THE NEED FOR THE WIRELESS APPLICATION PROTOCOL (WAP) IN CARS

Marc Bechler, Jochen Schiller
Institute of Telematics, University of Karlsruhe
Zirkel 2 – 76128 Karlsruhe – Germany
Tel: +49 721 608 [6400 | 6415] – Fax +49 721 388097
e-mail: [mbechler | schiller]@telematik.informatik.uni-karlsruhe.de

SUMMARY

Future cars will be integrated in large information systems, where various kinds of information might be sent to and from a car. Beyond information about the current traffic condition or weather forecast, cars should also offer access to the Internet, enabling electronic commerce or mobile commerce such as online banking and online brokerage, and various other applications. In order to handle this flood of information, a common platform is needed that allows for the integration of those different information streams. We explain, why the use of the common Internet protocols does not make sense and describe the Wireless Application Protocol (WAP), which was developed especially for the use in mobile and wireless networks, and why it is advantageous for the use in the area of traffic telematics.

MOTIVATION

Regarding the current traffic flow in cities and on motorways, we can state that the car density became more and more dramatic within the last few years. In 1996, a German driver was statistically involved in a traffic jam about 65 hours per year. The resulting economic damage was estimated at about 100 Billion €, while the consumption of gas additionally increased about 2.5 Billion litres, as cars have to wait in traffic jams or cruising through cities to find a place to park. This situation becomes even worse as the admission of new cars in Germany increases every year, 5.9% in 1998. Counteracting this situation by building new streets and, thus, enlarging the road infrastructure is in most cases not feasible, not accepted by the population, or often too expensive. Thus, the alternative is to improve the utilization of the roads by providing vehicles with dynamic information, e.g., about the current traffic situation. Therefore, the navigation unit within the car will be able to calculate both, the best route to the destination as well as the travelling time under the current circumstances.

In the future, we will see an evolution from vehicles to mobile information centres. Drivers enter their car in the morning, when the car already has cached the current road and weather conditions sent via DAB (Digital Audio Broadcasting [1]), while the driver's PDA (Personal Digital Assistant) uploads the new destination to the car's onboard navigation system. A Personal Travel Assistant (PTA) supports the driver planning his or her trip. While driving, traffic conditions on the highways will be sent to the car, the car forwards the selected information to the PTA, which can calculate a new route. The information is also sent to the PDA inside the car, which automatically rearranges the schedule for meetings. Similar scenarios are also conceivable for air traffic or railroad traffic. Beyond road information, future cars will offer more services to passengers, such as online banking, online brokerage, e-mail, maybe video conferencing, telephony, or browsing in the Internet. However, this plethora of new ca-

pabilities with specific requirements to the communication system arises the question how this information can be sent to the car by using a common platform.

EVOLUTION IN INFORMATION TECHNOLOGY

History shows that one single networking technology cannot cover all aspects of communication. Especially in wireless environments, different technologies are used for different kinds of applications, each fitted with highly different characteristics, capabilities, and coverage. Figure 1 gives an impression of this variety of different technologies. For broadband access, there are networks with a fixed infrastructure such as GSM [1], wireless LAN (e.g., IEEE 802.11 [2]) or maybe DECT [1] for cities, DAB [1], GPRS [1], or the future UMTS [1]. While GSM is optimised for mobile phones (but also offers data rates with a maximum of 14.4 kbit/s, and up to 57.6 kbit/s using HSCSD [1]), GPRS allows data rates up to 115 kbit/s, and DAB up to 1.5 Mbit/s, but in one direction only. Additionally, satellite links can be used for communication as well as GPS to determine the current position of a vehicle. In the future, cars are also able to form clusters in an ad-hoc fashion for exchanging information, such as a warning of the congestion in front of a car, so that the following cars are able to slow down in time. Inside a car, there may be pico networks (using IrDA [3] or Bluetooth [4]) used for the communication between mobile devices, e.g., between PDA, Laptop, and navigation unit.

