the connections and the level of currently buffered traffic awaiting transmission. When the BS does not have traffic to transmit through a particular connection, there is no downlink transmission time allocated for it. In such a case, it is impossible to use the downlink phase of WiMAX for communication with the ISS: it has to be conducted during the uplink phase.

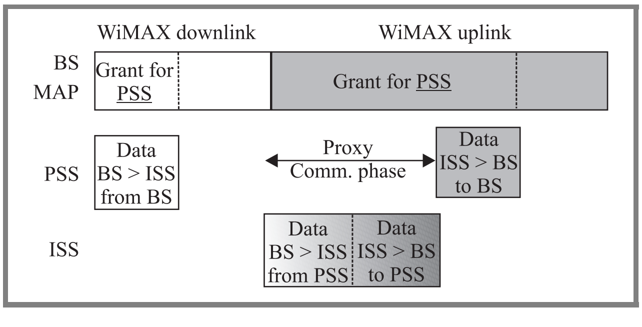


Fig. 7. SMM – variant 1: transmission organization.

Transmission time in the uplink phase is also granted by the BS, according to the traffic contracts of the SSs, but there are no optimizations made by the BS. That makes it possible for the PSS to obtain the necessary transmission time.

In variant 1 of SMM, the PS reserves uplink time to handle (Fig. 7):

- its own uplink traffic to the BS;
- retransmissions of the ISS uplink traffic from the PSS to the BS;
- ISS uplink traffic to the PSS;
- PSS downlink traffic to the ISS.

In variant 1 of SMM presented above, the only stations that require software modification and are aware of the SMM operation are the indirectly connected subscriber stations (ISSs) and their respective proxy subscriber stations (PSSs). The base station and other subscriber stations are neither modified nor aware of the SMM operation.

6.2. Support-mesh mode – variant 2

For variant 2 of SMM, we extend the BS functionality to make it aware of the indirectly connected subscriber stations. While this requires a modification of BS software, it also offers significant advantages in terms of efficiency and system control (Fig. 8).

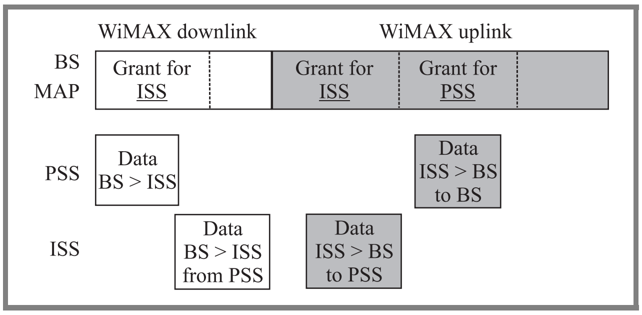


Fig. 8. SMM – variant 2: transmission organization.

In this case, ISSs communicate with the BS, with the proxy stations acting as repeaters, by retransmitting both control and user traffic between the ISS and the BS. This allows the ISSs to participate in the standard network entry procedure, establish WiMAX connections, and issue their own bandwidth requests. This approach allows us to retain the security and management capabilities of the classic PtMP WiMAX system, as the ISS stations are fully recognized by the BS. It also makes it possible to utilize both the downlink and uplink phases for communication with the ISSs, resulting in their much more balanced usage. This balance improves system reliability and can significantly improve the efficiency of WiMAX hardware implementations, which do not support a dynamic change of WiMAX frame division (a dynamic change of the uplink and downlink phase duration ratio).

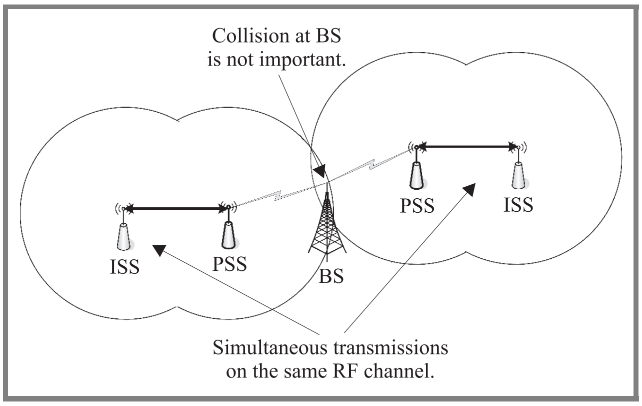


Fig. 9. Use of SDMA in variant 2 of SMM.

In this variant, there is also a possibility of utilizing spatial division multiple access (SDMA) to allow the utilization of a single frequency channel by many PSSs and ISSs at the same time (Fig. 9). This can be done due to the BS's complete knowledge of its network structure (including ISSs), by utilizing an additional raging phase to gather more information about the spatial separation of nodes. Ranging is the process of measuring the quality of the link between the BS and the SS, conducted during the time period especially reserved for this purpose in each WiMAX frame. This time period can be used by the BS to locate those SSs that are unable to interfere with each other, by conducting measurements of link presence and quality between pairs or groups of SSs. This task could be also accomplished by passive measurements by the SSs of the normal traffic generated by other stations, but the use of ranging mechanisms makes the process independent of station activity and their physical transmission characteristics, such as dynamically adjusted power and modulation.

7. Support-mesh mode – simulation results

To verify the usefulness of our solution we carried out a series of coverage tests, using a modified, WiMAX NLOS-compatible propagation model. As stated before, the classical propagation models are inefficient in detecting anomalies as small as the observed white spots and thus unfit for the task. Thus, we developed a modified propagation model. Its operation is supplemented by a file containing empirical real-world coverage data. That way the model is able to check the presence of small white spots, while still keeping calculation costs within acceptable limits.

We considered two urbanized area types:

– terrain *A*: dense urbanized area with 5–6 story buildings, 3.5 km^2: the Gdańsk-Wrzeszcz area;

– terrain *B*: sparse residential area with 12 story buildings (building length 30–300 m) and a limited number of smaller buildings and objects, 4 km^2: the Gdańsk-Zaspa area.

In these areas we randomly distributed 30 SMM-SS in two scenarios:

– scenario 1: 30 stationary SMM-SS on rooftops and building walls;

– scenario 2: 20 stationary SMM-SS on rooftops and building walls, 10 mobile SMM-SS slowly moving at street level.

Below we include the results of coverage modeling as the percentage of the previously uncovered area, which now has been covered by SMM-capable subscriber stations:

– area *A*, scenario 1: 80%;

– area *A*, scenario 2: 85%;

– area *B*, scenario 1: 95%;

– area *B*, scenario 2: 100%.

It is evident from these data that the support-mesh mode subscriber stations can provide a significantly better coverage in urbanized areas than a pure PtMP setup. It advantage is in the distributed layout of PSSs, which cover the target area from varied angles. The result is especially promising in the case of the relatively small number of large obstacles (area *B*). These results convinced us that WiMAX SMM could efficiently solve the coverage issues. Thus, we built simulation models for both of its variants. We employed a relatively simple WiMAX simulation model covering in detail only the ISO-OSI layer 2 network mechanisms, with very simplified layer 1 modeling. We are currently developing a more thorough simulation tool.

Our experiments confirmed the basic operation principles of the WiMAX support-mesh mode, but they also uncovered some limitations. Variant 1 of SMM, from the beginning designed as temporary, low-efficiency, emergency solution proved operable for up to 2 layers of proxies. Additional layers refuse to function due to the strict timing constraints of the system. Moreover, there is an additional, up to 8% per layer, performance degradation resulting from the repeated retransmission of data. This degradation applies only to indirectly connected stations and does not impact SSs in the classical PtMP setup.

Variant 2 provides better service and has been tested for up to 5 layers of proxy stations, at which point it was till operable. The loss of performance for indirectly connected stations was about 5% per proxy level, but in this case the BS could compensate for the loss and provide the ISS with a respectively higher bandwidth.

In both cases (variants 1 and 2) there is also a need for retransmission of data by PSSs, which is the main source of performance degradation, halving the bandwidth with each proxy level. Because of that, the performance of variant 2 with SDMA mechanisms is in most cases drastically better, as it allows the stations to conduct multiple retransmissions simultaneously. The exact results depend on station locations and terrain layout.

8. Conclusions

Based on our theoretical research and practical experiments, we observed a detrimental effect present in wireless networks based on WiMAX technology, resulting in small coverage holes in areas of otherwise good coverage. Such white spots are difficult to predict, even with the use of deterministic propagation models, which are among the most popular wireless network design support tools used today. They can lead to lower than expected service levels, requiring repositioning or installation of additional hardware and can be especially harmful for mobile users who will experience periodic losses of connectivity.

Despite the fact that the same coverage problem would affect mesh nodes (especially the mobile ones) potentially leading to topology instability, it is possible to design network control mechanisms to counter the effect. That would allow WiMAX mesh networks to provide continuous coverage of a given area eliminating the holes. The problem would be very difficult to fix in the classical BS-based architecture without large additional hardware costs.

We have proposed WiMAX mesh networks as a viable method of dealing with coverage difficulties in metropolitan areas. This kind of solution provides additional crucial advantages, e.g., well-scaling high network capacity, high reliability based on multiple redundancy, low cost of deployment.

As a fully functional mesh mode requires a significant number of additional, advanced mechanisms, and is still in a very early stage of research and development, we developed a hybrid PtMP-mesh solution dubbed the support-mesh mode. It provides the subscriber stations with proxy capabilities, thus bringing about many of the mesh mode advantages (including coverage and reliability) with considerably simpler mechanisms. Both variants of our solution require only limited modifications to subscriber station software, and only the more advanced variant 2 requires a modification of base station software.

Operation of both variants of our solution have been studied by simulation and yielded satisfactory results. Variant 1 should be treated as a temporary/emergency solution for currently available systems, while variant 2 can be considered for further development and incorporation as an option in the upcoming versions of WiMAX hardware.

References

[1] “IEEE Standard for Local and Metropolitan Area Networks”. Part 16: “Air Interface for Fixed Broadband Wireless Access Systems”, IEEE 802.16-2004.

[2] T. K. Sarkar *et al.*, "A survey of various propagation models for mobile communication", *IEEE Anten. Propag. Mag.*, vol. 45, no. 3, 2003.

[3] J.-P. Rossi and Y. Gabillet, "A mixed ray launching/tracing method for full 3-D UHF propagation modeling and comparison with wideband measurements", *IEEE Trans. Anten. Propag.*, vol. 50, iss. 4, pp. 517–523, 2002.

[4] ATDI, "The WiMax technologies with ICS telecom: fixed, nomadic and mobile!", 2007, http://www.adti.com/

[5] R. Olejnik, *Metody projektowania sieci WLAN a metoda projektowania parametrycznego*. Warszawa: WKŁ, 2007, t. 1, s. 347–356 (in Polish).

[6] K. Gierłowski and K. Nowicki, *Pilotażowa instalacja sieci bezprzewodowej standardu WiMax na Politechnice Gdańskiej*. Red. L. Kiełtyka. Warszawa: DIFIN, 2006 (in Polish).

[7] WiMAX Forum, "WiMAX Forum whitepapers", 2007, http://www.wimaxforum.org

[8] Alvarion, "BreezeMAX whitepapers", 2006, http://www.alvarion.com

[9] "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements" – Part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE 802.11-1999, Edition (ISO/IEC 8802-11:1999).

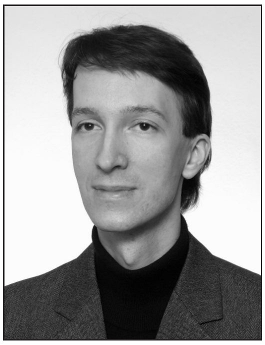
[10] M. T. Hewitt, "Surface feature data for propagation modeling", IEEE Colloquium on Terrain Modelling and Ground Cover Data for Propagation Studies, pp. 4/1–4/6, 1993.

[11] P. Matusz, J. Woźniak, K. Gierłowski, T. Gierszewski, T. Klajbor, P. Machań, T. Mrugalski, and R. Wielicki, "Heterogeneous wireless networks – selected operational aspects", *Telecommun. Rev. Telecommun. News*, no. 8-9, pp. 284–295, 2005.

[12] "Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services", IEEE P802.21/D05.00, Apr. 2007.

[13] P. Machań, A. Jordan, A. Wieczorkowska, and J. Woźniak, "IEEE 802.21 standard – mobility support mechanisms", *Telecommun. Rev. Telecommun. News*, no. 7, pp. 211–216, 2007.

[14] K. Gierlowski and K. Nowicki, "Wireless networks as an infrastructure for mission-critical business applications", in *Network-Based Information Systems: First International Conference, NBiS 2007, Regensburg, Germany, September 2007, Proceedings*, T. Enokido, L. Barolli, and M. Takizawa, Eds., *LNCS*. Berlin-Heidelberg: Springer, 2007, vol. 4658, pp. 49–58.



Krzysztof Gierłowski received his M.Sc. degree in electronics and telecommunications from the Faculty of Electronics, Gdańsk University of Technology (GUT), Poland, in 2002. He has authored or co-authored over 30 scientific papers and designed many production-grade network systems. His scientific and research interests include wireless local and metropolitan area networks, complex wireless network architectures, e-learning systems, network/application security and distributed computing.

e-mail: gierk@eti.pg.gda.pl
Gdańsk University of Technology
G. Narutowicza st 11/12
80-952 Gdańsk, Poland

Józef Woźniak is a full Professor in the Faculty of Electronics, Telecommunications and Computer Science at Gdańsk University of Technology, Poland. He received his Ph.D. and D.Sc. degrees in telecommunications from Gdańsk University of Technology in 1976 and 1991, respectively. He is author or co-author of more than 200 journal and conference papers. He has also co-authored 4 books on data communications, computer networks and communication protocols. In the past he participated in research and teaching activities at Politecnico di Milano, Vrije Universiteit Brussel, and Aalborg University, Denmark. In 2007 he was Visiting Erskine Fellow at the Canterbury University in Christchurch, New Zealand. He has served in technical committees of numerous national and international conferences, chairing or co-chairing several of them. He is a Member of IEEE and IFIP, being the Vice Chair of the WG 6.8 (Wireless Communications Group) IFIP TC6 and the Chair of an IEEE Computer Society Chapter (at Gdańsk University of Technology). His current research interests include modeling and performance evaluation of communication systems with the special interest in wireless and mobile networks.

e-mail: jowoz@eti.pg.gda.pl
Gdańsk University of Technology
G. Narutowicza st 11/12
80-952 Gdańsk, Poland

Krzysztof Nowicki received his M.Sc. and Ph.D. degrees in electronics and telecommunications from the Faculty of Electronics, Gdańsk University of Technology (GUT), Poland, in 1979 and 1988, respectively. He is the author or co-author of more than 100 scientific papers and author and co-author of four books, e.g., "LAN, MAN, WAN – Computer Networks and Communication Protocols" (1998), "Wired and Wireless LANs" (2002) (both books were awarded the Ministry of National Education Prize, in 1999 and 2003, respectively), "Protocol IPv6" (2003) and "Ethernet-Networks" (2006). His scientific and research interests include network architectures, analysis of communication systems, network security problems, modeling and performance analysis of cable and wireless communication systems, analysis and design of protocols for high speed LANs.

e-mail: know@eti.pg.gda.pl
Gdańsk University of Technology
G. Narutowicza st 11/12
80-952 Gdańsk, Poland

Paper

On a practical approach to low-cost ad hoc wireless networking

Paweł Gburzyński and Włodzimierz Olesiński

Abstract—Although simple wireless communication involving nodes built of microcontrollers and radio devices from the low end of the price spectrum is quite popular these days, one seldom hears about serious wireless networks built from such devices. Most of the commercially available nodes for ad hoc networking (somewhat inappropriately called "motes") are in fact quite serious computers with megabytes of RAM and rather extravagant resource demands. We show how one can build practical ad hoc networks using the smallest and cheapest devices available today. In our networks, such devices are capable of sustaining swarm-intelligent sophisticated routing while offering enough processing power to cater to complex applications involving distributed sensing and monitoring.

Keywords— *ad hoc wireless networks, sensor networks, operating systems, reactive systems, specification, simulation.*

1. Introduction

The vast number of academic contributions to ad hoc wireless networking have left a surprisingly tiny footprint in the practical world. For once, the industry is not much smarter, although for a different reason. While the primary problem with academic research, not only in this particular area, is its excessive separation from mundane and academically uninteresting aspects of reality, the industry appears to suffer from its inherent inability to think small and holistic. The net outcome is in fact the same in both cases: the popular and acknowledged routing schemes, as well as programmer-friendly application development systems, require a significant amount of computing power and are not suitable for small and inexpensive devices. As an example of the latter, consider a key-chain car opener. A networking "node" of this kind is typically built around a low-power microcontroller with $\sim$ 1 KB of RAM driving a simple transceiver. The combined cost of the two components is usually below $5. While it is not a big challenge to implement within this framework a functionally simple broadcaster of short packets, it is quite another issue to turn this device into a collaborating node of a serious ad hoc wireless system.

The plethora of popular ad hoc routing schemes proposed and analyzed in the literature [6, 22, 25, 27, 31, 32, 33, 38], addresses devices with a somewhat larger resource base. This is because those schemes require the nodes to store and analyze a non-trivial amount of information to carry out their duties. Moreover, none of them provides for "graceful downscaling," whereby a node with a smaller than "recommended" amount of RAM can still fulfill its obligation to the network, perhaps at a reduced level. With such systems, hardware resources must be overdesigned (i.e., wasted in typical scenarios), as an overrun leads to a functional breakdown.

The most popular commercial scheme originally intended for building ad hoc networks is Bluetooth. Its two fundamental problems are:

– a large footprint and, consequently, non-trivial cost and power requirements;

– arcane connection discovery and maintenance options, which render true ad hoc networking cumbersome.

Even though some attempts are still being made to build actual ad hoc networks based on Bluetooth [18], it is commonly agreed that the role of this technology is reduced to creating small personal-area hubs. ZigBee® (based on ad hoc on-demand distance vector (AODV) [34]) comes closer; however, despite the tremendous industrial push, it fails to catch on. The reason, we believe, is its isolation from the wider context of application development issues combined with the limited flexibility of AODV as a routing scheme.

If there is a place in the realm of low-end microcontrollers where the adjective "ad hoc" is well applicable, it is to software development. Typically, the software (firmware) designed for one particular project is viewed as a "one-night stand", and its re-usability, modularity, and exchangeability are not deemed interesting. This is because convenient, modular, and self-documentable programming techniques based on concepts like multi-threading, event handling, synchronization, object-orientedness are considered too costly in terms of resource requirements (mostly RAM) to be of merit in programming the smallest microcontrollers. Even TinyOS [26], which is the most popular operating system for networked microcontrollers, has many shortcomings in these areas. Moreover, its evolution (as it usually happens with systems driven by large communities and consortia), leans towards larger and larger devices.

Serious efforts to introduce an order and methodology into programming small devices often meet with skepticism and shrugs. The common misconception is that one will not have to wait for long before those devices disappear and become superseded by larger ones, capable of running Linux or Windows®. This is not true. Despite the ever decreasing cost of microcontrollers, we see absolutely no reduction in the demand for the ones from the lowest end of the spectrum. On the contrary: their low cost and low power requirements enable new applications and trigger even more demand.

One of our claims is that the only way to harness trivially small microcontrolled devices to performing complex communal tasks, amounting to effective and efficient ad hoc networking, is to follow a holistic approach to organizing their software. This is in fact the primary problem with the industry: their approach to solutions is *layered*, in the sense that separable components of the traget product are viewed by them as isolated subproblems to be attacked by different teams equipped with different tools and driven by different objectives. Consider a node of a wireless ad hoc network with the components listed in Fig. 1. In a typical production cycle, each of those components is viewed as an end in itself. From our perspective as academics, we tend to focus on the *protocols* component, forgetting that it is merely a fragment of something that may ever be useful. As it happens, the most mundane fragment of the whole picture, i.e., the hardware, directly determines the viability of the entire project as a commercial idea.

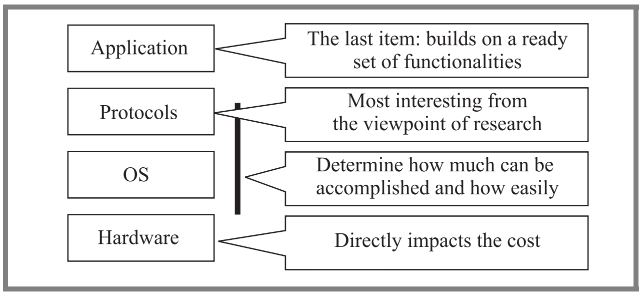

Application
The last item: builds on a ready set of functionalities
Protocols
Most interesting from the viewpoint of research
OS
Determine how much can be accomplished and how easily
Hardware
Directly impacts the cost

Fig. 1. Product “layers” of a wireless node.

A natural industrial reflex when it comes to protocols is standardization. While some of its benefits are unquestionable, e.g., the interoperability of diverse equipment, the main thrust of the standardization efforts that we in fact see in the area of low-cost wireless networking is aimed at the “soft” parts of the protocols (above the physical layer). Note that this has brought us ZigBee, which tries to impose on us ready network-layer paradigms for implementing operations as delicate as forwarding within unknown networks catering to unknown applications. Notably, in the very area where the standardization would be truly and indisputably useful, i.e., in the physical layer, we saw a complete lack of interest from the manufacturers, at least in the pre-ZigBee era. A stress on interoperability at that stage would have been considerably more beneficial to the community of users and developers of ad hoc wireless communication solutions. These days, it only happens in the context of ZigBee, which many of the manufacturers come to perceive as a curse of their membership in the consortium. Thus, they usually provide ways to bypass the ZigBee stack and enable various clumsy (but at least partially feasible) ad hoc networking scenarios, e.g., hubs or limited multi-hopping.

By their very nature, standards devised in isolation from the view of applications are bound to result in oversized footprints. This is because:

- They have to anticipate many circumstances that will occur marginally, if ever, in any particular application.
- They are devised by committees consisting of people with conflicting ideas and agendas, and tend to accommodate a little bit of everything – to satisfy all members.
- Their designers focus on functionality rather than feasibility: the lack of a reference point (application, hardware) makes it difficult to see the complexity of implementation.

In this paper, we outline our comprehensive platform for rapidly building wireless *praxes*, i.e., low-cost applications based on ad hoc networking. This platform comprises an operating system, a flexible, layer-less, and auto-scalable ad hoc forwarding scheme, and an emulator for testing the praxes in a high-fidelity virtual environment. We show how one can build well structured multithreaded programs operating within a trivially small amount of RAM and organize them into authentic ad hoc wireless applications.

2. The operating system

The foundation of our development platform is PicOS: a tiny operating system for small-footprint microcontrolled devices executing reactive applications [1, 39].

2.1. A historical perspective

The ideas that have found their way into PicOS originated as early as 1986. About that time, many published performance studies of carrier sense multiple access/collision detection (CSMA/CD)-based networks (like Ethernet) had been subjected to heavy criticism from the more practically inclined members of the community – for their irrelevance and overly pessimistic conclusions. The culprit, or rather culprits, were identified among the popular collection of models (both analytical and simulation), whose cavalier application to describing poorly understood and crudely approximated phenomena had resulted in worthless numbers and exaggerated blanket claims [5]. Our own studies of low-level protocols for local-area networks, on which we embarked at that time [8–12, 14–16], were aimed at devising novel solutions, as well as dispelling myths surrounding the old ones. Owing to the fact that exact analytical models of the interesting systems were (and still are) nowhere in sight, the performance evaluation component of our work relied heavily on simulation. To that end, we developed a detailed network simulator, called LANSF (local area network simulation facility) and its successor SMURPH (system for modeling unslotted real-time phenomena) [13, 19], which carefully accounted for all the relevant physical phenomena affecting the correctness and performance of low-level protocols, e.g., the finite propagation speed of signals, race conditions, imperfect synchronization of clocks, variable event latency incurred by realistic hardware. In addition to numerous tools facilitating performance studies, SMURPH was also equipped with instruments for dynamic conformance testing [3].

At some point we couldn't help noticing that the close-to-implementation appearance of SMURPH models went beyond mere simulation: the same paradigm could be used for implementing certain types of real-life applications. The first outcome of that observation was an extension of SMURPH into a programming platform for building distributed controllers of physical equipment represented by collections of sensors and actuators. Under its new name, SIDE (sensors in a distributed environment), the package encompassed the old simulator augmented by tools for interfacing its programs to real-life objects [20, 21].

A natural next step was to build a complete and self-sustained executable platform (i.e., an operating system) based entirely on SMURPH. It was directly inspired by a practical project whose objective was to develop a low-cost intelligent badge equipped with a low-bandwidth, short-range, wireless transceiver allowing it to communicate with neighbors. As most of the complexity of the device's behavior was in the communication protocol, its model was implemented and verified in SMURPH. The source code of the model, along with its plain-language description, was then sent to the manufacturer for a physical implementation. Some time later, the manufacturer sent us back their prototype microprogram for "optical" conformance assessment. Striving to fit the program's resources into as little memory (RAM) as possible, the implementer organized it as an extremely messy single thread for the bare CPU. The program tried to approximate the behavior of our high-level multi-threaded model via an unintelligible combination of flags, hardware timers and counters. Its original, clear, and self-documenting structure, consisting of a handful of simple threads presented as finite state machines, had completely disappeared in the process of implementation. While struggling to comprehend the implementation, we designed a tiny operating system providing for an easy, natural and rigorous implementation of SMURPH models on microcontrollers. Even to our surprise, we were able to actually *reduce* the RAM requirements of the re-programmed application. Needless to say, the implementation became clean and clear: its verification was immediate.

2.2. PicOS threads

The most serious problem with implementing non-trivial, structured, multitasking software on microcontrollers with limited RAM is minimizing the amount of memory resources needed to describe a thread. While the basic record of a thread in the kernel of an embedded system can be contained in a handful of simple variables (status, code pointer, data pointer, one or two links), the most troublesome component of the thread footprint is its stack, which must be preallocated to every thread in a safe amount sufficient for its maximum possible need. In addition to providing room for the automatic variables used by thread functions, including the implicit ones (like return addresses), the stack is an important part of the thread's context. When the thread is preempted, the stack preserves the snapshot of its trace, which will make it possible to resume the thread later, in a consistent and transparent manner.

At first sight, it might seem that microcontrollers with very small amount of RAM are condemned to running threadless systems. For example, in TinyOS [24, 26], the issue of limited stack space has been addressed in a radical manner – by avoiding multithreading altogether. Essentially, TinyOS defines two types of activities: event handlers (corresponding to interrupt service routines and callbacks) and the so-called *tasks*, which are simply chunks of code that cannot be preempted by (and thus cannot dynamically coexist with) other tasks.

One way to strike a compromise between the complete lack of threads on the one hand, and overtaxing the tiny amount of RAM with partitioned and fragmented stack space on the other, may be to reduce the flexibility of threads regarding the circumstances under which they can be preempted (i.e., lose the CPU). The idea is to create an environment where the thread is forced to relinquish its stack before preemption. That would restrict the preemption opportunities to a collection of *checkpoints* of which the thread would be aware. By stimulating a structured organization of those checkpoints, we could try to

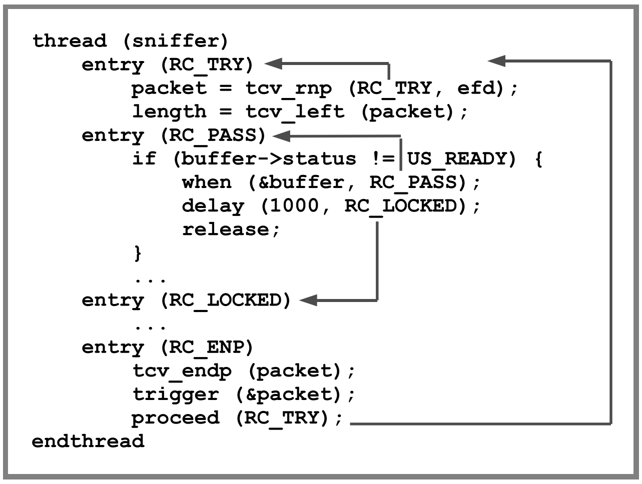
– avoid locking the CPU at a single thread for an extensive amount of time;

– turn them into natural and useful elements of the thread's specification, e.g., enhancing its clarity and reducing the complexity of its structure.

