An Optimized TCP for Internet Access of Vehicular Ad Hoc Networks

Marc Bechler, Sven Jaap, and Lars Wolf

Technical University of Braunschweig
Institute of Operating Systems and Computer Networks
`[bechler|jaap|wolf]@ibr.cs.tu-bs.de`

Abstract. Communication efficiency at the transport layer is of specific importance for ad hoc networks. Especially in vehicular ad hoc networks, vehicles will have a temporary and rather short-lived connectivity to the Internet, which has to be utilized efficiently. In this paper, we propose a TCP-based transport protocol called MCTP that is optimized for the Internet access in vehicular environments. Therefore, MCTP is combined with split performance enhancing proxy architectures, where a proxy separates the end-to-end TCP connection. This enables the deployment of optimized transport protocols while maintaining interoperability with TCP used in the Internet. For the evaluation, we emulated the communication characteristics of a "typical" vehicular scenario. This clearly shows the advantages of MCTP over traditional approaches; the overall data throughput is significantly higher when MCTP is used for communication between vehicle and proxy. The evaluation also emphasizes the usefulness of performance enhancing proxies in vehicular environments.

1 Introduction

Communication in vehicular environments will become very important and crucial for the future development in the automotive domain: it is considered as a key technology to increase traffic safety since vehicles will be able to distribute local information to other vehicles on the road. For example, emergency situations like an accident or a congestion behind a bend can be transmitted to succeeding vehicles. This way, the vehicles are able to slow down their speed in time. A key technology for inter-vehicle communication (IVC) is multi-hop ad hoc networking. Thereby, vehicles establish vehicular ad hoc networks (VANETs), which enable the local exchange of information without the need for infrastructure components like base stations. Examples for IVC systems are the FleetNet communication system [1] or CarNet [2].

With the introduction of VANETs, passengers also expect infotainment services as well as the access to Internet services using the IVC system. The transition between vehicles and the Internet is achieved by gateways installed on the road-side. The gateways thus provide a temporarily restricted access to the Internet for the passing vehicles traveling in a (specified) area around the gateways. Application scenarios are manifold, as illustrated by the following examples:

- businessmen likely want to send and download emails, and they may synchronize their personal information applications with their office systems,
- the navigation unit of a truck may want to communicate with the company's fleet management system in order to exchange time sensitive information.

In order to access Internet services, VANETs must be integrated into the Internet. This integration is typically achieved by performance enhancing proxies. For example, fig. 1 depicts the proxy architecture used for the Internet integration of the FleetNet IVC system [3]. Thereby, the VANET has connectivity to the Internet through gateways, which are itself connected to a gateway network. A proxy located at a fixed position in the Internet hides the characteristics of the VANETs and, thus, brings together the VANET and the Internet. The proxy also separates the end-to-end TCP connection into two segments: communication between proxy and Internet hosts using standard TCP, and communication between vehicles and proxy. This way, highly optimized transport protocols can be used for communication between proxy and vehicles in order to improve communication efficiency.

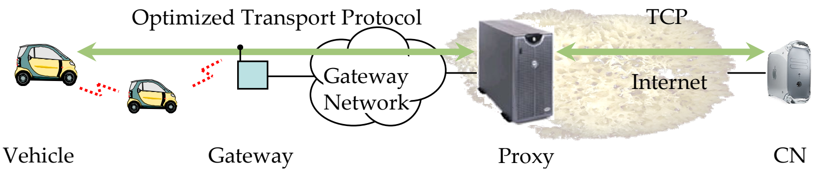


Fig. 1. Vehicular communication scenario

In this paper, we propose a TCP-based transport protocol called MCTP (Mobile Control Transport Protocol), which is optimized for proxy-based communication architectures used in vehicular environments. We describe the basic protocol mechanisms used in MCTP and compare its performance with traditional approaches in a test environment that emulates the characteristics of a typical communication scenario on a highway.