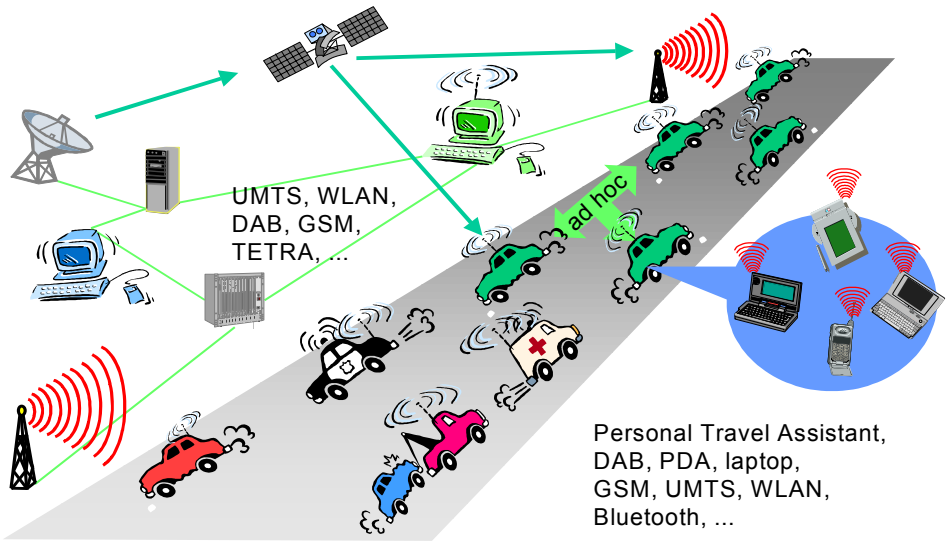

Figure 1: Future Road Traffic Scenario

In the future, we will see several orders of magnitude more mobile devices connected to the Internet. Each car, truck, aircraft, but also many people will carry a plethora of mobile devices connected to the Internet. The capabilities of those devices are very different. As an example, the screen size of a mobile phone is by far smaller compared to the one of a PDA, or the display of a navigation system. Usually, mobile devices have limited performance due to restricted power supply, or they have to take care of rough environments prevailed in vehicles.

This brief overview of the current evolution in the field of information technology shows that we have on the one hand a highly heterogeneous wireless environment with different characteristics. There are also many mobile devices with various capabilities working in this environment. On the other hand, we can see a growing number of applications with different requirements to mobile devices, security aspects, and transmission characteristics. The mobile

applications should not rely on one single communication system; the development of new applications becomes easier as developers do not have to take care of the technical aspects specific to one communication system, and the robustness of the application will be improved as it can use other communication systems for exchanging information. Thus, one single platform is needed that offers a reliable and possibly secure connection and fits to these different requirements.

THE WIRELESS APPLICATION PROTOCOL (WAP)

One obvious approach for such a common platform seems to be the Internet. The prevailed communication protocol in the Internet is TCP/IP, which offers a unified interface for transmitting data, independent of the underlying network. This idea has several advantages: All Internet-based applications such as WWW or e-mail can be used, and the integration of new applications is very easy by using TCP/IP, e.g., telephony via Internet using Voice over IP [5]. However, using the Internet has one main disadvantage: The protocols for communication are optimised for fixed networks with high reliability and low error rates. In mobile environments, this property is very disadvantageous and affects the behaviour of applications in several negative ways:

- TCP/IP works very ineffective in wireless environments. IP is based on a hierarchical addressing scheme; thus, supporting mobility is hard to achieve. A solution for this problem might be Mobile IP [6]; but the deployment of Mobile IP raises several other problems, e.g. in security support [6]. Compared to fixed networks, wireless links usually have higher delays and frequent transient interruptions. In those cases, TCP supposes congestion on the link and immediately slows down the data rate to its minimum. The slow-start-algorithm implemented in TCP prevents an increase in performance, thus the overall performance is by far lower than technically feasible.
- Security mechanisms in the Internet, such as SSL (Secure Socket Layer), are not sufficient for applications that handle personal information, such as online banking or electronic commerce. Mechanisms for the authentication of mobile devices are not provided, and meanwhile, further less expendable encryption algorithms exist.
- On the higher layers, HTTP that transports WWW content works stateless and does not perform any compression, which blows up the data volume that has to be transmitted. Additionally, HTML used for describing WWW pages contains a lot of information useless for today's mobile devices, such as colourful pictures or java applets.

