# A COMPARATIVE STUDY OF NETWORK TRANSPORT PROTOCOLS FOR IN-VEHICLE MEDIA STREAMING

*Mehrnoush Rahmani* [1], *Andrea Pettiti* [1], *Ernst Biersack* [2], *Eckehard Steinbach* [3], *Joachim Hillebrand* [1]

[1]BMW Group Research and Technology, Munich, Germany
[2]Institut Eurecom, Sophia-Antipolis, France
[3]Technische Universität München, Institute of Communication Networks, Munich, Germany

## ABSTRACT

We analyze and compare various transport protocols in the context of wireless in-vehicle IP-based audio and video communication. We determine the most appropriate transport protocol and discuss its benefits for an application in the car. The analyses are accomplished based on the IEEE 802.11 standard. A testbed is used to measure and compare quality of service values such as throughput, jitter and media quality at the receiver. In the experiments, the traditional protocols TCP and UDP showed the best performance.

## 1. INTRODUCTION

Today's premium cars include a multitude of interconnected audio and video devices. Examples are the telephone system that is linked to the radio and CD/MP3 player, digital television and the DVD player for both front and rear-seat receivers, telematics systems and driver assistance cameras for assistance and safety purposes. Current automotive communication networks consist of several network systems such as LIN, CAN, FlexRay, MOST, etc., as well as individual analogue cables for video devices such as cameras. In order to reduce the growing cost and complexity, [1] proposes an IP-based network for audio and video communication in the car. While the underlying technology in the wired network is selected to be Ethernet, for the wireless short-distance transmissions in the car, the IEEE 802.11g standard is used due to its wide availability in consumer electronic devices.

In order to define the network architecture in more detail, different transport protocols have been studied and analyzed from the quality of service (QoS) point of view for an application in the car. The objective of this work is to identify the most suitable transport protocol that fulfills the requirements of all in-vehicle audio and video transmitting devices and thus, to define the protocol stack for the future IP network.

## 2. TRANSPORT PROTOCOLS IN IP-NETWORKS

There are several transport protocols that can be used for audio and video transmission over wireless channels in the car. An overview of these protocols and their properties is given in the following.

**TCP, Transport Control Protocol:**
TCP (RFC 0793) is a connection oriented transport protocol that provides a reliable byte stream to the application layer. TCP uses an ARQ mechanism based on positive acknowledgments. It supports a congestion avoidance mechanism to reduce the transmission rate when the network is overloaded. Three different versions of TCP have been analyzed in this work. TCP Reno is currently the most used TCP implementation and is therefore, also called standard TCP (RFC 2581). Another TCP type with the same congestion control mechanism has been tested with increased window sizes[1]. In order to improve the TCP performance, the default and maximum size values are increased for both, receiver and congestion windows. The TCP version with increased windows is called 'TCP Reno increased windows' in this work. The third version of TCP used the WestWood congestion control algorithm.

**UDP, User Datagram Protocol:**
UDP (RFC 768) is a simple transport protocol. UDP does not guarantee any reliability and in order delivery of the packets. It allows multicast and broadcast transmissions. It is suitable for applications that need to define the sending rate, prefer packet losses to jitter or have strong delay requirements. UDP-lite (RFC 3828) is a lightweight version of UDP that delivers packets even if their checksum is invalid. This protocol is useful for real-time audio/video encoding applications that can handle single bit errors in the payload. For using this protocol, the user must disable the checksum on the link layer to forward the error-prone packets to the transport layer.

**DCCP, Datagram Congestion Control Protocol:**
DCCP (RFC 4340) is an unreliable datagram protocol, which

[1]There are three kinds of windows that are possible to modify. [2] explains how and what to change in order to increase the performance of the TCP protocol.

provides a congestion control mechanism[2]. Similar to UDP, it sends datagrams to the network and the application layer is responsible for the framing. Datagrams are acknowledged due to the congestion control mechanism. Unlike TCP, the sequence number in each packet is the sequence number of the packet and not the bytes sequence number. The protocol provides a reliable handshake for connection setup and tear down. The way how sequence numbers and connection initialization and termination are managed is secured against attacks. DCCP is suitable for applications such as media streaming over the Internet and supports the partial checksum option as described for UDP-lite.

**SCTP, Stream Control Transmission Protocol:**
SCTP (RFC 2960) is a reliable transport protocol that offers acknowledged, error-free and non-duplicated transfer of datagrams. Detection of corrupted data, loss and duplicated data is achieved by using checksums and sequence numbers. A selective retransmission mechanism is applied to recover losses and corrupted data. As opposed to TCP, STCP supports multihoming and the concept of several streams within a connection. Where in TCP a stream is referred to as a sequence of bytes, a SCTP stream represents a sequence of messages (datagrams) which may be very short or long. SCTP was developed for signaling applications over IP networks. The congestion control applied to this protocol is TCP like. However, [4] showed that the performance of SCTP is worse than a TCP flow under the same network conditions.