These ideas lie at the heart of PicOS's concept of threads, which are structured like finite state machines (FSM) and exhibit the dynamics of coroutines [4, 7] with multiple entry points and implicit control transfer.

For illustration, consider the sample thread code shown in Fig. 2. This is in fact a C function: any exotic keywords or constructs are straightforward macros handled by the standard C preprocessor. The states are marked by the `entry` statements. Whenever a thread is assigned the CPU, its code function is invoked in the *current state*, i.e., at one specific entry point.

State boundaries represent the checkpoints at which a thread can be preempted and resumed. The way it works is that a thread can only lose the CPU when it explicitly relinquishes control at the boundary of its current state. In particular, this happens when the thread executes `release`, as within state `RC_PASS` in Fig. 2. This has the effect of returning the CPU to the scheduler, which is then free to allocate it to another thread. A thread receiving the CPU will always find itself at the entry point to one of its states.



```
thread (sniffer)
    entry (RC_TRY)
        packet = tcv_rnp (RC_TRY, efd);
        length = tcv_left (packet);
    entry (RC_PASS)
        if (buffer->status != US_READY) {
            when (&buffer, RC_PASS);
            delay (1000, RC_LOCKED);
            release;
        }
        ...
    entry (RC_LOCKED)
        ...
    entry (RC_ENP)
        tcv_endp (packet);
        trigger (&packet);
        proceed (RC_TRY);
endthread
```

Fig. 2. A sample thread code in PicOS.

Typically, before executing `release`, a thread issues a number of *wait requests* specifying one or more conditions (events) to resume it in the future. Then, the effect of `release` is to block the thread until at least one of the conditions is fulfilled. If multiple events are awaited by the thread, the earliest of them will wake it up. Once that happens, all the pending wait requests are erased: the thread has to specify them from scratch at every wake-up. As a wait request, besides the condition, specifies the state to be assumed by the awakened thread, the collection of wait requests issued by a thread in every state describes the options for its transition function from that state.

In state `RC_PASS` (Fig. 2), if the `if` condition holds, the thread issues two wait requests: one with `when` and the other with `delay`. With `when`, the thread declares that it wants to be resumed in state `RC_PASS` upon the occurrence of an event represented by the address of a data object (`buffer`). Such events can be signaled with `trigger`, as illustrated in state `RC_ENP`. The `delay` operation sets up an alarm clock for the prescribed number of milliseconds (1000). The event waking the process up will be triggered when the alarm clock goes off.

A somewhat less obvious case of a wait request is operation `proceed` (at the end of state `RC_ENP`), which implements an explicit transition (a kind of "goto") to the indicated state. It can be thought of as a zero delay request (indicating the target state) followed by `release`. Thus, the transition involves releasing the CPU and re-acquiring it again, which gives other threads a chance to execute in the meantime.

The above paradigm of organizing tasks in PicOS has proved very friendly, versatile, and useful for describing the kinds of applications typical of embedded systems, i.e., reactive ones [1, 39]. The FSM layout of the praxis comes for free and can be mechanically transformed, e.g., into a statechart [17, 23], for easy comprehension or verification. Owing to the fact that a blocked thread needs no stack space, all threads in the system can share the same single global stack. The programmer-controlled preemptibility grain practically eliminates all synchronization problems haunting traditional multi-threaded applications.

2.3. The footprint

So far, PicOS has been implemented on the MSP430 microcontroller family and on eCOG1 from Cyan Technology. A port to ARM7 is under way. The size of the thread control block (TCB) needed to describe a single PicOS thread is adjustable by a configuration parameter, depending on the number of events E that a single thread may want to await simultaneously. The standard setting of this number is 3, which is sufficient for all our present applications and protocols. The TCB size in bytes is equal to $8+4E$, which yields 20 bytes for $E = 3$. The number of threads in the system (the degree of multiprogramming) has no impact on the required stack size, which is solely determined by the maximum configuration of nested function calls. As automatic variables are not very useful for threads (they do not survive state transitions and are thus discouraged), the stack has no tendency to run away. 96 bytes of stack size is practically always sufficient. In many cases, this number can be reduced by half.

2.4. System calls

In a traditional operating system, a thread may become blocked implicitly when it executes a system call that cannot complete immediately. To make this work with PicOS threads, which can only be blocked at state boundaries, potentially blocking system calls must incorporate a mechanism involving a combination of a wait request and `release`. For illustration, see the first statement in state `RC_TRY` (Fig. 2). Function `tcv_rnp` (belonging to VNETI – see Subsection 2.5) is called to receive a packet from a network session represented by descriptor `efd`. It may return immediately (if a packet is already available in the buffer), or block (if the packet is yet to arrive). In the latter case, the system call will block the thread and resume it in the indicated state when it makes sense to re-execute `tcv_rnp`, i.e., upon a packet reception.

Essentially, there are two categories of system calls in PicOS that may involve blocking. The first one, like `tcv_rnp`, may get into a situation when something needed by the program is not immediately available. Then, the event waking up the process will indicate a new acquisition opportunity: the failed operation has to be re-done. The second scenario involves a delayed action that must be internally completed by the system call before the process becomes runnable again. In such a case, the event indicates that the process may continue: it does not have to re-execute the system call. To keep the situation clear, the syntax of system call functions unambiguously determines which is the case. Namely, for the first type of calls, the state argument is first on the argument list, while it is last for the second type. Incidentally, all system calls that can ever block take more than one argument.

2.5. The versatile network interface (VNETI)

The interface of a PicOS application (praxis) to the outside world is governed by a powerful module called VNETI. VNETI offers to the praxis a simple and orthogonal collection of application programming interface (API), independent of the underlying implementation of networking. To avoid the protocol layering problems haunting small-footprint solutions, VNETI is completely layer-less and its semi-complete generic functionality is redefined by plug-ins.

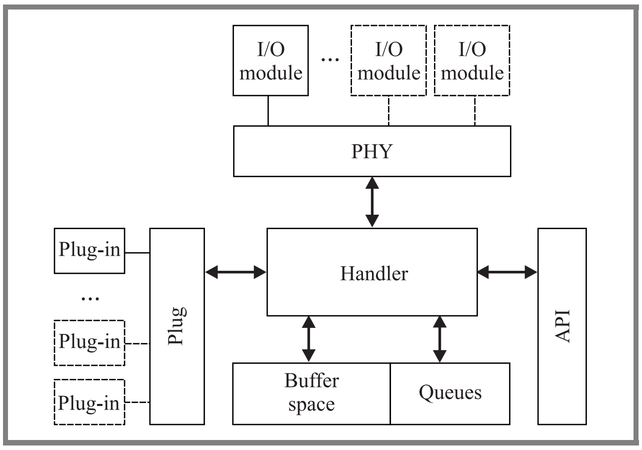


Fig. 3. The structure of VNETI.

The structure of VNETI is shown in Fig. 3. Standard sets of interfaces are provided for attaching drivers of physical communication modules (PHY), as well the plug-ins representing the *protocols* configured into the system. The API available to the praxis consists of a fixed set of operations that are independent of the configured assortment of plug-ins or the physical modules.

```
typedef struct {
  int (*tcv_ope) (int phid, int fd, va_list ap);
  int (*tcv_clo) (int phid, int fd);
  int (*tcv_rcv) (int phid, address buf, int len, int *ses,
                                             tcvadp_t *frm);
  int (*tcv_frm) (address packet, int phid, tcvadp_t *frm);
  int (*tcv_out) (address packet);
  int (*tcv_xmt) (address packet);
  int (*tcv_tmt) (address packet);
  int tcv_info;
} tcvplug_t;
```

Fig. 4. Plug-in interface.

A plug-in is described by a numerical identifier and a set of operations, as shown in Fig. 4. Generally, those operations intercept various requests coming from the praxis, as well as the packets, as they make their passes through the buffer storage of VNETI (Fig. 3). For example, when the praxis opens a communication session, by executing the `tcv_open` function of VNETI, it specifies the identity of the plug-in to be responsible for the session. Thus, VNETI will invoke the *ope* (`tcv_ope`) function provided by the plug-in, to carry out any specific administrative operations required to set up the session. Now, consider a packet being received from the network. The PHY module receiving the packet presents it to VNETI by invoking a standard interface function. In response, VNETI will in turn apply to the packet the *rcv* functions (`tcv_rcv` in Fig. 4) of all the configured plug-ins. Based on the packet content and the identity of the PHY module, the function may decide to *claim* the packet (by returning a special value) and assign it to a particular session (the return argument `ses`), which typically corresponds to one of the active session being handled by the praxis and associated with the plug-in. The last argument of `tcv_rcv` returns the pointers to the packet's low-level header and trailer, which will be discarded when the claimed packet is deposited by VNETI in its buffer space.

The set od operations available to plug-ins involve queue manipulations, cloning packets, inserting special packets, and assigning to them the so-called *disposition codes* representing various processing stages. Any sophisticated protocol (e.g., TCP/IP) can be implemented within this paradigm. Its underlying premise is to treat all packets "holistically" with no regard for any assumed *layers* of their processing.

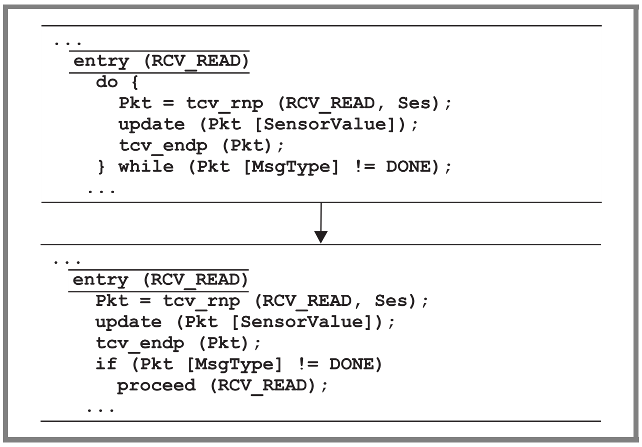
2.6. Real time considerations

One problem resulting from the limited preemptibility of threads is its potentially detrimental impact on the real-time behavior of the embedded application [2, 29]. This is because the maximum rescheduling time for any thread (regardless of its priority [37]) will include the maximum non-preemptibility interval for any other thread in the system.

PicOS scheduler admits several options, which can be selected at the time the system is compiled into a praxis. The most naive (and also the most popular) of those options implements a fixed priority (non-preemptive) scheme, whereby the threads are sorted in the decreasing order of their importance. Whenever a thread releases the CPU, the scheduler assigns it to the first thread that is not waiting for any event. This way, when multiple threads are ready to run, the one closer to the front of the list will win the CPU.

In most cases, this trivial approach to scheduling is quite adequate to fulfill the real time requirements of the application, especially if those requirements are soft. This is because reactive applications tend to do little computations (are not CPU bound) and focus on processing events, which actions typically take a small amount of time. Notably, the truly critical actions (like extracting data from time-constrained peripheral equipment) are carried out in interrupts, which are not subjected to the kind of postponement exercised by threads. Consequently, the limited preemptibility of the latter does not impair the rate at which external events can be formally absorbed by the PicOS application. The way interrupt service routines are organized in PicOS renders them interruptible: consequently an interrupt service routine can be preempted by another interrupt service routine, according to the pre-declared criteria of importance. To avoid inflating the bound on the maximum stack size, this feature is optional: it can be turned on selectively, on a per-interrupt basis, to cater to the hard real-time requirements of critical peripherals.

It is a good practice to organize PicOS threads in such a way as to keep the maximum execution time of a single state reasonably small. Note that the granularity of thread states is under the programmer's control. There is virtually no penalty for introducing extra states with the purpose of bringing down the maximum duration of a CPU burst exhibited by the thread. In many circumstances, in addition to improving the real-time behavior of the entire application, this approach enhances the clarity, self-documentability, and re-usability of the thread code.



```
...
 entry (RCV_READ)
   do {
     Pkt = tcv_rnp (RCV_READ, Ses);
     update (Pkt [SensorValue]);
     tcv_endp (Pkt);
   } while (Pkt [MsgType] != DONE);
  ...
```

```
...
 entry (RCV_READ)
   Pkt = tcv_rnp (RCV_READ, Ses);
   update (Pkt [SensorValue]);
   tcv_endp (Pkt);
   if (Pkt [MsgType] != DONE)
     proceed (RCV_READ);
  ...
```

Fig. 5. Avoiding non-trivial loops in PicOS threads.

It is also recommended to avoid non-trivial loops that do not cross state boundaries. Such loops can be always converted as shown in Fig. 5, i.e., by starting the loop at a state boundary and closing it with `proceed`, which has the effect of enabling preemptibility at every turn. Although `proceed` has the appearance of "goto," the operation involves an *actual* state transition, i.e., the thread function is exited and re-entered via the scheduler, which is thus allowed to interleave other threads with the loop. This way, any higher priority threads will be able to claim their share of the CPU in between the loop turns.

All internal functions (system calls) of PicOS have been programmed with consistent adherence to the principle of simplicity and orthogonality. This also applies to the internals of VNETI. Every function implements a loop-less action, whose execution time is approximately constant. This way, the timing of code referencing such functions is easy to estimate within a very narrow uncertainty margin. Consequently, it is possible to carry out meaningful real-time assessments, including hard real-time guarantees, by estimating the execution time of thread states. The latter can be often accomplished by a purely mechanical analysis of the compiler output, i.e., the tally of the CPU cycles in a loop-less sequence of machine instructions.

3. Communication

Most routing protocols for ad hoc wireless networks, as described in the literature, assume point-to-point communication, whereby each node forwarding the packet on its way to the destination sends it to a specific neighbor. Regardless of whether the scheme is proactive [6, 33] or reactive [22, 25, 27, 31, 32, 38], its primary objective is to determine the exact sequence of nodes to forward the packets from point *A* to point *B*. Despite the fact that the wireless environment is inherently broadcast, this free feature is rarely exploited during the actual forwarding of session packets, although all protocols necessarily take advantage of broadcast transmissions during various stages of route discovery (e.g., the periodic HELLO messages broadcast by all nodes to announce their presence in the neighborhood). For example, in AODV [34], a node *S* initiating packet exchange with node *D* broadcasts a request to its one-hop neighbors to start the so-called path discovery operation. Based on its current perception of the neighborhood and cached information collected form previous path discoveries, a node receiving such a request may decide to forward it elsewhere, or respond with a path information intended for the initiating node *S*. At the end, a single path between *S* and *D* has been established. A problem arises when the path is broken, because such a mishap effectively demolishes the entire delicate structure. When that happens, a new path discovery operation is essentially started from scratch.

On top of the susceptibility to node failures and disappearance (mobility), this generic approach requires the nodes to store a potentially sizable amount of elaborate routing information, which cannot be made fuzzy. For example, if a node is unable to store the identity of the next-hop neighbor for a particular session, then it will simply not be able to carry out its duties with respect to that session (thus breaking the path). It may confuse the network by offering a service that it is unable to deliver, resulting in stalled path discovery and, ultimately, communication failure.

3.1. Tiny ad hoc routing protocol (TARP): forwarding by re-casting

The idea behind our solution, dubbed TARP, is to embrace fuzziness as a useful feature and take full advantage of the inherently broadcast nature of the wireless medium. Traditional schemes view this nature as a rather serious problem and try to defeat its negative consequences (hidden/exposed terminals) via MAC-level handshakes intended to facilitate point-to-point transmission [28]. TARP, in contrast, turns it into an advantage.

Suppose that node S wants to send a packet to node D. With TARP, S simply transmits (broadcasts) the packet to its neighbors. A neighbor may decide to drop the packet (if it believes that its contribution to the communal forwarding task will not help) or retransmit it. This process continues until the packet reaches the destination D. An important property of this generic scheme (otherwise known as flooding) is that a retransmitted packet is never specifically addressed to a single next-hop neighbor. Needless to say, to make it useful, measures must be taken to limit the number of retransmissions to the minimum at which the desired quality of service is maintained. This part comes as a series of rules that determine when a node receiving a packet should rebroadcast it, as opposed to dropping. Some ideas for such rules are obvious, e.g., discarding duplicates of already seen packets and limiting the maximum number of hops that a packet can travel.

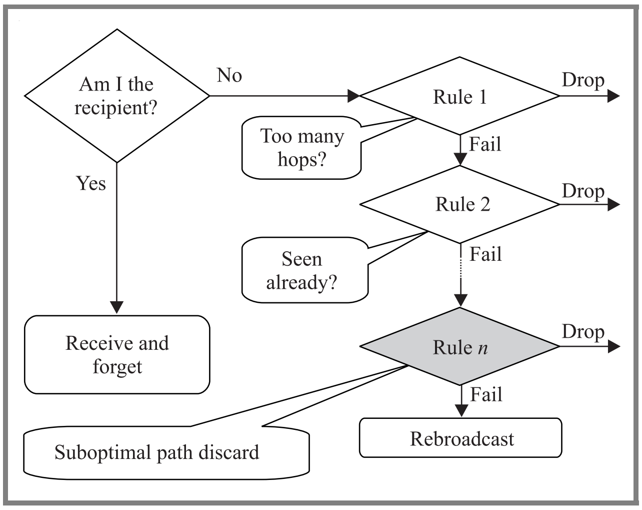


Fig. 6. Application of rules in TARP.

Figure 6 illustrates the way TARP applies its rules to an incoming packet. The important property of *any* rule implanted into TARP is its naturally conservative behavior in the face of incomplete information (uncertainty). This means that a rule that does not know what to do always *fails*, which is to say that the packet will not be dropped on its account. As most of the rules are cache driven, such conservative behavior provides for automatic scalability of the rule to the limitations of its resources. A better-equipped rule may tend to drop more packets and thus avoid polluting the neighborhood with superfluous traffic. The same rule executed on a device with smaller memory may not be as exacting, but if it errs, it does so on the safe side, i.e., it drops no packets that would not have been dropped, had the node been more resourceful. This is something that point-to-point forwarding protocols find difficult to accomplish. To them, a path is just a path: you either know the precise identity of your next hop neighbor, or you know nothing at all. There is no room for fuzziness in that kind of setup.

3.2. The selective packet discard (SPD) rule

The key to the success of our variant of flooding is the most representative rule of TARP, one that brings the paths traveled by forwarded packets down to a narrow (but intentionally fuzzy) stripe of nodes along the shortest route. This rule is named SPD, for suboptimal path discard.

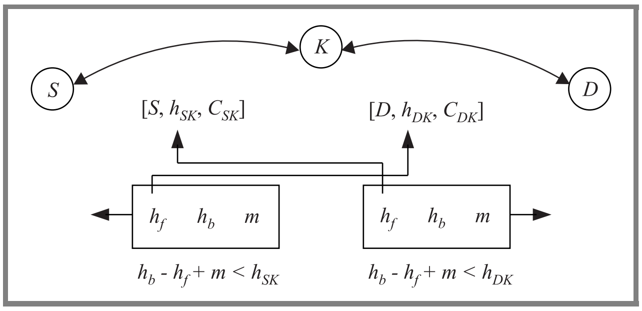


Fig. 7. The rule for selective packet discard.

Consider the three nodes shown in Fig. 7. K is contemplating whether it should re-broadcast an "overheard" packet sent by S and addressed to D. Suppose K knows this information: h_b – the total number of hops traveled by some packet that has recently reached S arriving from D (in the opposite direction); h_f – the number of hops traveled so far by the current packet; h_{DK} – the number of hops separating K from D. If $h_b < h_f + h_{DK}$, K can suspect that the packet can make it to D via a shorter path leading through another node. This is because, apparently, packets can make it from D to S in fewer hops than the combination of whatever the packet has already gone through with the number of hops it still must cover if forwarded via K. Thus, in such a case, K may decide to drop the packet.

The requisite information can be collected from the headers of packets that K overhears as the session goes on. To make it possible, the packet header should carry the current number of hops traveled by the packet as well as the number of hops traveled by the last packet that arrived at the sender from the opposite end. As a duplicate packet is always discarded at the earliest detection, a non-duplicate packet arriving at a destination makes it along the fastest (and usually the shortest) path. Until the network learns about a particular session (understood as a pair of nodes that want to communicate), the forwarding for that session may be overly redundant.

Owing to the inherent imperfections of the ad hoc wireless environment, K should not be too jumpy with negative decisions. TARP uses two adjustable ways to damp the behavior of the SPD rule to account for the uncertainty of knowledge. One of them is the slack parameter m shown in the inequality in Fig. 7. When $m > 0$, the rule will allow the node to forward the packet even though the path that it is able to offer appears to be slightly longer (by up to m hops) than the currently believed shortest path.

Each entry in the SPD cache, in addition to the node identifier and the current estimate for the number of hops, carries a counter (C_{SK} and C_{DK} in Fig. 7), which is incremented by 1 each time the rule succeeds on that entry (i.e., the packet is dropped). When the counter reaches a predefined threshold, the rule will forcibly fail, thus letting the packet explore other routing opportunities.

3.3. Smooth hand-offs

To see how a nonzero slack helps the network cope with node dynamics (mobility, failures), consider the scenario shown in Fig. 8. Packets traveling between U and V are forwarded within the clouded fragment of the network. Suppose that the arrows represent neighborhoods. In a steady state, the path A–B–C (of length 2) is the shortest route through the cloud.

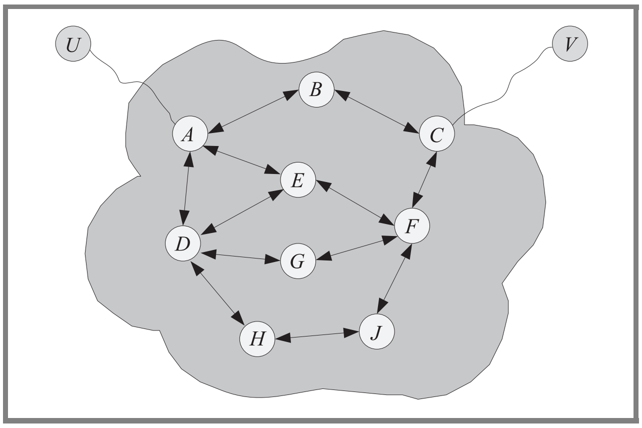


Fig. 8. A smooth handoff in TARP.

Let m be set to 1. This means that nodes E and F will also retransmit the packets because the route through them incurs a 1-hop increase over the best path. The worst thing that can happen is the disappearance of node B, which is a critical component of the current best path. Note, however, that this disappearance will not disrupt the traffic, because the second best path through the cloud, i.e., A–E–F–C, is also being used. The net outcome of this disappearance will be that a would-be duplicate arriving at A or C (from E or F), will be now *bona fide* received and forwarded towards the destination. After a short while, as the destinations update their h_b values in response to the increased number of hops along the best path, the nodes within the cloud will learn that A–E–F–C is the best path at the time. Then, nodes D and G will become involved as those located along the second best path (with 1-hop overhead), thus providing backup in case of subsequent mishaps.

3.4. Avoiding multiple paths with the same cost

One redundancy problem that *SPD* is unable to address is caused by possible multiple paths with the same smallest number oh hops. Consider the situation depicted in Fig. 9. Even with the most restrictive setting of the slack parameter, $m = 0$, both paths $<K_1, K_2, K_3>$ and $<L_1, L_2, L_3>$ will be occupied by the packets traveling between S and D. The duplicates will be eliminated at A (for the D–S direction) and B (for the direction from S to D); however, each of the K_i and L_i nodes will be consistently forwarding them because, according to *SPD*, each of those nodes is located on *a* shortest path between S and D. The problem is particularly nasty if the two rows of nodes can hear each other because then the redundant traffic contributes to the noise in their neighborhood and feeds into congestion.

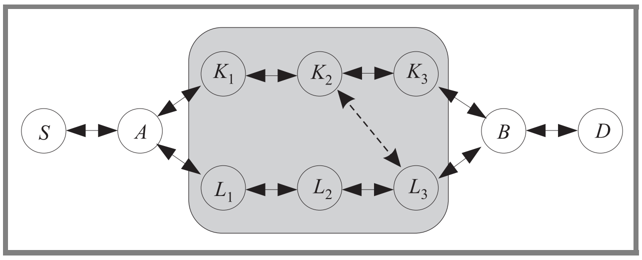


Fig. 9. Multiple paths with the same minimum cost.

To address this problem, TARP has an option whereby the packet header carries one extra bit labeled OPF (for optimal path flag). This flag is set by a forwarding node when it knows that the packet is being forwarded on one of the best paths, i.e., the SPD rule has failed non-forcibly. This means that the packet should normally reach the destination, unless some nodes have moved away or failed.

Consider nodes K_1 and L_1 in Fig. 9 receiving a packet from node A. Owing to the collision avoidance mechanism and randomized retransmission delays applied by the RF driver (the PHY module), one of these nodes, say K_1 will be first to re-broadcast the packet. The other node, L_1 will yield to this transmission and overhear (receive) the packet rebroadcast by K_1. Normally, that packet would be diagnosed as a duplicate and promptly discarded. However, if the OPF bit is set in the packet header, the rule in charge of discarding duplicates yields to another rule, which compares the signature of the received packet against the signatures of all packets currently queued for transmission. If a matching packet is found at L_1 and its h_f is not less than $h_f - 1$ in the received duplicate, then the packet at L_1 is dropped. In plain words, L_1 concludes that by forwarding its copy of the packet, it would not improve upon the forwarding opportunities already extended by K_1.

This mechanism will not help if the paths are disjoint, but it will kick in wherever they cross. Note that while long disjoint paths of the same length need not be rare in a realistic network, the ones for which the length is the shortest possible definitely are.

3.5. Re-casting versus point-to-point forwarding

The term “flooding” permeating the description of TARP may carry negative connotations in confrontation with the point-to-point forwarding protocols, which avoid that nasty problem by identifying precise paths within the unkempt mesh of nodes. This view is grossly misleading. First of all, no knowledge comes out of the blue, and a point-to-point scheme is not able to deliver packets until it has discovered the paths, which operation must necessarily involve some kind of flooding. Many cases of "sales pitch" (and even some "performance studies") either ignore those stages completely or misrepresent them.

If the network is perfectly stable and static, then TARP (with zero slack) is able to achieve essentially perfect convergence to a single shortest path between any pair of peers *A* and *B* (the rare scenarios mentioned at the end of Subsection 3.4 are statistically irrelevant). This is why the term "flooding" does not adequately reflect the nature of TARP, and we prefer to call our paradigm "re-casting." On the other hand, if the network undergoes changes, then the point-to-point schemes are forced to constantly recover from lost paths, which means resorting to various forms of broadcasting and flooding. Also, the standard broadcast component of any point-to-point scheme is the persistent transmission of HELLO packets allowing all nodes to keep track of their neighborhoods.

One may argue that the point-to-point protocols are able to exploit the benefits of handshakes (like RTS-CTS-DATA-ACK of IEEE 802.11) and, in particular, circumvent the hidden/exposed terminal problem, as well as use acknowledgments on every hop, thus enhancing the reliability of communication. However, owing to the fact that most traffic in low-cost wireless networks involves packets that are very short, an RTS/CTS-type handshake is going to be completely useless and likely harmful [35]. While hop-by-hop acknowledgments can help sometimes, they are not impossible in TARP, although the problem must be considered from a slightly different angle.

In contrast to a point-to-point hop, a TARP hop has no well-defined single recipient. Notably, an internal node, i.e., one that solely forwards packets and neither generates nor absorbs them, need not even be equipped with a network address. Thus, if it cares about feedback following its forwarding action, then it would like to know whether the packet has been picked up by one or more nodes in the neighborhood, which will *bona-fide* attempt to forward it towards the destination. The approach used in our first implementation of TARP was to listen for a copy of the transmitted packet (forwarded by one of the neighbors) and interpret it as an indication of success – in addition to timers and counters used to diagnose failures. There are two problems with this solution. First, depending on the load at the forwarding node, there can be a significant delay between packet reception and retransmission. Second, to make this idea work, the destination itself has to "forward" (i.e., retransmit) all received packets, which creates unnecessary noise in its neighborhood.

