An Edge-based Approach for Improving TCP in Wireless Mobile Ad Hoc Networks*

Ruy de Oliveira, Torsten Braun, Marc Heissenbüttel
Institute of Computer Science and Applied Mathematics
University of Berne
Neubrueckstrasse 10, CH-3012, Berne, Switzerland
{oliveira,braun, heissen}@iam.unibe.ch

Keywords: Wireless ad hoc networks, Round Trip Time, Congestion Control, Packet loss, End nodes.

Abstract

The Transmission Control Protocol (TCP) experiences severe performance degradation within wireless mobile ad hoc networks. The main reason is that the regular TCP does not discriminate packet losses due to medium error and disconnection (mobility) from congestion-induced losses. In this paper, we present the general ideas of our proposed end-to-end approach for enchancing TCP in this critical environment. The base of this approach is to involve only the end nodes in the congestion control performed by TCP. In other words, no specific cooperation from intermediate nodes is needed. Rather, a TCP sender monitors the network state and uses this input to react correctly. Using this approach, different monitoring techniques may be used to infer the internal network state. We propose to evaluate the reliability of using RTT variation monitoring as congestion indication. Some preliminary simulation evaluations in this direction are presented as well.

INTRODUCTION

In recent years, mobile IP networks with wireless support have experienced incredible widespread deployment all over the world. Nevertheless, the traditional cellular network architecture, where only the last hop is wireless, does not provide support for mobile nodes that desire or have to communicate directly among themselves. Thus, ad hoc networks come into place. In ad hoc networks, nodes do not need any existing infrastructure to communicate. The nodes work as both, host and router, as they not only exchange data with their peers but also forward traffic for other nodes. Thus, multiple hops can exist between sender and receiver.

Ad-hoc networks pose some tough challenges to TCP [1] as it was not designed to work in such highly dynamic and unpredictable environments. Rather, TCP was designed to work in wired networks where packet losses can safely be associated to network congestion. So, regular TCP relies on the assumption that any packet loss is the result of congestion inside the network, and appropriately reponses to it by slowing down its transmission rate.

In wireless mobile ad hoc scenarios, however, such losses may occur not only by congestion but also due to transmission errors and mobility. Transmission errors induce TCP to mistakenly reduce its sending rate (by halving its congestion window - CWND), and mobility-induced losses (link breakage) may lead TCP to incredibly long periods of inactivity due to its “exponential backoff mechanism”. This mechanism imposes exponentially increasing delay between every unsuccessfully consecutive retransmission attempt [1-2]. In situations in which a link restoration (due to mobility) takes place just after a long delay interval has been triggered, the exponential backoff mechanism will prevent TCP from retransmitting for a relevant period of time, which may last over "one minute". Thus, enhancements are indeed needed.

Improved TCP senders can either receive information from inside the network to learn about the reason of lost packets or infer it by doing permanent end-to-end measurements. Our approach relies on the latter case. The idea is to monitor the TCP flow and record useful data to infer the current state of the network when lost packets are perceived. In this way, changes are needed at the end nodes only.

The core of our proposal regards the mechanism to infer congestion inside the network. We evaluate the characteristics of round trip time (RTT) in this scenario to check whether this is a proper congestion indicator for this framework. We are fully aware that the simulation results herein are somewhat predictable, but our main goal has been to make sure that our simulations were consistent with what we expected. In addition, particular situations such as changes in the number of hops imposed by nodes movements had to be evaluated as we have not found any in the literature.

The remainder of this paper is organized as follows. In the next section we present the main related works in this area. The following section outlines the principles of our edge-based approach. The subsequent section presents the simulation evaluations, and lastly we conclude the work.

* The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

RELATED WORK

A number of research works have been devoted to improve TCP performance in a wireless environment. In terms of ad-hoc networks, however, not much has yet been done. In [3], we have discussed the main problems faced by TCP in such an environment and pointed out potential solutions. In general, the existing proposals can be divided into network-oriented and end-to-end approaches.

The main proposals for network-oriented approaches are [3-5]. Basically, these approaches rely on message notifications from inside the network to detect congestion or link interruptions by mobility. These messages are carried either by specially adapted routing protocol or by the regular TCP flow with bits explicitly set in the packet header (ECN scheme).

