A MANAGEMENT ENTITY FOR IMPROVING SERVICE QUALITY IN MOBILE AD-HOC NETWORKS

MARC BECHLER, BERNHARD HURLER, VERENA KAHMANN, LARS WOLF
Institute of Telematics, University of Karlsruhe
Zirkel 2, 76128 Karlsruhe, Germany
Phone: +49 721 608 [6397 | 6397 | 8072 | 8104] – Fax: +49 721 388097
Email: [bechler | hurler | kahmann | wolf]@tm.uka.de

Quality of Service in mobile ad-hoc networks has been considered so far with respect to improving routing protocols mostly. However, routing table updates may often be time-consuming and re-routing does not work in the case a mobile device offering an service (e.g. an Internet gateway) is shutting down. In this paper, we propose a flexible Management Entity that integrates both network state changes and service changes into a management information base within the IP layer. Service discovery protocols are helpful in detecting service information. We show that our Management Entity allows switching network technologies efficiently and thus can avoid breaking TCP connections.

1 Introduction

Due to recent advances in hardware technology and the evolution in mobile and wireless networking technologies, mobile devices gain more and more importance. Cellular phones, PDAs, or laptops become more and more powerful and popular as they are nowadays able to perform user's everyday tasks. On the networking point of view, there was a growing demand to interconnect those devices using ad-hoc networking technology, which allows a simplified use of the mobile devices without any configuration effort. Currently, several networking technologies for setting up ad-hoc networks coexist, such as Bluetooth, IEEE 802.11 (in ad-hoc mode), IrDA, or HIPERLAN. With UMTS, new and more powerful communication technologies might come up in the future, resulting in an extremely heterogeneous and variegated environment. A third trend can be seen in the growing importance of protocols for service location, especially for ad-hoc networks. Since static or manual configuration of servers, proxies or gateways may be impossible or at least annoying in many cases, those services have to be discovered within the ad-hoc networking range. Several service discovery protocols have been developed to alleviate the need of configuration for the user, like Jini [1], Bluetooth Service Discovery Protocol (SDP) [2], and the Service Location Protocol (SLP) [3] of the IETF, for example.

In current communication systems those functionalities coexist in terms of independent mechanisms and protocols. However, a combination and interaction of those mechanisms and protocols can provide for synergistic effects and, thus, significantly improve the quality of service support in an ad-hoc network. In this paper, we propose a flexible and efficient Management Entity (ME) located within mobile end systems operating in heterogeneous ad-hoc networking environments. The basis of this entity is a multiplexing mechanism that enables a mobile device to hand over connections between different bearers in a seamless way. A Management Information Base (MIB) supporting the multiplexing mechanism holds out information on the current status of each attached network interface card. In order to improve the service quality in ad-hoc networks, the ME exploits service location information, provided by an arbitrary service location protocol. This functionality, which is integrated within the operating system's network stack, works transparent to the applications so that they need not to be modified. Additionally, legacy applications are also supported and can take full advantage of this handover mechanism.

2 Approaches to Improve the Perceived Quality of Service

In contrast to [4], which shows the need for switching between different networks for wireless infrastructure-based technologies, multi-hop ad-hoc networks meet several new challenges: They are characterized by a highly dynamic topology without any infrastructure-based communication technologies. Communication should be possible even if the networking topology changes rapidly, as mobile devices sending data are moving around as well as mobile routers and mobile devices receiving the data. Mobile devices should also be able to communicate with each other using different networking technologies, resulting in a highly heterogeneous networking environment. Figure 1 demonstrates this heterogeneity in a typical ad-hoc networking scenario with five mobile devices. They are partially interconnected using IEEE 802.11 (in ad-hoc mode) and Bluetooth technology. Additionally, two mobile devices – the cellular phone on the left, and laptop B on the right-hand side – are connected to the Internet: The cellular phone uses GSM communication technology (represented by link (2) to the base station, could be alternatively GPRS, UMTS, DECT, etc.), whereas laptop B has a high-speed ADSL communication link (1) to an Internet Service Provider ISP2. In such a scenario, an improvement of the end-to-end quality of service in mobile and wireless ad-hoc networks can be achieved by exploiting the following two aspects: Utilization of Heterogeneity and consideration of service location information.

2.1 Utilization of Heterogeneous Environments

Although suitable routing protocols enable communication in multi-hop ad-hoc networks (e.g., DSR, TORA, AODV, c.f. [5] for an overview), communication paths between sender and receiver can break when the network is topology is partitioned due to the movements of the nodes. Consider the following situation in our scenario in Figure 1: In order to conserve energy, laptop A in the ad-hoc network initially communicates with laptop B using Bluetooth via the PDA (communication path (3)-(4)). If B becomes unreachable – for example when A moves out of the coverage of the Bluetooth network or when the PDA is switched off – communication is no longer possible. A's TCP connections will time out, even if B is still reachable using the IEEE 802.11 link (5) (which would increase the power consumption of both laptops A and B). The reason is that a TCP connection is uniquely identified by a quadruple (IP address A, port A, IP address B, port B) and switching to another network interface results in a new source IP address related with this interface.