A better solution employs the so-called *fuzzy acknowledgments*. When a node receives a packet, it first evaluates the rules and then, if all of them fail (i.e., the packet will be forwarded), the node responds with a short burst of RF activity (a simple unstructured packet) of a definite duration. This activity, if present, will tend to occur after a very short period of silence (analogous to SIFS in IEEE 802.11) needed by the node to evaluate the rules. When multiple recipients send their acknowledgments at (almost) the same time, the sender may not be able to recognize them as valid packets. However, it can interpret any activity (of a certain bounded duration) that follows the end of its last transmitted packet as an indication that the packet has been successfully forwarded. Even though the value of this indication may appear inferior to that of a "true" acknowledgment, it does provide the kind of feedback needed by the (informal) data-link function to assume that its responsibility for handling the packet has been fulfilled. When TARP operates with the fuzzy ACK option, any normal packet transmission is preceded by a short LBT (listen before transmit) period whose duration guarantees that fuzzy acknowledgments are not interfered with by regular packets.

Note that the implementation of fuzzy acknowledgments can be viewed as an example of inadequacy of layering in the wireless world. This is because the acknowledgment (which formally belongs to the data-link layer) can only be sent after the rules have been evaluated, i.e., the node concludes that its reception of the packet is going to contribute to its "network-layer" delivery. This is not the only place in TARP where layering gets in the way. Some rules operate best if their evaluation is postponed until the packet is about to be retransmitted, i.e., past the queuing in the data-link layer. Note that shortcuts of this kind are easily implementable within the plug-in model of VNETI.

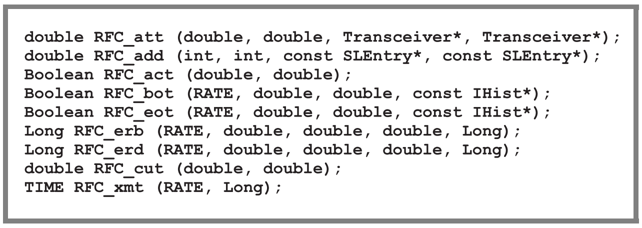
4. Execution and emulation

The large number of generic applications for the wireless devices, combined with the obvious limitations of field testing, result in a need for emulated virtual deployments facilitating meaningful performance assessment and parameter tuning. The close relationship between PicOS and SMURPH hints at the possibility of transforming PicOS praxes into SMURPH models with the intention of executing them virtually. Until recently, one element painfully missing from the scene was a detailed model of wireless channel (SMURPH was originally intended for modeling wired networks). Once that deficiency was eliminated, the circle could be closed, i.e., SMURPH became a vehicle for executing PicOS praxes in virtual settings, practically at the source code level.

4.1. Wireless extensions to SMURPH

Owing to the proliferation of wireless channel models, and the general confusion regarding their adequacy [36], SMURPH does not implement a fixed set of channel models, but instead allows the user to easily implement flexible models, potentially capturing all the aspects of signal propagation required for a detailed functional description of diverse RF modules.

A radio channel model in SMURPH is an object of type *RFChannel*. Its role is to interconnect *transceivers*, which provide the interface between nodes and radio channels. The built-in *RFChannel* class is intentionally open-ended: although it provides a complete functionality of sorts, that functionality is practically useless. It should be rather viewed as a generic parent type for building actual channel types, whose exact behavior is fully specified by a collection of virtual *assessment methods* provided by the user. The primary role of those methods (see Fig. 10) is to determine how a signal attenuates over distance, and how the levels of multiple signals perceived by the same recipient determine whether any of those signals can be recognized as a valid packet. In essence, they encapsulate the static (formula-like) component of the model, thus making its specification straightforward, while all the dynamic processing (like transforming the formulas into events) is hidden inside the SMURPH kernel.



```
double RFC_att (double, double, Transceiver*, Transceiver*);
double RFC_add (int, int, const SLEntry*, const SLEntry*);
Boolean RFC_act (double, double);
Boolean RFC_bot (RATE, double, double, const IHist*);
Boolean RFC_eot (RATE, double, double, const IHist*);
Long RFC_erb (RATE, double, double, double, Long);
Long RFC_erd (RATE, double, double, double, Long);
double RFC_cut (double, double);
TIME RFC_xmt (RATE, Long);
```

Fig. 10. Assessment methods of a wireless channel model in SMURPH.

For example, method `RFC_att` from Fig. 10 is responsible for calculating the received signal level, with the original signal strength and distance passed as the first two arguments. In some cases, the calculation may only depend on the distance (possibly involving randomized factors), in some others it may hinge on some intricate properties of the two transceivers involved, which are also made available to the method. Another method, `RFC_add`, carries out signal addition and is used to assess the level of interference into a reception. The multiple signals perceived by the *transceiver* are represented as an array of objects of type `SLEntry` (signal level entry). In addition to the numerical signal level, as determined by `RFC_att`, a signal level entry carries a generic user-definable attribute, which may introduce arbitrary factors into the operation, e.g., representing CDMA codes that impact the degree to which different signals contribute to the interference. Methods `RFC_bot` and `RFC_eot` are invoked to proclaim a success or failure for the action of perceiving the beginning and end of a packet. They base their decision on the so-called *interference histograms* (type `IHist`) reflecting the complete stepwise history of the interference suffered by the packet's preamble (in the first case) and the entire packet (for `RFC_eot`). `RFC_erb` and `RFC_erd` deal with bit errors and prescribe randomized occurrence of errors in preambles and packets, as well as the timing of user-definable events depending on errors. For example, the model can trigger an event on the first occurrence of a symbol error, thus aborting a packet reception in progress.

In contrast to many popular network simulators, e.g., ns-2 [30], where the fate of every packet is essentially determined at the moment of its departure, our model makes it possible for the virtual RF module to perceive a variety of events depending on dynamic levels of interference and changing predictions of bit errors. A packet reception is not a single indivisible episode, but can be split into stages affecting the module's response. Any physical action of the real counterpart of the module's virtual incarnation can be expressed and meaningfully accounted for in the model.

4.2. The virtual underlay execution engine (VUE^2)

Figure 11 shows the layout of a complete PicOS system implanted into a microcontrolled node. In particular, TARP (described in Subsection 3.1) can be seen as a plug-in to VNETI (Subsection 2.5).

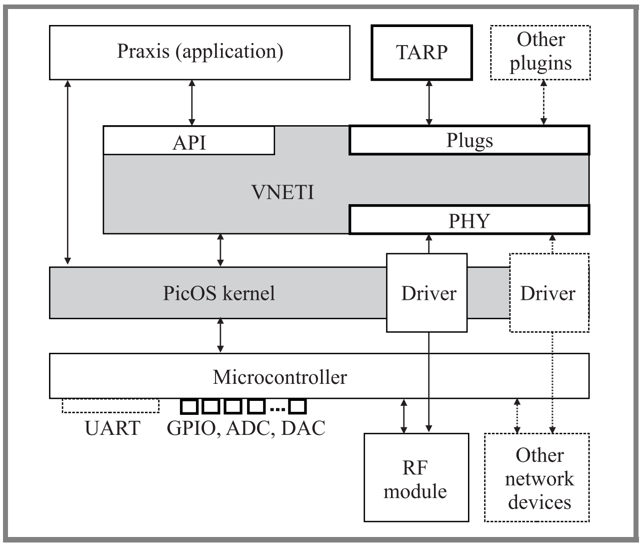


Fig. 11. System structure.

By imposing a certain software layer on SMURPH, which provides a collection of event-driven interfaces representing the environment of a PicOS praxis, and implementing a set of macros transforming PicOS keywords into their SMURPH counterparts, one can render the praxis source code acceptable as a SMURPH program. This is even possible without a formal converter[1], as long as the praxis has been coded with adherence to certain rules. This way, a PicOS praxis can be compiled and executed in the environment shown in Fig. 12, with all the physical elements of its node replaced by their detailed SMURPH models. Notably, exactly the same source code of VNETI is used in both cases.

The VUE^2 has been built with surprising ease because of the similarity in the thread models in PicOS and SMURPH. In both environments, a thread describes a finite state machine, with the state transition function specified in terms of event wait operations. The rules for aggregating such operations and waking up the threads based on the occurrence of the awaited events are practically identical in both

[1] Such a converter would be helpful, of course, and is being implemented.

systems. In SMURPH, viewed as a simulator, the awaited events are delivered by abstract objects called *activity interpreters,* while in PicOS they are triggered by actual physical phenomena (a packet reception, a character arrival from the universal asynchronous receiver/transmitter (UART), and so on).

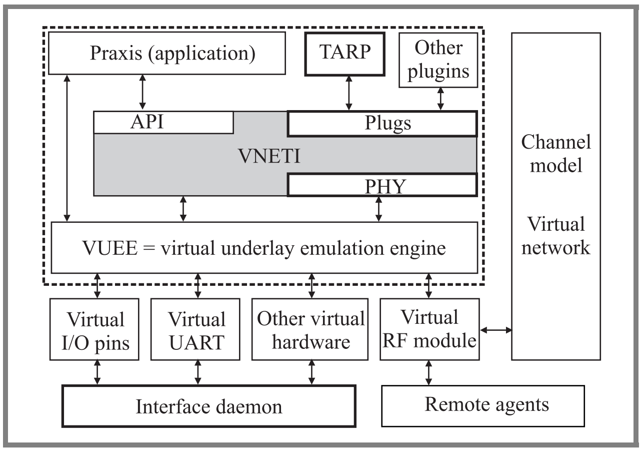


Fig. 12. The structure of a VUE^2 model.

The first significant difference between the two systems is in the interpretation of time flow. In SMURPH, time is purely virtual, which means that formally nobody cares about the actual execution time of the simulation program, but only about the proper marking of the relevant events with virtual time tags. As in all event-driven simulators, the virtual time tags have nothing to do with the real time. Consequently, the useful semantics of time for SMURPH and PicOS threads are different. The actual execution time of a SMURPH thread is essentially irrelevant (unless it renders the model execution too long to wait for the results) and all that matters is the virtual delays separating the artificially triggered events. For example, two threads in SMURPH may be semantically equivalent, even though one of them may exhibit a drastically shorter execution time than the other (due to more careful programming and/or optimization). In PicOS, however, the threads are not (just) models but they run the "real thing." Consequently, the execution time of a thread may directly influence the perceived behavior of the PicOS node. In this context, the following two assumptions make the VUE^2 project worthwhile:

1. PicOS programs are reactive, i.e., they are practically never CPU bound (see Subsection 2.6). In other words, the primary reason why a PicOS thread is making no progress is that it is waiting for a peripheral event rather than the completion of some calculation.

2. If needed (from the viewpoint of model fidelity), an extensive period of CPU activities can be modeled in SMURPH by appropriately (and explicitly) delaying certain state transitions.

In most cases, we can safely ignore the fact that the execution of a PicOS program takes time at all and only focus on reflecting the accurate behavior of the external events. With this approximation, the job of porting a PicOS praxis to its VUE^2 model can be made trivially simple. To further increase the practical value of such a model, SMURPH provides for the so-called *visualization mode* of execution. In that mode, SMURPH tries to map the virtual time of modeled events to real time, such that the user has an impression of talking to a real application. This is only possible if the network size and complexity allow the simulator to catch up with the model execution to real time. If not, a suitable slow motion factor can be employed.

A VUE^2 model can be dynamically interfaced to various *remote agents* (Fig. 12) implementing its interfaces to the real world. The behavior of those agents can be driven from scripts or manually, possibly over the Internet, by a human experimenter. For example, nodes can be powered up and down, their I/O pins can be examined and set, their sensors can be set to specific values, their UARTs or USB interfaces can be mapped to user-accessible virtual terminals or made available to other programs. In particular, an external operations support system (OSS) prepared to talk to the real network can be authoritatively tested in the virtual setup. Networking practitioners should immediately recognize the potentials of VUE^2 when applied to diverse areas of software development, from rapid prototyping to test automation. The virtual nodes can be rendered mobile in response to explicit commands or driven by programmable scenarios. The latter feature comes courtesy of SMURPH and is not VUE^2-specific.

5. Sample application blueprints

Our collection of PicOS praxes includes a set of generic wireless applications that can be easily adapted for various "typical" custom deployments. Those generic applications are called *blueprints*, even though they are fully working, demonstrable systems. For illustration, we present here two such blueprints: routing tags (*RTags*), and tags and pegs (*T&P*). They cover two large classes of applications with different mobility aspects and traffic patterns. They also illustrate how TARP, owing to its rule-driven behavior, can be optimized to different characteristics of the application.

5.1. Routing tags

Routing tags (Fig. 13) is characterized by the presence of an "elevated" node type called *master.* Any node can become master at any time, either self proclaimed or elected by other nodes. Usually, the network is partitioned among the masters with OSS interfaces for external (human or computer) operators. In a typical deployment, masters send messages to other nodes to solicit replies or to trigger some actions. This does not preclude other nodes (any nodes) exchanging messages: the traffic originating or terminating at masters is merely "highlighted," which is to say that some of TARP's parameters are optimized for its presence. In the case of multiple OSS masters, the partitioning is functional (a given group of sensors communicates with a designated master), without being actual: the sensors and/or RTags-routers can intersperse geographically and route traffic in a group-transparent fashion.

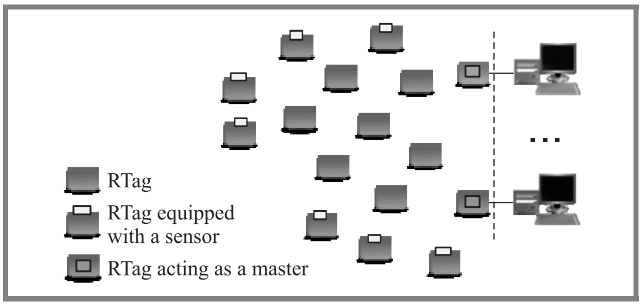


Fig. 13. Routing tags.

Masters usually send messages meant to update the context of newcomers, synchronize time-stamped functionality, and repaint fragments of the network topology for the recipients, i.e., keep the SPD caches filled with useful information (Subsection 3.2). Routing is optimized for relatively infrequent traffic and low mobility. A typical representative of this application class is an on-demand low-mobility asset monitoring system.

5.2. Tags and pegs

With tags and pegs (Fig. 14), the network consists of two types of nodes. Pegs are intentionally immobile, at least compared to tags. Some pegs can play the role of OSS gateways. Their primary purpose is to provide a kind of semi-fixed infrastructure for tracking the whereabouts of tags. Depending on the requirements, the assortment of tools facilitating this tracking may include specialized sensors deployed at tags (e.g., accelerometers, magnetic sensors) reporting their status to pegs. A significant degree of accuracy for many instances of location tracking can be achieved by measuring and correlating the received signal level (RSSI) at multiple pegs perceiving the same tag.

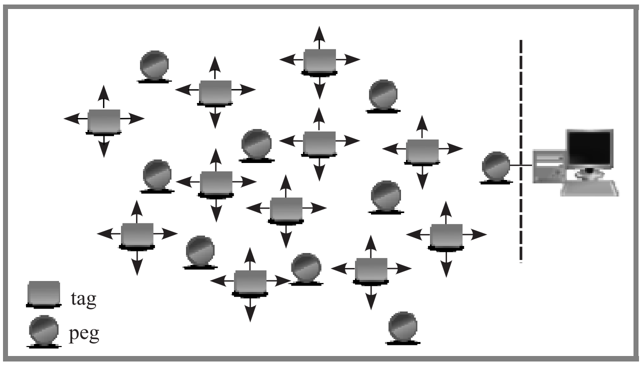


Fig. 14. Tags and pegs.

The tracking may involve various predicates applied to dynamic configurations of tags perceived in the same neighborhood. For example, a gathering of, say, 4+ people with certain attributes in an airport washroom can be detected and signaled as an event calling for special attention. The class of applications covered by *T&P* deals with mobile objects (assets, luggage, people, hazardous materials), whose mobility patterns may have to be classified by event-triggering predicates: mutual exclusion, avoidance of certain spots, time restrictions, etc.

One observation from our experiments with the various "communication modes" of the network is that practically any attempt at classification yields interesting transgressors, i.e., useful application patterns that weld fragments from seemingly distinct areas into innovative functionality. For example, *T&P* clouds embedded into an *RTags* mesh bring about the capacity for distributed self-monitoring. Envision groups of art exhibits at an exposition. A group can raise an alarm if a neighbor becomes mobile; also, it can signal the presence of an unknown member, e.g., one being removed from another area. A traveling exposition can be made self-configurable, enforcing identical setups on every stop.

We started with *RTags*, providing a generic blueprint of a monitoring system, and only after implementing *T&P* did we notice this additional and attractive distribution (or localization) of previously centralized functionality. From this point of view, our framework not only facilitates practical ad hoc networking, but also uncovers its hidden applications. Owing to the high flexibility of a TARP node, such "reconfigurations" can be often soft and dynamic, e.g., available through a sequence of commands injected into the network from an OSS agent.

6. Summary

Using the technology described in this paper, we have been able to build several practical ad hoc networks, including serious industrial deployments. By a "practical network" we understand one that works and meets the expectations of its users.

Customary, we extend the notion of practicality onto technologies, e.g., we say that Ethernet technology is practical, even if some of its botchy specimens fail. Networks acquire practicality via technological progress and industrial acceptance, but this acquisition need not be universal. Ethernet or IP networks are indisputably practical, so are ATM and Bluetooth, even if their cases illustrate the fact that practicality not always follows common sense. On the other hand, IP extensions (meant to make it a "one for all" choice), should, after all these years, be denied practicality. So, we are afraid, must some wireless ad hoc networking schemes, notably ZigBee®, despite powerful industrial sponsors behind them. In the latter case, most of the harm is inflicted by confusing a general scheme with a complete solution, of which the scheme is merely a (likely quite suboptimal) part.

We have presented here a technology that, in our opinion, makes ad hoc networks practical, i.e., functional and deployable in a variety of industrial frameworks. While we do not claim that ours is the only possible approach to practical ad hoc networking, we couldn't find a better one despite honest attempts. Our present library of application blueprints (working, demonstrable, open-ended data-exchange patterns) makes us confident that the combination of tools and methodologies comprising our platform is powerful enough to handle many practically interesting cases of distributed sensing, monitoring, industrial process control, and so on. We are proud of the fact that every detail described in this paper has found its way into real (deployed) ad hoc networks. Large fragments of our research and development were stimulated, or even directly triggered, by the findings and wishes of their users.

References

[1] K. Altisen, F. Maraninchi, and D. Stauch, "Aspect-oriented programming for reactive systems: a proposal in the synchronous framework", Res. Rep. no. tr-2005-18, Verimag CNRS, Nov. 2005.

[2] F. Balarin *et al.*, "Scheduling for embedded real-time systems", *IEEE Des. Test of Comput.*, vol. 15, no. 1, pp. 71–82, 1998.

[3] M. Berard, P. Gburzyński, and P. Rudnicki, "Developing MAC protocols with global observers", in *Proc. Comput. Netw.'91*, Kraków, Poland, 1991, pp. 261–270.

[4] G. M. Birthwistle, O. J. Dahl, B. Myhrhaug, and K. Nygaard, *Simula Begin*. Oslo: Studentlitteratur, 1973.

[5] D. R. Boggs, J. C. Mogul, and C. A. Kent, "Measured capacity of an Ethernet: myths and reality", WRL Res. Rep. 88/4, Digital Equipment Corporation, Western Research Laboratory, 100 Hamilton Avenua, Palo Alto, California, 1988.

[6] T.-W. Chen and M. Gerla, "Global state routing: a new routing scheme for ad-hoc wireless networks", in *Proc. ICC'98*, Atlanta, USA, 1998.

[7] O. J. Dahl and K. Nygaard, *Simula: A Language for Programming and Description of Discrete Event Systems. Introduction and User's Manual*, 5th ed. Oslo: Norwegian Computing Center, 1967.

[8] W. Dobosiewicz and P. Gburzyński, "Improving fairness in CSMA/CD networks", in *Proc. IEEE SICON'89*, Singapore, 1989.

[9] W. Dobosiewicz and P. Gburzyński, "Performance of piggyback Ethernet", in *Proc. IEEE IPCCC*, Scottsdale, USA, 1990, pp. 516–522.

[10] W. Dobosiewicz and P. Gburzyński, "On the apparent unfairness of a capacity-1 protocol for very fast local area networks", in *Proc. Third IEE Conf. Telecommun.*, Edinburgh, Scotland, 1991.

[11] W. Dobosiewicz and P. Gburzyński, "An alternative to FDDI: DPMA and the pretzel ring", *IEEE Trans. Commun.*, vol. 42, pp. 1076–1083, 1994.

[12] W. Dobosiewicz and P. Gburzyński, "On two modified Ethernets", *Comput. Netw. ISDN Syst.*, pp. 1545–1564, 1995.

[13] W. Dobosiewicz and P. Gburzynski, "Protocol design in SMURPH", in *State of the Art in Performance Modeling and Simulation*, J. Walrand and K. Bagchi, Eds. Gordon and Breach, 1997, pp. 255–274.

[14] W. Dobosiewicz and P. Gburzyński, "The spiral ring", *Comput. Commun.*, vol. 20, no. 6, pp. 449–461, 1997.

[15] W. Dobosiewicz, P. Gburzyński, and V. Maciejewski, "A classification of fairness measures for local and metropolitan area networks", *Comput. Commun.*, vol. 15, pp. 295–304, 1992.

[16] W. Dobosiewicz, P. Gburzyński, and P. Rudnicki, "On two collision protocols for high-speed bus LANs", *Comput. Netw. ISDN Syst.*, vol. 25, no. 11, pp. 1205–1225, 1993.

[17] D. Drusinsky, M. Shing, and K. Demir, "Creation and validation of embedded assertions statecharts", in *Proc. 17th IEEE Int. Worksh. Rap. Syst. Protot.*, Chania, Greece, 2006, pp. 17–23.

[18] D. Dubhashi *et al.*, "Blue pleiades, a new solution for device discovery and scatternet formation in multi-hop Bluetooth networks", *Wirel. Netw.*, vol. 13, no. 1, pp. 107–125, 2007.

[19] P. Gburzyński, *Protocol Design for Local and Metropolitan Area Networks*. Upper Saddle River: Prentice-Hall, 1996.

[20] P. Gburzyński and J. Maitan, "Simulation and control of reactive systems", in *Proc. Wint. Simul. Conf. WSC'97*, Atlanta, USA, 1997, pp. 413–420.

[21] P. Gburzyński, J. Maitan, and L. Hillyer, "Virtual prototyping of reactive systems in SIDE", in *Proc. 5th Eur. Concur. Eng. Conf. ECEC'98*, Erlangen-Nuremberg, Germany, 1998, pp. 75–79.

[22] M. Gunes, U. Sorges, and I. Bouazizi, "ARA – the ant-colony based routing algorithm for manets", in *Proc. Int. Worksh. on Ad Hoc Netw. IWAHN*, Vancouver, Canada, 2002.

[23] D. Harel, "On visiual formalisms", *Commun. ACM*, vol. 31, no. 5, pp. 514–530, 1988.

[24] J. Hui, "TinyOS network programming (version 1.0)", TinyOS 1.1.8 Documentation, 2004.

[25] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks", in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Norwell: Kluwer, 1996, vol. 353.

[26] P. Levis *et al.*, "TinyOS: an operating system for sensor networks", in *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer, 2005, pp. 115–148.

[27] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing", in *Proc. ACM/IEEE Int. Conf. Mob. Comput. Netw. MOBICOM'00*, Boston, USA, 2000, pp. 120–130.

[28] X. Ling *et al.*, "Performance analysis of IEEE 802.11 DCF with heterogeneous traffic", in *Proc. Consum. Commun. Netw. Conf.*, Las Vegas, USA, 2007, pp. 49–53.

[29] J. Liu and E. A. Lee, "Timed multitasking for real-time embedded software", *IEEE Contr. Syst. Mag.*, vol. 23, no. 1, pp. 65–75, 2003.

[30] D. Mahrenholz and S. Ivanov, "Real-time network emulation with ns-2", in *Proc. Eighth IEEE Int. Symp. Distrib. Simul. Real-Time Appl.*, Budapest, Hungary, 2004, pp. 29–36.

[31] V. D. Park and M. S. Cors, "A performance comparison of TORA and ideal link state routing", in *Proc. IEEE Symp. Comput. Commun.'98*, Athens, Greece, 1998.

[32] C. Perkins, E. Belding Royer, and S. Das, "Ad-hoc on-demand distance vector routing (AODV)", Febr. 2003, Internet Draft: draft-ietf-manet-aodv-13.txt.

[33] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers", in *Proc. SIGCOMM'94*, London, UK, 1993, pp. 234–244.

[34] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing (AODV)", in *Proc. IEEE Worksh. Mob. Comp. Syst. Appl. WMCSA*, New Orleans, USA, 1999, pp. 90–100.

[35] A. Rahman and P. Gburzynski, "Hidden problems with the hidden node problem", in *Proc. 23rd Bienn. Symp. Commun.*, Kingston, Canada, 2006, pp. 270–273.

[36] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication", *IEEE Anten. Propag. Mag.*, vol. 45, no. 3, pp. 51–82, 2003.

[37] K. Schwan and H. Zhou, "Dynamic scheduling of hard real-time tasks and real-time threads", *IEEE Trans. Softw. Eng.*, vol. 18, no. 8, pp. 736–748, 1992.

[38] C.-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing", in *Proc. IEEE 15th Ann. Int. Phoenix Conf. Comp. Commun.*, Phoenix, USA, 1996, pp. 480–486.

[39] B. Yartsev, G. Korneev, A. Shalyto, and V. Ktov, "Automata-based programming of the reactive multi-agent control systems", in *Int. Conf. Integr. Knowl. Intens. Multi-Agent Syst.*, Waltham, USA, 2005, pp. 449–453.

Paweł Gburzyński holds a Ph.D. in informatics from the University of Warsaw, Poland. Before emmigrating to Canada in 1984, he had been involved in a number of software development projects in Poland, including Loglan, in collaboration with Dr. Andrzej Salwicki and his team. Since 1985 he has been with the faculty of the Department of Computing Science, University of Alberta, Canada, where he is a Professor. In 2002, he cofounded Olsonet Communications Corporation, where he acts as Chief Scientist. Dr. Gburzynski's contributions include SMURPH (a high-fidelity modeling/emulation package for communication systems) and PicOS (an operating system for embedded platforms). He has devised a number of communication protocols, including ad hoc wireless routing schemes and custom protocols for high-performance systems, and implemented many practical wireless networking solutions. His research interests are in telecommunication, embedded systems, operating systems, simulation and performance evaluation. As a hobby, he develops effective anti-spamming tools.

e-mail: pawel@cs.ualberta.ca
Department of Computing Science
University of Alberta
Edmonton, Alberta, T6G 2E8 Canada

Włodzimierz Olesiński (Partner, President & CEO) cofounded Olsonet Communications Corporation, Canada, in 2002. Prior to founding the company, he worked within R&D organizations at Alcatel, Newbridge, Nortel Networks, and Bell-Northern Research, taking part in software development for PSTN and packet networks. His areas of expertise extend over network management, protocol design, operating systems, embedded systems, real-time systems, and mission-critical networks. He is holding an M.Sc. in informatics from the Jagiellonian University of Kraków, Poland. He has lived and worked in Canada, Germany, and Poland.

e-mail: wlodek@olsonet.com
Olsonet Communications Corporation
51 Wycliffe Street
Ottawa, Ontario, K2G 5L9 Canada

Paper

A simple vehicle classification framework for wireless audio-sensor networks

Baljeet Malhotra, Ioanis Nikolaidis, and Janelle Harms

Abstract—Vehicle tracking is one of the important applications of wireless sensor networks. We consider an aspect of tracking: the classification of targets based on the acoustic signals produced by vehicles. In this paper, we present a naïve classifier and simple distributed schemes for vehicle classification based on the features extracted from the acoustic signals. We demonstrate a novel way of using Aura matrices to create a new feature derived from the power spectral density (PSD) of a signal, which performs at par with other existing features. To benefit from the distributed environment of the sensor networks we also propose efficient dynamic acoustic features that are low on dimension, yet effective for classification. An experimental study has been conducted using real acoustic signals of different vehicles in an urban setting. Our proposed schemes using a naïve classifier achieved highly accurate results in classifying different vehicles into two classes. Communication and computational costs were also computed to capture their trade-off with the classification quality.