Those disadvantages show the need for an alternative architecture with standardised protocols that are optimised for the use in mobile and wireless environments. Thus, in 1997, the WAP-Forum [7] was founded by a few companies in order to create an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly. This aspect makes it interesting for the use in vehicles. Meanwhile, over 200 companies joined the forum, representing over 90 % of the global handset market, carriers with more than 100 million subscribers, leading infrastructure providers, software developers and other organizations providing solutions to the wireless industry.

Figure 2 gives an overview of a typical WAP infrastructure. A mobile device communicates with, e.g., a Web server via a WAP proxy. Thus, a set of new techniques can be used for the wireless network, such as protocols optimised for wireless links, or, as shown in figure 2, the use of the Wireless Markup Language (WML) instead of the unsuitable HTML for describing

the information. The WAP proxy is also integrated in the Internet environment, i.e., it can access Web Servers by simply using the Internet protocols, i.e., TCP/IP and HTTP. The basic communication interaction between mobile device and Web Server works in the following way. The mobile device sends an encoded request to the WAP Proxy, which decodes the request and translates it from the WAP protocol stack to the Internet protocol stack. The WAP Proxy passes the new request to the specified Web Server, which sends the requested information in a response back to the WAP Proxy. The information will be translated to the WAP protocols, encoded, and finally sent to the mobile device. There are two ways to create WML content. The first is to write raw WML code, which is stored directly on a Web Server. The WAP Proxy downloads this code via HTTP and sends it directly to the mobile device, using the protocols defined in the WAP architecture. Alternatively, the WAP Proxy requests common HTML code, and converts it to WML code using specific filters. From the outset, WAP integrates speech services for telephony using WTA Servers (Wireless Telephony Application). This empowers the WAP architecture as one common platform for supporting voice and data communication and, thus, takes the preceding integration of voice and data services into account.

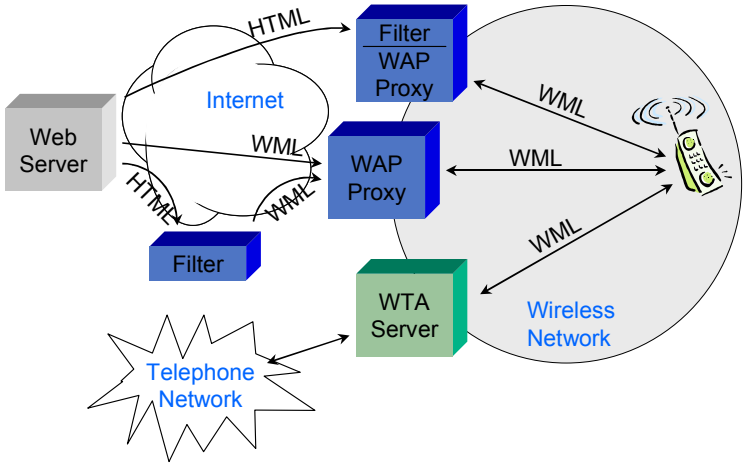

Figure 2: WAP Infrastructure Overview

The following sections describe the Wireless Application Protocol in more detail and outlines how it can be used for sending various types of information to different devices in vehicles.

The WAP Architecture

Starting with version 1.0 in 1999, version 1.1 is currently implemented in common mobile devices. Meanwhile, the standardisation of version 1.2 has been finished. Basically, version 1.1 and version 1.2 of WAP describe the same architecture and protocols; version 1.2 can be seen as an extension in order to support more features.

The WAP architecture comprises six layers as can be seen in figure 3. The stack on the left hand compares the protocols used in the Internet with the layers of the WAP architecture. One main idea of WAP is the independence of communication protocols from the employed bearer service used for transmitting data. WAP only specifies the adaptation to those different bearers. In WAP 1.2, the adaptation to the following bearers is specified: various GSM services (e.g., GSM-CSD, GSM-GPRS, GSM-SMS, etc.), IS-136, CDPD, CDMA, PDC, iDEN, FLEX and ReFLEX, PHS, DataTAC, TETRA, and DECT. The WAP architecture is open in a way,

that services and applications can be implemented using parts of the architecture, or have direct access to the bearer services, which is shown on the right side in figure 3.

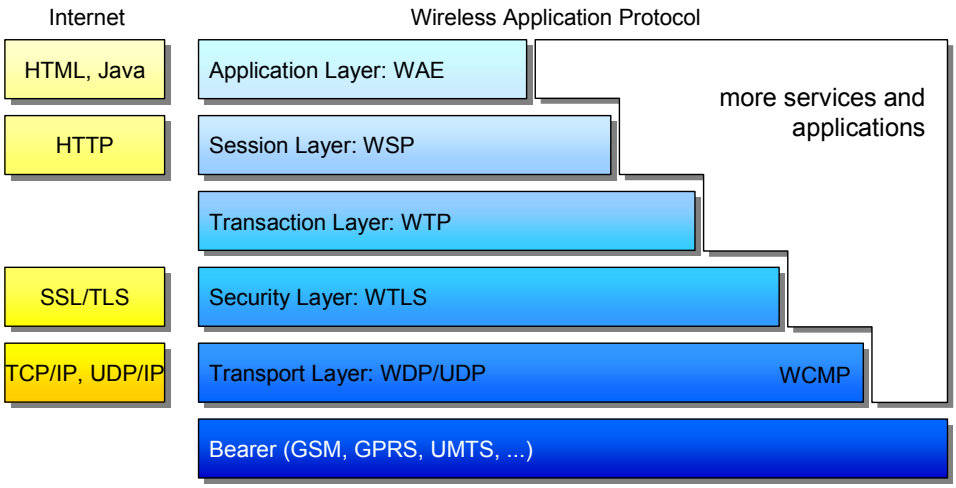

Figure 3: Architecture of the Wireless Application Protocol

Corresponding to TCP/IP in the Internet, WAP specifies the *Wireless Datagram Protocol* (WDP) for the transportation of data. WDP provides an unreliable and connectionless service via a consistent service to the upper layer protocols, so they are able to function independent of the underlying wireless networking technology. The transport layer contains an adaptation layer which supports specific features of the different bearer services. If the underlying bearer service is an IP-based network, such as GSM-CSD, WDP complies to the User Datagram Protocol (UDP) used in the Internet. In this case, the WDP standard provides the same communication scenario used for dial-in via a modem: PPP lies on top of the GSM data channel, followed by the Internet Protocol (IP) and by UDP (see also figure 4).

The Internet Control Message Protocol (ICMP) has its correspondence in the *Wireless Control Message Protocol* (WCMP) that allows to exchange management information or notifications about certain events. WCMP could be used for notification if, e.g., a destination is unreachable, messages are too big for transmission over a certain bearer service, a reassembly failure occurred, or simply for exchanging information via echo request/replies.

Located on top of the transport layer, the *Wireless Transport Layer Security* (WTLS) cares about the security aspects of the communication. The WTLS layer is modular; the mechanisms used for security depend on the required security level of the given application, i.e., applications are able to selectively enable or disable WTLS features depending on their security requirements. WTLS is based on TLS/SSL, but has been optimised for the use over narrowband links. It takes care about the following security aspects:

- *Privacy:* WTLS assures that the information transmitted between the terminal and an application server is confidential. Other participants who might record this information will not be able to read it.
- *Data integrity:* WTLS guarantees that the information transmitted between terminal and server could not be modified or damaged in a way that the receiver does not recognise the modifications.

- *Authentication:* WTLS features a mechanism that both application server and mobile terminal can authenticate each other. Thus, it is ensured that the mobile device communicates with the application server it has contacted, and vice versa.
- *Denial-of-service protection:* WTLS supports mechanisms to identify and reject data which might be replayed by a third party, and makes many typical denial-of-service attacks more difficult in order to protect the higher layers.

For the initialisation of a secure connection, a handshake (HS) protocol (either a full HS or an optimised HS) is used, where client and server agree on a protocol version, select the cryptographic algorithms, optionally authenticate each other, and use public-key encryption techniques to generate a shared secret key. In order to support the features listed above, WTLS has to perform four classes of cryptographic operations:

- *Digital Signing*, by using one-way hash functions as input for a signing algorithm for authentication during the initial handshake operation.
- *Stream cipher encryption*, where plaintext is XORed with an identical amount of output generated from a cryptographic secure-keyed pseudo-random number generator.
- *Block cipher encryption*, where every block of plaintext is encrypted to a block of ciphertext using cipher block chaining (CBC), e.g., RC5, DES, 3DES, or IDEA.
- *Public-key encryption*, using a public-key algorithm to encrypt data in such a way that it can be decrypted only with the matching private key. In WTLS, algorithms such as Diffie Hellman (DH) or RSA can be used for those purposes, especially for exchanging a shared secret key during the initial handshake.

After creating a secure connection, the messages for transmission are optionally compressed, authenticated using a MAC (Message Authentication Code, negotiated during the initial handshake), and afterwards encrypted. Hence, the receiver has to perform the inverse transformation of those steps to receive the original message.

The *Wireless Transaction Protocol* (WTP) on top of the WTLS layer has no explicit complement in the Internet (however, the seldom used T/TCP [6] has similar characteristics). WTP was defined to provide the services that are necessary for interactive “browsing”, i.e., typical request/response applications. Such a request/response pair is called a transaction. The objective of WTP is to deliver a transaction in a reliable way while balancing the amount of reliability required for the application with the cost of delivering the reliability, resulting in an improved reliability over datagram services. WTP provides three different classes of lightweight transaction services:

- *Class 0: Unreliable Request,* which might be used for simple push services.
- *Class 1: Reliable Request,* used for reliable push services. Thereby, the sender will be acknowledged that her or his message was delivered to the receiver.
- *Class 2: Reliable Request/Response,* the basic transaction. A reliable request will be sent to the receiver, which itself responds with the requested data.

In order to achieve the objectives mentioned above, WTP supports protocol features such as message transfer instead of byte-stream transfer, asynchronous transactions, error handling, concatenation and separation, as well as no explicit connection set up or tear down phases (causing excessive overhead). Although WTP is an optional layer, it is very important for,

e.g., mobile banking applications as the user has to be aware whether the transaction s/he adjoined was performed or not.

As described above, HTTP does not perform well in wireless communications. Thus, the WAP-Forum specified the *Wireless Session Protocol* (WSP), which is the last layer underneath the applications. The objective is to enable an organised exchange of information between co-operating client/server applications by establishing and releasing a reliable session and capability negotiation. WSP provides a consistent interface for two session services: A *Connection Mode*, which defines a connection-oriented service that directly accesses WTP functionality, and a *Connectionless Mode*, which specifies a connectionless service, based on a secure (WTLS) or non-secure (WDP) datagram service. The essential properties of WSP are full HTTP 1.1 functionality and semantics, but using compact binary encoding (e.g., using definitions for well-known headers to reduce protocol overhead, or header code pages), a long-lived session state including session migration, and a facility for pushing data to mobile devices.

On top of the WAP stack, the *Wireless Application Environment* (WAE) is located which includes all elements of the WAP architecture related to application specification and execution. This comprises networking schemes, content formats, programming languages as well as shared services, caching, and device profiles. The WAE model consists of four components: WAE User Agents, e.g. a browser that interprets content delivered from the underlying WAP layers. This can be WML (Wireless Markup Language), or WMLScript similar to Javascript. The filters (second component) within the WAP Proxy (see figure 2) act as a converter for WAP content, as they are able to convert HTML to WML. The third component, WTA (Wireless Telephony Applications), is a set of telephony specific extensions that allows for the integration of voice services in the WAP model.

New Features in WAP 1.2

The functionality of each layer described above is the same in WAP 1.1 and WAP 1.2, i.e., WAP 1.1 enabled devices could be used in a WAP 1.2 environment and vice versa, but cannot take advantage of the new features specified in version 1.2. Additionally to version 1.1, WAP 1.2 introduces a WAP pushing architecture. The pushing architecture allows for pushing contents from a server to a mobile client using the push-over-the-air protocol between WAP Proxy and WAP-enabled mobile device. On application layer, the WTA interface was enhanced to support further telephony services. In order to make use of the security aspects at the client, a WMLScript Crypto Library was specified in WAP 1.2. An important progress of WAP 1.2 is the introduction of a Framework for Composite Capability/Preference Profiles (CC/PP) which allows a description of a mobile device's capabilities (so called User Agent Profiles or Capability and Preference Information (CPI), comprising hardware capabilities, software characteristics, application's and user's preferences, WAP-specific settings, as well as the characteristics of the network technology the mobile device is currently using for communication. According to this feature, the relevant mechanisms of WSP were specified in more detail in order to support an efficient transmission and caching of the CPI specified for a mobile device.

The next generation of the Wireless Application Protocol will comprise several interfaces to support further technologies, such as SIM-cards or billing and accounting functionality.

COMMUNICATION WITH CARS USING WAP

The deployment of the Wireless Application Protocol in cars is illustrated in figure 4, which is based on GSM-CSD (Circuit Switched Data, 9.6 kbit/s) for wireless networking. The WAP Proxy is hosted by a (GSM) Service Provider (SP) and is, thus, connected to a GSM gateway (green circle) by PSTN (Public Switched Telephone Network). Note that GSM-CSD is an IP-based bearer; thus, WDP complies to UDP/IP. After requested by the car, the WAP Proxy downloads information from a Web Server in the Internet using HTTP, and sends this information via the Wireless Applications Protocols to the GSM gateway. The GSM gateway forwards the information to the base station the car is currently attached to, which transmits it over the wireless link (CSD-RF) to the GSM module inside the car. This GSM module is connected to the car's onboard communication system, e.g. a MOST or a CAN bus. The information (e.g. traffic information) will be tunnelled to the corresponding device inside the car, such as a WAP-enabled navigation unit that handles it. For browsing in the Internet, WML pages (so called "decks") might be sent via an IrDA port or a Bluetooth module that are also connected to MOST/CAN to a WML browser running on a PDA, as shown in figure 4. Conversely, the transmission from the car to the WAP Proxy (and then into the Internet) follows the inverse communication path.

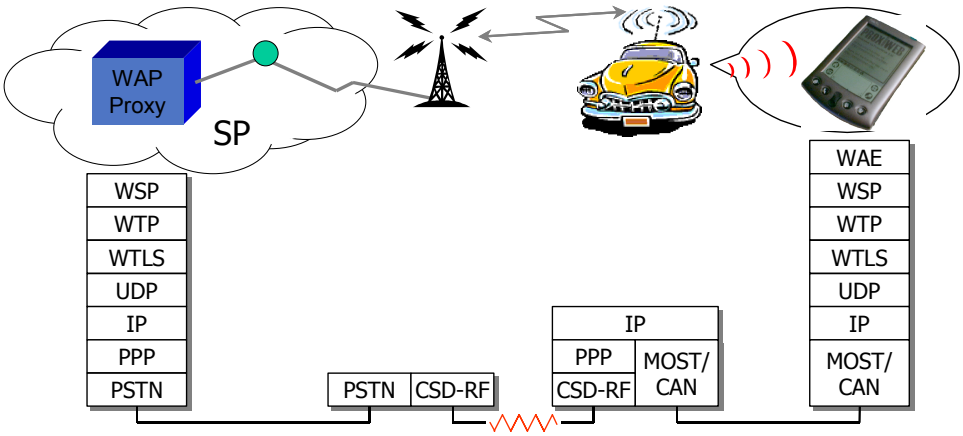

Figure 4: Deployment of WAP in Cars

CONCLUSION

In order to handle the information flood for future cars, the current protocol suite of the Internet as a common platform is unusable, as the protocols are not optimised for mobile and wireless environments. In this paper, we described the architecture of the Wireless Application Protocol and how mobile applications can benefit from the mechanisms and protocols provided by WAP. Those properties empower WAP as a common platform for implementing new applications for future cars.

[1] European Telecommunications Standards Institute (ETSI), http://www.etsi.org/, 2000
[2] Institute of Electrical and Electronics Engineers (IEEE), http://www.ieee.org/, 2000
[3] Infrared Data Association (IrDA), http://www.irda.org/, 2000
[4] Bluetooth Consortium, http://www.bluetooth.com/, 2000
[5] U. Black: *Voice over IP*, Prentice Hall, 2000
[6] J. Schiller: *Mobile Communications*. Addison Wesley, 1999
[7] Wireless Application Protocol Forum, http://www.wapforum.org/, 2000