In the following, we first describe related work on improving TCP performance in section 2. Section 3 introduces our transport protocol MCTP, which is evaluated in section 4. Finally, section 5 concludes this paper.

2 Related Work

TCP was developed for networks with a fixed topology. This way, it works well in wired networks and provides an acceptable performance in terms of data throughput. However, the characteristics of mobile networks like VANETs differ fundamentally from wired networks: On the one hand, vehicles are highly mobile and therefore the topology of the VANET is subject of permanent reconfigurations and partitionings. On the other hand, communication is based on wireless radio technology, which shows high variations in the transmission quality. Internet access also will not be available continuously resulting in potentially long periods of disconnections. Several studies investigated the impact of these aspects on the performance of TCP. The investigations showed that TCP provides poor throughput in multi-hop ad hoc networks although a higher throughput might be possible in theory [4]. The performance degradation mainly results from the conservative flow and congestion control mechanisms deployed in TCP. For example, TCP interprets transmission errors as a congestion situation and thus reduces the throughput. The algorithms used are slow start and congestion avoidance [5]. Over the years, TCP was enhanced by several new protocol features. TCP Reno introduced fast retransmit/fast recovery, which was further improved in TCP New Reno according to RFC 2582. Furthermore, TCP was enhanced by selective acknowledgements (RFC 2018). These extensions are already integrated in TCP implementations of common operating systems like Linux. However, such extensions do not solve the basic problems of TCP in mobile environments. This way, TCP still provides a poor performance in the VANET scenario, i.e. for communication between a vehicle and the proxy [3]. In order to improve end-to-end communication efficiency at the transport layer, related work can be classified into three categories (RFC 2757): (i) pure congestion control modifications, (ii) utilization of information from intermediate systems, and (iii) completely new transport protocols not based on TCP. We do not consider snoop-based approaches since they are not expected to provide significant improvements in networks with a high frequency of handoffs.

An obvious way to increase performance is to modify the congestion control in TCP. A noticeable amount of work tries to predict different situations based on local information. With the help of this information, the congestion control algorithms of TCP are modified to react accordingly depending on the predicted situation. Several approaches like TCP Westwood [6] try to estimate the available bandwidth in an intelligent way, which is used to optimize the TCP flow control. Other approaches like TCP DOOR [7] modify the congestion control based on the arrival of out-of-order packets, or they even examine inter-packet arrival times for using a rate-based congestion control mechanism (e.g., Wireless TCP [8]). Approaches like ADTCP [9] additionally measure short term throughput, packet loss ratio, and packet out-of-order delivery ratio, and they use a modified TCP state machine to react efficiently in these situations. Another common solution is to completely modify the algorithms used for slow start, congestion avoidance, and various timeout calculations like, e.g., TCP Vegas [10]. Approaches like ATP [11] completely replace the congestion control of TCP by different algorithms. In Freeze-TCP [12], the receiver notifies the sender in case of an impending congestion. The sender then “freezes” TCP to prevent further transmissions.

A general drawback of this category is that predictions about potential congestion situations are based on local information, which may not reflect the current state of the VANET. This misprediction potentially reduces TCP performance. Moreover, the congestion control algorithms do not provide mechanisms to handle both short-term and longer-term periods of disconnections.

The second possibility is to utilize information from intermediate systems, if the network is able to detect different situations. A common mechanism is Explicit Congestion Notification (ECN, RFC 3168), where intermediate nodes are able to detect pending congestions and signals them to the communicating end systems. This way, an ECN-enabled TCP may use this information to optimize communication efficiency. The utilization of information from intermediate systems is a promising approach to improve TCP in VANETs. The network information provides a better accuracy of the estimations compared to the predictions of pure congestion control modifications. This concept implicitly includes the consideration of notifications, which enables TCP to react quickly to various situations in the network. However, TCP extensions like ECN basically do not solve the general problems of TCP in VANETs since these approaches are still based on exponential backoff timers to calculate the retransmission timeouts. This mechanism is not suitable to handle long-term disconnections from the Internet appropriately since they may cause either a reset of the TCP connection or a long recovery phase after a reconnection to the Internet.