End-to-end approaches, in which only the end nodes exchange control data, are relatively recent. In [3], we have outlined our basic approach in this sense, and in this paper we give more details on it including basic simulation results. Recently, [7] has proposed a mechanism based on multiple metric in terms of monitoring parameters, which we also intend to develop in our approach. That work used four distinct metrics to isolate more accurately the reason of any packet lost.

With regards to RTT evaluations, [6-8] have worked on that, considering different methodologies, and their results seem to be a little divergent.

EDGE-BASED APPROACH

The key idea of this approach is to provide the TCP sender with updated information about the congestion state of the network. In this way, the sender can combine such information with other elements (e.g, length of last interval without any packet receipt, number of recent retransmissions, etc.) and react appropriately. Provided that the network state information is reliable, we do not change the normal behavior of TCP in dealing with congestion, which has proved over the years to be somewhat efficient.

Several algorithms might be used for inferring the congestion state inside the network such as loss rate and/or RTT variation measurements, and explicit probing. At first, we intend to use RTT measurements as a congestion indication parameter. Our idea is not to interfere with the normal TCP operation, but only enhance it, whenever it is possible. So, the sender has to record the recent past measurements and use them when lost packets are detected by either 3 duplicate acknowledgements (DACKs) or timeout. The pseudo code in Fig.1 outlines the changes in TCP sender actions.

According to our general algorithm shown in Fig. 1, when the sender receives 3 DACKs, it checks the network state for congestion. If it gets a positive response, then the normal congestion control (CC) mechanism is invoked, because such a condition indicates loss due to congestion, in which case the sender should keep its original behavior (half CWND). Otherwise, a random loss caused by the medium is inferred, and it justifies a direct retransmission without invoking the CC mechanism, since it would impose a needless decrease on the sending rate.

```
3 Dacks                                  ; ack loss
if (congestion)
    invoke CC                            ; loss due to cong.
else
   simply retransmit                     ; medium loss
Timeout                                  ;pck or ack losses or disc.
if (congestion & recent pck received)    ; recent pck rec. ~ RTO
    invoke CC                            ; loss due to cong.
else
    put sender into "probe mode"         ; disconn. under cong.
                                         ; or not
```

Figure 1. Pseudo code of TCP sender actions.

The TCP sender may be led into timeout by losses due to congestion or disconnection (mobility). In order to distinguish between timeout by congestion and timeout by disconnection, the sender keeps track of the instant of the last received packet for the specific session. This is represented by the "recent pck received" variable in the pseudo code. Using such information (that may be derived from RTO), the sender can determine whether a recent packet has been received, which indicates that there is connectivity. If so, the loss is associated with congestion, and then the CC mechanism is called. Otherwise, a disconnection is inferred and the sender is put into "probe mode", in which the sender will probe the network until it receives a reply from the receiver.

Hence, using this algorithm, the sender will be prevented from invoking the CC when it is not the best choice. This implies both avoiding the problem of the "exponential backoff mechanism" and improving the end-to-end throughput.

The state diagram of Fig. 2 shows the possible states for the TCP sender. Note that the sender only leaves the "normal" state if a packet loss is perceived, by either timeout or 3 DACKs, under non-congestion situations.

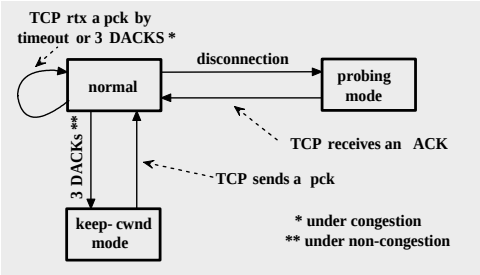


Figure 2. State diagram.

The state “keep-cwnd” is a transitory one in which the sender simply retransmits the missing packet without decreasing the current CWND or changing any other variable.

SIMULATION ANALYSIS

In this section, we present our preliminary simulation results using the ns-2 simulator. This evaluation has been carried out to evaluate the viability of using RTT measurements to infer congestion inside the network. As mentioned earlier, RTT features have already been evaluate in previous researches, but not in the exact sense as done here.

Basic simulation setup

Unless otherwise mentioned, the basic simulation settings consist of a flat area of 1000m x 1000m, in which the communication range of every node is 250 meters. The wireless bandwidth is 1 Mbps, and except in one case, explicitly mentioned, DSR [11] is the routing protocol in place. The sender transfers continuously a file (infinite FTP) to the receiver.