Keywords— sensor networks, vehicle classification, acoustic signals.

1. Introduction

Networked sensors can be equipped with various sensing devices, as well as memory, processor, radio, and a power supply. However, they are still constrained by limited memory, processing power, channel capacity, and, most importantly, energy reserves. When tracking is considered as an application, data-intensive sources (e.g., high frame rate/high resolution video) is usually avoided as being more energy expensive than low data rate sources. For this reason tracking using audio signals is usually preferable. Vehicle tracking on acoustic data is based on the fact that different vehicles produce distinctly different acoustic signals because their engine and propulsion mechanisms are unique [12]. The problem of vehicle detection using the acoustic signature has been extensively studied [2, 12, 14]. Recently, target classification based on acoustic signals in wireless sensor networks has been addressed in [4, 9]. The advantage of sensor networks is that they provide redundancy in terms of sensing and processing units. Hence, they can operate together in a distributed and coordinated fashion to detect and report the presence of a target vehicle, possibly refining the tracking and classification quality as the target is moving.

We should add that vehicle tracking includes various objectives that must be supported by a number of steps. These steps include vehicle detection, identification and/or classification, and localization. Depending on the specific tracking objectives, all or combinations of these steps may be required. In this paper we restrict our attention to classification alone. Classification is necessary because sensors can report on a specific moving vehicle only after they recognize vehicles that are of some interest. Classification is naturally more challenging if there are multiple targets of various types (e.g., tanks, jeeps, other types of military vehicles, civilian vehicles, etc.). Furthermore, there may be a number of vehicles of the same type, e.g., tanks of a particular make. We define as *classification* the problem of identifying which class a vehicle belongs to. Identifying a *particular* vehicle goes one step further and is not within the scope of the current paper.

Various techniques have been proposed to address the classification problem [2, 4, 12], relying on *feature extraction* that differ in the way features are extracted. For example [2], proposed a wavelet based method for feature extraction, which works as follows: three different types of acoustic signature are extracted: squeak sound, sound under motion, exhaust sound.

The data points contained by each of these signatures are decomposed to a 12 element feature vector using the multiresolution analysis [10]. These 12 element feature vectors represent the energy concentration of the signature signal at 12 different resolution levels. The continuous wavelet transform (CWT) and the Short-Term Fourier transform (STFT) plots were used for two other feature vectors. Finally, these feature vectors are used to compute the distance between the reference and unknown signatures. Wu *et al.* in [14] proposed a principle component analysis based method for recognition of acoustic signatures. The basic idea of their proposed method is to use together the mean adjusted sound spectrum, and key eigenvalues of the covariance matrix to characterize an acoustic signature.

An adaptive threshold based algorithm is proposed in [3] for vehicle detection, based on the average energy of an acoustic signal crossing a threshold value before a decision on the detection of the vehicle can be made. The threshold is updated adaptively. Also, [4] details experiments carried out during the 3rd SensIT situational experiment (SITEX02) organized by DARPA/IXO. Various military vehicles were used in these experiments, real word data was generated and archived for future studies. The objective was to detect and accurately locate vehicles using energy-based localization algorithm. Frequency spectrum based features are extracted from the acoustic and seismic signals captured by the sensors. These features were extracted by using 512-point fast Fourier transform (FFT). Then, for classification purposes, three different approaches are used, i.e.:

- *k*-nearest-neighbor classifier,
- maximum likelihood classifier,
- support-vector-machine classifier.

In [9] a framework for collaborative signal processing in sensor networks is designed for the purpose of multiple targets detection, classification, and tracking.

In this paper we study the impact of increasing the complexity and memory footprint of a classification algorithm to achieve better classification accuracy. We consider this to be a reasonable trade-off since what is usually assumed to be expensive in terms of energy is communication, and not computation/storage (within reason of course). While we are also increasing the computation cost, we argue that as long as the computation is allowed to be completed within a reasonable amount of time, computation can be spread over a longer period of time by proper reduction of the CPU clock [16].

One distinct contributions of our work is that in order to maximally benefit from the information collected by a sensor, we consider multiple representations of features. Some, are well known (FFT, PSD, etc.) but we also introduce *Aura matrices*. Aura matrices [6] have been used in the past for analyzing and predicting texture patterns [15]. In our study we create "artificial" 2-dimensional "textures" by arranging PSD data into matrices, and then using Aura matrices to summarize the information of the 2-dimensional matrix. In other words, Aura matrices attempt to visually approximate the arrangement of PSD values. For details about the construction of Aura matrices the reader is referred to [6]. To exploit the inherently distributed environment of the sensor networks we also propose dynamic PSD features, which are generated on the *run* by the sensors as they capture the acoustic signals. The distinct contribution of the dynamic PSD features is that they are quite low on dimension, yet, effective to produce good classification results.

We start by describing a naïve classification scheme and elementary forms for a distributed implementation over a sensor network in Section 2. Section 3 presents the details on the acoustic features used in this study. Section 4 presents performance evaluation results in terms of classification accuracy and energy expenditure trade-offs for the different distributed implementations. Finally, Section 5 summarizes the findings of the paper and outlines future research objectives.

2. A naïve classifier

Existing techniques such as *k*-nearest neighbor (*k*-NN) can be used by sensors to perform the classification. *k*-NN is based on the idea that similar objects are closer to each other in a multidimensional feature space. *k*-NN is one of the simplest, yet accurate, classification methods and recently it has been used in sensor networks for target classification [4, 9]. Unfortunately, finding *k* for the optimum solution is non-trivial. In contrast to *k*-NN algorithm we adopt a naïve approach where first a training set $|U|$ is defined for each class of the vehicles. Equal number of samples are assumed to be in the training set of these classes. Note that class labels of the training samples are known in advance. Now in order to classify an unknown sample using a particular feature, the naïve classifier does the following: for each class, and for each sample of the $|U|$ samples in the training set of each class, it calculates the distance of the unknown sample from the training set samples. Classifier determines the average distance of the unknown sample from the training set samples of each class. The unknown sample is determined to be in the class with the smallest average distance. If we assume $m \times n$ to be the size of the feature vectors extracted from each of the samples from the set $|U|$ of a class. Furthermore, if we assume there are total c training classes, then the number of computations performed by the classifier to compute the similarity measure for all training classes is proportional to $|U| \times c \times m \times n$. It is clear from this discussion that the dimensionality of the feature vectors is important to the naïve classifier in terms of computational cost. Feature selection is also important for a classifier to achieve good classification results [5]. We discuss more on features selection in Section 3.

Classification process. As a vehicle crosses through an area monitored by a sensor network, the nodes self-organize in neighborhood "clusters" using a technique similar to [1] but where the tie breaking criteria for selection of the "master" node is the signal quality of the monitored vehicle. The master is the node with higher average power of received signal, hence possibly closest to the vehicle. Multiple neighborhoods may be formed, but with a single master node per neighborhood. After the selection, a master node prepares a *schedule*, and broadcasts it in its neighborhood to initiate the classification process. A *schedule* basically consists of *classification assignments* for all sensors in the neighborhood. A typical *classification assignment* for a sensor is to compute the similarity measure of an unknown sample w.r.t. the training samples as specified in the *schedule*. Sensors in a neighborhood after completing their assignments reply back to their master node with their results. After collecting the results the master node makes a decision on the class of the unknown vehicle. Each of the individual neighborhoods can perform a classification method independently of the other neighborhoods. However, multiple neighborhoods may collaborate with each other for two main reasons:

- better accuracy in classifying a vehicle,
- sharing the costs associated with classification.

In our study we examined various scenarios of single and multiple neighborhoods based classification. We propose four basic schemes:

- **Single neighborhood using local signatures (SN-LS).** In this scheme each sensor in a neighborhood predicts the class of the unknown vehicle using a vehicle's local signature captured by the sensor itself. The master node collects results from all sensors in the neighborhood and classifies the unknown vehicle based on the majority of predictions.

- **Single neighborhood using global signatures (SN-GS).** In this scheme each sensor in a neighborhood predicts the class of the unknown vehicle using a vehicle's global signature. A global copy of vehicle's signature is transmitted to a sensor by the master node of its neighborhood. A sensor after receiving global signature from the master node fuses it with its own local signature by using an appropriate averaging function. Master node collects results from all participating nodes in the neighborhood and classifies the unknown vehicle based on the majority of predictions.

- **Multiple neighborhood using local signatures (MN-LS).** In this scheme a master node not only collects results from sensors in its own neighborhood, but it also invokes its adjoining neighborhoods to seek the classification results. All sensors in participating neighborhoods use their local copy of a vehicle's signature.

- **Multiple neighborhood using global signatures (MN-GS).** The basic difference between this scheme and the previous scheme (MN-LS) is that sensors in a particular neighborhood use a global copy of a vehicle's signature provided to them by their respective master node.

3. Features extraction

Sensors perform classification using the features extracted from the acoustic signatures they capture locally or provided to them by their master node. A vehicle's sound is a stochastic signal. The sound of a moving vehicle observed over a period of time will not be a stationary signal. However, a signal of fairly short duration can be treated as a stationary signal [14]. In our case we chose the signal's duration to be 11.06 ms, i.e., 256 data points sampled at a frequency of 22 kHz. In our study we considered six acoustic features that are generated using FFT and PSD of the time series data of a given signal.

1. **Linear FFT feature (LFFT).** This feature is generated using FFT of 256 data points that gave us a linear vector (of size 256) representing frequencies with a resolution of 85.93 Hz.

2. **Linear PSD feature (LPSD).** This feature is generated by taking power spectral density estimates of 256 data points. With a resolution of 85.93 Hz this method gave us a linear vector (of size 128) to form a linear PSD feature.

3. **Multidimensional FFT feature (MFFT).** In this case 10 blocks of 256 FFT data points are used to form a multidimensional FFT feature. This feature can be seen as a matrix of size 256 × 10. The size 10 was determined by trial and error method.

4. **Multidimensional PSD feature (MPSD).** In this case 10 blocks of 128 PSD data points are used to form a multidimensional PSD feature. This feature can be seen as a matrix of size 128 × 10.

5. **Aura of a multidimensional PSD feature (AMPSD)**. It has been demonstrated in [7] that PSD is not an optimal feature for signal recognition. We sought to improve the PSD based feature using some established statistical techniques, namely Aura matrices. In order to construct AMPSD features we simply compute Aura of a MPSD matrix. For computing the Aura of a matrix the reader is referred to [6].

6. **Dynamic multidimensional PSD feature (DMPSD)**. One limitation of the multidimensional features is their size. Consider the MPSD feature which is a 128 × 10 matrix. In order to classify an unknown sample, the naïve classifier must compute the similarity measure of the unknown sample w.r.t. all training samples in all the classes. That may make the naïve classifier computationally expensive for any real time application. Sensors can adopt a dynamic approach here. After constituting a MPSD feature, each sensor may choose only selective PSD dimensions. One criterion for selection is to choose only those dimensions that have the maximum value in each of the blocks (of 128 PSD points). For example, if there were only two blocks of PSD data, and if the first PSD block had a maximum value in the d_1 dimension, and the second PSD block had the maximum value in the d_5 dimension, then only d_1 and d_5 dimensions are selected for both the blocks (of 128 PSD points) to create DMPSD (dynamic MPSD) feature. In this particular example, the DMPSD feature is a matrix of size 2 × 2.

The FFT and PSD data of each of the training samples from all the training classes can be extracted *off-line* and uploaded to the sensors in advance before their deployment. After deployment sensors must extract FFT/PSD features from the unknown samples *on-line*. In the case of LFFT, LPSD, MFFT, MPSD, and AMPSD features, the dimensions of the training FFT/PSD data are fixed. However, in the case of DMPSD feature, sensors must adjust the dimensions of the training PSD data according to the dimensions of the DMPSD feature of the unknown sample that is being classified. In our experimental study, which is presented next, we evaluate the performance of the above discussed features in terms of their accuracy, communication, and computational costs.

4. Experimental study

We used acoustic signal samples of various urban ground vehicles, recorded using a Panasonic US395 microphone. Approximately 50 samples of various vehicles were recorded at two main locations of Bonnie-Doon mall and the University of Alberta bus stop in the city of Edmonton. The samples included ETS buses (part of the public transportation system at the city of Edmonton), different types of cars, small trucks, SUVs, and mini vans. All samples were transferred to MATLAB for the simulation of classification algorithms. We standardized our acoustic dataset to remove any shifting and scaling factors by using the *normal form* [8] of the original time series data.

We assume that every sensor has a copy of training set, U for each class. A sensor's captured signal of an unknown vehicle, which needs to be classified, may be different from other sensors signal of the same unknown vehicle captured approximately at the same time because of the different sensors positions. In order to create a local copy of an unknown signature for a sensor, we attenuate the original signal based on the distance of the sensor from the moving vehicle. Then, we introduce time difference of arrival (TDOA) lags for multiple sensors capturing the same signal based on their relative position, and also add white noise. A vehicle's sound can also be degraded by reverberations, however, we considered an outdoor open environment, so we have neglected the effect of reverberations. In our simulation we considered various scenarios for sensor setup. In these experiments sensors are assumed to be placed along two straight parallel lines, i.e., as they would be deployed along the sides of a street. Sensors are placed 5 m apart and their sensing and radio range is 15 m. A vehicle is considered to be moving with a speed of 53±2 km/h.

4.1. Performance metrics

We consider three performance metrics:

- classification accuracy,
- communication overheads,
- computation cost.

Classification accuracy is computed based on the leave-one-out policy. Under this policy one sample is removed from the acoustic dataset consisting of all samples in a class. This sample is called the testing sample. The rest of the samples in the dataset constitute the training set U for that class. Class label of the testing sample is assumed to be unknown. Then, the distance of the testing sample is computed from all the samples in the training set of each class. This process is repeated for all samples in our acoustic dataset. If two samples are represented by matrices $X_{m\times n}$ and $X'_{m\times n}$, then the distance between them is computed as follows:

$$d = \sum_{q=1}^{|n|} \sum_{p=1}^{|m|} |x_{pq} - x'_{pq}| \,. \tag{1}$$

Classification accuracy is calculated as a percentage of testing samples that are correctly classified from the total number of testing samples. In the experiments we used a simplistic case of only two classes. The objective was to classify the previously mentioned vehicles into two classes, i.e., ETS buses and other vehicles that are not ETS buses. Communication overheads are computed based on the number of bits transmitted by a sensors per classification event. Computational costs are based on the number of computations performed by a sensor per classification event to measure the similarity difference between the unknown sample and the training samples in all classes.

4.2. Single neighborhood case

The results for classification accuracy in SN-LS and SN-GS schemes are presented in Fig. 1. In these experiments we vary the number of sensors in a single neighborhood such that all sensors in the cluster are able to communicate to the master node. In that way we vary the cluster size from 3 sensors to 90 sensors in the cluster. The classification accuracy using most of the features, except DMPSD feature, remains the same as the cluster size changes from 3 to 90 sensors. The classification accuracy of DMPSD feature improved from 77% to 90% as the cluster size changed. The reason for improved accuracy is that sensors dynamically select the PSD points as they capture the unknown signal. When the large number of sensors are available in the cluster, the probability of sensors selecting the effective features increases. As the sensors in a neighborhood make their individual decision, an increase in the number of sensors selecting the effective features increases the probability of that particular neighborhood making a correct prediction.

The reason for the lack of improvement using the rest of the feature extraction schemes is that sensors use only a fixed set of features on the training dataset. Adding more sensors into the neighborhood improves the approximation of the distance measurements collected from the multiple sensors from a neighborhood. However, in the case of naïve classifier, these improved approximations did not improve the classification accuracy much. An accuracy of 98% is achived using MPSD and AMPSD features that is better than the accuracies reported in [11], which uses k-NN classifier. As shown in Fig. 1, DMPSD feature's performance improved clearly in both the SN-LS and SN-GS schemes. With a better copy of vehicle's signal available to all participating sensors, features in SN-GS scheme performed slightly better than SN-LS.

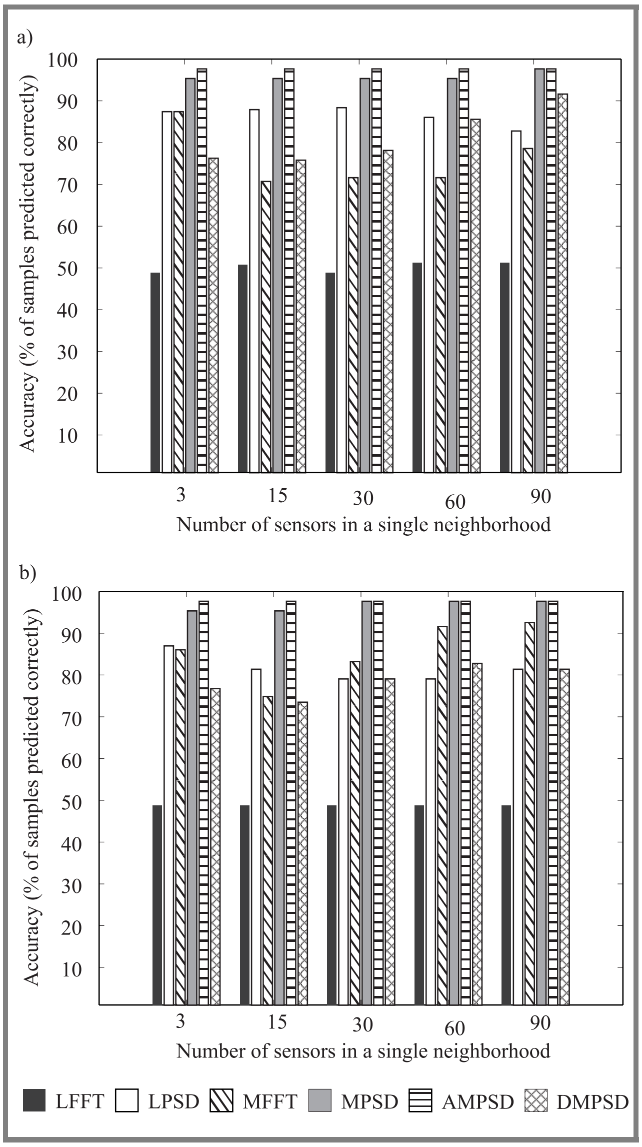


Fig. 1. Classification accuracy of the single neighborhood case: (a) SN-LS; (b) SN-GS.

Increasing the number of sensors in a neighborhood has more impact on the communication costs than on the classification accuracy. The results for SN-LS and SN-GS schemes are presented in Fig. 2a and 2b, respectively. As the number of sensors increases the number of messages exchanged increases. Therefore, costs increase for both the single neighborhood based schemes. Communication costs are much higher in the SN-GS scheme due the transmission of the signature by the master node to its neighborhood.

The benefits of smaller feature size in terms of computational cost are summarized in Fig. 2c. The average size of the DMPSD feature is 6×10, which is almost 4, 2, 42, 21, and 21 times less than LFFT, LPSD, MFFT, MPSD, and AMPSD features, respectively. Due to the reduced feature vectors size, the cost for computing similarity measure in naïve classifier using DMPSD feature is the least as compared to the other feature vectors. On the other hand

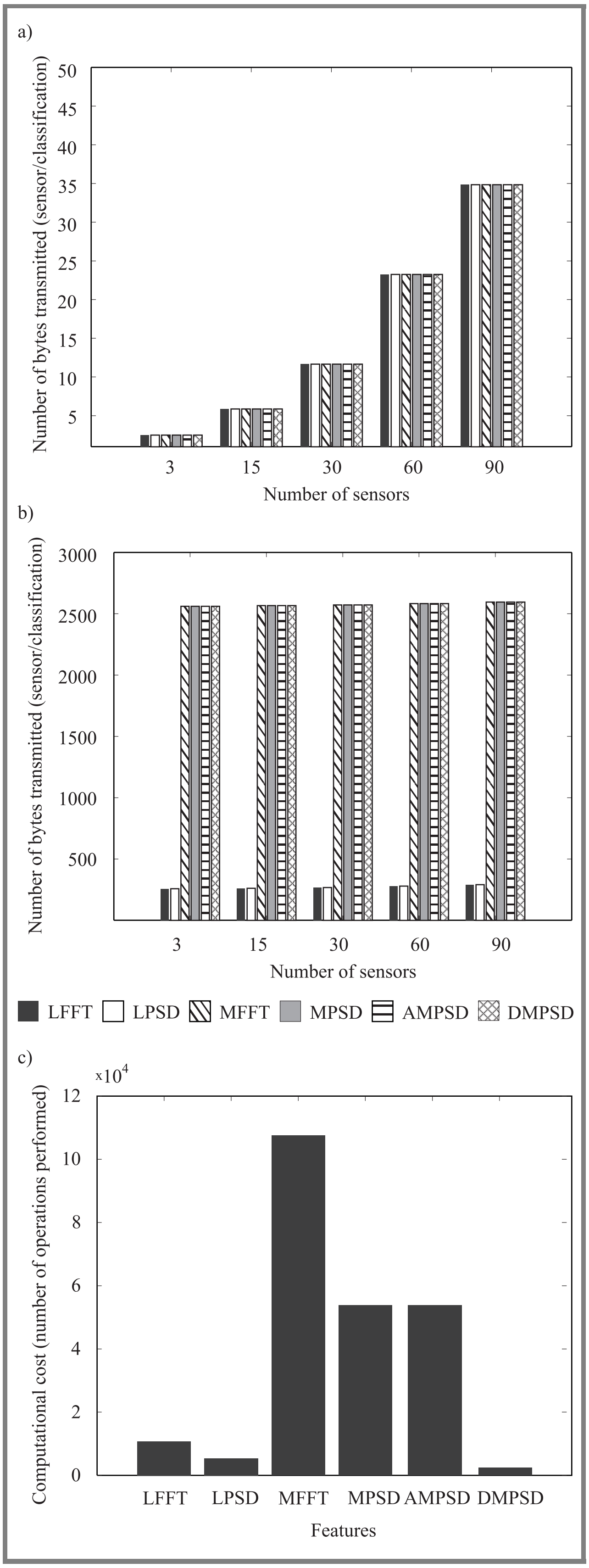


Fig. 2. Cost for the single neighborhood case: (a) communication cost for various features in SN-LS; (b) communication cost for various features in SN-GS; (c) computation cost of similarity measure for various features using the naïve classifier.

MFFT and MPSD are the most expensive features to use. The cost of computing Aura of MPSD is not included in the results of AMPSD feature shown in Fig. 2c.

4.3. Multiple neighborhood case

In the experiments for the multi-neighborhoods based schemes we simulate various scenarios of neighborhood formation. In these experiments we increase the number of neighborhoods by decreasing the number of sensors available per neighborhood while keeping the total number of sensors fixed at 60. For example in the first scenario we form 2 neighborhoods with 30 sensors in each of those neighborhoods. In the second scenario 3 neighborhoods are formed with 20 sensors in each of those neighborhoods. Similarly we generated the rest of the scenarios. We generate these scenarios by adjusting the parameters such as transmission range of the sensors.

Classification based on the multiple neighborhoods may arise in various situations. Consider the case in which the transmission range of the sensors is limited to communicate at shorter distances only. It may restrict the sensors to form neighborhoods within their vicinity only. However, this particular situation is favorable for energy conservation [13]. As shown in Fig. 3 performing classification in smaller sized neighborhoods is more efficient in MN-LS scheme. The reason for lesser cost in the multiple neighborhood case is that setting up smaller sized neighborhoods is less expensive in comparison to forming the larger sized neighborhoods. However, in the case of MN-GS scheme the savings from the smaller sized neighborhoods are marginalized by the heavy costs of transmitting the global signatures. As expected communication costs are much higher in MN-GS scheme. These results are presented in Fig. 3b.

The results for classification accuracy in MN-LS and MN-GS schemes are presented in Fig. 4. There is not much dif-

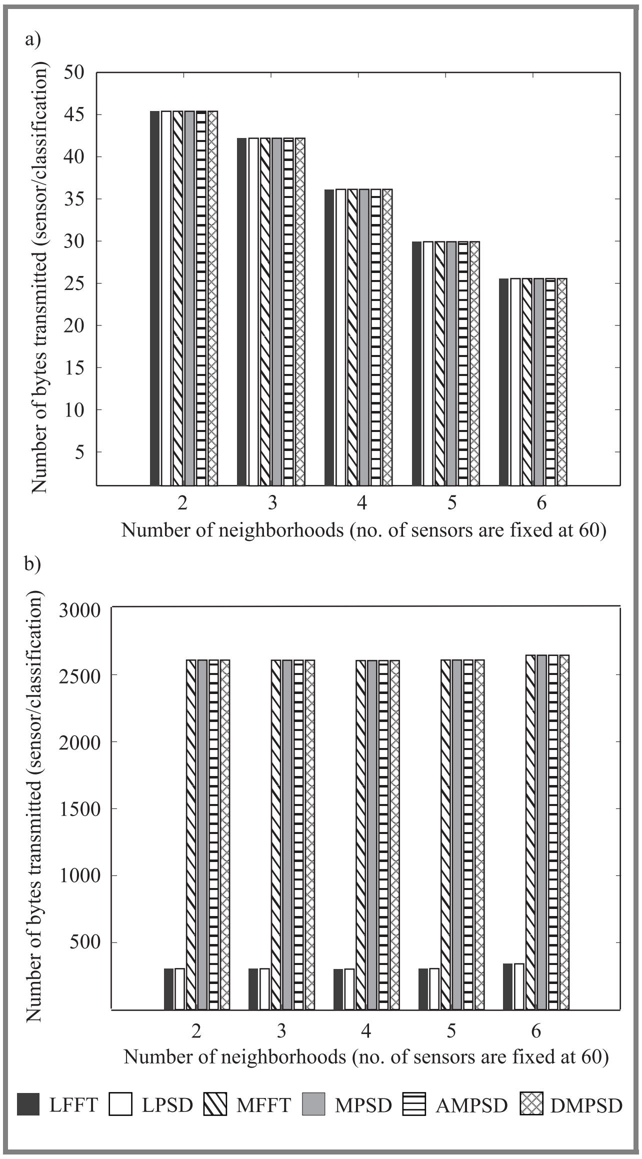


Fig. 3. Communication cost of the multiple neighborhood case: (a) MN-LS; (b) MN-GS.