The third category comprises transport protocols not based on TCP. A typical example is the Stream Control Transmission Protocol (SCTP, RFC 2960). In contrast to TCP, the connection-oriented SCTP supports multi-streaming and multi-homing capabilities. This category is not discussed further on since such protocols do not provide a socket-like API, which requires new network programming paradigms that aggravate the deployment of existing applications in vehicular environments.

3 MCTP

An optimized transport protocol for vehicular environments must be able to distinguish between error-prone links and network congestions in order to handle packet losses appropriately. Moreover, it must be able to utilize information from both intermediate systems and from underlying protocols. This is necessary for an efficient treatment of both short-term network partitions and longer-term periods of disconnections from the Internet. However, none of the existing related work fulfils these requirements sufficiently. This way, we developed the transport protocol MCTP (Mobile Control Transport Protocol) for communication between vehicles and a fixed proxy in the Internet. MCTP combines several TCP enhancements proposed in section 2. Its core functionality belongs to the category of utilizing information from intermediate systems, which is extended by modifications of the TCP congestion control mechanisms. In general, MCTP is based on the principles of Ad Hoc TCP (ATCP [13]), which relies on information on pending congestions in the network. This idea is combined with an approach similar to TCP Feedback [14] and TCP Stop-and-Go proposed by Ritter [15]. Like ATCP, MCTP implements a sublayer between TCP and IP as depicted in fig. 2. The basic principle of MCTP is that it observes the IP packet flow between sender and receiver in order to react appropriately. Therefore, MCTP considers notifications from underlying protocols as well as from intermediate systems:

- ECN indicates pending congestions detected by intermediate systems.
- Intermediate systems indicate a partitioned network using ICMP destination unreachable messages. This information is relevant for local communication between vehicles only, i.e. for communication without Internet access.
- The mobility management protocol [16] we used is able to notify MCTP in case of disconnections very efficiently.

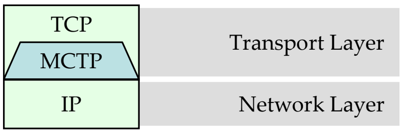


Fig. 2. MCTP in the TCP/IP model

The available information enables MCTP to distinguish between link errors, congestions, network partitions, and disconnections from the Internet. Besides the available information, MCTP also takes into account events caused by TCP itself. Such events are the retransmission timeouts for segments and the arrival of (duplicate) acknowledgements for successfully transmitted segments. Based on this knowledge, MCTP controls the transmission procedure of TCP in different situations by controlling retransmissions and timeouts, and by probing for the network characteristics. MCTP therefore implements its own protocol state machine, which comes into operation after TCP successfully established a connection between the end systems.