In order to create congestion conditions, two other TCP connections (also with FTP traffic generator) are used. We call the main connection C1 and the other two C2 and C3. The flow through C1 exists during the whole simulation time which lasts between 150 and 200 seconds.

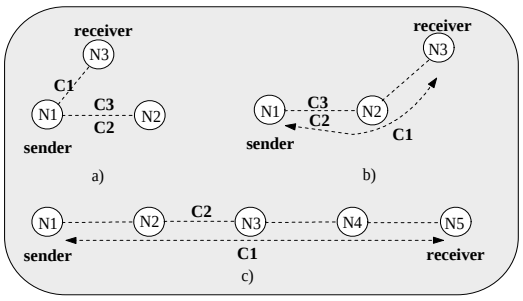


Figure 3. Simulation scenarios.

End nodes communicating directly (1-hop)

The idea here is to observe how RTT behaves in a full wireless shared medium, where all nodes share a common access medium (1-hop). For that, we use the scenario with three nodes depicted in Fig. 3a. Node 1 (N1) is the sender and Node 3 (N3) is the receiver. The other two connections for congestion are set up between N1 and N2 and start at 50 and 100 seconds, respectively.

Fig. 4a shows the measured RTTs for three different packet sizes, namely 250, 1000 and 1500 bytes. The results are somewhat obvious because for larger packet sizes higher RTTs should indeed be measured. Additionally, Fig. 4a shows clearly that during the period the network is facing relatively low congestion with the insertion of connection 2 only, the observed RTTs experience stable oscillations around an average. After 100 seconds, however, the magnitude of the oscillations becomes much higher as the result of high level of packet losses and retransmissions.

End nodes communicating via a third node (2-hops)

In this scenario, we have one more hop involved in the main connection between the end nodes (C1). Fig. 3b shows the topology used, in which the bottleneck is built between nodes N1 and N2. Connections C2 and C3 start at the same instants as in the previous scenario. The purpose of these simulations is to determine how much one extra hop can influence the RTT variations.

In principle, the results are also predictable in that they reveal that the RTTs here experience the same trend of the previous case. They also show that the main connection faces intervals of breakage due most probably to the sequential invocation of the exponential backoff mechanism. Such intervals are shown as straight lines (since the simulator repeats the last measured RTT) in the last period of Fig. 4b.

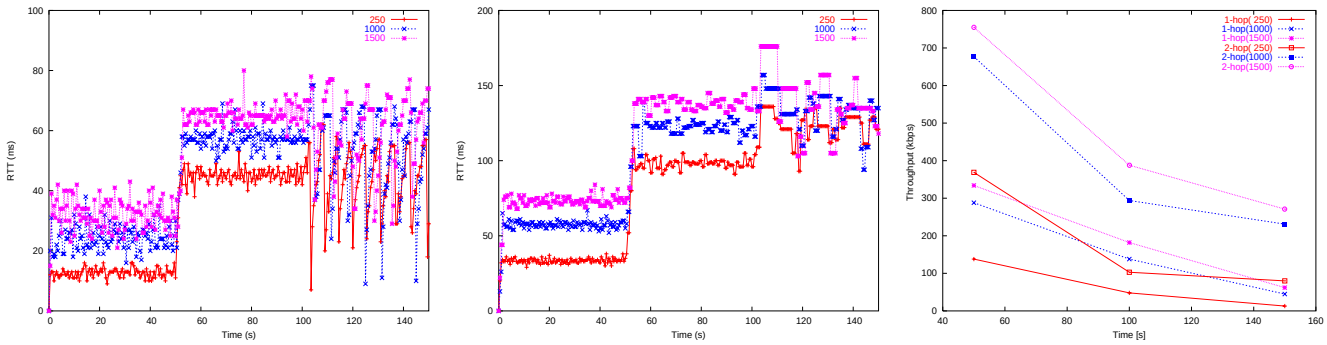
Throughput

It is well-known that the throughput experienced by a connection in a multi-hop environment decreases abruptly with the number of hops crossed by the connection due to the shared medium in place [2], [12]. Fig. 4c confirms this characteristic by showing that the throughputs of the second scenario (Fig. 3b) are lower than the ones from the first scenario (Fig. 3a). The results are computed taking the average over intervals of 50 seconds. Note that in the first interval the throughput decreases more significantly. Yet, in the 1-hop scenario, the level of throughput decrease is really abrupt, being about 50% in all cases. Comparing item a) with b) in Fig. 4, we can see that the trend in both scenarios is the same, the magnitude of the values are different though.