## 3. SYSTEM DESCRIPTION

### 3.1. In-Vehicle Wireless Channel Analysis

The wireless communication channel provided by the IEEE 802.11g standard creates a bottleneck for in-car multimedia streaming due to interferences with other wireless technologies in the car, e.g., Bluetooth and the low throughput capacity compared to the wired Fast-Ethernet network. Several parameters such as the maximum achievable load in real wireless channels in the car and the amount of losses and corrupted packets have been measured in an experimental testbed. According to [5], the Access Point (AP) queue size has been measured to be 120 MTU-sized packets large. Thus, losses due to the AP buffer overflow could be computed and separated from channel losses. As AP, the LinkSys Broadband Router Wireless-G 2.4 GHz WRT54GL and as wireless receiver device the Intel IPW2200 have been applied. This test setup shows the possible application of a wireless receiver, e.g., a laptop in the car. The throughput has been measured for three different average values of receiver signal power: $-20$ dBm, $-35$ dBm and $-60$ dBm. The noise level at the receiver side is set to $-80$ dBm. While $-20$ dBm and $-35$ dBm can be considered as possible values in the car, $-60$ dBm is a value to study the performance in the worst case. In order to explain the receiver signal power values in a more comprehensive way, the receiver can be considered 10 cm away from the AP for $-20$ dBm, around 5 meters from the AP for $-35$ dBm and around 20 meters from the AP for $-60$ dBm which could show the case when the wireless receiver exits the car. Table 1 shows the results of the throughput with UDP and TCP as transport protocols[3]. These results are also representative for DCCP and SCTP protocols. In all performed tests,

| Protocol | **-20 dBm** | **-35 dBm** | **-60 dBm** |
|---|---|---|---|
| UDP | 4.24 | 4.18 | 3.6 |
| TCP | 3.2 | 3.1 | 2.4 |

**Table 1**. Throughput (Mbyte/s) measured in the wireless environment using both UDP and TCP with three different values of receiver signal power.

UDP reached a higher throughput than TCP, because TCP occupies the channel more often by sending ACKs and due to the slow-start function of TCP for newly established connections and after retransmissions due to timeouts. Fig.1 shows

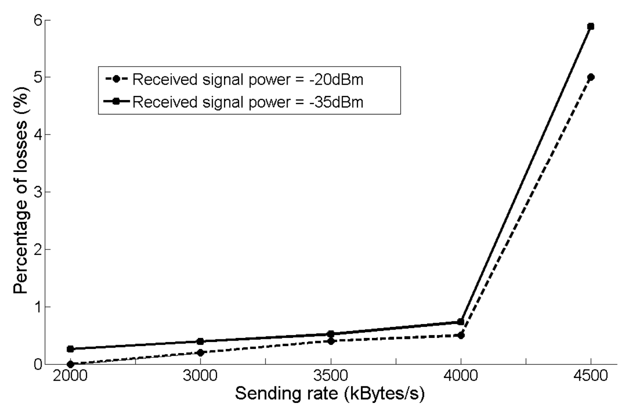


**Fig. 1**. Packet losses in the car's wireless channel

the percentage of losses observed in the UDP tests for $-20$ dBm and $-35$ dBm of receiver signal power. Each point is the average number of lost packets observed during a test of 300 seconds. The line obtained with $-60$ dBm is not shown, because the variance between consecutive seconds is too high so that the mean value is not significant. Until the saturation of the wireless channel (around 4.2 Mbyte/s) the amount of losses is very low. Beyond, the receiver experiences a large number of losses which is mainly caused by AP buffer overflow.

[2]DCCP provides two different congestion control mechanisms, the TCP-like or CCID2 and the TCP-Friendly Rate Control (TFRC) or CCID3. However, due to Linux implementation errors of the TFRC and because TFRC is adapted for constant bit rate applications [3], the analysis of this work is based on the CCID2.

[3]For UDP a constant bit rate sender application with 5 Mbyte/s has been applied while for TCP a 100 MByte file transfer is performed.

Since the lite versions of UDP and DCCP can tolerate the corrupted packets, the amount of packets with bit errors have been measured in order to justify their application in our network. The NetGear 108 Mbit/s Wireless PC Card 32-bit card bus WG511T has been applied in the monitor mode that works under Linux. The source code of this driver software has been modified to ignore the link layer checksum. After recompiling it was possible to deliver corrupted packets to the transport layer. The percentage of corrupted packets did not differ for different sending rates. The largest amount of corrupted packets was observed for the receiver signal power of $-60$ dBm that was on average $0.18\%$ which is very small and negligible. Additionally, if the channel is encrypted, as it should be in the car, a single bit error makes the whole packet useless. Thus, the application of the lite protocols in the car is not recommended. Therefore, these protocols are not further considered in the analysis of this work.