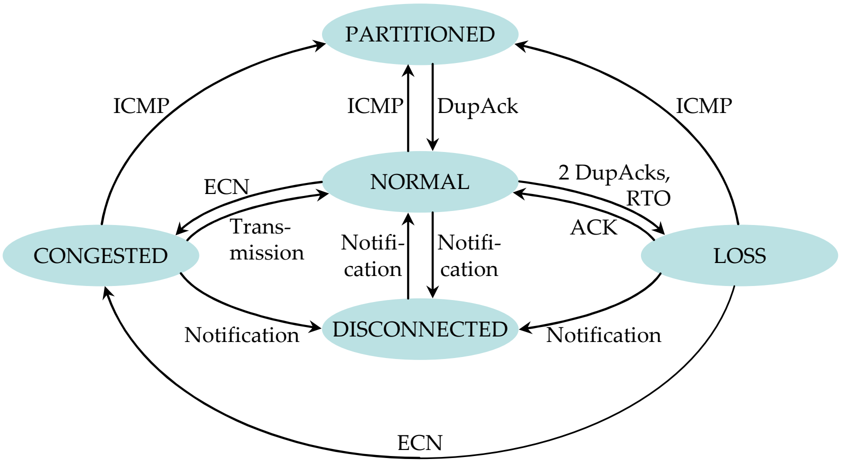


Fig. 3. MCTP protocol state machine

3.1 MCTP Protocol State Machine

A basic feature of MCTP is that it explicitly differentiates between segment losses caused by congestion and segment losses caused by single transmission errors for ongoing connections. MCTP also distinguishes between a partitioned network and a disconnection from the Internet in case of temporary communication breakdowns. A partitioning appears only if a vehicle communicates with another vehicle via multi-hop communication, whereas disconnections occur when a vehicle communicates with a proxy in the Internet. This way, both states can be seen as orthogonal from each other. Fig. 3 shows the protocol state machine. The states NORMAL, LOSS, and CONGESTED are the common operation modes of MCTP in case a data flow is possible. PARTITIONED and DISCONNECTED are only entered when communication is broken.

An important goal of MCTP is to minimize the number of TCP slow starts caused by segment losses. A TCP sender considers a segment as being lost in the following cases:

- receipt of three duplicate acknowledgements (DupAck) for a segment,
- a retransmission timeout (RTO) occurs for a segment.

In the NORMAL state, MCTP counts the number of DupAcks received for a segment. If ECN does not indicate a pending congestion, a segment loss was likely caused by a transmission error. If MCTP receives two DupAcks for a segment in this situation, it enters the LOSS state. Since the TCP congestion control reacts only after the third DupAck, it does not interfere with MCTP in this situation. Similarly, MCTP enters the LOSS state if an RTO expires. In the LOSS state, MCTP forces TCP to freeze its state temporarily. This way, TCP does not invoke congestion control, which would be the wrong thing to do in this situation. Instead, MCTP retransmits the unacknowledged TCP segment. It therefore controls the retransmission timers for the segment accordingly. If an acknowledgement for the segment arrives from the communication peer, MCTP forwards the acknowledgement to TCP, which also recovers TCP, and returns to NORMAL. A different situation occurs when ECN indicates a pending congestion in an intermediate system. Then, MCTP switches to CONGESTED and does nothing: hence, MCTP leaves the congestion control completely to TCP, which handles this situation very efficiently. After the TCP sender transmits a new segment, MCTP returns to NORMAL. This operation mode is similar to ATCP. Differences occur in the handling of DupAcks; whereas ATCP waits for three consecutive DupAcks, MCTP only waits for two DupAcks. Furthermore, MCTP is not based on TCP Reno but uses TCP New Reno with an improved fast retransmit/fast recovery mechanism and selective acknowledgements.

Vehicular mobility may stall ongoing connections in the VANET for a temporary period of time. These communication disruptions are typically caused by a network partitioning or if a gateway becomes unavailable and an alternative gateway cannot be discovered. MCTP considers these two situations and controls TCP appropriately in order to improve the recovery after a connection breakdown. The PARTITIONED state represents a network partitioning that is relevant for inter-vehicle communication only. In contrast, the DISCONNECTED state is entered when the vehicle gets disconnected from the Internet (i.e. the proxy). In case of a network partitioning, an intermediate vehicle will throw an ICMP destination unreachable message if it detects a broken link. If MCTP receives this ICMP message, it moves into the PARTITIONED mode and freezes the current state of TCP. Additionally, it performs a window probing mechanism similar to the zero window probing used in TCP. Thereby, MCTP probes the connection with constant period (the last RTO value). This is in contrast to TCP, which would exponentially backoffs the probing period. If MCTP receives a DupAck from the receiver, the connection is apparently reestablished and communication can be continued. In this case, MCTP recovers TCP, activates the slow start phase of TCP without reducing the slow start threshold, and moves itself back to NORMAL. The PARTITIONED state is also entered from the LOSS state and the CONGESTED state upon receiving an ICMP destination unreachable message. The explicit probing of the connection in case of a network partitioning is optional since it cannot be assumed that a location-based ad hoc routing protocol can detect the reestablishment of the end-to-end routes.