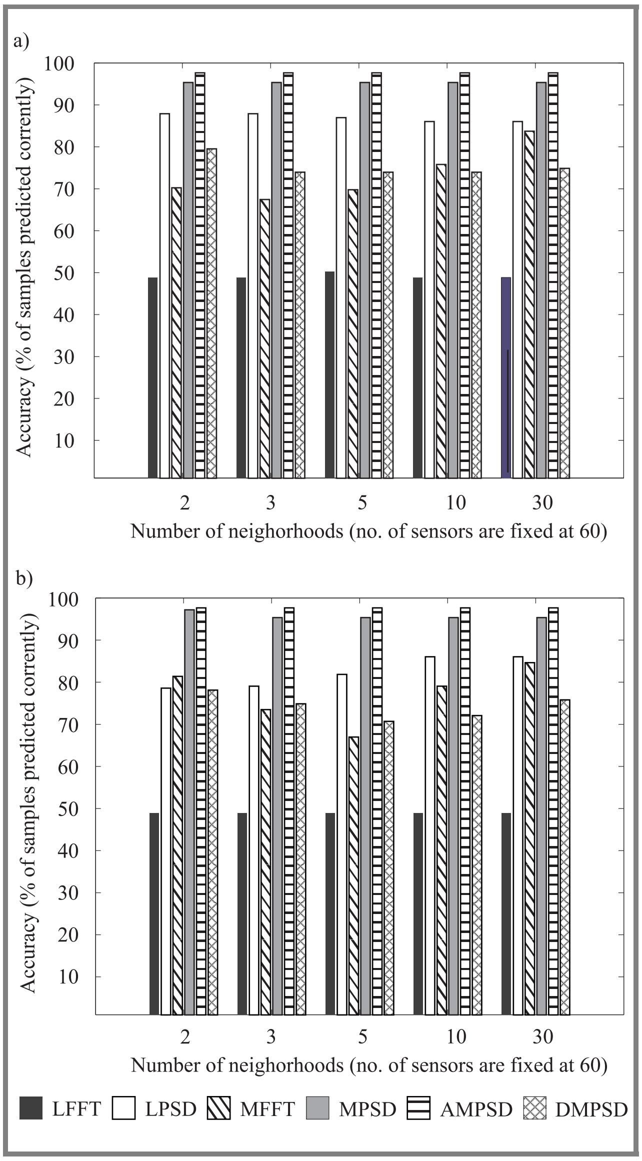


Fig. 4. Classification accuracy of the multiple neighborhood case: (a) MN-LS; (b) MN-GS.

ference between the classification results for single neighborhood scheme (e.g., SN-LS) and multiple neighborhood scheme (e.g., MN-LS) for all features, except DMPSD feature. The accuracy with DMPSD feature decreases as the number of neighborhoods increases. The reason for this behavior is that in the multiple neighborhood schemes the number of sensors per neighborhood decreases. That also means a lesser number of sensors in the neighborhood have the effective features, which affects the overall decision of the neighborhood as a single unit. However, when the number of neighborhoods are large, accuracy improves slightly. In the case of global signatures, a similar trend can be seen in multiple neighborhood schemes for the DMPSD feature. Overall, with a better copy of vehicle's signal available to all participating sensors in a neighborhood, MN-GS scheme performs slightly better than MN-LS.

The results presented here suggest that there is a tradeoff between communication costs, computational costs, and achieving a higher classification accuracy. A higher classification accuracy comes at a higher communication and computational costs for sensors. We also note that when the number of training samples are fixed, then varying the number of sensors per neighborhood affects the classification accuracy for some features. Having more sensors in a neighborhood increases the classification accuracy but at a higher cost of communication. On the other hand selection of features is also an important decision. Some features are more expensive to use than others, but their classification results are better. Our proposed DMPSD feature produced the best combination of accuracy and efficiency, respectively, in terms of classification results and computational costs.

5. Conclusions and future directions

Classifying ground vehicles is an important application of wireless sensor networks. Features extracted from the acoustic signatures of these vehicles form the basis for classification. Whether sensor networks provide for efficient implementation of tracking, depends on whether necessary operations, such as classification, can be performed efficiently in a distributed fashion, achieving high classification accuracy at reasonable communication and computational costs. In this paper, we proposed several distributed schemes for vehicle classification. These schemes are based on the idea of collaborations in single and multiple neighborhoods. One distinct contribution of this paper is dynamic acoustic features, which exploit the inherently distributed nature of a sensor network. These features are extracted by the sensors independently of each other in a distributed fashion, which are simple, yet, effective. We conducted a simulation study using real acoustic signals of urban ground vehicles. Simulation results have revealed the performance of our proposed schemes. Our proposed schemes achieved up to 98% accuracy for a binary classification using a naïve classifier. These results are even better than some of the existing results obtained through the k-NN classifier. In the future we would like to improve the efficiency of our proposed schemes. We also plan to conduct an experimental study where we consider more than two classes of ground vehicles.

Acknowledgements

We thank the kind support of NSERC through its Discovery Grants program. Baljeet Malhotra's work has been supported by an NSERC postgraduate scholarship.

References

[1] S. Basagni, "Distributed clustering for ad hoc networks", in *Proc. 1999 Int. Symp. Parall. Archit., Algor. Netw. ISPAN*, Fremantle, Australia, 1999.

[2] R. Braunlin, R. M. Jensen, and M. A. Gallo, "Acoustic target detection, tracking, classification, and location in a multiple-target environment", in *Proc. SPIE Conf. Peace Wart. Appl. Tech. Iss. Unatt. Ground Sens.*, Orlando, USA, 1997, vol. 3081, pp. 57–66.

[3] J. Ding, S.-Y. Cheung, C.-W. Tan, and P. Varaiya, "Signal processing of sensor node data for vehicle detection", in *Seventh Int. IEEE Conf. Intell. Transp. Syst.*, Washington, USA, 2004.

[4] M. Duarte and Y.-H. Hu, "Vehicle classification in distributed sensor networks", *J. Parall. Distrib. Comp.*, vol. 64, no. 7, pp. 826–838, 2004.

[5] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[6] I. M. Elfadel and R. W. Pickard, "Gibbs random fields, cooccurences, and texture modeling", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, no. 1, pp. 24–37, 1994.

[7] L. German, "Signal recognition: both components of the short time Fourier transform vs. power spectral density", *Patt. Anal. Appl.*, vol. 6, no. 2, pp. 91–96, 2003.

[8] D. Goldin and P. Kanellakis, "On similarity queries for time-series data: constraint specification and implementations", in *Int. Conf. Princ. Pract. Constr. Programm.*, Marseille, France, 1995, pp. 137–153.

[9] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets in distributed sensor networks", *IEEE Sig. Proces. Mag.*, vol. 19, no. 2, pp. 17–30, 2002.

[10] S. G. Mallat, "A theory for multiresolution signal decomposition: a wavelet representation", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, 1989.

[11] B. Malhotra, I. Nikolaidis, and J. Harms, "Distributed classification of acoustic targets in wireless audio-sensor networks", *Comput. Netw.* (special issue on Wireless Multimedia Sensor Networks), 2007.

[12] G. Succi and T. K. Pedersen, "Acoustic target tracking and target identification – recent results", in *Proc. SPIE Conf. Unatt. Ground Sens. Technol. Appl.*, Orlando, USA, 1999, vol. 3713, pp. 10–21.

[13] W. H. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", in *Proc. Hawaii Int. Conf. Syst. Sci.*, Maui, Hawaii, USA, 2000, pp. 1–10.

[14] H. Wu, M. Siegel, and P. Khosla, "Vehicle sound signature recognition by frequency vector principal component analysis", *IEEE Trans. Instrum. Measur.*, vol. 48, no. 5, pp. 1005–1009, 1999.

[15] X. Qin and Y.-H. Yang, "Similarity measure and learning with gray level Aura matrices (GLAM) for texture image retrieval", in *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Patt. Recog. CVPR*, Washington, USA, 2004.

[16] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy", in *Proc. 1st USENIX Symp. Oper. Syst. Des. Implem.*, Monterey, USA, 1994, pp. 13–23.

Baljeet Malhotra is a Ph.D. student in the Computing Science Department at the University of Alberta, Canada. He received the B.T. degree in computer science and engineering in 1999 from the National Institute of Technology, Jalandhar, India. He was a senior software engineer at Satyam Computer Services Ltd., Hydrabad, India, during 1999–2002. He received the M.Sc. degree in computer science in 2005 from the University of Northern British Columbia, Canada. His research interests include acoustic classification, localization and tracking, distributed estimation, and data management techniques in the wireless sensor networks. He is also interested in machine learning and its applications in wireless sensor networks.
e-mail: baljeet@cs.ualberta.ca
Computing Science Department
University of Alberta
Edmonton, Alberta T6G 2E8, Canada

Ioanis Nikolaidis is an Associate Professor with the Computing Science Department at the University of Alberta, Canada. He received his B.Sc. from the University of Patras, Greece, in 1989 and his M.Sc. and Ph.D. in computer science from Georgia Tech in 1991 and 1994, respectively. Between 1994 and 1996 he worked for the European Computer-Industry Research Center in Munich, Germany, in the area of distributed computing. He joined the University of Alberta in January 1997. He has published more than sixty articles in books, journals, and conference proceedings in the area of computer networking. His research interests range from network modeling and simulation, large scale data delivery systems, to mobile and secure networking. Since 1999 he has been a member of the editorial board (and is currently the Editor in Chief) of the "IEEE Network" magazine. He is also a member of the editorial for the "Computer Networks" journal (Elsevier), and of the "Journal of Internet Engineering" (JIE). He has served in the technical program committees of numerous conferences, including ICC, Globecom, INFOCOM, LCN, IPCCC, PerCom, IFIP Networking, and CNSR. He is in the steering committee of WLN (co-located annually with IEEE LCN) and in the steering committee of the ADHOCNOW conference. He was the conference co-chair of ADHOCNOW 2004. He is a member of IEEE and ACM.
e-mail: yannis@cs.ualberta.ca
Computing Science Department
University of Alberta
Edmonton, Alberta T6G 2E8, Canada

Janelle Harms is an Associate Professor in the Computing Science Department at the University of Alberta, Canada. She received her Ph.D. in computer science from the University of Waterloo, Canada, in 1992. She was working in the area of resource allocation and performance analysis of networks. She has served on technical program committees of numerous conferences, including ICC, INFOCOM, ICCCN, IPCCC, ADHOCNOW, MASCOTS and CNSR. Her research interests include performance aspects of network resource allocation, routing and network design problems in mobile and fixed-topology networks.
e-mail: harms@cs.ualberta.ca
Computing Science Department
University of Alberta
Edmonton, Alberta T6G 2E8, Canada

Paper

Multi-threshold signature

Bartosz Nakielski, Jacek Pomykała, and Janusz Andrzej Pomykała

Abstract—The work presents a new signature scheme, called the multi-threshold signature, which generalizes the concept of multisignature and threshold signature. This scheme protects the anonymity of signers in a way the group signature does – in exceptional circumstances the identities of signers may be revealed. Due to the new party – completer, in our scheme the threshold size may vary together with the message to be signed. The presented scheme is based on the RSA signature standard, however other signature standards might be applied to it as well.

Keywords— public key cryptography, threshold signature, multisignature, secret sharing.

1. Introduction

Threshold and multiparty cryptography represent a wide and important area of the modern cryptography. The large part of it deals with the signature schemes such as threshold signatures and multisignatures.

Threshold signatures (c.f. [3, 9]) allow any group of l users to create a signature provided $l \geq t$ (where t is a threshold level, fixed in advance). The multisignature allow any group of members to sign the given message. The identities of signers are recognized in the verification phase and then the decision if the signature is accepted is made (see [1, 4–6]). The verification of the signature applies the public keys of corresponding signers.

This paper is motivated by the following problem: given the group G of cardinality l and the pair (m,t) we are interested in the cryptographic scheme that allows any subgroup of at least t members to sign the message m. In distinction to the common (t,l)- threshold type scheme, here the value of t is not fixed in advance, but may vary together with the message m to be signed. Thus it might be very useful in applications, where the number of members required to agree upon the given document depends for instance on the document's "priority".

Another motivation is to propose the flexible signature scheme, which according to the requirement is anonymous or admits the signer's identification. This flexibility was not the subject of the previous papers, which generally speaking treat both solutions in separate schemes (c.f. [2] for example). From the practical point of view this ability seems to be significant in applications, and the proposed scheme provides the essential computational savings by joining both options within one cryptographic scheme. Therefore as an input for the signing algorithm is the triple (m,t,b), where $b \in \{0,1\}$ points out if the signature should be anonymous or with the signer's identification. The resulting signature is to be verified by any user applying the public key related to G. Similarly as in the conventional threshold signature scheme we require, that any subgroup of cardinality less than t is not able to generate the valid (i.e., accepted in the verification phase) signature, attached to the pair (m,t) (in fact it is not able to obtain any nontrivial information about the group G secret value related to its public key).

In the conventional threshold signature the group public key corresponds to the given value of the threshold size t. The idea of our solution relies on the enlarging somewhat the original group G, so that the public key corresponds to the bigger threshold size t'. Then the additional shares (handled by the additional (trusted) party C) will ensure the valid threshold size $t \leq t'$ of the original signer's group. One could extend the above idea considering not necessarily trusted completer (e.g., C being another group of signers). Such a development in direction of dynamic groups was considered in [10]. The presented scheme is based on the RSA [7] cryptosystem, and the Shamir secret sharing protocol [8]. The paper contains the detailed description of the corresponding multi-threshold signature scheme and the proof of its correctness.

2. General system model

2.1. Participants

We assume there are three parties involved in the protocol:

1. Group $G = \{P_1, P_2, \ldots, P_t\}$.

2. The trusted dealer D responsible for the generation of private and public key of G and the corresponding shares for the group members $P_1, \ldots, P_t$ and completer C.

3. The trusted completer C responsible for flexibility of the threshold level.

We assume that the dealer D is connected with the members P_i and the completer C by the secure channels. Communication between C and members of G goes through group message board (GMB) where all the partial signatures are published (only C and G have an access to it).

2.2. Notation

Throughout the paper N is a positive integer such that: $N = pq$, where $p = 2p' + 1$, $q = 2q' + 1$ and p, q, p', q' are prime numbers satisfying $\min(p'q') > 2t$, where $t = |G|$.

By λ we denote the Carmichael lambda function defined as

$\lambda(\prod_p(p^{\alpha_p}) = \mathrm{lcm}_p\lambda(p^{\alpha_p})$, where:

$\lambda(2) = 1 \quad \lambda(2^\alpha) = 2^{\alpha-2}$ for $\alpha \geq 3$,

$\lambda(4) = 2 \quad \lambda(p^\alpha) = p^{\alpha-1}(p-1)$ if p is an odd prime number.

Conventionally the elements e and d are mutually inverse elements in $Z^*_{\lambda(N)}$, i.e., $ed \equiv 1 \mod \lambda(N)$.

We assume that every member $P_i \in G$ is equipped with the RSA keys (N_i, e_i, d_i), respectively, needed for the authentication process within corresponding parties or members. To assure the uniqueness of m' at the end of the verification process we assume that $N_1 < N_2 < ... < N_t$:

$$A_i = \prod_{\substack{j=1 \\ j\neq i}}^{2t}(x_i - x_j) \mod \lambda(N) \text{ for } i = 1,2,...,2t$$

(x_i are numbers assigned to P_i).

Moreover we let $A'_i = A_i/2^{\varpi_i}$, where ϖ_i is the highest power of 2 dividing A_i and $\varpi = \max_{i\in I}\varpi_i$.

Throughout the paper h will denote a given secure hash function.

3. Initialization phase

In the initialization phase the dealer D performs the following steps:

1. Generates the key pair (d, e) and a random polynomial:

   $f(x) = d + c_1x + c_2x^2 + ... + c_tx^t \in Z^*_{\lambda(N)}[x]$.

2. Computes the shares $s_i = f(x_i)(A'_i)^{-1} \mod \lambda(N)$ and sends them to P_i $(i \leq t)$ and to C (for $t+1 \leq i \leq 2t$).

   Remark 1. Since $\min(p', q') > 2t$ the odd numbers A'_i are invertible $\mod \lambda(N)$.

3. Selects $g \in Z^*_N$ of order equal to $\lambda(N)$ and sends g^{-1} $\mod N$ and $z_i = (g^{s_i})^{-1} \mod N$ to the completer C.

4. He publishes the group public key $gpk = (N, e, \varpi)$.

4. The anonymous signing phase

Assume that the tuple (m,t,b) (m is the message, $t \in \{1,2,...,K\}$ is the threshold level and $b \in \{0,1\}$ points out the signature type (anonymous or with signers identification)), is given to G and C in order to be signed by a given subgroup of G. Then the following steps are performed:

1. **Completer's computation**. The completer C computes $m^* = h(m,t,b)$ and applies the partial signature generation algorithm to compute and publish in the *GMB* the following partial signatures: $\sigma_{i_{t+1}}(m^*)$, $\sigma_{i_{t+2}}(m^*), ..., \sigma_{i_{t+1}}(m^*)$ (where $\sigma_i(m^*) = (m^*)^{s_i}$ $\mod N$) together with the sequence $i_{t+1}, i_{t+2}, ..., i_{t+1}$ of terms contained in the interval $(t, 2t]$.

2. **Group signing**. The group members who decide to sign m, compute $m^* = h(m,t,b)$ and publish their partial signatures $\sigma_{i_1}(m^*), \sigma_{i_2}(m^*), ..., \sigma_{i_t}(m^*)$

   $(1 \leq i_1 < i_2 < ... < i_t \leq t)$ in the *GMB*.

3. **Partial signature verification**. The completer selects a random $r \in Z_{\lambda(N)}$ computes $v^* = (\frac{m^*}{g})^r$ $\mod N$ and sends it to the members participating in the signature generation. Next he computes $v_j = (z_{i_j}\sigma_{i_j}(m^*))^r \mod N$. Each member compute $v^*_j = (v^*)^{s_j} \mod N$ and sends it to the completer. Completer accepts the partial signature σ_{i_j} if and only if $v_j = v^*_j$.

4. **Generation of the full signature**. With the aid of the share combining algorithm the t valid signatures $\sigma_{i_1}(m^*)$, $\sigma_{i_2}(m^*), ..., \sigma_{i_t}(m^*)$ allow any delegated signer to compute the anonymous signature $((m,t,0), \sigma_B(m^*))$, where:

   $\sigma_B(m^*) = \prod_{j\in B}\sigma^{a_i}_{i_j}(m^*) \mod n$ and

   $a_i = 2^{\varpi-\varpi_i}\prod_{\substack{j\in B\\ j\neq i}}(0 - x_j)\prod_{j\notin B}(x_i - x_j)$ is the appropriate Lagrange coefficient for the group B.

   When the signature $\sigma_B(m^*)$ (verified by any signer using gpk) occurs in the *GMB*, the anonymous signing is finished and $\sigma_B(m^*)$ is published.

5. The authorization phase

In the following part the members $P_i \in B$ authorize subsequently their signature using the private keys d_i. We remark that the description of B contains the subscripts of the corresponding signers. They perform the following steps:

1. P_1 computes the message $m' = h(m^*, \sigma, B)$, signs it using his private key d_1 and sends the obtained ciphertext $\delta_1 = (m')^{d_1} \mod N_1$ to the second member P_2.

2. P_2 verifies if $(\delta_1)^{e_1} \equiv m' \mod N_1$ if so, he computes $\delta_2 = (\delta_1)^{d_2} \mod N_2$ and sends it to P_3 (otherwise he publishes in *GMB* information about this disagreement and stops the protocol).

3. Similarly P_3 verifies the obtained ciphertext δ_2 using the public keys (e_2, N_2) and (e_1, N_1), respectively, computes $\delta_3 = (\delta_2)^{d_3} \mod N_3$, sends it to P_4 and so on.

4. The last member $P_t \in B$ verifies δ_{t-1} using the public keys:

$(e_{t-1},N_{t-1}),(e_{t-2},N_{t-2}),\ldots(e_1,N_1)$ and, if the verification is correct, he computes $\delta_t=(\delta_{t-1})^{d_t} \mod N_t$ and publishes it in *GMB*.

5. The full signature of the message m is the 5 – tuple: $((m,t,\sigma),B,\delta_t)$, where $P_i \in B$ are ordered as obove.

According to the requirements, the chosen member of the group G publishes the anonymous signature $((m,t,0),\sigma)$ or the full signature $((m,t,1),\sigma,B,\delta_t)$.

The anonymous signature does not imply any information about the identities of the members of B. It proves only that at least t members of group G have signed the document m.

6. The verification phase

After receiving the anonymous signature (m,t,σ) the verifier uses the group public key (e,N,ϖ) to compute $m^*=h(m,t,b) \mod N$ and then accepts it provided $\sigma^e \equiv (m^*)^{2^{\varpi}} \mod N$. To verify the full signature (m,t,σ,B,δ_t) one first computes $m'=h(m^*,\sigma,B)$ and then accepts the full signature provided $(\ldots((\delta_t^{e_t} \mod N_t)^{e_{t-1}} \mod N_{t-1})\ldots)^{e_1} \equiv m' \mod N_1$.

Theorem 1. The correctly created signature will be accepted in the verification phase.

Proof. First let us consider the anonymous signature (m,t,σ). It is sufficient to prove that $\sigma^e \equiv (m^*)^{2^{\varpi}} \mod N$. By definition we have $\sigma=\prod_{i\in B}\sigma_i^{a_i}=(m^*)^{\sum_{i\in B}s_ia_i}$, where:

$$s_i=f(x_i)(A_i')^{-1} \mod \lambda(N)\,, \tag{1}$$

$$a_i=2^{\varpi-\varpi_i}\prod_{j\in B,j\neq i}(0-x_j)\prod_{j\notin B}(x_i-x_j)\,. \tag{2}$$

It remains therefore to show that $\sum_{i\in B}s_ia_i=2^{\varpi}d=2^{\varpi}f(0)=F(0) \mod \lambda(N)$.

In this connection we apply the Lagrange interpolation formula for

$F(x)=2^{\varpi}f(x)\in Z^*_{\lambda(N)}[x]$ whose graph passes by the points $(x_{i_1},F(x_{i_1}))$,

$(x_{i_2},F(x_{i_2})),\ldots,(x_{i_{t+1}},F(x_{i_{t+1}}))$, where $B=\{i_1,i_2,\ldots,i_t,i_{t+1},\ldots,i_{t+1}\}$.

We have $F(x)=\sum_{i\in B}f(x_i)(2^{\varpi}\Lambda_i(x)) \mod \lambda(N)$, where:

$$2^{\varpi}\Lambda_i(x)=2^{\varpi-\varpi_i}\prod_{j\in B,j\neq i}(2^{\varpi_i}(\frac{x-x_j}{x_i-x_j}))=$$

$$=2^{\varpi-\varpi_i}\prod_{j\in B,j\neq i}(x-x_j)\prod_{j\notin B}(x_i-x_j)\prod_{\substack{j=1\\ j\neq i}}^{2t}\frac{2^{\varpi_i}}{(x_i-x_j)} \mod \lambda(N).$$

Hence in view of Eqs. (1) and (2) and definition of A_i' we obtain:

$$F(0)=\sum_{i\in B}f(x_i)(A_i')^{-1}\cdot 2^{\varpi-\varpi_i}\prod_{j\in B,j\neq i}(0-x_j)\prod_{j\notin B}(x_i-x_j)=$$

$$=\sum_{i\in B}s_ia_i \mod \lambda(N), \text{ as claimed.}$$

To verify the full signature (m,t,σ,B,δ_t) we use the bijectivity of transformation $x\longmapsto x^{d_i} \mod N_i$ $(1\leq i\leq t)$ and the inequalities:

$N_1<N_2<\ldots<N_t$ (that assure the uniqueness of m' at the end of the verification process).

Taking δ_t to the power e_t we obtain the unique $\delta_{t-1} \mod N_{t-1}$ then (using e_{t-1}) the unique $\delta_{t-2} \mod N_{t-2}$ and finally the unique

$((m')^{d_1})^{e_1}=m' \mod N_1$ as required.

■

7. Conclusions

Two basic benefits of the presented scheme are the scaleability (in threshold size) and generality – it might be useful for the applications typical for the threshold-type signatures or multisignatures.

The final output is the pair: anonymous G-signature and the full signature (containing the signers' identifications).

The completer can be regarded as a well protected machine which for the input value (m,t,b) outputs the corresponding partial signatures.

As proved in [10] the multi-threshold device with C regarded as another group of signers could be developed in the direction of dynamic groups signatures schemes.

Appendix – an example

1. System parameters:

$p=23$	$q=47$	$N=1081$	$\lambda(N)=506$
$p\prime=11$	$q\prime=23$	$t=3$	$\min(p',q')>6=2t$
$e=13$	$d=39$	$m^*=7$	$G=\{P_1,P_2,P_3\}$

2. Dealer generates random polynomial:

$f(x)=3x^3+5x^2+7x+39$ and sets $x_i=i$ which implies:

$$A_i=\prod_{\substack{j=1\\ j\neq i}}^{6}(i-j) \mod 506$$

3. We have:

$A_1=386$	$A_1'=193$	$\varpi_1=1$	$(A_1')^{-1}=409$	$f(1)=54$
$A_2=24$	$A_2'=3$	$\varpi_2=3$	$(A_2')^{-1}=169$	$f(2)=97$
$A_3=494$	$A_3'=247$	$\varpi_3=1$	$(A_3')^{-1}=295$	$f(3)=186$
$A_4=12$	$A_4'=3$	$\varpi_4=2$	$(A_4')^{-1}=169$	$f(4)=339$
$A_5=482$	$A_5'=241$	$\varpi_5=1$	$(A_5')^{-1}=21$	$f(5)=68$
$A_6=120$	$A_6'=15$	$\varpi_6=3$	$(A_6')^{-1}=135$	$f(6)=403$

$\varpi=\max_i\varpi_i=3$

4. Dealer, using the table above, computes:

   $s_i = f(i) * (A_i')^{-1} \mod \lambda(N)$

   $s_1 = (54 * 409) \mod 506 = 328$

   $s_2 = (97 * 169) \mod 506 = 201$

   $s_3 = (186 * 295) \mod 506 = 222$

   $s_4 = (339 * 169) \mod 506 = 113$

   $s_5 = (68 * 21) \mod 506 = 416$

   $s_6 = (403 * 135) \mod 506 = 263$

5. Dealer selects $g = 3$ and sends to the completer the following values:

   $g^{-1} \mod N = 3^{-1} \mod 1081 = 721$ and

   $z_1 = (g^{s_1})^{-1} \mod N = 331$

   $z_2 = (g^{s_2})^{-1} \mod N = 259$

   $z_3 = (g^{s_3})^{-1} \mod N = 639$

   $z_4 = (g^{s_4})^{-1} \mod N = 949$

   $z_5 = (g^{s_5})^{-1} \mod N = 538$

   $z_6 = (g^{s_6})^{-1} \mod N = 647$

6. We assume that $m^* = h(m, 2, 0) = 7$ and $B = \{2, 3, 4, 6\}$.

7. P_2 and P_3 generate and send to the completer their partial signatures:

   $\sigma_2 = (m^*)^{s_2} \mod N = 7^{201} \mod 1081 = 711$

   $\sigma_3 = (m^*)^{s_3} \mod N = 7^{222} \mod 1081 = 3$

8. Completer verifies partial signatures created by P_2 and P_3.

   He selects $r = 5$ and sends v^* to P_2 and P_3, where

   $v^* = (m^* g^{-1})^r \mod N = (7 \cdot 721)^5 \mod 1081 = 732.$

   Next the completer computes:

   $v_2 = (z_2 \cdot \sigma_2(m^*))^r \mod N = (259 \cdot 711)^5 \mod 1081 = 948$

   $v_3 = (z_3 \cdot \sigma_3(m^*))^r \mod N = (639 \cdot 3)^5 \mod 1081 = 16$

9. Members P_2 and P_3 compute and send to the completer:

   $v_2^* = (v^*)^{s_2} = 732^{201} \mod 1081 = 948$

   $v_3^* = (v^*)^{s_3} = 732^{222} \mod 1081 = 16$

10. Completer accepts σ_2, σ_3 and creates two missing partial signatures:

    $\sigma_4 = (m^*)^{s_4} \mod N = 7^{113} \mod 1081 = 964$ and

    $\sigma_6 = (m^*)^{s_6} \mod N = 7^{263} \mod 1081 = 79$

11. P_2 (as a delegated user) computes the interpolation coefficients:

    $a_2 = 2^1(0-3)(0-4)(0-6)(2-1)(2-5) \mod 506 = 216$

    $a_3 = 2^2(0-2)(0-4)(0-6)(3-1)(3-5) \mod 506 = 262$

    $a_4 = 2^1(0-2)(0-3)(0-6)(4-1)(4-5) \mod 506 = 216$

    $a_6 = 2^0(0-2)(0-3)(0-4)(6-1)(6-5) \mod 506 = 79$

    and finally he computes the anonymous signature:

    $\sigma = \prod_{i \in B} \sigma_i^{a_i} =$

    $(711^{216} * 3^{262} * 964^{216} * 79^{386}) \mod 1081 = 354$

12. To verify the signature $((m, 1, 0), 354)$ we use the public key $(1081, 13, 3)$ and compute:

    $\sigma^e \mod N = 354^{13} \mod 1081 = 909$

    $(m^*)^{2^{\varpi}} \mod N = 7^8 \mod 1081 = 909$ and accept the signature.

