Fuzzy Logic Based Handoffs in Vehicular Communication Environments

Marc Bechler, Lars Wolf
Institut für Betriebssysteme und Rechnerverbund
Technische Universität Braunschweig
Braunschweig, Germany
[bechler, wolf]@ibr.cs.tu-bs.de

***Abstract*— Vehicles typically travel in heterogeneous overlay networks where several communication systems may be available simultaneously. This way, a vehicle has the choice to decide which communication system to be used for Internet access. In order to utilize this heterogeneity, we propose a management entity called MoccaMuxer located in the network layer of a communication system. This management entity integrated itself seamlessly into a proxy-based communication architecture, which is typically used for the Internet intagration of inter-vehicle communication systems. The MoccaMuxer is able to determine the most suitable communication system using fuzzy logic, and it dynamically hands off connections after the decision to the new communication system in a seamless way.**

I. Introduction

Traffic telematics is a key technology for the future development in the automotive domain. Inter-vehicle communication (IVC) systems will become very important for modern vehicular applications. An example technology is the FleetNet communication system [1]. Such systems are based on self-organising ad hoc networks enabling autonomous multi-hop communications between vehicles. This way, vehicles are able to exchange data locally even over multiple intermediate vehicles. Vehicles are also able to communicate with gateways acting as transitions points to the Internet. An important technology to to integrate IVC systems into the Internet is to deploy a proxy-based communication architecture [2]. Thereby, a (transparent) proxy located in the Internet brings together communication in the IVC network and communication in the Internet: the proxy handles the mobility of the vehicles using a respective mobility management protocol, and it allows to deploy an optimized transport protocol between vehicles and proxy whereas communication between proxy and Internet hosts is still based on standard TCP. Fig. 1 depicts the integration

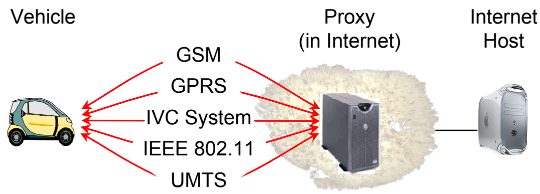

Fig. 1. Proxy-based vehicular communication scenario

of a proxy-based communication architecture in the IVC scenario.

The IVC system is not the only communication system a vehicle can use for Internet access. Vehicles typically move in highly heterogeneous communication environments, and other (cellular and ad hoc based) communication systems might be available simultaneously in an overlaid fashion. As illustrated in fig. 1, the vehicle may have the choice between GSM/GPRS, the IVC system, IEEE 802.11, a UMTS network and potentially other communication technologies to access resources in the Internet. This heterogeneity can be used to improve both connectivity and quality of service (QoS) support for the applications as illustrated by the following examples:

- A vehicle can use either the IVC system or alternatively a GSM, GPRS, or UMTS network for Internet access.
- If a gateway to the Internet becomes unavailable in the IVC network, a vehicle can hand off communications to another available communication system.
- If a vehicle passes a gas station, it may have a temporary high-speed Internet access via the IEEE 802.11 WLAN of the gas station. In this case, it can hand off its connections temporarily to this IEEE 802.11 network.

However, the different communication systems have very different communication characteristics and capabilities. In order utilize communication heterogeneity in this vehicular communication scenario for Internet access, “intelligent” management mechanisms are required to determine the most suitable alternative and to handoff communications to this new communication system dynamically. In this paper, we propose such a management entity called MoccaMuxer (Mobile communication architecture Multiplexer) that is integrated seamlessly into a proxy-based communication architecture for the Internet integration of IVC systems. The MoccaMuxer considers several performance input parameters and deploys a fuzzy expert system to determine the most suitable communication system.

This paper is organized as follows: Section II identifies the challenges, defines the objectives and provides a survey on related work. The MoccaMuxer is proposed in section III, where the basic protocol mechanisms as well as the implementation of the fuzzy system are outlined. Section IV briefly provides a qualitative evaluation of the MoccaMuxer, and section V finally concludes our paper.

II. Objectives and Related Work

The overall objective is to utilize heterogeneity in order to improve quality of service (QoS) support for communication with the Internet in a vehicular environment. An important characteristic in this heterogeneous communication scenario is that the available communication systems typically have different characteristics and capabilities: Whereas an IEEE 802.11 network may provide data rates of up to 54 Mbit/s in a restricted geographical area, cellular communication systems like GPRS provide area-wide coverage with data rates of up to 384 kbit/s with highly varying data rates and delays. The communication costs also differ significantly; whereas Internet access by the IVC system is very cheap or even free of charge, the costs for UMTS are expected to be very high and may depend on connection times and/or data volume. The QoS support also depends on the requirements of the IP-based applications running on the communication platform located in the vehicle. In order to bring together these different issues, protocol mechanisms are needed which fulfil the following basic requirements [3][4]:

- It must be possible to hand off connections seamlessly between different communication systems, which is also called a vertical handoff.
- It must be possible to determine the most suitable communication system.

A noticeable amount of work exists to providing seamless handoffs in mobile and wireless networks. Examples are the BARWAN project [5], MosquitoNet [6], a handoff scheme for mobile networks proposed by Pahlavan et al. [7], the Smart Decision Model proposed by Chen et al. [8], and approaches using a dynamic network reconfiguration of mobile devices [9]. A detailed description and discussion of this research with respect to the requirements for utilising heterogeneity can be found in [10][11]. In summary, these approaches share two basic drawbacks: First, they provide vertical handovers in overlay networks with only a few communication systems. Second, these approaches do not define mechanisms to decide which network fits best to the QoS requirements of running applications.

III. MoccaMuxer

In order to utilize heterogeneity, we implemented a management entity called MoccaMuxer (Mobile communication Multiplexer). The MoccaMuxer is able to hand off connections dynamically between different network interfaces. The integration of the MoccaMuxer into the proxy-based communication architecture needs to be seamless, i.e. invisible for existing applications and application-specific protocols. We therefore deploy the MoccaMuxer on the communication platform in the vehicles and in the proxy. Fig. 2 illustrates the resulting communication scenario. If the MoccaMuxer in the vehicle realizes that the proxy becomes unreachable in the current network, it hands off its communications to an alternative communication system like GPRS. For example, this situation may occur in the IVC system when the vehicle leaves the service area of its current gateway an alternative gateway is not available. The MoccaMuxer then tunnels the communication flow through GPRS to the proxy instead of the inter-vehicle communication system. If the connectivity between vehicle and proxy via the IVC network becomes available again when entering the service area of a new gateway, the ongoing connections can be switched back to the IVC system (i.e., another handoff takes place). The deployment of the MoccaMuxer in the vehicles and the proxy has the important benefits that available communication systems can be used alternatively to the IVC system and Internet hosts need not to be modified, which alleviates the deployment of the MoccaMuxer in a real-world scenario.

The MoccaMuxer is located in the network layer of vehicles and the proxy. Fig. 3 illustrates the in-

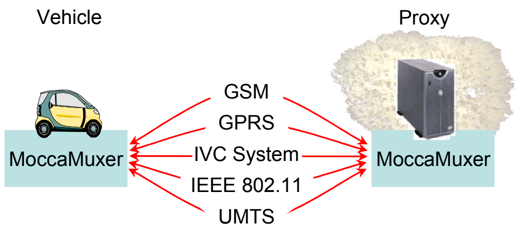

Fig. 2. Integration of the MoccaMuxer in a Proxy Architecture

tegration and realization of the MoccaMuxer. The handoffs are realized by a central element called DeviceMultiplexer, which responsible to switch between the different network interfaces dynamically. If the receiver becomes unreachable in the network actually used, the DeviceMultiplexer queries a Management Information Base (MIB) providing information about alternative networks to communicate with the proxy. This information can be hard-coded into the MIB of the MoccaMuxer in the vehicle since the vehicle always uses the same proxy for Internet access [2]. Afterwards, the DeviceMultiplexer determines the most suitable alternative (see next section), establishes an IP tunnel to the alternative IP address of the proxy, and forwards the IP packets through this tunnel. The tunnelling is necessary since otherwise the "relabeled" IP packets cannot be associated with the ongoing transport layer connection. The tunnelling mechanism, which is the main principle of Mobile IP (RFC 3344), is performed by a redirector in the DeviceMultiplexer that calls the IP stack a second time and, thus, redirects every IP packet to the respective network device.

When an encapsulated packet arrives at the proxy, the MoccaMuxer within the proxy first establishes a reverse tunnel back to the sending vehicle. Hence, the communication path from the MoccaProxy to the vehicle is also tunnelled through the alternative communication system. This way, asymmetrical communication is avoided since common wireless communication systems usually provide duplex communication channels. After establishing the reverse tunnel, the DeviceMultiplexer in the proxy unpacks the original IP packet and forwards it to the upper layer protocols. A detailed description of the DeviceMultiplexer implementation and its performance can be found in [12].

A. MIB

The MIB contains information about the available communication systems. A monitor within the MoccaMuxer is responsible to keep the MIB entries up-to-date. Therefore, it collects relevant information from both the network layer and the link layer, which can be used to determine the current state of a network interface. Further important parameters are the tunnel endpoints of the proxy, which are needed in case of handoffs and which can be configured statically in the MIB since a vehicle is always related with exactly one proxy. This way, the MIB in the vehicle reflects the IP addresses the proxy can be reached in the respective communication system.

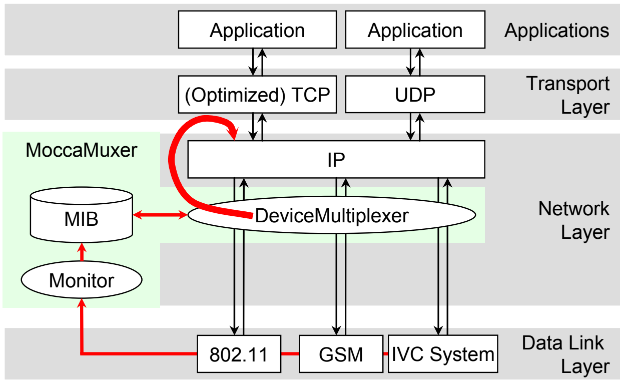

Fig. 3. MoccaMuxer

The information stored in the MIB is classified in the following two ways:

- *Static Information* such as tariffs for the transmission costs or the maximum data rate of a network. For example, in case of GSM this would be the price per minute and a data rate of 9.6 kbit/s (duplex) per connection.
- *Dynamic Information* for each connection/communication system like the currently available bandwidth or the current error rate. This information is typically be derived from the kernel and the device drivers of the operating system as well as by passive measurements.

Based on this information, the MoccaMuxer is able to determine the most suitable alternative communication system.

B. Decision Process

The information provided for the decision process may be partially complete or diffuse. For example, in case of an inactive GPRS network device the current utilization may not be determined in order to avoid additional costs. This way, the decision process has to take this incomplete information base into account. The decision process in the MoccaMuxer is thus based on fuzzy logic: The basic available information is the input for a fuzzy expert system. The use of fuzzy logic is a suitable method for the decision

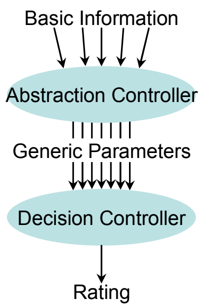

Fig. 4. MoccaMuxer decision process

process, because it describes a system intuitively using linguistic variables. In contrast, mathematical optimization approaches typically are not able to cope with diffuse sets, whereas neural networks are highly complex and may have problems with varying and indeterministical communication characteristics.

The fuzzy controller in the MoccaMuxer implements a two-tier decision process as illustrated in fig. 4. For each network device, an Abstraction Controller in the first tier processes the basic information and generates a set of generic parameters. These parameters are the input for the second tier, the Decision Controller. The Decision Controller finally determines the "rating" of a communication system represented by a numerical value. Such a decision is performed for each available communication system. Finally, the MoccaMuxer finds the most suitable network device by comparing the ratings of the network devices attaches to the end system. If a more suitable communication system is available, the DeviceMultiplexer performs a handoff to this communication system as described in the previous section.

C. Prototype Implementation

The Abstraction Controller is fed with various basic parameters provided by the Linux operating system we used for the implementation of the MoccaMuxer. In the following description of the basic parameters, the mnemonic 'rx' represents the receiving state, 'tx' the transmission state. For the decision process, the following basic parameters were used:

- *Device State:* The parameter `devState` represents the state of a network device, which indicates whether the network device is active or inactive.
- *Packet and Byte Statistics:* These parameters provide statistical information about the packets/bytes transmitted and received. Examples are packet counters and data rates.

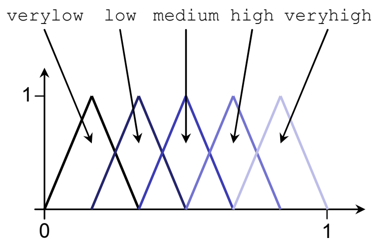

Fig. 5. Fuzzy set for the fuzzyfication

- *Maximum Data Rate:* The maximum data rate is necessary for the interpretation of the statistical data. For network devices with a shared medium, this value has to be used with caution.
- *Output and Input Queue:* Another important parameter are the number of bytes in the output and input queue. Therefore, respective parameters specify the current utilization of these queues and their maximum length.
- *Error Counters:* Transmission and receive errors are also considered for the decision process.
- *Incoming Tunnels:* Usually, tunnels are symmetrical. Hence, the decision process has to consider incoming tunnel requests.

Since the basic parameters may be unreliable and diffuse, we cannot use them immediately for the decision process. Hence, the Abstraction Controller first determines a set of generic and hardware-independent parameters like the relative error rate, relative queue utilization, and relative data rates. The Abstraction Controllers maps these parameters onto a fuzzy set with five states of a triangular pattern as illustrated in fig. 5. The five states are `verylow`, `low`, `medium`, `high`, and `veryhigh`.

With the help this information, the Abstraction Controller determines link quality and load of a network device. In the following, we will focus on the link quality only; details for the load calculation can be found in [2]. For the link quality, the Abstraction Controller uses the relative transmission error counter (relTxErrorCount) together with the device state `devState` in the following way to determine the transmission quality `txQuality`:

```
if (devState == inactive) then
   txQuality is verylow;
else case relTxErrorCount is
   veryhigh then txQuality is verylow;
   high then txQuality is low;
   medium then txQuality is medium;
   low then txQuality is high;
   verylow then txQuality is veryhigh;
end;
```

The `rxQuality` for the receive quality is calculated accordingly. Finally, the overall link quality is determined by the minimum of `rxQuality` and `txQuality`.

The Decision Controller first determines with the help of the load parameter the `loadGain`, which prefers network devices with existing tunnels in order to reduce handoffs [2]. Together with the link quality, the Decision Controller finally determines the overall rating of a network device. Based on this rating, the attached network devices can be compared: The MoccaMuxer then triggers the DeviceMultiplexer to hand off the network device with the highest rating.

IV. Quantitative Evaluation Summary

The MoccaMuxer enables seamless handoffs between different communication systems in order to utilize heterogeneity. Thereby, communication between vehicle and proxy can be switched over to alternative communication systems if the proxy becomes unreachable in the network currently used. An evaluation of the DeviceMultiplexer showed that the delay caused by the dynamical switching mechanism is very low and does not affect the performance of TCP significantly [12].

An important concept of the MoccaMuxer is the consideration of the communication characteristics. In contrast to related work in this area, static and dynamic information is used for determine the most suitable alternative network in order to continue communication. The computational overhead of the fuzzy expert system is very low, and it is expected that the system is able to find more suitable solutions than a greedy (or random-based) decision approach. However, a final evaluation with real measurements is almost impractical since the results highly depend on real communication characteristics, the hardware and hardware drivers being used, and the reliability of the available information.

V. Conclusions

Overlay networks often are considered as a challenge, not as a chance for quality of service support. Especially in vehicular network environments where communication with the Internet may be available temporarily, heterogeneity can significantly improve the Internet access: if a network becomes unavailable, alternative communication technologies can be used instead. In this paper, we propose a management entity called MoccaMuxer, which is able to utilize heterogeneity. The MoccaMuxer is located in the network layer, where it can hand off connections dynamically between different networks. In order to find suitable alternatives, a fuzzy-based expert system considers several available parameters for the decision process.

Future work will address the consideration of additional important parameters to improve the decision process. Moreover, quality of service requirements from applications need to be addressed as well in order to optimize the decisions further on.

References

[1] W. Franz, R. Eberhardt, and T. Luckenbach, "FleetNet – Internet on the Road," in *Proceedings of the 8th World Congress on Intelligent Transport Systems*, Sydney, Australia, Oct. 2001.

[2] M. Bechler, *Internet Integration of Vehicular Ad Hoc Networks*, Dissertation, Logos-Verlag Berlin, ISBN 3-8325-0750-7, 2004.

[3] D. Wisley, H. Aghvami, S. L. Gwyn, T. Zahariadis, J. Manner, V. Gazis, N. Houssos, and N. Alonistioti, "Transparent IP Radio Access for Next-Generation Mobile Networks," *IEEE Wireless Communications*, Aug. 2003.

[4] L. Taylor, R. Titmuss, and C. Lebre, "The Challenges of Seamless Handover in Future Mobile Multimedia Networks," *IEEE Personal Communications*, Apr. 1999.

[5] E. A. Brewer, R. H. Katz, Y. Chawathe, S. D. Gribble, T. Hodes, G. Nguyen, M. Stemm, T. Henderson, E. Amir, H. Balakrishnan, A. Fox, V. N. Padmanabhan, and S. Seshan, "A Network Architecture for Heterogeneous Mobile Computing," *IEEE Personal Communications*, Oct. 1998.

[6] S. Cheshire and M. Baker, "A Wireless Network in MosquitoNet," *IEEE Micro*, Feb. 1996.

[7] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J.-P. Makela, R. Pichna, and J. Vallström, "Handoff in Hybrid Mobile Data Networks," *IEEE Personal Communications*, Feb. 2000.

[8] L.-J. Chen, T. Sun, B. Chen, V. Rajendran, and M. Gerla, "A Smart Decision Model for Vertical Handoff," in *Proceedings of the 4th ANWIRE International Workshop on Wireless Internet and Reconfigurability*, Athens, Greece, May 2004.

[9] J. Inouye, J. Binkley, and J. Walpole, "Dynamic Network Reconfiguration Support for Mobile Computers," in *Proceedings of the 3rd ACM International Conference on Mobile Computing and Networking (Mobicom)*, Budapest, Hungary, Sept. 1997.

[10] M. Bechler and H. Ritter, "A flexible Multiplexing Mechanism for Supporting Quality of Service in Mobile Environments," in *Proceedings of the 34th Annual Hawai'i International Conference on System Sciences (HICSS)*, Maui, Hawai'i, USA, Jan. 2001.

[11] J. Ylitalo, T. Jokikyyny, T. Kauppinen, A. J. Tuominen, and J. Laine, "Dynamic Network Interface Selection in Multihomed Mobile Hosts," in *Proceedings of the 36th Hawai'i International Conference on System Sciences (HICSS)*, Big Island, Hawai'i, USA, Jan. 2003.

[12] M. Bechler, B. Hurler, V. Kahmann, and L. Wolf, "A Management Entity for Improving Service Quality in Mobile Ad-Hoc Networks," in *Proceedings of the 1st International Conference on Wireless LANs and Home Networks (ICWLHN)*, Singapore, Dec. 2001.