The PARTITIONED mode is of relevance for inter-vehicle communication only. This mode is similar to ATCP; differences between MCTP and ATCP occur in the probing and freezing mechanisms. The PARTITIONED mode is not used when a vehicle communicates with a host in the Internet. In this case, the mobility management protocol is able to detect disconnections very efficiently [17]. If a vehicle looses contact to a gateway, MCTP is notified about the disconnection and switches into the DISCONNECTED mode. In this mode, MCTP completely stops the TCP transmissions and freezes RTO timers. Both TCP and MCTP remain in this state until MCTP is notified about the availability of a new gateway. It then restores TCP and moves itself back to NORMAL. In addition, MCTP activates the slow start phase of TCP without modifying the threshold for the slow start. This allows TCP to converge its data rate to the new situation. Finally, MCTP triggers TCP to retransmit queued segments immediately. If such segments are not available, MCTP sends two acknowledgements in order to generate a DupAck.

4 Evaluation

The goal of the evaluation is to determine the performance of our MCTP Linux implementation together with the communication characteristics of a typical VANET scenario. The VANET communication characteristics were modeled by the NISTNet emulator, which shapes network traffic flows according to configurable parameters like bandwidth, delay, jitter, packet drop rate, and packet duplication rate. Fig. 4 shows our test environment consisting of five connected Linux hosts: on the left-hand side, the mobile node (MN) represents the vehicle that communicates via the proxy (middle) with a correspondent node (CN) in the Internet on the right-hand side. The VANET emulator between MN and proxy emulated the communication characteristics a vehicle experienced, and a second emulator between proxy and CN emulated the Internet characteristics.

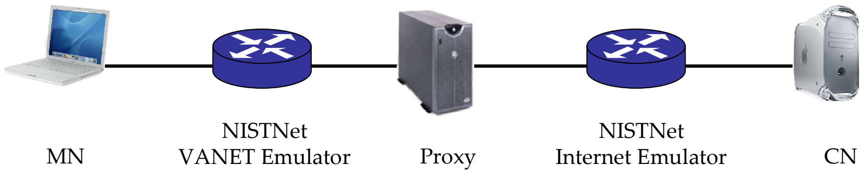


Fig. 4. Test environment used for the evaluation

The communication characteristics in the Internet are highly complex, which make the realistic model almost impossible for the Internet emulator. Instead, we used the following parameters derived from investigations in [18]:

- The bandwidth between proxy and CN is assumed to be higher compared to the bandwidth in the VANET.
- The delay is assumed to be 200 ms with a jitter of ±10 ms.
- The IP packet error rate is 0.2 %. Duplicates are not assumed.