References

[1] C. Boyd, "Digital mutisignatures", in *Cryptography and Coding*, H. Baker and F. Piper, Eds. Oxford: Oxford University Press, 1989.

[2] C. C. Chang and H. F. Huang, "An efficient and practical (t,n) threshold signature scheme with known signers", *Fundam. Informat.*, vol. 53, no. 3, pp. 243–253, 2003.

[3] Y. Desmedt and Y. Frankel, "Threshold cryptosystems", in *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology, LNCS*. Berlin: Springer, 1989, vol. 435, pp. 307–319.

[4] K. Itakura and K. Nakamura, "A public key cryptosystem suitable for digital multisignatures", *NEC Res. Develop.*, no. 71, pp. 1–8, 1983.

[5] S. Micali, K. Ohta, and L. Reyzin, "Accountable – subgroup multisignatures", in *Proc. 8th ACM Conf. Comput. Commun. Secur. CCS'01*, Philadelphia, USA, 2001.

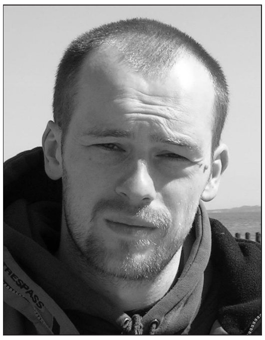
[6] T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems", *ACM Trans. Comput. Syst.*, vol. 6, iss. 4, pp. 432–441, 1988.

[7] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[8] A. Shamir, "How to share a secret", *Commun. ACM*, vol. 22, pp. 612–613, 1979.

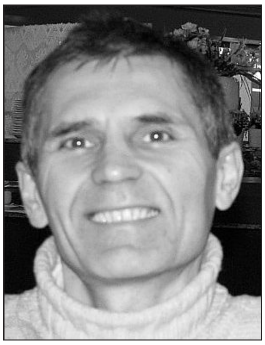
[9] V. Shoup, "Practical threshold signatures", in *Proc. Adv. Cryptol. EuroCrypt 2000*, Bruges, Belgium, 2000.

[10] J. Pomykała and T. Warchoł, "Threshold signatures in dynamic groups", in *Proc. Fut. Gener. Commun. Netw.*, Jeju-Island, Korea, 2007.



Bartosz Nakielski was born in Warsaw, Poland, in 1979. He received his M.Sc. in mathematics from Department of Mathematics, Mechanics and Informatics of Warsaw University. His thesis was titled "Arithmetical aspects of digital signatures". He was working as a certificate authority administrator in Information Security Department in Social Insurance Institution (years 2004–2007). Since January 2008 he works in Security Department in National Bank of Poland.

e-mail: barteknakielski@aster.pl



Jacek Pomykała works at the Faculty of Mathematics, Informatics and Mechanics of Warsaw University, Poland, since 1985. He has defended his Ph.D. thesis in 1986 in the area of sieve methods in number theory. In 1997 he has made the habilitation in the area of automorphic L-functions and distribution of their zeros. Now his main area of subject is cryptology and number theory. He has published over 20 papers mainly in mathematical journalas and has been directed two KBN grants on the arithmetics of elliptic curves and zeros distributions of Hecke's L-functions, respectively. He is also one of the authors of the book concerning the information systems and cryptography. He had many research visits (two long term visits) in Europe, America, Asia and has been an invited speaker in many international conferences in mathematics.
e-mail: pomykala@mimuw.edu.pl
Institute of Mathematics
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
Banacha st 2
02-097 Warsaw, Poland

Janusz Andrzej Pomykała received B.Sc. and M.Sc. in mathematics from the University of Warsaw in 1983, focusing on the thesis about iterated forcing method, written under prof. W. Guzicki. He has taught and done research at various colleges and universities in Poland. He was working also in ILLC at the University of Amsterdam, due to Tempus program. His fields of interest include mathematical logic, algebra, use of novel methods of data analysis, information systems and databases, design of safe systems. He is also active in the popularization of mathematics and computer science on elementary level. He works in the Department of Applied Computer Science and Management in Warsaw Management Academy. He has written about 20 scientific articles and one book (in Polish). He was on more then 10 international conferences and workshops in the field of computer science.
e-mail: pomykala_andrzej@mac.edu.pl
Department of Applied Computer Science and Management
Warsaw Management Academy
Kawęczyńska st 36
03-772 Warsaw, Poland

Paper

Temperature dependence of polarization mode dispersion in tight-buffered optical fibers

Krzysztof Borzycki

Abstract—Experiments and theoretical analysis of influence of temperature on polarization mode dispersion (PMD) in single mode optical fibers and cables are presented. Forces generated by contracting buffer create optical birefringence and increase fiber PMD at low temperatures. Single mode fiber (SMF) in 0.9 mm polymeric tight-buffer can exhibit an extra component of PMD exceeding 0.3 ps/$\sqrt{\text{km}}$ in such conditions. On the other hand, tight-buffered spun nonzero dispersion-shifted fibers (NZDSF) and optical units with stranded single mode fibers have showed good stability of PMD over wide range of temperatures. This is due to presence of circular strain in the core, blocking accumulation of mechanically induced birefringence.

Keywords— single mode optical fiber, polarization mode dispersion, birefringence, environmental testing, tight-buffered fiber, optical fiber cable, optical ground wire, temperature cycling, polymer.

1. Introduction

Polarization mode dispersion (PMD) is detrimental to high speed optical fiber transmission, thus attracting considerable interest. This phenomenon results from imperfections in fiber geometry and external mechanical disturbances: pressure, bending, twisting, etc. While improved manufacturing processes have reduced PMD of single mode fibers down to 0.02–0.05 ps/$\sqrt{\text{km}}$, this performance is guaranteed only when the fiber is well protected against deformations and external forces, e.g., in a loose-tube cable. Situation is different, when the fiber has thick, tightly applied polymeric protective buffer. Forces transferred from such buffer to fiber are non-negligible and depend strongly on operating temperature because of mismatch between thermal expansion coefficients of silica and polymers. Buffer material, method of buffer application and variations in polymer crystalline phase content all have considerable influence here.

An important, but mostly neglected factor is the presence of circular strain in the fiber core, introduced either by stranding of cable components or fiber spinning during drawing. Such strain, if large enough, can dramatically improve stability of PMD in a fiber being subject to mechanical disturbances and variable temperatures.

Experiments described in previous paper on the same subject, presented in *Journal of Telecommunications and Information Technology*, no. 3, 2005 [1] and elsewhere [2, 3] revealed strong increase of PMD in most – but not all tight-buffered fibers at low temperatures. This paper covers:

- additional experiments on fibers and cables;
- fiber-buffer interactions and resulting PMD changes;
- influence of fiber twisting and spinning on PMD stability.

Some test data are presented in report [4]. Detailed descriptions and analysis are included in Ph.D. thesis by the same author [5].

2. Experiments

2.1. PMD measurement setup

Both the procedure for testing temperature-induced effects in optical fibers and instruments used at National Institute of Telecommunications (NIT) were presented in [1]. A new PMD measurement setup (Figs. 1 and 2), still based on the fixed analyzer method [6] was introduced. Use of superluminescent light emitting diode (SLED) source and optical spectrum analyzer (OSA) reduced measurement time, extended spectral range to 1250–1650 nm and improved resolution to 0.01 ps. Instruments were controlled by a personal computer (PC) through the general purpose interface bus (GPIB).

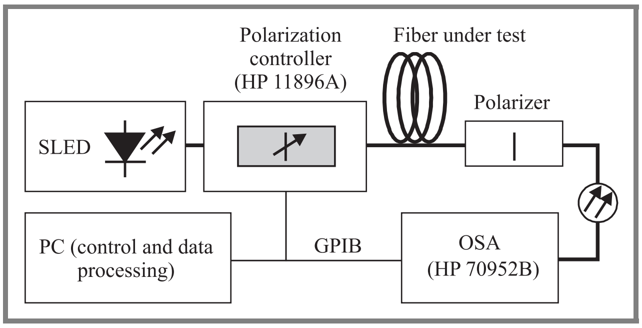

Polarization controller (HP 11896A)
Fiber under test
Polarizer
SLED
PC (control and data processing)
GPIB
OSA (HP 70952B)

Fig. 1. Block diagram of PMD measurement setup with SLED source.

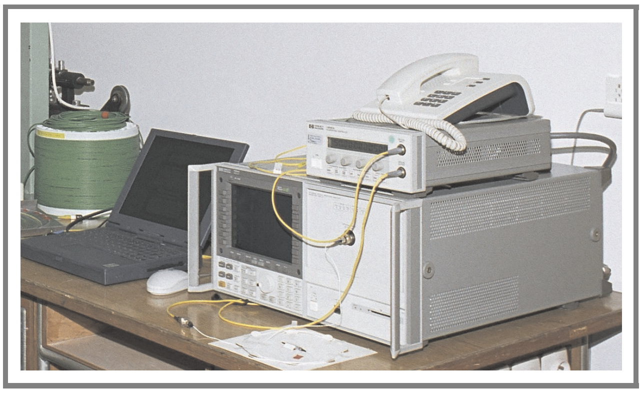


Fig. 2. PMD measurement setup: laptop PC (left), polarization controller (upper right), OSA (right), polarizer with connectors (bottom). SLED source not shown.

2.2. Preparation of samples

To minimize PMD variations due to pressure between layers of fiber or cable, several samples under test have been loosely hanged on a supporting tube or coiled on a flat plate rather than wound on a spool or drum (see Fig. 3). This ensures a more realistic approximation of operating conditions.

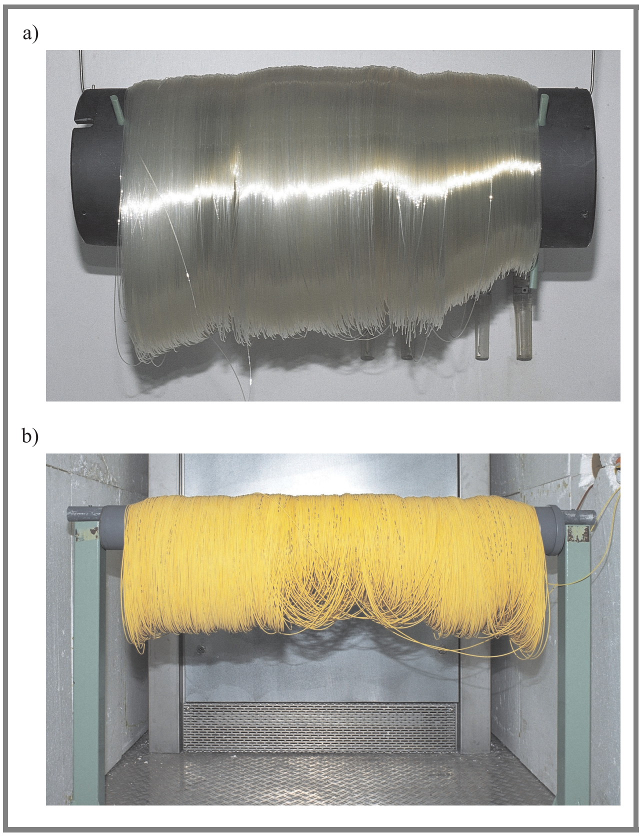


Fig. 3. Samples of tight-buffered fiber (a) and indoor cable (b) in the environmental chamber.

Cables with loose tubes effectively protecting the fibers from external crush forces were tested on standard drums. Care was taken to minimize fiber movements, in particular vibrations and dancing caused by forced air flow inside the environmental chamber. Fibers suspended in the air were particularly prone to such disturbances, resulting in random bending and fluctuations of state of polarization. This in turn produced rapid variations in spectra recorded during PMD measurements (Fig. 4).

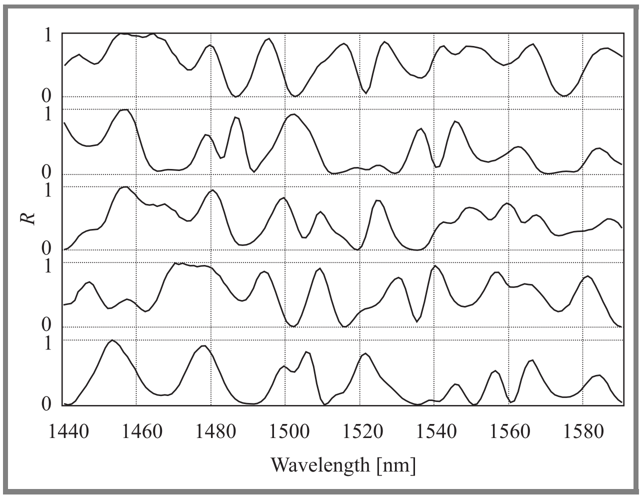


Fig. 4. Transmission spectra $R(\lambda)$ acquired during 5 fixed analyzer scans at 5 min intervals. Single mode fiber (SMF) in 0.9 mm semi-tight buffer, 6177 m long. Forced air circulation. The fiber was hanging loosely and air flow caused dancing.

Without forced air movement, transmission spectra variations in fixed temperature regime were smaller and occurred gradually (Fig. 5), primarily due to slowly changing temperature of the sample. This, however, did not reduce variability of PMD results in successive measurements. In fact, minor mechanical and thermal disturbances to fiber under test are believed to improve accuracy of PMD measurement, as they ensure collection of wider and more representative set of test data [7].

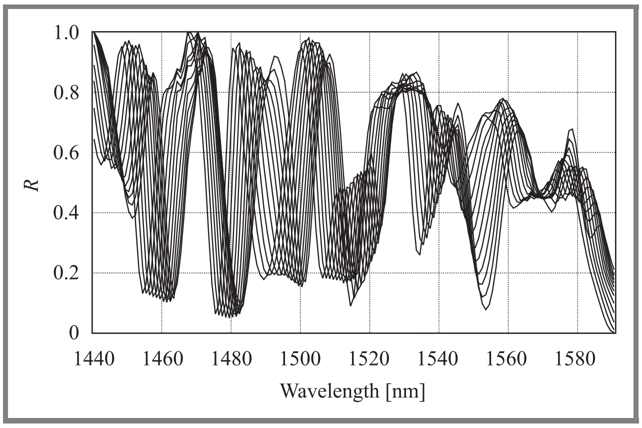


Fig. 5. Twelve transmission spectra $R(\lambda)$ acquired at 5 min intervals. Sample temperature rose by approximately 1°C during this time. Sample as in Fig. 4. Air circulation was switched off.

Uncertainty of PMD measurements was reduced by averaging 6–20 results obtained at each temperature. No significant change in fiber attenuation was observed during any experiment presented here.

2.3. PMD-temperature characteristics: tight-buffered fibers

2.3.1. Indoor cable with G.652 fiber

This cable (Tele-Fonika W-NOTKSd 1J 2,0) had a single non dispersion-shifted (SMF, ITU-T G.652) [8] fiber in a 0.9 mm extruded poly-butylene-terephtalate (PBT) tight buffer, aramide strength member, low smoke zero halogen (LSZH) jacket and diameter of 2 mm. The sample (Fig. 3) was 4040 m long.

Previous test on this cable [1] revealed rise of PMD at temperatures below 0°C, and large, permanent increase of PMD after cooling from +60°C to room temperature. This effect was attributed to increase of content of crystalline phase and related shrinkage of PBT buffer.

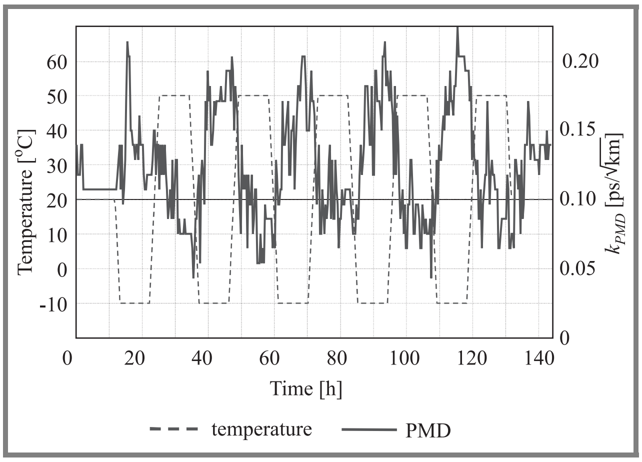


Fig. 6. Changes to PMD coefficient (k_{PMD}) of G.652 fiber in extruded buffer during temperature cycling test no. 1.

To investigate this issue further, the cable was initially subjected to 5 temperature cycles between –10°C and +50°C, with a controlled rate of 20°C/h, and dwell time of 9 h. Figure 6 shows the results.

Conclusions and observations were as follows:

- There was a small increase of PMD in each cycle and at all temperatures.
- Dwell time was too short for complete stabilization of PMD, so the results are not accurate.

In the next test, the same cable was subjected to 3 temperature cycles, with temperature range extended to –20°C…+60°C and rate of change 20°C/h. Dwell time was increased to 24 h, in order to ensure full stabilization of PMD. Results are shown in Fig. 7.

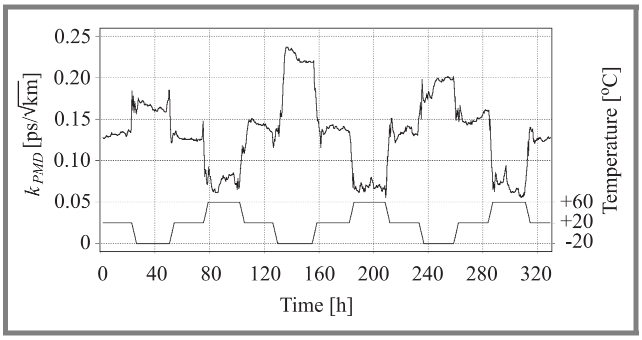


Fig. 7. Variations of PMD coefficient of indoor cable with G.652 fiber in extruded buffer during temperature cycling test no. 2. Lower graph shows temperature profile.

Conclusions and observations after this test:

- Large variations of PMD between temperature cycles.
- Thermal history has most effect on PMD at –20°C, and none at +60°C. This is due to reduction of forces acting on the fiber at high temperature and "reset" of polymer condition when the glass transition temperature T_g (approx. +50°C for PBT) is exceeded (see Subsections 3.2–3.4).
- Reduction of PMD in the last cycle proves the observed process is reversible, ruling out polymer oxidation, decomposition, cracking, etc. Variations in crystalline phase content due to differences in cooling conditions in each cycle are the most likely reason for effects observed.

2.3.2. Tight-buffered G.652 fiber

This test was preformed on 460 m long fiber with a 0.9 mm extruded polyamide tight buffer, made by Sumitomo (Japan). The experiment was carried out during COST-291 short-term scientific mission (STSM) [4]. Figure 8 shows the PMD characteristics in thermal equilibrium conditions.

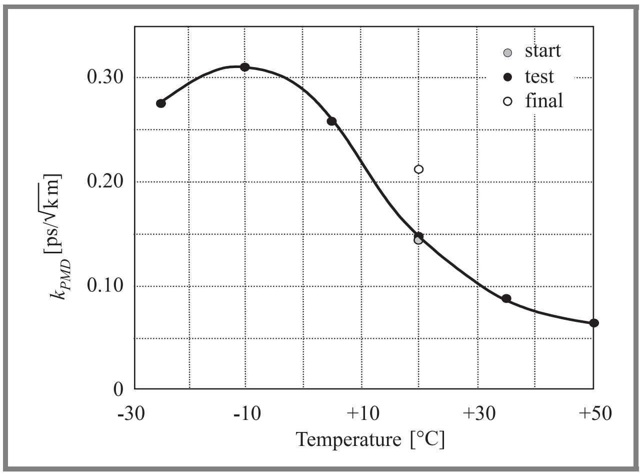


Fig. 8. PMD-temperature characteristics of G.652 fiber in nylon buffer. Steady-state data.

The following effects have been detected:

- Strong rise of PMD with decrease of temperature, but this effect is reversed below –10°C.
- Thermally induced component of PMD has exceeded 0.30 ps/$\sqrt{\text{km}}$ at –10°C.
- Permanent increase of PMD after test, probably due to crystallization and shrinkage of polyamide.

A graph plotted using all data (Fig. 9) shows strong PMD transients. There was always a temporary change of PMD in a direction **opposite** to what was observed in steady-state condition. This is likely due to finite time necessary for relaxation of coating and buffer polymers and reduction of internal strains.

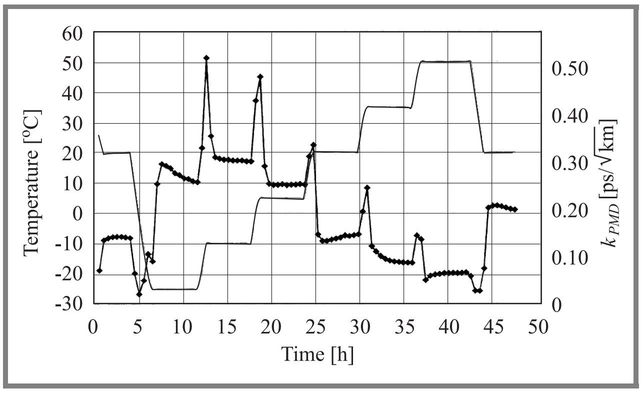


Fig. 9. Fiber PMD coefficient (bold line) and temperature (thin line) during temperature cycling of G.652 fiber in tight nylon buffer. PMD transients at each temperature transition are well visible.

2.3.3. Indoor cable with G.655 fiber

Cable selected for this test (TP SA OTO Lublin, Poland, W-NOTKSdD 1J5 2,0) had a single nonzero dispersion-shifted (NZDSF, ITU-T G.655) [9] fiber in a 0.9 mm UV-cured acrylate buffer, aramide strength member, low-smoke zero halogen jacket and external diameter of 2 mm. It was 4083 m long. Test results are presented in Fig. 10.

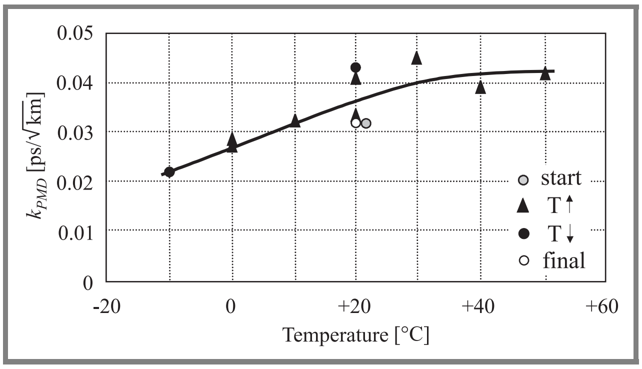


Fig. 10. PMD-temperature characteristics of indoor cable with G.655 fiber in UV-cured buffer.

This sample exhibited very interesting behavior:

- **Reduction** of PMD at temperatures below +20°C. This is opposed to **increase** of PMD in identical cable with G.652 fiber [1]; the difference must be due to special properties of G.655 fiber.
- No permanent increase of PMD. The buffer is made of cross-linked material with good dimensional stability.

2.3.4. Tight-buffered optical unit with stranded G.652 fibers

The test was performed on 170 m long 12-fiber optical unit extracted from optical ground wire (OPGW) made by AFL Telecommunications (USA). Design of this OPGW and its optical unit were presented in paper [1]. Fibers are stranded helically around a rigid strength member with a 125 mm pitch. For the test, all fibers were fusion spliced together, giving a total length of 2040 m.

OPGWs used in overhead high voltage lines have wide range of operating temperatures: –40…+85°C, so test conditions were chosen accordingly. This test was performed during COST-291 STSM [4].

Figure 11 shows results collected in steady-state conditions; transient data have been deleted.

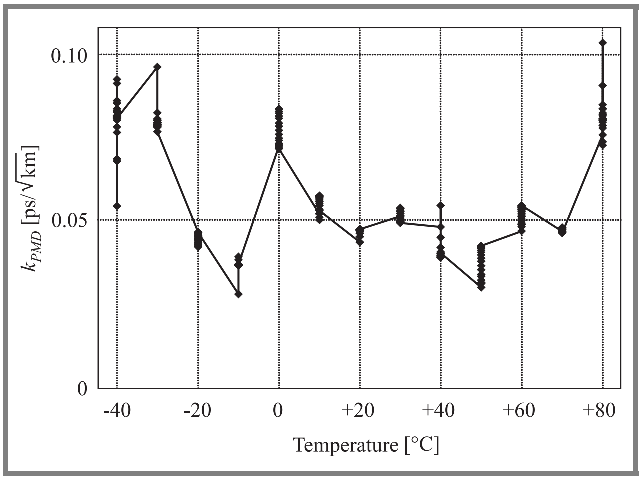


Fig. 11. PMD-temperature characteristics of tight-buffered OPGW unit with stranded G.652 fibers.

This sample has shown remarkable stability of PMD over a wide range of temperatures; some increase occurred only at **both** extreme temperatures: –40°C and +80°C. Most measured PMD values were close to 0.05 ps/$\sqrt{\text{km}}$.

2.4. PMD-temperature characteristics: loose tube cable

This was an optical ground wire with gel-filled central loose tube (Fig. 12) holding 24 loose optical fibers of non dispersion-shifted type (ITU-T G.652). Large space for movement of fibers minimize strain, bending and external pressure.

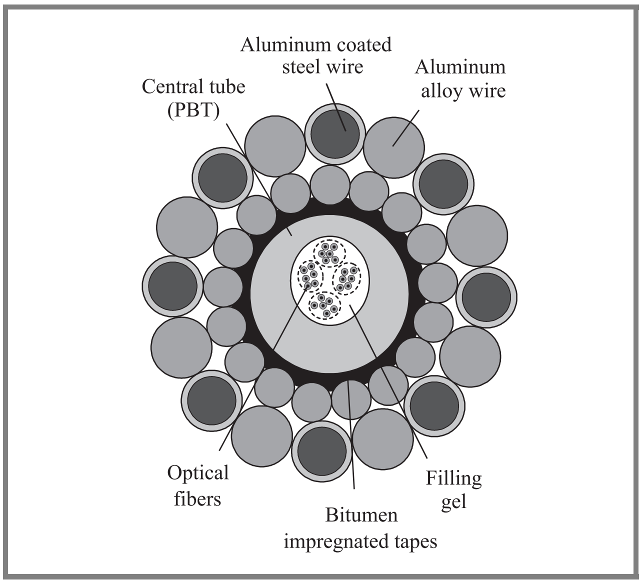


Fig. 12. Cross-section of loose-tube OPGW (NK Cables AACSR/AW SS-24F 64/34).

All 24 fibers in the 1310 m long OPGW were fusion spliced together, having a total length of 31.7 km. As seen on Fig. 13, there was little variation of fibers' PMD with temperature and no permanent increase after test. This confirms assumption that forces generated by shrinking or expanding tight buffer are primarily responsible for PMD instability observed in several other samples.

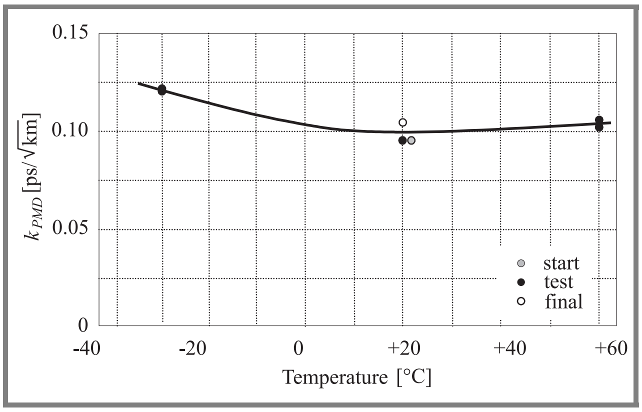


Fig. 13. PMD-temperature characteristics of loose-tube OPGW with G.652 fibers.