### 3.2. Testbed Description

In order to compare the performance of the different transport protocols for multimedia streaming, two types of applications, i.e., a constant bit rate (CBR) and a variable bit rate (VBR) application have been studied. For the CBR application, the packet size is set to 1328 bytes. The packet interarrival time is then computed for each desired bit rate. The inter-arrival times and bit rates are listed in Table 2. For the VBR application, a MPEG-2 compressed video sequence is used. Fig.2 shows our testbed. The wired environment has

| **Rate (kbyte/s)** | **450** | **700** | **1200** | **2000** |
|---|---|---|---|---|
| **Inter-arrival time ($\mu$s)** | 2951 | 1897 | 1106 | 664 |

**Table 2**. Inter-arrival times between consecutive packets to achieve the desired sending rate with a packet size of 1328 bytes (payload)

been considered, because it is a suitable channel to compare the protocols' performance without having the problem of frequent channel changes due to interferences or reflections as in the wireless channel. The network emulator NetEm has been applied in the Linux kernel to introduce a fixed amount of packet losses. Several loss rates have been applied, i.e., $0.4\%$, $1.5\%$ and $5\%$. $0.4\%$ is the average value observed in the channel during the channel analysis tests. $1.5\%$ can be considered as a worst case when the wireless channel is saturated. While $5\%$ was chosen to test the protocols in a very lossy channel. In the wireless channel tests, one extra sender and receiver are used to provide a disturbing CBR UDP flow as it could be the case in the car (see Fig.2). These tests are performed for receiver signal power values of $-20$ dBm, $-35$ dBm and $-60$ dBm.

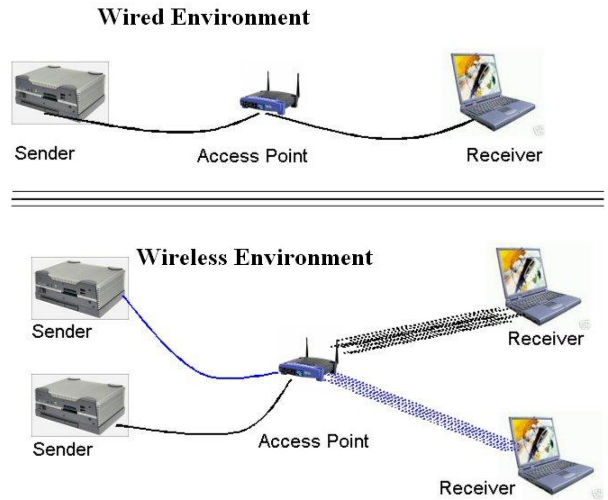


**Fig. 2**. Testbed implementation for the wired and wireless environments

## 4. MEASUREMENT RESULTS

In this section, the most significant evaluation results are presented for both environments. The throughput and the packet inter-arrival times are measured for the CBR application while the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) are used to measure the video quality for the VBR application. Fig.3 and 4 show the throughput values measured in the wired and wireless environments, respectively. In Fig.3, TCP WestWood and TCP Reno with increased windows perform the best with just a small difference to UDP while Fig.4 shows a better performance of UDP in the wireless environment. The solid line shows the ideal value, i.e., the loss free behavior in each figure. Packet inter-arrival

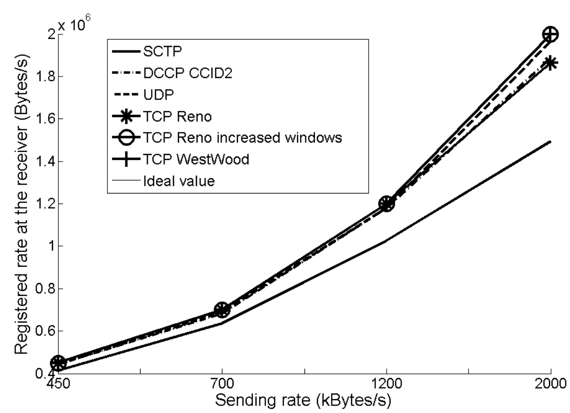


**Fig. 3**. Achieved throughput in the wired environment for $1.5\%$ loss rate. The ideal curve is covered by the TCP WestWood curve.

jitter has been measured at the receiver application layer. The

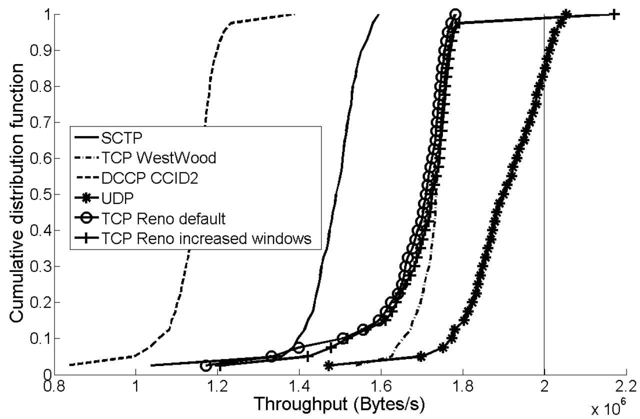


**Fig. 4**. Cumulative distribution function of the throughput values in the wireless environment for a sending rate of 2000 kbyte/s and a receiver signal power of -35 dBm.

average inter-arrival jitter values turn out to be lower than 4 ms in both environments. Reliable protocols show a larger jitter, because they need to wait for acknowledgments before retransmissions.

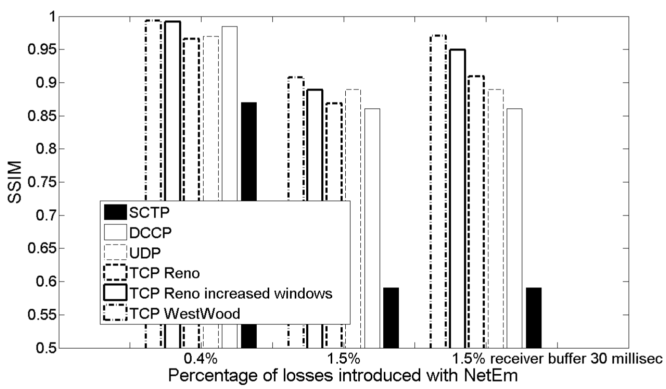


**Fig. 5**. SSIM values obtained when comparing the received video and the reference one. The video was streamed using the standard Video LAN Client application.

From Fig.5 and Table 3, it can be seen that TCP specially with the WestWood congestion control performs the best. However, the difference between TCP and UDP is very small. DCCP and SCTP perform worse than the TCP versions and the UDP protocol. Accordingly, we conclude that TCP and UDP are more suitable for an application in the car.

## 5. CONCLUSION

In this work, the IEEE 802.11g wireless technology has been investigated for media streaming applications in a future IP-based, wired/wireless in-car network architecture. The packet loss and corruption characteristics of this channel in the car

| **UDP** | **TCP** | **TCP 1** | **TCP 2** | **DCCP** | **SCTP** |
|---|---|---|---|---|---|
| **PSNR (dB), Introduced loss:** $1.5\%$ | | | | | |
| 24.28 | 23.91 | 24.93 | 24.58 | 23.69 | 14.97 |
| **PSNR (dB), Introduced loss:** $1.5\%$**, rec. buffer: 30 ms** | | | | | |
| 24.28 | 24.04 | 28.11 | 30.88 | 23.69 | 14.97 |
| **PSNR (dB), receiver signal power:** $-35$ **dBm** | | | | | |
| 32.84 | 25.92 | 24.56 | 37.67 | 20.28 | 24.23 |

**Table 3**. PSNR values obtained from the comparison between the received and sent video sequences (MPEG-2 format) in the wired and wireless environments. TCP 1 is the TCP version with increased windows while TCP 2 stands for the TCP WestWood.

are measured and discussed. The lite protocols, i.e., UDP lite and DCCP lite are not suitable for in-car applications, because they are not adapted for secured channels as it would be the case in the car. For the transmission of time critical media streams over reliable wired channels, UDP is a suitable protocol. But for real-time media streaming over unreliable wireless channels, the best protocol would be a modified version of TCP in that the application controls the time a frame is to be sent. If a frame is not completely transfered when the end-to-end delay threshold expires, the sender application must start transmitting the next frame in order to keep the real time restrictions.

## 6. REFERENCES

[1] M. Rahmani, J. Hillebrand, W. Hintermaier, E. Steinbach and R. Bogenberger, "A Novel Network Architecture for In-Vehicle Audio and Video Communication," *IEEE Broadband Convergence Networks Workshop, Munich, Germany*, May 2007.

[2] Lawrence Berkeley National Laboratory, *TCP Tuning Guide*, 2007.

[3] J. Van Velthoven, K. Spaey and C. Blondia, "Performance of Constant Quality Video Applications using the DCCP Transport Protocol," *Performance Analysis of Telecommunication Systems Research Group, University of Antwerp, IBBT, Middelheimlaan 1, B-2020 Antwerpen, Belgium*.

[4] R. Brennan and T. Curran, "SCTP Congestion Control: Initial Simulation Studies," http://www.eeng.dcu.ie/ opnet/papers/, 2001.

[5] F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki, "Measuring Queue Capacities of IEEE 802.11 Wireless Access Points," *Proceedings of the Fourth IEEE International Conference on Broadband Communications, Networks and Systems (BroadNets07)*, September 2007.