Multi-hop scenario

The aim of this scenario is to verify whether RTTs can also reflect congestion state inside the network with multiple hops. We use the string topology shown in Fig. 3c. Sender and receiver are connected at the ends, and only connection C2 starting at 100 seconds between nodes N2 and N3 are used. For the sake of readability, only RTT measurements for packet sizes of 250 are shown in Fig. 5. We can see that, despite the relatively high magnitude of the RTT variation in the interval 0-100 seconds, it is still possible to detect some anomaly in it after the instant of 100 seconds. Fig. 6 depicts the thoughput achieved with the different packet sizes.

Unlike the previous cases, in which the simulations using higher packet sizes achieve higher throughput, here the same



a) b) c)

Figure 4. RTT and throughput variations under two levels of congestion for packet sizes of 250, 1000 and 1500 bytes: a) 1-hop scenario, b) 2-hop scenario, c) Throughput of both scenarios

does not happen when the network is free of congestion. Observe that the measurements for packet size 250 get the highest throughput in the interval 0-100 seconds. In the next interval, however, it does not follow this trend due to the congestion imposed by connection C2.

Such a behavior for the network without congestion has been detected in [12], in which the authors show that it is related to the specific "hidden-node" problem inherent in multi-hop environments. Nevertheless, they do not address congestion situations which seem to have a different outcome as we note here.

3-nodes scenario with movement

In all prior scenarios, the nodes are fixed. Here, the node N3 in Fig. 3a moves away from node N1 at the speed of 10 m/s, so that the routing protocols has to act to redirect the flow between the end nodes via node N2. The final arrangement is shown in Fig. 3b, where there are two hops between sender and receiver. In these measurements there are no competing flows for the main flow. The goal here is to compare the RTT behavior under congestion (Figs. 4a, 4b) with the RTT behavior under changes in the number of hops (Fig. 7a). As expected, both conditions impose similar pattern to RTT variations, which has to be taken into account by the TCP sender as far as our proposed algorithm is concerned. Likewise, the throughput results are similar to the previous case in terms of trends, which is found by comparing Fig. 4c with Fig. 7c.

Large scenario with movements

The former simulations allow us to have a representative idea about RTT behavior in simple scenarios. However, in future ad-hoc networks, a large number of nodes and also diversified routing protocol strategies are foreseeable. This means that new approaches should take such factor into account. That is why, the simulation performed here considers a larger number of nodes. In this scenario, 50 nodes nodes move freely and both, the DSR and AODV routing protocols, are considered The single competing traffic starts at 100 seconds.

Fig. 7c shows that in the run with DSR the connection is most of the time up and that the congestion is detected. On the other hand, the results for AODV [13] reveal that most of the time the connection is down (straight lines), and traces of congestion seem to be detected about the instant 160 seconds. Thus, it appears that indeed RTT can say something useful to TCP sender in terms of congestion.

As found here, the routing strategy in place may also play a key role on the end-to-end performance, in that it can lead a given flow toward congested links or not.

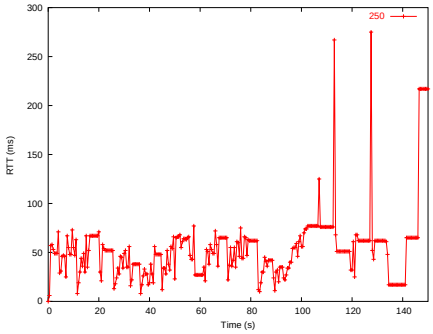


Figure 5. RTT of the multi-hop scenario.