2.5. Strain-temperature characteristics: tight-buffered fibers

Temperature-induced changes in axial fiber strain were measured for two indoor cables, each with a single fiber having a 0.25 mm primary coating and 0.9 mm tight buffer:

A) Tele-Fonika W-NOTKSd 1J 2,0; fiber buffer was extruded of PBT;

B) OTO Lublin W-NOTKSdD 1J5 2,0; fiber buffer was made of UV-cured acrylate.

Both cables were subjected to temperatures from –20°C to +50°C and optical length of fibers was measured with optical time domain reflectometer (OTDR). Sample lengths were 4040 m (A) and 4083 m (B). OTDR measurements were performed at 1310 nm and 1550 nm wavelengths. Figure 14 shows variations in fiber length relative to reference measurements at +20°C. Results obtained at both wavelengths were averaged and corrected for elastooptic effect with a factor of 1.25.

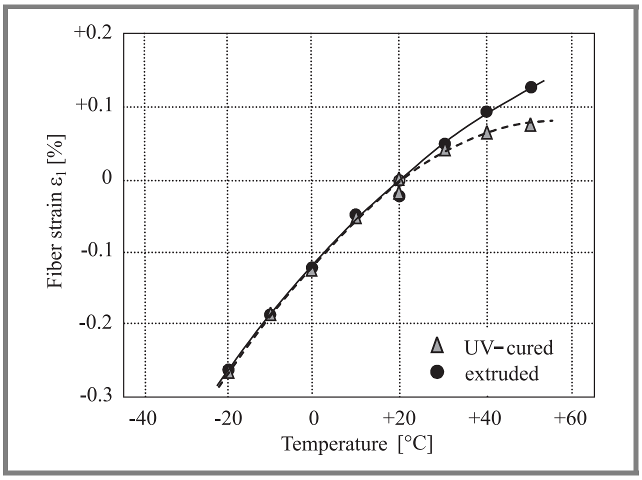


Fig. 14. Relative axial strain versus temperature: fibers in extruded and UV-cured 0.9 mm tight buffers.

Thermal expansion coefficients of both fibers at low temperatures are similar, about $7.5 \cdot 10^{-5}$/K at –10°C. However, fiber B has much lower zero-strain temperature (T_R), at which the strain approaches zero. Estimated T_R of this fiber is +60°C. Estimated T_R of fiber A in extruded buffer exceeds +100°C.

Lower T_R means smaller absolute compressive strain in fiber B – estimated to be 0.28% at the lowest operating temperature of –10°C, reduced tendency to buckling plus superior stability of PMD and attenuation. On the other hand, this fiber is also less tolerant to elongation caused by tensile forces.

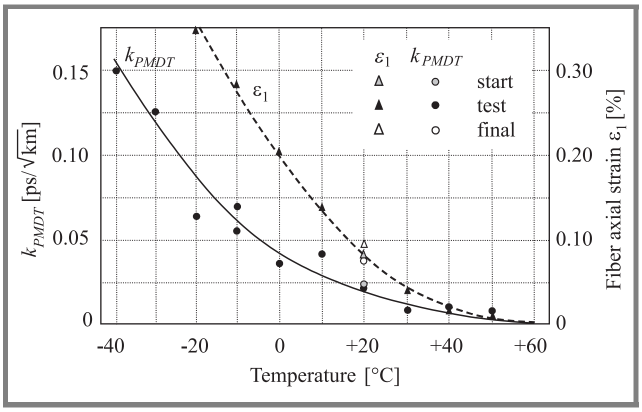


Fig. 15. Fiber strain (ε_1) and coefficient of thermally induced PMD (k_{PMDT}) in a G.652 fiber caused by shrinkage of 0.9 mm UV-cured acrylate buffer.

Data from this test allowed to analyze results of tests on PMD temperature dependence in a cable identical with "B", but containing a G.652 (unspun) fiber, presented earlier in paper [1]. Figure 15 shows a comparison between temperature dependence of fiber axial (compressive) strain and thermally induced PMD component. It is evident that induced PMD is roughly proportional to fiber strain.

3. Thermal and mechanical properties of fiber in tight buffer

3.1. Design and manufacturing of tight buffer

Typical tight-buffered fiber used in indoor, "universal" and some military cables is shown in Fig. 16. A thick layer of hard polymer is added over a standard primary coated fiber to make it stronger and easier to handle, especially when fitting optical connectors. Some buffer designs involve two layers: soft inner one, e.g., made of silicone and hard outer one, e.g., made of nylon.

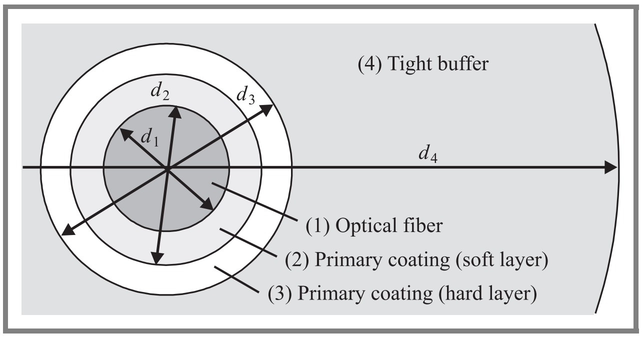

(4) Tight buffer
d_2
d_3
d_1
d_4
(1) Optical fiber
(2) Primary coating (soft layer)
(3) Primary coating (hard layer)

Fig. 16. Cross-section and dimensions of fiber in a typical 0.9 mm tight buffer.

Typical dimensions of buffered fiber for indoor cables and material data are listed in Table 1. Comparison of those data clearly indicates that it is the tight buffer, whose properties dictate mechanical forces acting on the glass fiber in variable temperatures; contribution from primary coating is marginal.

Tight buffers are applied to fibers using two methods:

1) extrusion of polymer, most often PBT, PVC or nylon – predominantly PA12 and its blends (Fig. 17);

2) UV-curing of liquid acrylate compounds on the fiber.

Commonly used extrusion process requires temperature of approximately +250°C. Melted polymer is usually rapidly cooled with cold water, in order to limit crystallization of PBT or polyamide and post-extrusion shrinkage. The partially ordered crystalline phase has a density approximately 2% higher than amorphous one [10], so this approach reduces undesirable buffer shrinkage and fiber strain. However, operation of fiber above the glass transition temperature of polymer (T_g), being about +40…+60°C for PBT and PA12, and subsequent slow cooling causes crystalline phase content to increase and the buffer to shrink further. This shrinkage may be reversed by heating above T_g again and rapid cooling, but results of such treatment are hardly predictable, as presented in Subsection 2.3.1. This problem is absent in cured and amorphous polymers.

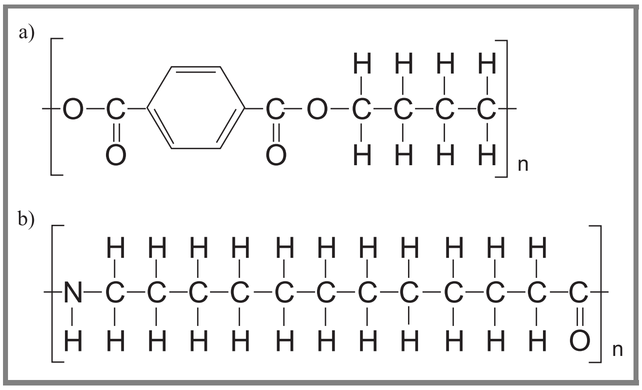


Fig. 17. Structures of common buffer materials: PBT (a) and PA12 (b).

Table 1
Thermo-mechanical data of Corning SMF-28 fiber in extruded PBT tight buffer at +20°C; layer numbering in accordance with Fig. 16

No.	Component (layer) of buffered optical fiber	Young modulus (E)	External diameter (d)	Cross-section	Thermal expansion coefficient (α)
		[MPa]	[mm]	[mm^2]	[K^{-1}]
1	Silica glass fiber	73000	0.125	0.0123	$5.5 \cdot 10^{-7}$
2	Soft primary coating	1.4	0.190	0.0161	$2.2 \cdot 10^{-4}$
3	Hard primary coating	700	0.250	0.0207	$8.0 \cdot 10^{-5}$
4	Tight buffer (PBT)	2500	0.900	0.5871	$1.3 \cdot 10^{-4}$

Buffered fiber is subject to axial compressive strain of 0.05–0.35% at room temperature, with zero stress temperature (T_R) between +50°C and +190°C [5]. This strain can be reduced if a suitable tension is applied to fiber during application of a buffer.

Curing of liquid material with ultraviolet light is in theory a "cold" process, but intense radiation required for high-speed curing can heat the fiber up to 100°C, and subsequent shrinkage is increased. Nevertheless, UV-cured buffer generally produces less strain on the fiber (see Subsection 2.5) and its circularity is better.

3.2. Temperature-dependent axial strain

Axial fiber strain increases with falling temperature due to contraction and rise of modulus of buffer material, according to simplified formula [5]:

$$\begin{aligned}\varepsilon_1 &= \frac{(d_4^2-d_3^2)E_4}{d_1^2E_1+(d_4^2-d_3^2)E_4}\int_T^{T_R}(\alpha_4-\alpha_1)dT \\ &\approx \frac{(d_4^2-d_3^2)E_4}{d_1^2E_1+(d_4^2-d_3^2)E_4}(T_R-T)\,\alpha_{4eff}\,, \qquad (1)\end{aligned}$$

where: ε_1 is fiber strain, T – operating temperature, T_R – zero stress temperature, d_4, d_3 – outer and inner diameter of hard buffer (Fig. 16), α_1, α_4 – thermal expansion coefficients of silica and buffer material, E_1, E_4 – Young moduli of silica and buffer, α_{4eff} – effective thermal expansion coefficient of buffer material between T and T_R.

Beginning from zero level at T_R, compressive strain increases with decrease of temperature, as shown in Figs. 14 and 15. Pure axial strain does not produce difference between refractive indices for polarization modes in the fiber core, so it doesn't affect fiber PMD.

3.3. Lateral strain

Internal strain in the fiber in the direction perpendicular to its axis produces a mechanically induced optical birefringence M and differential group delay $\Delta\tau$ in accordance with formulae below:

$$M = n_x - n_y = R(\sigma_x - \sigma_y)\,, \qquad (2)$$

$$\Delta\tau = \frac{[R]L}{c}(\sigma_x - \sigma_y)\,, \qquad (3)$$

where: M – mechanically induced birefringence, n_x, n_y – refractive indices in x and y directions, σ_x, σ_y – strains in x and y directions, R – the elastooptic coefficient, equal to $-3.15\cdot10^{-6}$/MPa for SiO$_2$ glass at 1550 nm, L – fiber length.

The coefficient of polarization mode dispersion component mechanically induced in a long fiber with uniform strain distribution is described by the formula:

$$k_{PMDM} = \frac{R}{c}\sqrt{\frac{h}{1000}}(\sigma_x - \sigma_y)\,, \qquad (4)$$

where: c – speed of light in vacuum, h – coupling length of polarization modes [m].

For a typical $h = 10$ m, a 1 MPa lateral differential strain produces PMD as high as 1.05 ps/$\sqrt{\text{km}}$.

In general, lateral strain is proportional to axial strain, as both result from polymer shrinkage. The latter is easier to measure, however – see Subsection 2.5.

At room temperature, the uneven lateral shrinkage or ellipticity of buffer affects the fiber relatively little, because the soft inner layer of coating prevents transfer of any significant pressure to glass fiber. This protection disappears below glass transition temperature (T_g) of coating, when it stiffens. Depending on particular material of primary coating, its T_g can be between 0°C and –80°C (see Fig. 18).

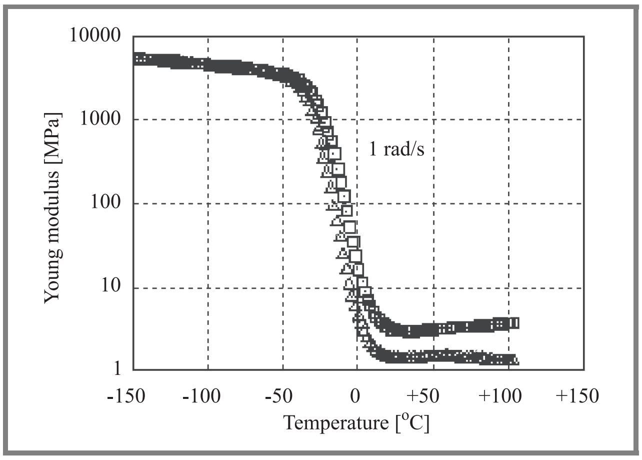


Fig. 18. Temperature dependence of modulus for two UV-cured acrylate compounds used for soft primary coatings [11].

When the buffer lacks perfect circularity, mechanically induced PMD appears and quickly grows at low temperatures approaching T_g of primary coating. This process is augmented by buffer shrinkage. However, it is possible that further cooling causes increased mixing of polarization modes due to unevenly distributed pressure, resulting in saturation of PMD growth or even some reduction of PMD, as presented in Subsection 2.3.2.

3.4. Fiber bending and buckling

Eccentricity of coating and buffer results in non-symmetry of longitudinal forces due to polymer shrinkage, causing fiber bending; bend curvature rises with falling temperature. Deformation of fiber produces mechanical strains and induces PMD in accordance with Ulrich formula [12]:

$$k_{PMDB} = \frac{k}{r_B^2}\,, \qquad (5)$$

where: k_{PMDB} – the bending-induced PMD coefficient [ps/$\sqrt{\text{km}}$], k – proportionality factor, being 200–3500 ps$\cdot$mm^2/$\sqrt{\text{km}}$ for G.652 fibers [5], r_B – fiber bending radius.

Excessive axial strain leads to fiber **buckling**: loss of mechanical stability and deformation of fiber into a helix or wave pattern. With a high quality buffer this process occurs suddenly at critical low temperature, in order of –40°C, causing abrupt increase of fiber loss and PMD.

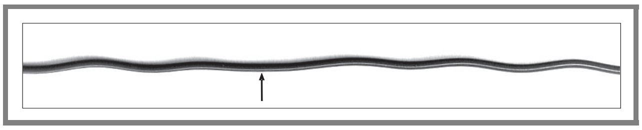


Fig. 19. Buckled fiber in a 0.9 mm buffer. The fiber has formed a reversible spiral, with arrow showing a reversal point. The image is shortened by a factor of 1:4 for better visibility.

If the buffer has frequent imperfections producing minute fiber bends, the latter grow gradually with decreasing temperature. In a bad quality product, buckling can occur even at room temperature, with distinct deformation and helix diameter up to 3 mm (see Fig. 19). Embedding of fiber in a cable severely restricts such deformation.

Helically deformed fiber moves inside soft coating and is pressed against the hard coating above. At the same time, a formerly rigid, straight fiber becomes a spring offering little resistance to axial shrinkage of polymeric buffer with decreasing temperature. Assuming that helix diameter is restricted by buffer size and cable design to a certain value d_{TS} (usually 0.1–1 mm), fiber bending radius can be estimated as

$$r_B = \frac{d_{TS}}{4\alpha_4 \cdot (T_B - T)} + \frac{d_{TS}}{2}, \qquad (6)$$

where: T_B is the critical buckling temperature, and T – operating temperature.

Substituting formula [5] into [6], one can estimate the PMD resulting from such a temperature-dependent bending:

$$k_{PMDT} \approx \frac{16k \cdot \alpha_4^2 \cdot (T_B - T)^2}{d_{TS}^2}. \qquad (7)$$

It must be noted, however, that a high quality buffered fiber shall **not** experience any significant buckling over the full range of specified operating temperatures. This still leaves the induction of PMD due to lateral strain, however.

Equation (7) explains why several manufacturers of industrial and military-grade cables have adopted 2-layer buffers, usually of 0.4/0.9 or 0.5/0.9 mm size: a soft inner layer of material like silicone allows to increase d_{TS} to 0.4–0.5 mm, dramatically reducing detrimental fiber bending at low temperatures.

Total PMD induced in the fiber due to mechanical influence of tight buffer shrinking with decrease of temperature can be estimated using a simplified, semi-empirical formula:

$$k_{PMDT} = k_1(T_1 - T)^2 + k_2(T_2 - T), \qquad (8)$$

where: k_{PMDT} – PMD coefficient resulting from thermal contraction of buffer and coating, k_1, k_2 – proportionality factors, T_1, T_2 – threshold temperatures.

Data obtained during temperature cycling tests of tight-buffered, un-spun single-mode fibers presented in paper [1] and here (see Figs. 8 and 15) confirm applicability of formula (8).

Fibers in indoor and field deployable cables sometimes have a "semi-tight" buffer – extruded loose tube of 0.9 mm diameter with a single fiber surrounded by a layer of gel some 0.05–0.10 mm thick. Unlike tight designs, the buffer is not mechanically bonded to fiber, so its extrusion shrinkage and thermal expansion are about twice as large in comparison to tight buffer. Unless considerable tension is applied during extrusion, fiber often suffers from excessive overlength and has high PMD due to bending and pressure against buffer walls, even across a full range of operating temperatures.

Total PMD coefficient of tight-buffered fiber at low temperature is given by the following formula:

$$k_{PMD} = \sqrt{k_{PMDG}^2 + k_{PMDT}^2}, \qquad (9)$$

where: k_{PMDG} is the "natural" PMD coefficient dictated by non-perfect fiber geometry, close to what is measured in a primary coated fiber at room temperature.

Contribution of thermo-mechanically induced PMD is noticeable only when it becomes comparable to "natural" PMD of a particular fiber, therefore high PMD fibers tend to exhibit superior PMD stability than low PMD ones, which need better mechanical protection to retain their properties.

4. Effects of fiber spinning and stranding on PMD

4.1. Spun fibers

Introduction of dispersion-shifted and nonzero-dispersion shifted single mode fibers (ITU-T G.653 and G.655) [9], characterized by smaller core size and increased doping levels with respect to standard non dispersion-shifted fiber (ITU-T G.652) has created difficulties with ensuring perfect fiber geometry and low PMD, especially as demands of customers kept growing. An elegant and almost universally adopted solution is "spinning" of fiber – rotation at constant or modulated rate, applied to fiber during drawing from a hot preform [13], first introduced to mass production of telecom fibers by AT&T in 1993. Semi-molten glass drawn from the tip of the preform is easily and permanently deformed. Spinning prevents accumulation of birefringence in a fiber drawn from preform with non-circular core, as this deformation is regularly rotated in different locations along the fiber and their individual contributions to birefringence cancel each other. Spinning, in fact, has been originally applied to make low-birefringence sensor fibers back in 1979 [14].

The simpler non dispersion-shifted fibers have largely continued to be drawn without spinning, albeit the process is becoming more common since 2005.

4.2. Stranding of optical units

Stranding of multiple units to form a cable or its core is a common process in the cable industry, extensively used in manufacturing of optical fiber cables. As no untwisting is usually applied later, this step leaves a permanent circular strain in the parts being stranded. This is normally not an issue with copper wires, aramide yarns or plastic parts, e.g., loose tubes in optical fiber cable.

While fibers inside stranded tubes can move and relax – especially when stranding is of reversible type, stranding of fibers to form a tight-buffered multi-fiber unit leaves a permanent strain, as any fiber movement is prevented by rigid coatings and buffers holding fibers and other parts together. The same can happed in a multi-fiber indoor cable, where fiber units have common sheath and yarn filling. As typical rates of twist do not affect fiber reliability or attenuation, this effect is usually ignored.

4.3. Circular strain and fiber PMD

Twisting of optical fiber creates circular strain and related circular birefringence, meaning a rotation of light polarization plane with rate proportional to fiber twist rate γ:

$$\delta = 2\pi g_s \gamma z = 2\pi \frac{g_s}{\Lambda_S} z, \tag{10}$$

where: δ – rotation angle of polarization plane, g_s – stress-optic rotation coefficient of optical fiber, γ – fiber twist rate [rev/m], Λ_S - stranding pitch [m], z – distance from the beginning of twisted fiber section.

The stress-optic rotation coefficient is defined by the following equation:

$$g_s = \frac{E_{SF} \cdot R}{n_2}, \tag{11}$$

where: E_{SF} – modulus of rigidity (shear modulus) of fiber core material, n_2 – refractive index of fiber core.

Substituting the following parameters of single-mode silica fiber operating at 1550 nm: $E_{SF} = 31000$ MPa, $R = -3.15 \cdot 10^{-6}$/MPa and $n_2 = 1.47$, one obtains $g_s = 0.0665$. Measurements of single mode telecom fibers give $g_s \approx 0.07$ [19, 20].

Fiber twisting produces two circular polarization modes with opposite rotation directions and circular birefringence [16], causing a differential group delay $\Delta\tau$ and twist-induced PMD (k_{SPMD}). For uniform twist, their values can be calculated as follows [5]:

$$\Delta\tau = 0.065\gamma L, \tag{12}$$

$$k_{SPMD} = 0.065\sqrt{\frac{h}{1000}}. \tag{13}$$

In the formulae above, h is expressed in [m], $\Delta\tau$ in [ps], and the PMD coefficient k_{SPMD} in [ps/$\sqrt{\text{km}}$].

Assuming a typical coupling length $h = 10$ m, fiber twisting at 8 rev/m rate, measured for the 12-fiber tight buffered optical unit of OPGW tested in our lab produces PMD with $k_{SPMD} = 0.052$ ps/$\sqrt{\text{km}}$. This corresponds pretty well to surprisingly repeatable PMD values measured in several experiments on this type of OPGW and optical unit extracted from it (see [1] and Subsection 2.3.4).

Besides generation of PMD by twisting, excellent stability of PMD suggests there is additional mechanism present in twisted fibers, capable of reducing PMD contributions resulting from mechanical influences presented in Section 3.

Periodic rotation of polarization plane prevents accumulation of $\Delta\tau$ in a fiber being subject to perpendicular mechanical stress with a (relatively) fixed direction, because locally generated contribution to birefringence experiences "modulation" and change of sign with respect to birefringence accumulated in the preceding length of fiber. The resulting reduction of mechanically-induced PMD is described by the following PMD reduction factor [5]:

$$\zeta_S \approx \frac{1}{\sqrt{1 + 0.0196\left(\frac{L_B}{\Lambda_S}\right)^2 + 0.0182\left(\frac{L_B}{\Lambda_S}\right)^4}}, \tag{14}$$

where: L_B – fiber beat length [m].

With a relatively strong twist, when $L_B/\Lambda_S \geq 5$, the PMD reduction factor is approximately proportional to square of the L_B/Λ_S ratio:

$$\zeta_S \approx 7.4\left(\frac{\Lambda_S}{L_B}\right)^2. \tag{15}$$

For the majority of telecom type single mode fibers, characterized by $L_B \geq 5$ m, a 100-fold reduction of PMD generated by external crush forces ($\zeta_S = 0.01$) can be ensured by twisting the fiber with a $\Lambda_S = 183$ mm pitch; this corresponds to a twist rate $\gamma = 5.44$ rev/m.

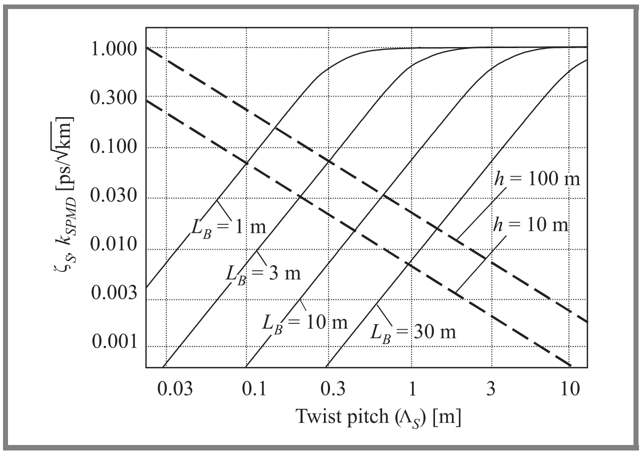


Fig. 20. Characteristics of PMD reduction factor, applicable to PMD induced by fixed direction crush forces (ζ_S) (continuous lines) and PMD introduced by twisting in a nonzero dispersion-shifted fiber (k_{SPMD})(λ = 1550 nm, dashed lines) versus fiber twist pitch.

Characteristics of PMD reduction factor and twist-generated polarization mode dispersion are plotted in Fig. 20. One shall notice, that excessive twisting, besides possible problems with mechanical design and reliability of optical fiber cable, can easily produce PMD high enough to severely degrade transmission performance of fiber, negating the beneficial effects of improved tolerance to bending and external forces.

It was also necessary to explain the apparently odd results obtained during low-temperature tests on tight-buffered nonzero dispersion-shifted fibers (NZDSF, ITU-T G.655), presented in Subsection 2.3.3 and paper [1]. Those fibers were not twisted, but still exhibited greatly superior resistance to mechanical induction of PMD in comparison to standard single-mode fibers (SMF, ITU-T G.652) operating in comparable conditions.

As indicated in Subsection 4.1, the difference lies in spinning of G.655 fibers. A by-product of this process is the presence of residual circular strain in the spun fiber, as the fiber cannot fully relax once it is colder and its glass becomes more viscous when leaving the melting zone.

Measurements of unidirectionally spun G.652 fibers [17] suggest that residual strain corresponds roughly to twisting of fiber at a rate equal to 10% of spinning rate. For a typical spinning rate of 5 rev/m, we get $\Lambda_S = 2$ m, which (see Fig. 20) provides at least a 10-fold reduction of mechanically induced PMD in a good quality fiber with $L_B \geq 20$ m. This explains superior stability of PMD in (spun) G.655 fibers in comparison to (unspun) G.652 fibers we have tested.

5. Conclusions

Both laboratory experiments and theoretical analysis have clearly proved that single-mode fibers placed in thick “tight” or “semi-tight” polymeric buffers of type commonly used in optical fiber cables can be susceptible to detrimental mechanical forces generated by the buffer, particularly at low temperatures. While fiber attenuation is usually not affected, its polarization mode dispersion in cold environment can increase dramatically, by up to 0.3 ps/$\sqrt{\text{km}}$ and even more. This effect has so far been overlooked by researchers and engineers responsible for cable and system specifications.

While harmful to operation of high-speed networks, this phenomenon can be used in research and diagnostics of optical fiber cables.

Induction of PMD by forces exerted by fiber buffer can be eliminated or reduced, when the fiber is either appropriately twisted during cable manufacture or spun during drawing. While both processes have been introduced by the industry for other reasons, they have an unintended beneficial side effect.

In particular, use of spun single-mode fibers in cables of tight buffer design intended for use at low temperatures or being subject to severe crush and bending is strongly recommended.

Acknowledgements

Most research work presented here was performed within COST Action 270 “Reliability of Optical Components and Devices in Communications Systems and Networks”, and financed by the Polish Government research grant no. 632/E-242/SPB/COST/T-11/DZ198/2003-2005. Additional test data were gathered during short-term scientific mission (STSM) to Politecnico di Torino and Instituto Superiore Mario Boella (ISMB), Turin, Italy, within COST Action 291 “Towards Digital Optical Networks”. The author is grateful to ISMB researchers Silvio Abrate and Augustino Nespola for their valuable assistance.

References

[1] K. Borzycki, “Influence of temperature and aging on polarization mode dispersion of tight-buffered optical fibers and cables”, *J. Telecommun. Inform. Technol.*, no. 3, pp. 96–104, 2005.

[2] K. Borzycki, “Temperature dependence of PMD in optical fibres and cables”, in *Proc. ICTON 2005 Conf.*, Barcelona, Spain, 2005, vol. 1, paper Tu.C3.7, pp. 441–444.

[3] K. Borzycki, M. Jaworski, and M. Marciniak, “Temperature dependence of PMD in tight buffered G.652 and G.655 single-mode fibers”, in *Proc. OC&I-2006/NOC-2006 Conf.*, Berlin, Germany, 2006, pp. 21–28.

[4] K. Borzycki, “Report on COST-291 short term scientific mission to Politecnico di Torino (Turin, Italy)”, July 2005.

[5] K. Borzycki, “Wpływ temperatury na dyspersję polaryzacyjną (PMD) jednomodowych włókien światłowodowych w pokryciach ścisłych” (Influence of temperature on polarization mode dispersion (PMD) of single-mode tight-buffered fibers), Ph.D. thesis, Warsaw, National Institute of Telecommunications, March 2006 (in Polish).