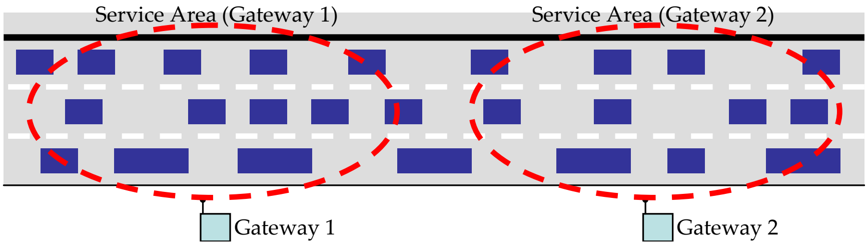


Fig. 5. Highway segment assumed for evaluation

The VANET emulator models a highway segment with a high traffic flow as depicted in fig. 5. Thereby, Internet access is provided by two gateways. The VANET emulator models the communication characteristics a vehicle v experiences while passing this segment. Due to multi-hop communication with an assumed transmission range of 100 m, v is able to communicate with the Internet in the service area (2 km diameter) around each gateway. Fig. 6 shows the "distance" in hops between gateways and v traveling at the right lane. The contact to the first gateway is assumed at 14 hops. v first approaches the gateway resulting in a decrease of the distance every 3 s on average. After 40 s, v enters the direct transmission range of the gateway and contact is lost for a short time after it leaves this range. After 80 s, the first gateway gets unavailable for v and communication is no longer possible for the next 50 s, until v enters the service area of the second gateway. After 215 s, v leaves this service area and communication breaks again. The second scenario assumes a vehicle driving on the left lane at a higher speed resulting in overtaking maneuvers and, thus, a more unsteady distance graph (cf. fig. 7 (b)). For inter-vehicle communication, we assumed the FleetNet system [1] that has the following characteristics: 588 kbit/s (shared) link layer bandwidth, 40 ms delay, 1 % IP packet error rate per link, and 1 % duplicates, symmetrical communication. On the network layer, we assumed the overhead caused by an optimized gateway discovery protocol for VANETs [16] and a respective mobility protocol for VANETs described in [19]. Thereby, the available bandwidth is shared equally among 27 communicating vehicles, resulting in 21.57 kbit/s on average per vehicle. The path between gateway and proxy was not considered since we assumed an ATM network that connects the gateways to the proxy.

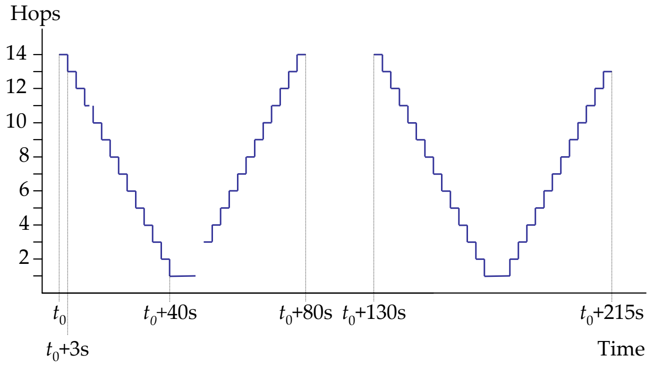
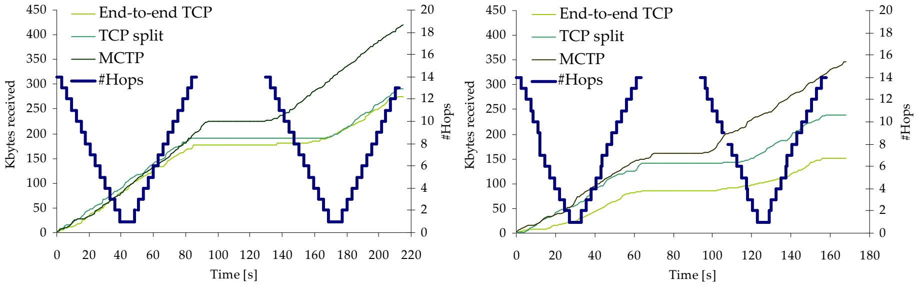


Fig. 6. Distance between vehicle and gateways

In our test environment, we evaluated three configurations: end-to-end TCP between MN and CN, a proxy that segments the connection into two TCP connections (“TCP split”), and a split proxy using MCTP for communication between MN and proxy. For each configuration, we transferred data from the MN (vehicle) to the CN through both emulators, which reflect the communication characteristics a vehicle experiences in the above scenario. We repeated each measurement three times and took their mean value in order to minimize statistical variations of the NISTNet emulator. Fig. 7 shows the results of the three configurations for the right (a) and left (b) lane. The charts also depict the distance between the vehicle and the gateways to show the correlations. The three graphs in fig. 7 (a) showed similar characteristics in the beginning. This behavior can be expected since decreasing error rates and packet delays typically do not cause slow starts in TCP. The throughput of the three tests decreases slightly when the number of hops increases in the time interval between 50 s and 80 s. This chart also depicts the effects of a longer period of disconnection between 90 s and 130 s: After the reconnection through the second gateway at 130 s, it takes a long time until TCP detects the reconnection and continues with its transmission. Interestingly, end-to-end TCP had a slightly quicker response time, which is explained by statistical deviations of the NISTNet emulator; it took about 35 s until end-to-end TCP and TCP split recovered after the reconnection. The MCTP measurements show a smooth and continuous behavior over the total simulation run. An interesting observation is that MCTP is able to transmit data until the disconnection from the first gateway occurs (at about 90 s) whereas communication in case of end-to-end TCP and TCP split stalled about 10 s before the disconnection from the first gateway occurred. This effect can be explained with the high packet error rates at this distance, which reduces the TCP throughput significantly. After the reconnection to the second gateway at 130 s, MCTP reacts quickly and continues its transmission in the same way than in the beginning of the simulation run. In this phase, the data throughput also increases continuously.



(a) Right lane

(b) Left lane

Fig. 7. Evaluation results

The measurements for the left lane in fig. 7 (b) show that the throughput is lower than on the right lane. This is caused by the shorter connection times to the Internet and the higher variations in the communication characteristics. End-to-end TCP seems to have problems especially in the beginning of the simulation run. It takes about 20 s until end-to-end TCP is able to transmit a noticeable amount of data. This chart also illustrates the problem of TCP with longer periods of disconnections. It takes about 35 s until TCP recovers after the reconnection to the second IGW at about 95 s. In contrast, TCP split has a significantly better performance since the data throughput increases more steadily in the beginning. The TCP split measurement also converge more quickly after the reconnection to the Internet through the second gateway, which takes on average 25 s. The MCTP measurements showed a characteristic similar to the measurements for the right lane. Thereby, the transmission of data segments continues steadily while the vehicle is connected to the Internet. After the reconnection to the second gateway, MCTP reacts quickly and the transmission is continued with a very short delay but suffers from the high packet losses in the beginning.

The measurements showed that MCTP improves communication efficiency at the transport layer in this scenario. MCTP is able to retransmit lost segments very efficiently and, in contrast to TCP, it reacts quickly to disconnections from and reconnections to the Internet and, thus, does not pass up the available bandwidth. In both scenarios, the performance of MCTP is significantly higher compared to the other tests: Over the simulation time, end-to-end TCP transmitted 274.155 Kbyte (left lane: 150.592 Kbyte), TCP split transmitted 291.531 (left lane: 237.955 Kbyte), and MCTP was able to transfer 420.885 Kbyte (left lane: 346.072 Kbyte) of data. Since segment losses and temporary disconnections from the Internet are quite common in vehicular communication scenarios, we can carefully conclude that MCTP is able to improve communication between vehicles and Internet hosts.

5 Conclusion

Communication efficiency is an important issue in vehicular ad hoc networks. In this paper, we propose an optimized transport protocol called MCTP for the Internet access of vehicles through VANETs. MCTP was developed for proxy-based communication architectures where vehicles communicate with a proxy using MCTP, whereas communication between proxy and Internet host is based on standard TCP. MCTP distinguishes different network situations and is, thus, able to control TCP appropriately: MCTP handles segment losses efficiently and reacts to disconnections very quickly. Our evaluation based on an emulated highway segment with a high traffic flow shows that MCTP is able to increase data throughput by a factor of 2.3 compared to traditional end-to-end TCP, and by a factor of 1.5 compared to a split TCP approach. Our evaluation also showed that performance enhancing proxies improve communication performance in vehicular environments.

In our future work, we will examine additional "typical" vehicular communication scenarios. The current status of the MCTP prototype includes the basic protocol mechanisms. We are planning to improve this prototype further on by considering additional available information, e.g. from the routing protocol. This allows us to optimize the slow start phases after disconnections or after a network partitioning. We are also planning additional comparisons with different TCP variants and TCP optimizations. However, most of them are not compatible with our test environment.

References

1. Franz, W., Eberhardt, R., Luckenbach, T.: FleetNet – Internet on the Road. In: Proceedings of the 8th World Congress on Intelligent Transport Systems, Sydney, Australia (2001)

2. Morris, R., Jannotti, J., Kaashoek, F., Li, J., Decouto, D.: CarNet: A Scalable Ad Hoc Wireless Network System. In: Proceedings of the 9th ACM SIGOPS European Workshop, Kolding, Denmark (2000)
3. Bechler, M.: Internet Integration of Vehicular Ad Hoc Networks. Dissertation, Logos-Verlag Berlin, ISBN 3-8325-0750-7 (2004)
4. Bae, S., Xu, K., Lee, S., Gerla, M.: TCP Behavior across Multihop Wireless and Wired Networks. In: Proceedings of the 2002 IEEE GLOBECOM, Taipei, Taiwan (2002)
5. Schiller, J.: Mobile Communications. Addison Wesley (2003)
6. Gerla, M., Sanadidi, M.Y., Wang, R., Zanella, A., Casetti, C., Mascolo, S.: TCP Westwood: Congestion Window Control Using Bandwidth Estimation. In: Proceedings of the 2001 IEEE GLOBECOM, San Antonio, Texas, USA (2001)
7. Wang, F., Zhang, Y.: Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response. In: Proceedings of the 3rd ACM MobiHoc, Lausanne, Switzerland (2002)
8. Sinha, P., Venkitaraman, N., Sivakumar, R., Bharghavan, V.: WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks. In: Proceedings of the 5th ACM/IEEE MOBICOM, Seattle, Washington, USA (1999)
9. Fu, Z., Greenstein, B., Meng, X., Lu, S.: Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks. In: Proceedings of the 10th International Conference on Network Protocols (ICNP), Paris, France (2002)
10. Brakmo, L.S., Peterson, L.L.: TCP Vegas: End to End Congestion Avoidance on a Global Internet. IEEE Journal on Selected Areas in Communications (1995)
11. Sundaresan, K., Anantharaman, V., Hsieh, H.Y., Sivakumar, R.: ATP: A Reliable Transport Protocol for Ad-hoc Networks. In: Proceedings of the 4th ACM MobiHoc, Annapolis, Maryland, USA (2003)
12. Goff, T., Moronski, J., Phatak, D.S., Gupta, V.: Freeze-TCP: A true End-to-End TCP Enhancement Mechanism for Mobile Environments. In: Proceedings of the 19th IEEE Conference on Computer Communications (Infocom), Tel Aviv, Israel (2000)
13. Li, J., Singh, S.: ATCP: TCP for Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications (2001)
14. Chandran, K., Raghunathan, S., Venkatesan, S., Prakash, R.: A Feedback Based Scheme For Improving TCP Performance in Ad-Hoc Wireless Networks. In: Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS), Amsterdam, The Netherlands (1998)
15. Ritter, H.: Bedarfsorientierte Dienstgüteunterstützung durch adaptive Endsysteme. VDI Verlag (2001) [German].
16. Bechler, M., Jaap, S., Wolf, L.: Mobility Management for Vehicular Ad Hoc Networks. In: Proceedings of the 61st IEEE Semiannual Vehicular Technology Conference (VTC 2005 Spring), Stockholm, Sweden (2005)
17. Bechler, M., Franz, W.J., Wolf, L.: Mobile Internet Access in FleetNet. In: Proceedings of the 13th Fachtagung Kommunikation in Verteilten Systemen (KiVS), Leipzig, Germany (2003)
18. Bolot, J.C.: End-to-End Packet Delay and Loss Behavior in the Internet. In: Proceedings of the 1993 ACM SIGCOMM Conference, San Francisco, California, USA (1993)
19. Bechler, M., Storz, O., Franz, W., Wolf, L.: Efficient Discovery of Internet Gateways in Future Vehicular Communication Systems. In: Proceedings of the 57th IEEE Vehicular Technology Conference (VTC), Jeju, Korea (2003)