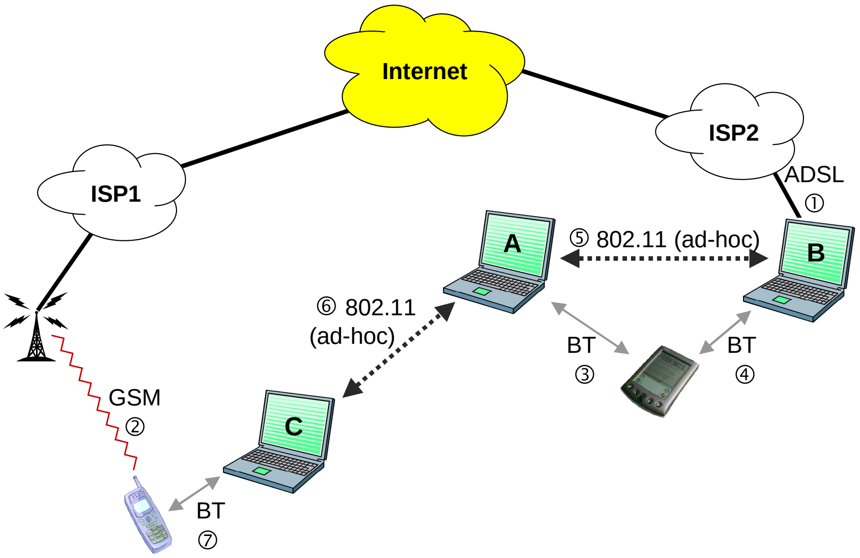

Figure 1. Ad-hoc Networking Scenario

However, it is also harmful to change the bindings of IP addresses to a networking device due to three reasons:

- The mobile node becomes unreachable as the new address bound to the network interface might be topologically incorrect
- The process of binding, unbinding, and the update of the internal routing table(s) is not very efficient.
- Caching of ARP information is not possible as the matching between IP address and MAC address changes after the modification.

Thus, a seamless and transparent switching mechanism between different networking interfaces is needed, which is currently not supported by common operating system's TCP/IP implementations.

2.2 Utilization of Service Information

Several service location protocols have been developed in the last few months. Among the most famous are Jini, UPnP (Universal Plug and Play), Bluetooth SDP, Salutation and SLP. Most service location protocols may be used in ad-hoc networks (see [6] for a characterization of the protocols). Some protocols (Jini, UPnP) even provide service access in addition to service discovery, in case of Jini even without the need of pre-configured drivers for a service. For ad-hoc networks it is also important that a central service manager is not required because in a dynamic environment a centralized entity is always a single point of failure. Thus, service location protocols that implement distributed service managers or enabling direct discovery of services at a particular device should be preferred in ad-hoc networks. Our architecture does not depend on a particular service discovery protocol; however, some protocols are restricted on particular technologies (e.g. Bluetooth SDP).

As described in the previous section, seamless switching between different networks is a basic feature for improving the quality of a perceived service. However, the heterogeneity also implies that the services are also distributed over the accessible ad-hoc networks. Due to device mobility, the services (in each network) need to be regularly discovered and their availability is not ensured as mobile devices (providing services) can be frequently switched off and on by their users. In the scenario described in figure 1, the laptop A has a connection to the Internet, using a gateway from ISP2. This gateway can be reached by means of IP routing via laptop B. If B is switched off (or becomes unreachable due to a link breakdown) A’s connection to the Internet terminates. In this case, the service location protocol running on A has to discover an alternative proxy providing Internet access service in the heterogeneous ad-hoc network. In this case, a proxy from ISP1 will be used which can be reached via the route (6)-(7)-(2). Afterwards, the network settings of A need to be reconfigured and the (communicating) application must be restarted as the source address might have changed. Of course, similar scenarios are conceivable with various other services, such as wireless and mobile telephony in ad-hoc networks using registration services.

In our example, we only need to discover a service and retrieve information about it. This information is promoted to the IP layer via the MIB, so in our scenario, the IP layer will use the information that ISP1 can be used as another proxy if ISP2 is unreachable. Thus, the ongoing communication with hosts in the Internet can be handed over seamlessly to the new proxy and need not be reestablished.

2.3 Architecture: The Management Entity

We designed and implemented a *Management Entity* (ME) that is able to avoid such breakdowns of ongoing connections as described in the example above by handing over the established communication links to a new transmission path. It is also able to consider service location information to find alternative services within the ad-hoc network. Figure 2 shows the integration and the components of this ME in more detail.

On top of the communication layers are the applications that use the network protocols for communication. Additionally, an application daemon implementing a service location protocol is running on the end system, which discovers the available services within the ad-hoc networks the mobile device is currently participating. Within common operating systems, such as Windows 2000 or Linux, TCP and/or UDP is used at the transport layer, and IP at the network layer, located above the link layer. The lowest layers, physical layer and link layer, realize the different transmission technologies for ad-hoc networks. As can be seen in Figure 2, our ME covers link layer, network layer, and aspects of the application layer. This results in an advantageous behavior of the communication protocols, as information from other layers could be utilized for optimizing the protocol mechanisms of the network layer, resulting in an improved quality of service support for the applications.