[6] “Transmission media characteristics – Optical fibre cables: Definitions and test methods for statistical and non-linear related attributes of single-mode fibre and cable”, ITU-T Rec. G.650.2 (01/2005).

[7] A. F. Judy, A. H. McCurdy, R. K. Boncek, and S. K. Kakar, “Fiber PMD – room for improvement”, in *Proc. NFOEC 2003 Conf.*, Orlando, USA, 2003, pp. 1208–1217.

[8] “Transmission media characteristics – Optical fibre cables: Characteristics of a single-mode optical fibre and cable”, ITU-T Rec. G.652 (06/2005).

[9] “Transmission media characteristics – Optical fibre cables: Characteristics of a non-zero dispersion-shifted single-mode optical fibre and cable”, ITU-T Rec. G.655 (03/2006).

[10] I. Gruin, *Materiały polimerowe*. Warszawa: Wydawnictwo Naukowe PWN, 2003 (in Polish).

[11] C. Aloisio, A. Hale, and K. Konstadinidis, “Optical fiber coating delamination using model coating materials”, in *Proc. 51st IWCS Conf.*, Orlando, USA, 2002, pp. 738–747.

[12] R. Ulrich, S. C. Rashleigh, and W. Eickhoff, “Bending-induced birefringence in single-mode fibers”, *Opt. Lett.*, vol. 5, no. 6, pp. 60–62, 1980.

[13] D. A. Nolan, X. Chen, and M.-J. Li, “Fibers with low polarization-mode dispersion”, *J. Lightw. Technol.*, vol. 22, no. 4, pp. 1066–1077, 2004.

[14] S. R. Norman, D. N. Payne, and M. J. Adams, “Fabrication of single-mode fibres exhibiting extremely low polarisation birefringence”, *Electron. Lett.*, vol. 15, no. 11, pp. 309–311, 1979.

[15] A. J. Barlow and D. N. Payne, “The stress-optic effect in optical fibers”, *IEEE J. Quant. Electron.*, vol. QE-19, no. 5, pp. 834–839, 1983.

[16] T. Chartier, C. Greverie, L. Selle, L. Carlus, G. Bouquest, and L.-A. de Montmorillon, “Measurement of the stress-optic coefficient of single-mode fibers using a magneto-optic method”, *Opt. Expr.*, vol. 11, no. 20, pp. 2561–2566, 2003.

[17] D. Sarchi and G. Roba, “PMD mitigation through constant spinning and twist control: experimental results”, in *Proc. OFC’2003 Conf.*, Atlanta, USA, 2003, paper WJ2, pp. 367–368.

Krzysztof Borzycki was born in Warsaw, Poland, in 1959. He received the M.Sc. degree in electrical engineering from Warsaw University of Technology in 1982 and a Ph.D. degree in communications engineering from National Institute of Telecommunications (NIT), Warsaw in 2006. He has been with NIT since 1982, working on optical fiber fusion splicing, design of optical fiber transmission systems, measurement methods and test equipment for optical networks, as well as standardization and conformance testing of optical fiber cables, SDH equipment and DWDM systems. Other activities included being a lecturer and instructor in fiber optics and technical advisor to Polish fiber cable industry. He has also been with Ericsson AB research laboratories in Stockholm, Sweden, working on development and testing of DWDM systems for long-distance and metropolitan networks between 2001 and 2002 while on leave from NIT. He has been engaged in several European research projects since 2003, including Network of Excellence in Micro-Optics (NEMO) and COST Actions 270, 291 and 299. This included research on polarization mode dispersion in fibers and cables operating in extreme environments, and testing of specialty fibers. Another area of his work is use of optical fibers in electric power cables and networks. Dr. Borzycki is an author or co-author of 1 book, 2 Polish patents and over 40 scientific papers in the field of optical fiber communications, optical fibers and measurements in optical networks, as well as one of "Journal of Telecommunications and Information Technology" editors.
e-mail: k.borzycki@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

Paper

Rain precipitation in terrestrial and satellite radio links

Jan Bogucki and Ewa Wielowieyska

Abstract—This paper covers unavailability of terrestrial and satellite line-of-sight radio links due to rain. To evaluate the rain effects over communication system, it is essential to know the temporal and spatial evolution of rainfall rate. Long-term 1-min average rain-rate characteristics necessary for the design of microwave radio links are determined for central Poland. 1-min rain rate distributions are presented. It also describes comparison results of predicted attenuation obtained from ITU-R formula and empirical data at frequency bands 11 GHz and 18 GHz and satellite 12 GHz. The National Institute of Telecommunications stores 11-year long rain intensity characteristics (1985–1996), based on data derived on 15.4 km long experimental path. In this paper chosen experimental data are presented.

Keywords—satellite and line-of-sight radio links, propagation, rain fading.

1. Introduction

The quality of microwave signal transmission above 8 GHz is dependent on precipitation [1, 2]. The most significant contribution to atmospheric attenuation is due to the rain [3]. Nowadays numerous prediction models for rain attenuation exist. Generally, these prediction methods employ two stages of modeling. The first stage involves the prediction of rain fall rate probability distribution. The second stage relates the path attenuation with the specific attenuation, which is expressed in terms of the rain fall rate and path length.

Fading due to rain is the dominating degradation factor on radio-relays at the frequencies above 20 GHz and is essentially comparable to fading due to the layering of the atmosphere in the frequency range from 8 GHz to 20 GHz.

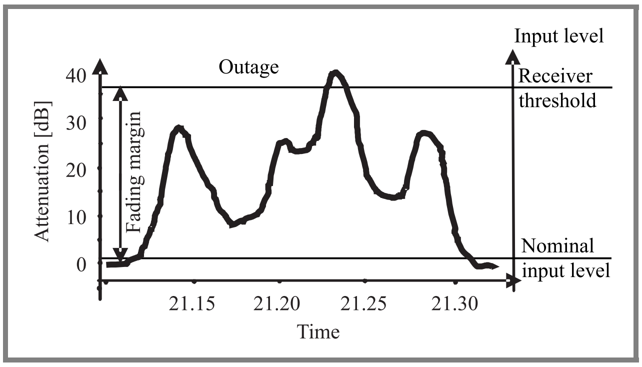


Fig. 1. An example of rain attenuation of 18.6 GHz terrestrial path.

The received signal varies with time and the system performance is determined by the probability for the signal to drop below the radio threshold level (Fig. 1).

It is very important to estimate degradation of radio-relays. Rain fall in the space separating the transmitter and receiver may sometimes cause detrimental effects to the received signal.

This type of fading has been well characterized by yearly measurements resulting in system design methods that limit outages within international standard limits [4]. For this reason it is necessary to develop an experimental network which provides the adequate data to study, prevent and compensate for rain fade. In this paper an experimental rain fall characteristics are presented.

2. Rain fading

2.1. Rain predictions

There are many methods which permit to estimate rain attenuation on radio link paths. The most popular are the ITU-R models [5, 6, 7]. At the National Institute of Telecommunications (NIT) TrasaZ and TraSat computer programs which implement these models have been worked out. They are radio frequency propagation computer programs for transmission path between RF transmitter and receiver. The TrasaZ program calculates fades expected from rain and multipath, attenuation by atmospheric gases, interference, diversity, Fresnel zone as well as determines antenna heights, reduction of cross-polar discrimination and power budget. Earth curvature for standard or sub-standard atmosphere is taken into account. The program frequency range is from 1 GHz to 60 GHz. The TraSat program calculates fades expected from rain, attenuation by atmospheric gases, and power budget of the satellite link for different angles of elevation and geographic location [8].

In the ITU-R model, input data to compute attenuation due to rain are: rain rate $R_{0.01}$ exceeded by 0.01% of the time, effective path length and geographic location [9] and also 0° isotherm height for the satellite path.

2.2. Measurement system

Taking into consideration the nature and the necessity of research on propagation effects occurring in radio links, detailed instructions and requirements, which should be met to prepare a self-operating measuring site, have been described in [9].

The radio links have been developed to test propagation at frequencies 11.5 GHz and 18.6 GHz terrestrial and 12.5 GHz satellite links. Along the path 5 rain gauges were situated (*s1*, ..., *s5*) – see Fig. 2. Terrestrial and satellite receivers were situated at the NIT.

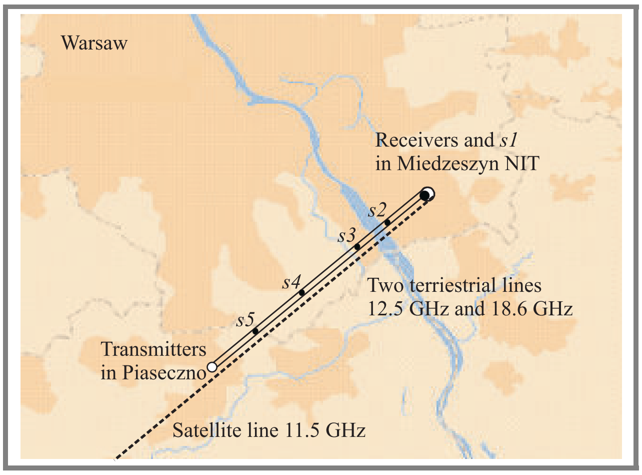


Fig. 2. Map of situated experimental links.

This paper describes the example test results of wave attenuation in above mentioned radio links. National Institute of Telecommunications carried out research on all phenomena causing fading on line-of-sight radio links and then selected only rain fading.

Rain rates and rain attenuation in mentioned radio links were measured in NIT in Miedzeszyn, near Warsaw and statistic distributions were carried out [10].

2.3. Rain measurements

Rainfall is measured in millimeters and the rain intensity in millimeters per hour. An important parameter is the integration time. Typical values of the integration time are 1 min, 5 min, 10 min and 1 hour. An integration time of 1 min should be used for rain intensity in link calculations.

Two types of rain gauges are used in the propagation experiments: capacitor gauges and typing bucket gauges. They have different dynamic ranges, integration time and calibration problems. In our measurement system typing bucket gauges were applied. Their parameters are:

- 1 tip/min corresponded to rain rate of 2.8 mm/h.
- Rain rates from this value down to 0.28 mm/h were calculated by programme application, which averaged single tips in the gaps shorter than 10 min. Longer gaps were considered as the breaks between the rain events.
- Rain intensity was measured in mm/h with values for integration of 1 min.

3. Measurement results and prediction

3.1. Data processing

Received signal samples were used for calculation of monthly and annual fading distributions as well as distributions for worst months. In this last case the formula for selected fading level A [dB] was applied:

$$p_{nm} = \max(p_1, p_2, \ldots, p_{12}),$$

where: $p_1, p_2, \ldots, p_{12}$ – percentage of A [dB] level exceedances in successive months of the year.

Software package was written to analyze propagation data.

3.2. Some rain statistical characteristics

The rain data are important because the percentage of time for which given value of rain attenuation is exceeded can be calculated from the rainfall rate $R_{0.01}$. The $R_{0.01}$ is the rain rate expressed in mm/h exceeded at the considered location by 0.01% of an average year.

Attenuation produced by rain can be caused by rain anywhere along the path where the air temperature is warm enough to maintain liquid raindrops. Rain can occur over the rain gauge but not cover the rest of the path or, conversely, be over most of the path but not over the rain gauge.

Figure 3a presents the results of minute-by-minute comparison of attenuation and rain-rate measurements. Along the path 5 rain gauges were situated as shown in Fig. 3a where rain rates in successive minutes at the five sites during a severe storm are presented. The path attenuation at 18.6 GHz and 11.5 GHz on the line-of-sight radio links and 12.5 GHz satellite path are also shown. In this example the line of columns passed the path at different times.

Figure 3b shows the area to time distribution of rainfall intensity along the path of this rainfall storm. The duration of storm lasted about 18 minutes.

Figures 4 and 5 present the empirical rain rate characteristics. Annual and averaged rain rate distributions for the period of five years are presented in Fig. 4. For the small rain rates, below 10 mm/h, results are the same in each year. Rain rates distributions for high rain rates changed markedly. For example at 0.001% the rain rate threshold increased from 43 in second year up to 106 mm/h in third year.

The stabilization of distributions cumulated in periods of one, two, three, four and five years is presented in Fig. 5.

3.3. Prediction and empirical data at the 12.5 GHz satellite link

The measurements of 12.5 GHz beacon signal from Lucz 1 were conducted and simultaneously of 1-minute average rain rate under the Earth-Lucz path. The projection of the satellite link on the Earth agrees with the terrestrial path. Received antenna elevation angle was 22° and azimuth 224.3°.

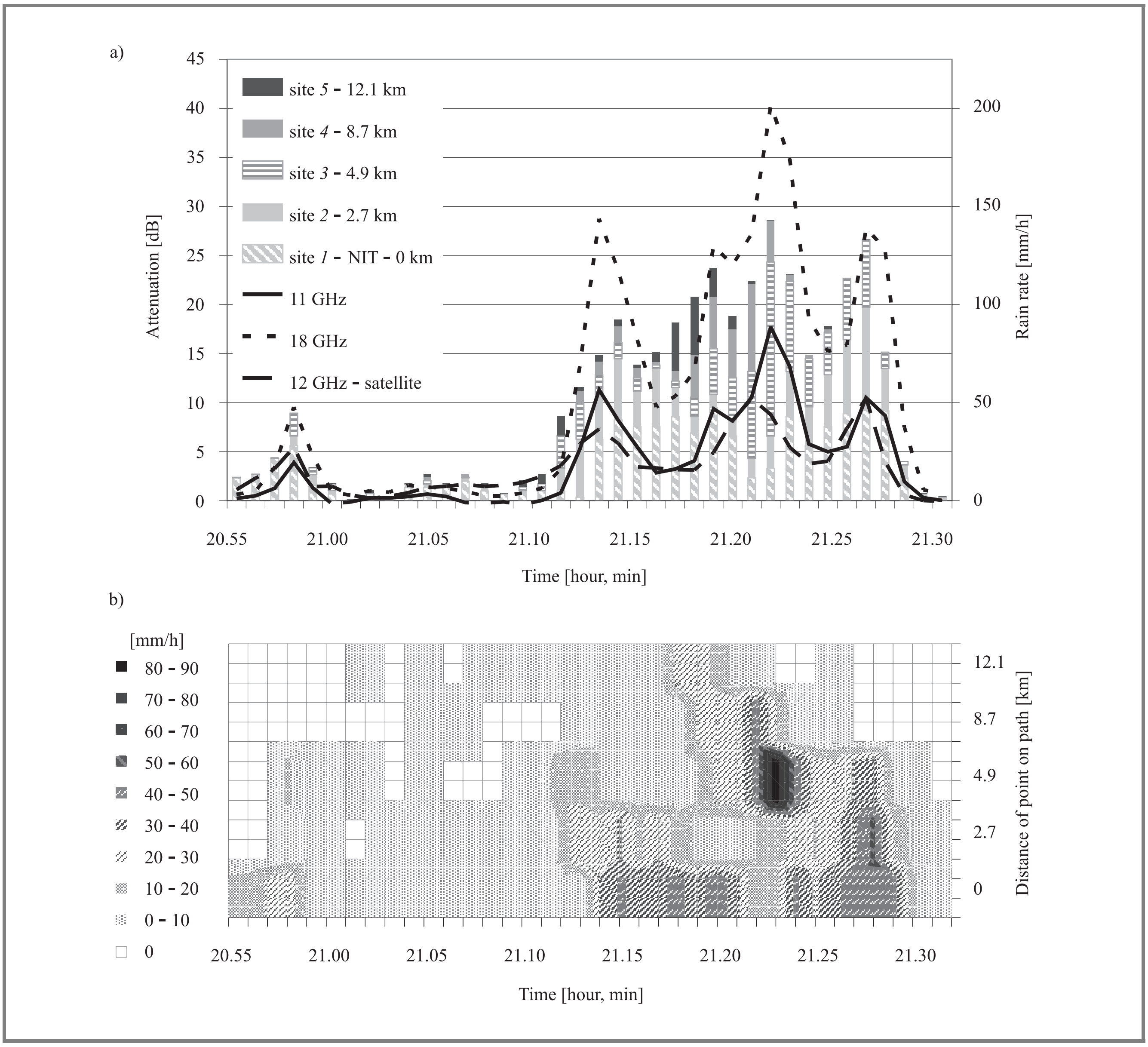


Fig. 3. An example of the storm along 15.4 km path: (a) point rain-rates along the path and path attenuation at 18.6 GHz, 11.5 GHz terrestrials and 12.5 GHz satellite; (b) spatial-temporal distribution of rainfall intensity along the path.

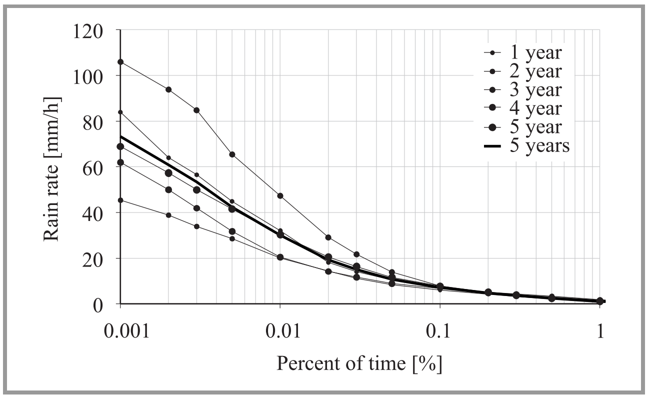


Fig. 4. Annual and averaged rain rate distributions for period of five years.

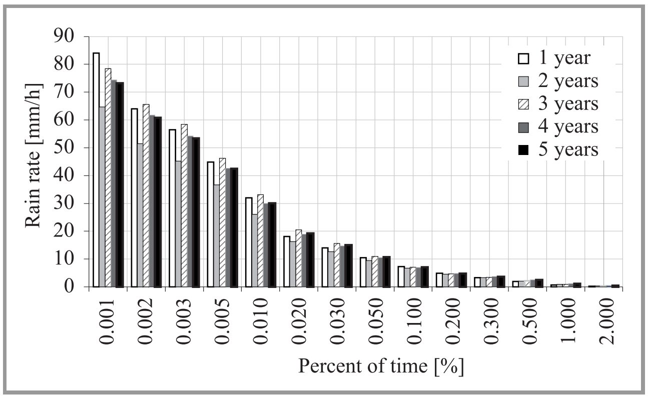


Fig. 5. Cumulated distributions of rain rate in consecutive years.

Having in mind unstable position of satellite and the lack of antenna tracking facility, the signal samples were processed in a special way in order to obtain zero level during the event. 1 minute average values of samples in several minutes before the event and several minutes after the event were calculated. Straight line, which connected both average levels, was taken as zero level for the event.

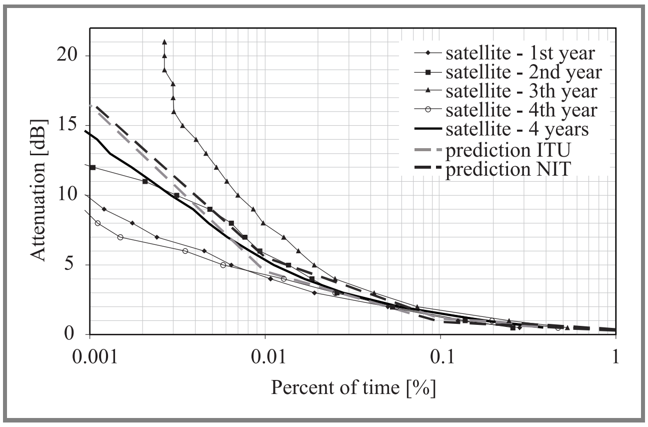


Fig. 6. Measured and calculated distributions of attenuation at 12.5 GHz satellite path.

Annual distributions of attenuation due to rain are presented in Fig. 6. During one storm with very intense rainfall the signal exceeded 20 dB level in this case during 10 minutes. The empirical annual attenuation distribution was compared with predicted distribution, based on model of ITU-R [6].

In this prediction model the input data to calculate attenuation due to rain are:

- data in world database recommended by ITU-R (prediction ITU Fig. 6);
- 0° isotherm height recommended by the Institute of Meteorology and Water Management (IMGW) in Warsaw and rain intensities measured by NIT in Warsaw (prediction NIT Fig. 6).

The biggest difference between empirical average of 4 years and prediction distributions is at 0.001% and is less than 2.4 dB for both prediction distributions. For bigger percentage, it was found out that prediction NIT model at 12.5 GHz satellite links is better than prediction ITU model. The absolute value of the biggest difference from empirical distribution for ITU distribution is 1 dB and for NIT distribution is 0.5 dB.

3.4. Prediction and empirical data at frequencies 11.5 GHz and 18.6 GHz terrestrial links

The terrestrial path of 15.4 km length operated continuously at frequencies of 11.5 GHz and 18.6 GHz [11]. The attenuations due to rain have been selected from other events. Figure 7 presents the rain attenuation statistics, show the percentages of the year that attenuation level *A* [dB] has been exceeded in case of rain on those two paths. Statistics do not include the events with melting snow. The annual attenuation distributions in five year period at 11.5 GHz and 18.6 GHz are presented in Fig. 7.

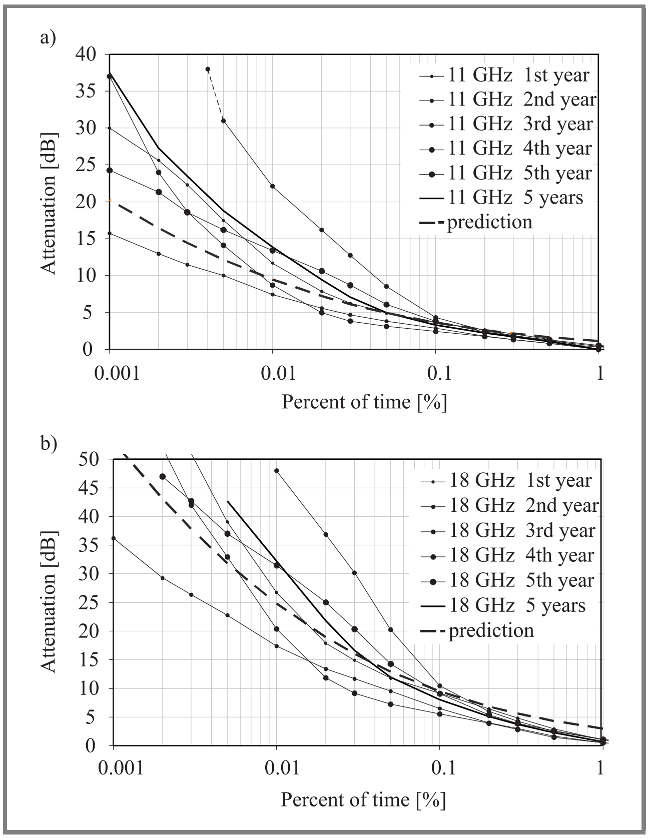


Fig. 7. Measured and calculated distributions of attenuation at (a) 11.5 GHz and (b) 18.6 GHz.

Mathematical description of 11.5 GHz paths corresponds to experimental data of attenuation caused by rain for percents greater than 0.03% time; the difference is below 1 dB (Fig. 7a). But measurement results done in the third year of measurement differ a lot.

Mathematical description of 18.6 GHz paths corresponds to experimental data of attenuation caused by rain for percentage greater then 0.03%; maximum difference measured is 2.3 dB. The maximum difference is 10.8 dB for 0.005% time (Fig. 7b).

4. Conclusion

Microwave radio links can be properly and precisely engineered to overcome potentially detrimental propagation effects. One of the characteristics that must be taken into consideration is rain attenuation. Some power margin has to be incorporated into the network design to allow for the amount of power reduction of received carriers due to rain. Prediction model does not entirely correspond to measurement results because only statistical relationship between rain rate and attenuation is possible.

The data calculated using the ITU-R model differ from experimental data for 11.5 GHz band for small percentage.

It was found out that predicted attenuations at 11.5 GHz radio links are less than measured. And for 18.6 GHz band attenuation measured for small percentage is larger than predicted. ITU-R model corresponds to measurement results in 12.5 GHz satellite link.

The results of rain measurements done in the third year differ a lot – 32 mm/h greater than average value for 0.001%. According to research done at NIT in Warsaw the measured rain intensities are larger than data in world database recommended by ITU-R.

Knowledge of the fading statistics is extremely important for the design of wireless systems. For microwave radio link design it is best to get local cumulative rainfall data. Unfortunately, for many places, there are no data on rain intensities averaged in 1 minute, and we are forced to use global values according to the climatic zone of the link.

If reception frequently cuts in and out of during light rainstorm, this is a good indication that the system has not been peaked to maximum performance. Most of existing rain attenuation prediction models do not appear to perform well in high rainfall regions. Further work should be devoted to parameterize the effects of rain path attenuation in order to provide a quantitative estimation useful for communication system design.

References

[1] J. Bogucki and E. Wielowieyska, "Niezawodność horyzontowych linii radiowych – aspekt praktyczny", in *KKRRiT Conf.*, Wrocław, Poland, 2003, pp. 121–124 (in Polish).

[2] J. Bogucki and E. Wielowieyska, "Multipath in line-of-sight links – prediction vs reality", in *Proc. 16th Int. Conf. Radioelektronika 2006*, Bratislava, Slovakia, 2006, pp. 259–262.

[3] H. R. Anderson, *Fixed Broadband Wireless*. Chichester: Wiley, 2003, pp. 127–164.

[4] R. K. Crane, *Propagation Handbook for Wireless Communications System Design.* London: CRC Press, 2003, pp. 225–280.

[5] "Propagation data and prediction methods required for the design of terrestrial line-of-sight systems", ITU-R Rec. P.530-12 (02/2007).

[6] "Propagation data and prediction methods required for the design of Earth-space telecommunication systems", ITU-R Rec. P.618-9 (08/2007).

[7] "Specific attenuation model for rain for use in prediction methods", ITU-R Rec. P.838-3 (03/2005).

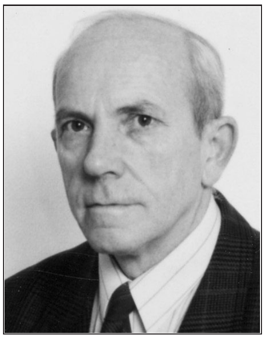
[8] J. Bogucki and E. Wielowieyska, "Elewacja a tłumienie propagacyjne na trasie nachylonej w zakresie fal milimetrowych", in *XI Symp. URSI 2005*, Poznań, Poland, 2005, pp. 411–414 (in Polish).

[9] J. Bogucki and E. Wielowieyska, "Propagation reliability of line-of-sight radio relay systems above 10 GHz", in *17th Int. Wrocław Symp. Exhib. Electromagn. Compatib.*, Wrocław, Poland, 2004, pp. 37–40.

[10] A. Kawecki, "Finalne charakterystyki propagacji mikrofal na trasie doświadczalnej Instytutu Łączności", *Prace IŁ*, no. 109, 1997 (in Polish).

[11] J. Bogucki, "Precipitation in radio communication", in *Proc. 17th Int. Conf. Radioelektronika 2007*, Brno, Czech Republic, 2007, pp. 325–328.

[12] J. Bogucki and E. Wielowieyska, "Reliability of line-of-sight radio-relay systems", *J. Telecommun. Inform. Technol.*, no. 1, 2006, pp. 87–92.



Jan Bogucki was born in Warsaw, Poland. He graduated Eng. degree at Technical University of Warsaw (1972). Since 1973 he has been employed at National Institute of Telecommunications, Warsaw, where he has been engaged in radiolinks digital, digital television and microwave propagation in the troposphere.

e-mail: J.Bogucki@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland

Ewa Wielowieyska was born in Warsaw, Poland. She finished Mathematics Faculty of Warsaw University. Since 1981 she has been employed at National Institute of Telecommunications, Warsaw, where she has been engaged in microwave propagation in the troposphere, propagation digital radio signals on short, medium and long waves.

e-mail: E.Wielowieyska@itl.waw.pl
National Institute of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland