

CER/TER - THE NEW METRIC FOR TCP CONNECTION ROBUSTNESS EVALUATION AND COMPARISON

Ondrej VONDROUS, Peter MACEJKO, Zbynek KOCUR

Department of Telecommunication Engineering, Faculty of Electrical Engineering,
Czech Technical University in Prague, Zikova 1903/4, 166 36 Prague 6, Czech Republic

ondrej.vondrous@fel.cvut.cz, peter.macejko@fel.cvut.cz, zbynek.kocur@fel.cvut.cz

DOI: 10.15598/aeee.v13i5.1500

Abstract. *This article presents new metric for TCP connection robustness evaluation and comparison. This metric is focused on TCP connection and transmission continuity rather than on maximal throughput or minimal RTT. This metric is developed especially for evaluation of narrow band networks. That is why it is very convenient to use this metric for networks such as Internet of Things networks or industrial sensor networks. Our metric is based on observing if connections or transmissions are successfully finished or not. It is possible to optimize this metric for specific situations. This metric can be used in both the real networks and in discrete simulation environments.*

Keywords

Connection error rate, narrow band network optimization, TCP metric, transmission error rate.

1. Introduction

We are focused on a behavior of TCP [1] protocol under heavy load in narrow band networks in our research. Under this circumstance communication over TCP protocol can easily fail. There are situations when robustness of TCP connection is more important than overall data throughput.

In this paper we would like to present new metric for TCP robustness evaluation. This metric is based on observing the connection error rate (CER) and transmission error rate (TER) per time slice or observation duration.

2. Transmission Control Protocol

The TCP (Transmission Control Protocol) is one of the most important protocols used in current networks. Despite the fact that the new technologies and new protocols are developed, the TCP protocol is still used in most common network services today. This protocol is well designed and works well in ordinary communication.

The TCP protocol is reliable by its nature. It has implemented algorithms such as three way handshake, packet acknowledgments and retransmissions for ensuring that data are successfully transferred to destination. But in some situations it is not enough. Sometimes it is necessary to use more than one TCP connection on the same transport path and that is why algorithms for congestion control and avoidance are implemented. These algorithms help to make TCP flows more stable and friendly to other coexisting data flows along the same path as opposed to protocol such as UDP (User Datagram Protocol). Basic algorithms are slow start, congestion avoidance, fast retransmit and fast recovery [2].

2.1. Congestion Control and Avoidance

Forementioned RFC 5681 [2] describes these congestion control mechanisms.

1) Slow Start

In order to avoid network congestion the TCP at the beginning of new connection slowly probe the network to determine available capacity. The slow start algorithm is used for this purpose at the beginning of a

transfer, or after repairing loss detected by the retransmission timer.

2) Congestion Avoidance

When congestion on the network is detected the congestion avoidance algorithm is used, and it is active for the whole duration of network congestion. This algorithm is based on the Congestion Window (CWND) variable. CWND limits the amount of unacknowledged data a TCP can send. When congestion is detected then the congestion window is only incremented by roughly 1 full-sized segment per round-trip time (RTT). CWND can be increased more progressively when congestion is over to reach maximum available transmission capacity faster. CWND is decreased in situations where segment loss is detected.

3) Fast Retransmit

This algorithm detects and repairs segment loss, based on incoming duplicate acknowledgments (ACKs). The Fast Retransmit algorithm uses the arrival of 3 duplicate ACKs as an indication that a segment has been lost. After receiving 3 duplicate ACKs, TCP performs a retransmission of what appears to be the missing segment, without waiting for the retransmission timer to expire.

4) Fast Recovery

After the fast retransmit algorithm sends what appears to be the missing segment, the fast recovery algorithm governs the transmission of new data until a non-duplicate ACK arrives. In most cases, it is not necessary to fall back to slow start algorithm and drop congestion window size down to 1.

2.2. Implementation Drawbacks

There are several reasons why TCP protocol is not efficient in some situations as it could be. It is quite common that adoption of new TCP versions is delayed by years or even decades. Especially in Supervisory Control And Data Acquisition (SCADA) Remote Terminal Units (RTUs). It is typical that there is the only basic implementation of TCP protocol in RTUs, and it is not possible to configure TCP parameters like Initial Window (IW) or Maximum Segment Size (MSS). The situation is even worse because mechanism such as scale-able Congestion Window (CW) is not used, and CW size is fixed to some specific value. Also, the adoption of new algorithms in more up to date systems is not optimal.

There are also many transport protocols, which have implemented special algorithms to ensure TCP-friendly behavior. These protocols are a potential source of problems too. Majority of algorithms such as AIMD [3], CUBIC [4], CTCP [5], DCCP [6] and SCTP [7] are designed to be friendly to former TCP using AIMD algorithm. However, AIMD is deployed in less than 26 % of web serves TCP implementation [8] nowadays.

The TCP protocol can also fail in extreme conditions such as poor wireless path (high latency and small throughput) between nodes, heavy overload of network and narrow band network with extreme delays. Under this conditions it is probable that TCP retransmissions of loss and corrupted packet make thing even worse and connections can time out. Commonly used TCP protocols can still perform well in such conditions but in industrial deployment change of communication protocol is a expensive and long term task.

3. TCP Performance and Robustness Evaluation

In our research, we are focused on optimizing performance of narrow band wireless networks. We are especially interested in improving TCP connection robustness in extreme conditions as mentioned at the end of the previous section. In specific industrial networks, it is more important to avoid connection disconnection than to achieve maximal network throughput and that is why our methods are based on observing TCP stream continuity rather than maximum overall throughput.

Many studies are focused on comparing throughput of different TCP implementation such as Tahoe, Vegas, Reno, New Reno, Sack [9], [10], [11]. We also found metrics for evaluation of flow fairness very important. One of the most used metric for fairness evaluation is Jain's fairness index (JFI) [12] defined by Eq. (1) nevertheless it is only fairness oriented metric. (This index could be used for many different single network parameters such as throughput or RTT, but it is sometimes difficult to identify the right parameter for specific case). Instead of fairness we are more interested in some qualitative metrics which incorporates stability and robustness of communication.

$$f(x_1, x_2, x_3, \dots, x_n) = \frac{\left(\sum_{i=1}^n x_i\right)^2}{n \cdot \sum_{i=1}^n x_i^2}. \quad (1)$$

In this paper, we would like to introduce our new metric for TCP data stream robustness comparison under different network conditions. We are focused on ob-

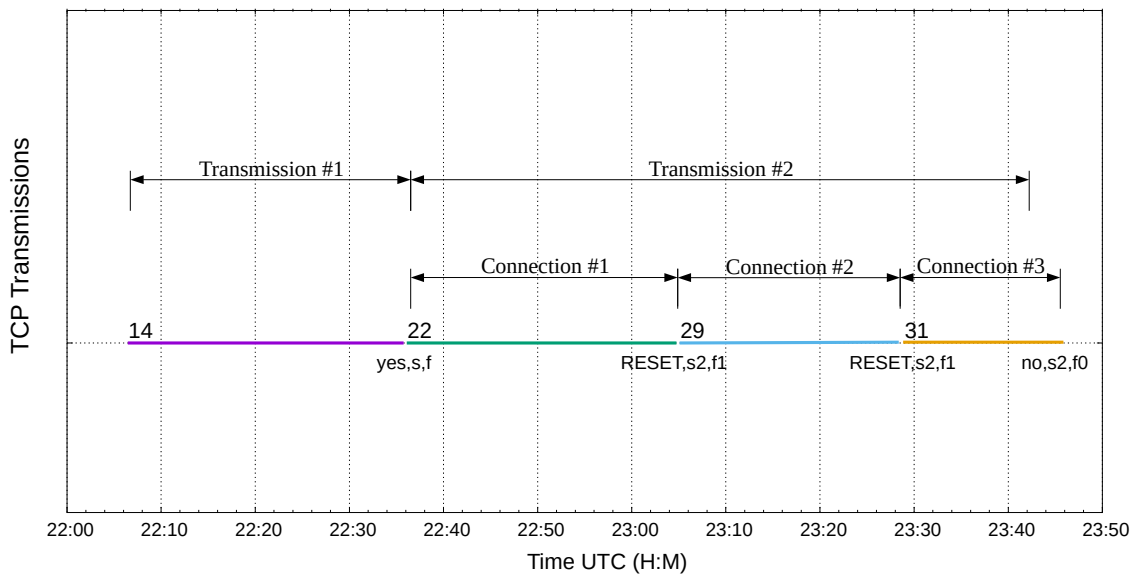


Fig. 1: TCP transmission vs. TCP connection.

servicing connection and transmission continuity of TCP data flows. The outputs of our new metric are values of Connection Error Rate (CER) and Transmission Error Rate (TER).

3.1. Connection vs. Transmission

For purposes of this paper we have to strictly define what is a connection and what is a transmission. The difference between connection and transmission is shown in Fig. 1.

The transmission is defined as successfully finished TCP communication between two network endpoints. One transmission can consist of multiple connections.

The connection is defined as a sub part of transmission. Connection begins when connection is established and ends when close, reset or timeout of connection occurs.

In Fig. 2 the TCP analysis of multiple (16) hosts communication is shown. There are several cases shown where transmission consists of several connections. Every single row represents one or more (but only one transmission at a time) transmissions between specific client and server. Successful transmission is indicated by string "yes" at the end of connection. In some cases the transmission is broken into several connections such as transmission consisting of connections 15, 18 and 21. In this case the transmission was not completely successful as indicated by "no" at the end of the last connection.

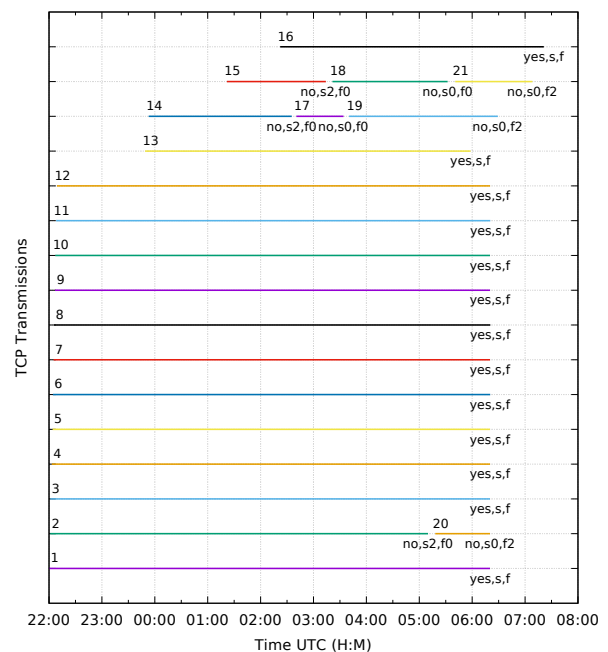


Fig. 2: TCP time analysis.

3.2. TER - Transmission Error Rate

In this metric we observe the ratio of successful vs. failed transmissions of specific data set per time slice or observation duration. The transmission is typically started by 3-way TCP handshake and finished by 4-way TCP handshake. If the transmission is broken into several connections and finished successfully with 4-way TCP handshake, it is considered as successful transmission. We define TER as a percentage of failed

transmission to all transmissions. An example of TER output is shown in Fig. 3 where the TCP transmission robustness in network stressed by the communication of 10 to 30 hosts is compared.

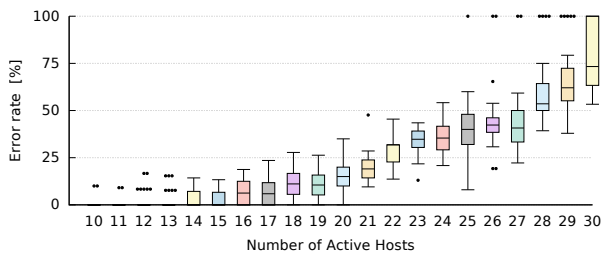


Fig. 3: Transmission error rate - 4 min. interval of inactivity.

$$TER = \frac{T_{bad}}{T_{good} + T_{bad}} \cdot 100, \tag{2}$$

where

- T_{good} - number of successfully finished transmissions,
- T_{bad} - number of failed transmissions.

3.3. CER - Connection Error Rate

In this metric, we observe the behavior of TCP transmission of specific data set between client and server. In this specific case, we observe how many times are single transmission break into several connections until all data are transferred, or the whole transmission fails. There are several reasons for that as shown in list below:

- RTO timeout,
- reception of SYN packets during active connection,
- reception of FIN packets during active connection,
- reception of out-of-order TCP packets.

An example of CER output is shown on Fig. 4 where the TCP connection robustness in network stressed by communication of 10 to 30 hosts is compared.

We define CER as a percentage of failed (unfinished) connections to all connections.

$$CER = \frac{C_{bad}}{C_{good} + C_{bad}} \cdot 100, \tag{3}$$

where

- C_{good} - number of successfully finished connections,
- C_{bad} - number of terminated (unfinished) connections.

4. Methodology for CER/TER Usage

Our methods are optimized to use captured data in PCAP format. The advantage of PCAP format is the fact that data can be captured in the same format in real network and also in Omnet++/INET framework [13].

The PCAP file can be analyzed by TCPTrace [14] utility which returns information about every single TCP connection recorded in PCAP file. It is also possible to define the timeout when connection is considered as failed, which will be discussed in following section. Finally CER/TER metrics are computed and prepared for comparison with other results.

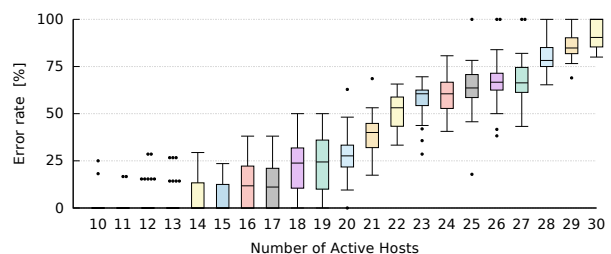


Fig. 4: Connection error rate - 4 min. interval of inactivity.

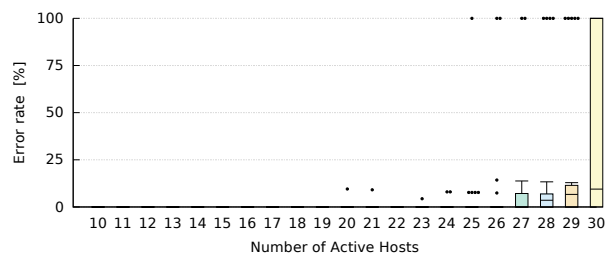


Fig. 5: Connection error rate - 30 min. interval of inactivity.

4.1. Optimization of CER/TER Metrics

When using CER/TER metrics results are strictly dependent on duration of test and load of the network. It is necessary to optimize the duration of the test to provide results, which can be compared with other test results. If the time interval of observation is too short, there is not enough data for comparison. On the other hand if the time interval of observation is too long, we can suppose (if the data transmission is long enough) that almost every single transmission in the congested environment can fail. In this case it is not possible to compare measurement results. We suggest optimizing the measurement to place CER/TER values between 20 and 50 percent. In this case, it is possible to observe when target network stability is increased or decreased.

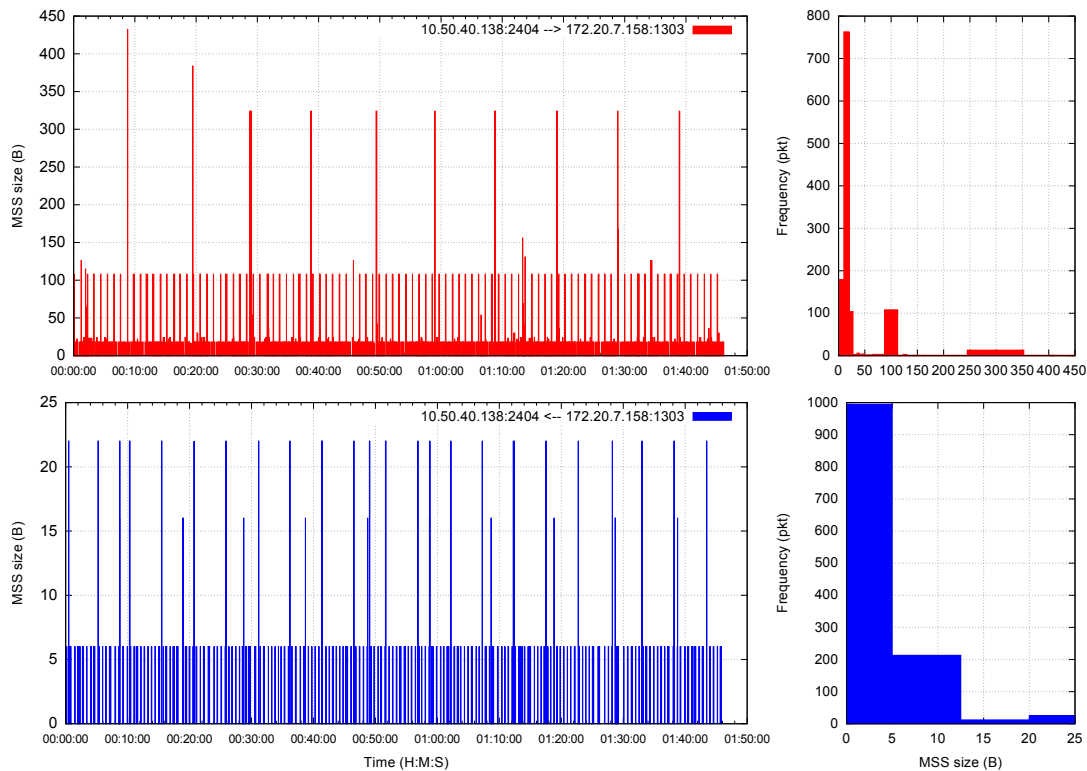


Fig. 6: Representation of typical Up/Down link IEC 60870-5-104 data profile.

The important parameter when optimizing CER/TER metric is TCP connection interval of inactivity (interval after which an open connection is considered closed) which controls output results of tcptrace utility. When connection is inactive for duration of this interval it is still considered as connected but after this interval the connection is considered as failed. By selecting proper interval of inactivity we can increase or decrease sensitivity of CER/TER metric as demonstrated on Fig. 4 and Fig. 5.

We achieve best results when comparing optimization in communication network stressed by unified communication data profiles. We focused our research on sensor and industrial automation processes networks used for data acquisition and control. It is quite common that IEC 60870-5-104 protocol is used in such networks. This protocol is encapsulated into TCP and it is used by SCADA systems for RTUs communication with control ICT systems.

The Fig. 6 represents typical uplink/downlink IEC 60870-5-104 communication profile between SCADA system and RTU. This specific profile was acquired from real network.

On one hand it is not possible to use this metrics as out of the box but on the other hand it allows user to optimize this metric for specific use.

4.2. Optimization of Network Simulations

When using CER/TER metric in discrete simulation environment such as Omnet++ results show the strong dependency of measured results and pseudo-random number sequence even in the case of long 24 h simulation time runs. We recommend to repeat measurements with different seeds for pseudo random generator and then statistically evaluate results to get more precise and valid representation of CER/TER metric as shown in Fig. 3, Fig. 4, Fig. 5.

5. Area of CER/TER Usage

This metric is convenient for optimization of narrow band networks where congestion occurrence is quite common due to the fact that available bandwidth is limited, and a number of communication requests can exceed network capacity. Under these conditions increased RTT, increased packet loss and decreased throughput can cause transmission failures. That is why we found interesting to use CER/TER metric for optimization in the area of Internet of Things (IoT) or industrial sensor networks utilizing TCP protocol. The CER/TER metric is well suited to get a complex overview of TCP behavior in the high loaded communication channel.

6. Conclusion

The presented CER/TER metric and methodology are useful when it is necessary to validate or compare transmission and connection robustness. This metric is well suited for both the data acquired from the real network and for data acquired from discrete simulation environment such as Omnet++. It is very convenient to use this metric for optimization process of communication systems via Omnet++ simulations. In this case it is possible to use highly parallel computing environment such as National Grid Infrastructure MetaCentrum.

The advantage of CER/TER metric is possibility to compare real network measurements with measurements provided by simulations and it is possible to validate how close is network simulator implementation to real network behavior.

Acknowledgment

This work was supported by the Grant of the Technology Agency of the Czech Republic, No. TA04011571, Radio for Smart Transmission Networks, and was researched in cooperation with RACOM and Brno University of Technology. Also this work was supported by Ministry of Interior of the Czech Republic in project System of signaling protocols identification and processing (VF20132015029) and by Students grant of Czech Technical University in Prague, No. SGS13/200/OHK3/3T/13. We also thanks for access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Infrastructure for Research, Development, and Innovations" (LM2010005), is greatly appreciated.

References

- [1] RFC 793 - Transmission Control Protocol: Protocol specification. *RFC-Base* [online]. Available at: <http://www.rfc-base.org/rfc-793.html>.
- [2] RFC 5681 - TCP Congestion Control. *RFC-Base* [online]. Available at: <http://www.rfc-base.org/rfc-5681.html>.
- [3] CHIU, D. and R. JAIN. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*. 1989, vol. 17, iss. 1, pp. 1–14. ISSN 0169-7552.
- [4] HA, S., I. RHEE and L. XU. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review*. 2008, vol. 42, iss. 5, pp. 64–74. ISSN 0163-5980. DOI: 10.1145/1400097.1400105.
- [5] TAN, K., J. SONG, Q. ZHANG and M. SRIDHARAN. A compound TCP approach for high-speed and long distance networks. In: *Proceedings of IEEE International Conference on Computer Communications*. Barcelona: IEEE, 2006, pp. 1–12. ISBN 1-4244-0221-2. DOI: 10.1109/IN-FOCOM.2006.188.
- [6] RFC 4340 - DCCP: Datagram Congestion Control Protocol. *RFC-Base* [online]. Available at: <http://www.rfc-base.org/rfc-4340.html>.
- [7] RFC 4960 - Stream Control Transmission Protocol. *RFC-Base* [online]. Available at: <http://www.rfc-base.org/rfc-4960.html>.
- [8] PENG, Y., L. WEN, X. LISONG, J. DEOGUN and Y. LU. TCP Congestion Avoidance Algorithm Identification. In: *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*. Minneapolis: IEEE, 2011, pp. 310–321. ISBN 978-1-61284-384-1. DOI: 10.1109/ICDCS.2011.27.
- [9] ISOBE, T., N. TANIDA, Y. OISHI and K. YOSHIDA. TCP acceleration technology for cloud computing: Algorithm, performance evaluation in real network. In: *Proceedings of International Conference on Advanced Technologies for Communications (ATC)*. Hanoi: IEEE, 2014, pp. 714–719. ISBN 978-1-4799-6955-5. DOI: 10.1109/ATC.2014.7043480.
- [10] DE CICCO, L. and S. MASCOLO. TCP congestion control over HSDPA: An experimental evaluation. In: *Proceedings of Mediterranean Conference of Control and Automation*. Palermo: IEEE, 2014, pp. 1043–1048. ISBN 978-1-4799-5900-6. DOI: 10.1109/MED.2014.6961512.
- [11] TROPEA, M. and P. FAZIO. Evaluation of TCP versions over GEO satellite links. In: *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*. Toronto: IEEE, 2013, pp. 86–90. ISBN 978-1-56555-352-1.
- [12] BENNET, J. C. R. and H. ZHANG. WF2Q: worst-case fair weighted fair queueing. In: *Proceedings of Conference of the IEEE Computer Societies. Networking the Next Generation*. San Francisco: IEEE, 1996, pp. 120–128. ISBN 0-8186-7293-5. DOI: 10.1109/INFCOM.1996.497885.

- [13] VARGA, A. and R. HORNIG. WF2Q: An overview of the omnet++ simulation environment. In: *Proceedings of International conference on Simulation tools and techniques for communications, networks and systems & workshops*. Brussels: ICST, 2008, pp. 60:1–60:10. ISBN 978-963-9799-20-2.
- [14] BLUM, R. *Network Performance Open Source Toolkit Using Netperf, Tcptrace, NISTnet and SSFNet*. New York: John Wiley & Sons, 2003. ISBN 978-0471433019.

About Authors

Ondrej VONDROUS was born in Czech Republic in 1981. He received his M.Sc. degree in electrical engineering from the Czech Technical University in Prague in 2011. Since 2011 he has been studying Ph.D. degree in telecommunication engineering. His research

interests include network transmission control, data flow analysis and data flow optimization.

Peter MACEJKO was born in Czech Republic in 1980. He received his M.Sc. degree in electrical engineering from the Czech Technical University in Prague in 2006. He is teaching networking technologies and distributed systems. His research is focused on scheduling in distributed systems and data flow and protocol analysis. He is currently actively involved in projects focused on high speed data transmission from fast moving objects.

Zbynek KOCUR was born in 1982. He received his M.Sc. degree in electrical engineering from the Czech Technical University in Prague in 2008 and Ph.D. degree in electrical engineering in 2014. He is teaching communication in data networks and networking technologies. His research is focused on wireless transmission and data flow analysis, simulation and optimization.