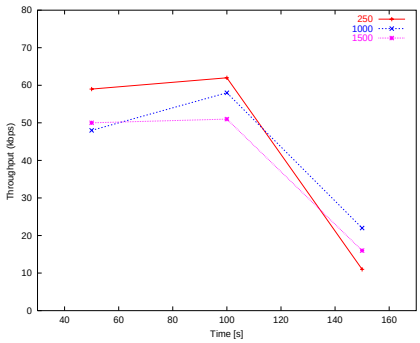
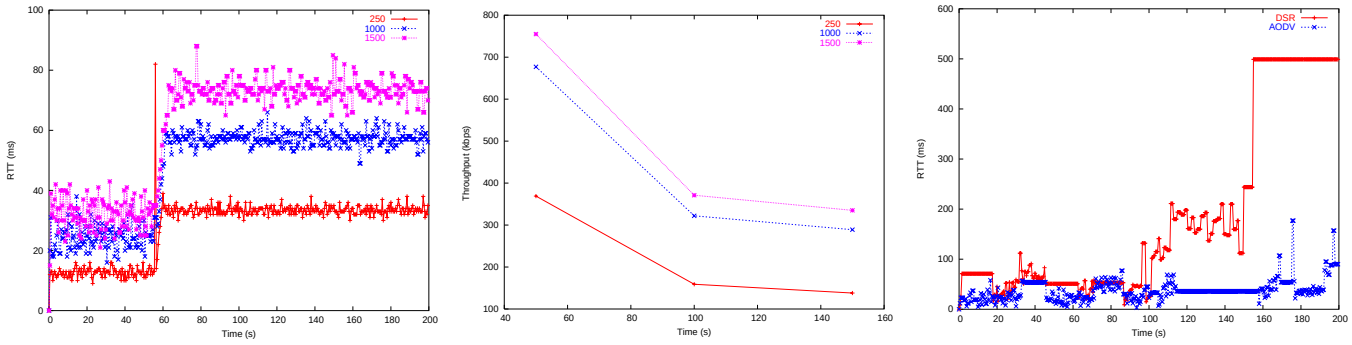


Figure 6. Throughput of the multi-hop scenario.



a) b) c)

Figure 7. Scenarios involving node movements:
a) RTT of a topology changing from one to two hops due to a node movement, and b) its throughput.
c) RTT of the scenario with 50 nodes moving at random and considering two distinct routing protocols.

Discussions

In general the results obtained show that RTT measurements may reveal congestion conditions. However, it has to be smoothed because of its high oscillations shown in the measurements, which could led to instability. The sender needs to differentiate the RTT variations by congestion from the ones due to changing in the number of hops. Our early proposal to address this problem is to monitor the number of packets retransmitted in the following short interval after a sharp RTT increasing detection. In this way, the sender will be able to distinguish between both cases since congestion should trigger significantly more retransmissions.

The high magnitude of oscillations observed under high congestion should also to be detected by the sender congestion detection mechanism, since this characteristic indicates a severe condition inside the network. For that, we intend to make use of appropriate mathematical approaches.

Finally, a very positive aspect of the results refers to the fact that under low/medium congestion the measurements are really representative. Thus, we propose to rely only on reliable measurements. Under non-reliable measurements, we infer congestion. This is important because our approach aims at effectively improving the performance of regular TCP when facing low/medium congestion. In conditions of heavy congestions we intend to let the regular mechanism react, which makes the proposed approach more conservative than aggressive. In addition, the mechanism will slow down the transmission rate whenever changes in the number of hops are perceived. The idea here is to probe the new path for availability of resources.

CONCLUSIONS

We have described the general principles of our end-to-end approach for improving TCP in ad hoc networks. Additionally, we have presented early simulation results carried out to evaluate whether RTT monitoring can be a good indicator of network congestion. The main advantages of our approach are simplicity and independence of intermediate nodes cooperation.

The simulation scenario focused on congestion conditions with variable number of hops between the end nodes, and also on path changes on the fly due to mobility. The main outcome is that RTT measurements may well reflect congestion state inside the network but only under certain particular cases. Specifically, the measurements revealed that RTT variations may indicate network congestion accurately in circumstances in which the network is initially free of congestion or facing light congestion. Under heavy congestion, RTT variations proved not to indicate much about the changes inside the network.

This factor, however, is not a limitation to our edge-based approach since the regular TCP congestion control should be invoked under high congestion anyway. In other words, we propose to trust RTT measurements only in situations in which they are really representative, in all other conditions we infer congestion and let the normal TCP congestion control react. Thus, in general, we believe that useful data for our TCP sender can be extracted from RTT behavior over time. We are going to assess that in more detail in future work.

REFERENCES

[1] Stevens, W. R. 1994. *TCP/IP Illustrated Volume 1.* Addison Wesley.

[2] Oliveira, R. and T. Braun. 2002. “TCP in Wireless Mobile Ad Hoc Networks.” Technical Report IAM-02-003. Inst. of Computer Science, University of Berne, July. (available at ftp://ftp.iam.unibe.ch/pub/TechReports/2002/iam-02-003.pdf).

[3] Chandran, K.; S. Raghunathan; S. Venkatesan; R. Prakash. 1997. “A Feedback Based Scheme For Improving TCP Performance In Ad-Hoc Wireless

Networks." In *Proceedings of International Conference on Distributed Computing Systems- ICDCS '98* (Amsterdam, The Netherlands, May 26-29). IEEE, 472-479.

[4] Liu, J.; S. Singh. 2001. "ATCP: TCP for Mobile Ad Hoc Networks." *IEEE Journal on selected areas in communications, vol. 19, No. 7*, July:1300-1315.

[5] Holland, G. and N. H. Vaidya. 1999. "Analysis of TCP Performance over Mobile Ad Hoc Networks." In *Proceedings of 5th Annual International Conference on Mobile Computing and Networking* (Seattle). 219-230.

[6] Oliveira, R. 2002. "Approaches for improving TCP in Wireless Mobile Ad Hoc Networks.' In Technical Report IAM-02-004. Inst. of Computer Science, University of Berne, September. (available at ftp://ftp.iam.unibe.ch/pub/TechReports/2002/iam-02-004.pdf).

[7] Fu, Z.; B. Greenstein; X. Meng; S. Lu. 2002. "Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks." *10th IEEE International Conference on Network Protocols* (Paris, France). IEEE, 212-225.

[8] Liu J.; I. Matta and M. Crovella. 2003. "End-to-End Inference of Loss Nature in a Hybrid Wired/Wireless Environment." in *Proceedings of Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'03)*.

[9] Ni, Q.; T. Turletti and W. Fu. 2002. "Simulation-based Analysis of TCP Behavior over Hybrid Wireless & Wired Networks." *1st International Workshop on Wired/Wireless Internet Communications (WWIC 2002), in conjunction with Internet Computing 02*. Las Vegas, Nevada, U.S.A.

[10] Biaz, S. and N. H. Vaidya.1998. "Distinguishing Congestion Losses from Wireless Transmission Losses:A Negative Result." In *Proceedings of EEE 7th Int. Conf. on Computer Communications and Networks* (Lafayet, LA, Oct.).

[11] Johnson, D. and D. Maltz. 1996. "Dynamic source routing in ad hoc wireless networks." *Mobile Computing.* Chapter 5, August: 153-181.

[12] Xu, S. and T. Saadawi. 2001. "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *IEEE Communications Magazine,* Vol 39, No. 6, June:30 -137.

[13] Perkins, C. E.; E. M. Royer and S. R. Das. 2001. "Ad hoc on-demand distance vector (AODV) routing." Internet Daft draft-ietf-manet-aodv-08.txt. March.

BIOGRAPHY

Ruy de Oliveira graduated as an Electronic Engineer from the Federal Engineering School of Itajubá (EFEI), Brazil, in 1992. He received his M.Sc. degree in computer networks area from the University of Uberlândia, Brazil, in 2001. Since then, he has been pursuing his Ph.D. degree and working as a research assistant in the Computer Science Institute at the University of Berne, Switzerland. In 2001 he was involved in an European Project (Sequin) for QoS in the Internet. Currently, he is taking part in a long term Swiss research project (NCCR-MICS) on self-organizing networks. His research interests focus on transport layer protocols for such networks.

Torsten Braun got his Diploma Degree and Ph.D. degree from the University of Karlsruhe (Germany) in 1990 and 1993, respectively. From 1994 to 1995 he has been a guest scientist at INRIA Sophia-Antipolis (France) working in the area of high speed protocol implementations. From 1995 to 1997 he has been working at the IBM European Networking Center Heidelberg (Germany), at the end as a project leader and senior consultant with focus on broadband network services and next generation Internet protocols. Since 1998, he has been a full professor of Computer Science at the Institute of Computer Science and Applied Mathematics at the University of Bern (Switzerland), where he is heading the "Computer Networks and Distributed Systems" research group.

Marc Heissenbüttel received his M.Sc. degree from the University of Bern, Switzerland, in 2001 in computer science and mathematics. He is a research assistant and Ph.D. student at the Institute of Computer Science and Applied Mathematics at the University of Bern. His research interests include mobile ad-hoc networks, routing, and IPv6. He is currently involved in a long term Swiss research project (NCCR-MICS) on self-organizing networks.