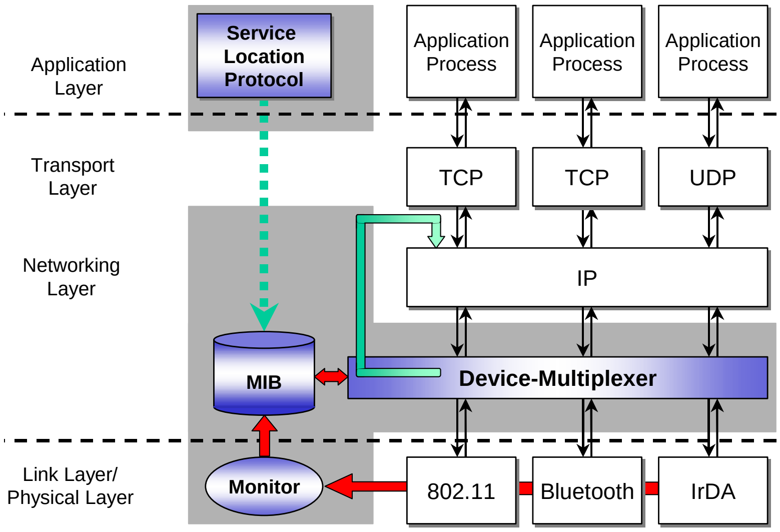

Figure 2. The Management Entity

On the link layer, a *Monitor* collects relevant data from the different networks. As an example, the monitoring of an IEEE 802.11-based ad-hoc network comprises parameters like the availability, or the current packet loss rate (as a result of a calculation based on various other physical parameters such as signal strengths of different access points, etc.). Those parameters are stored in a *Management Information Base* (MIB), which is updated by the Monitor. Up to now, we only consider whether an attached networking interface is currently able to send and receive data to/from other mobile stations. A *Service Location Protocol* running in user space also interacts with the MIB. The MIB is regularly updated with the discovered services, i.e., the ME has an overview of the currently available services within the ad-hoc network and their location (in combination with the routing protocol, the ME knows which networking interface it has to use for communication with this service). If a communication link breaks, the ME is able to find an alternative path to the receiver using another network – assuming that the receiver is accessible in general. If the mobile device uses a particular service within the network and the communication with the device that supports this service fails (e.g., when it is switched off), the ME has immediate access to the information about other stations offering a similar service in the ad-hoc network. The ME is then able to tunnel the packets to another device supporting this service.

2.3.1 Multiplexer

The basic element in our ME for achieving a seamless handover between different communication technologies is a *Multiplexer,* which is also shown in Figure 2. Note that the ME addresses end-to-end quality of service aspects. In multi-hop ad-hoc networks, communication paths between sender and receiver can change frequently due to the dynamics of the network topology. In this case, the ad-hoc routing protocol is responsible for ensuring the adaptation of the forwarding behavior from each node to the reconfiguration of the network. If the receiver becomes unreachable with the actually used networking technology (which can be detected by overflowing transmission buffers), the Multiplexer queries the MIB for alternative communication technologies supporting a communication path to the receiver. In order to achieve transparency for the applications, the TCP/UDP ports and the IP addresses that are in use must not change if the IP packets are sent via another network device, as described above. Hence, our Multiplexer encapsulates IP datagrams in new IP packets with the IP address of the chosen networking device as the source address. This tunneling mechanism, which is the main principle of Mobile IP, is performed by using a redirector within the multiplexing entity, that calls the IP stack a second time and, thus, redirects every packet to another networking device which sends them out. Due to the encapsulation of packets, the new created packet could exceed the maximum size for IP packets. Hence, the created IP packets must be additionally segmented and reassembled if necessary.

When an encapsulated packet arrives at the receiving node, the Multiplexer first establishes a tunnel back to the sender (if such a tunnel does not yet exist), as we assume that a breakdown affects the direction from sender to receiver and vice versa. Afterwards, the Multiplexer unpacks the original IP packet and forwards it to the upper layer protocols using the original mechanisms of the IP implementation. Notice that our approach, all communicating mobile nodes must have a ME.

3 Related Work

Heterogeneous network environments are basically grasped as a challenge, not as a chance for QoS support. There are several projects focusing on the provision of an improved support for mobile hosts at different layers of the network stack, such as [7]. However, none of them addresses the end-to-end QoS aspect in mobile ad-hoc networks. A different approach is to reconfigure the network dynamically, which is the basic of, e.g., [8]. Mobility support for computers is achieved by a dynamic network reconfiguration. The authors define an abstract set of device characteristics, which could also be useful for our Management Entity. In order to achieve mobility, guards within the operating system detect the validation and invalidation of the device characteristics. Depending on those parameters, the system enables an "intelligent adaptation" on network layer, transport layer, and application layer. At the network layer, the adaptation process reacts to invalidations of routes by modifying the routing tables within the kernel of the operating systems. This is different to our approach as we provide a more flexible approach by directly choosing the NIC without changing the bindings from IP addresses and routes.

In contrast to our approach, QoS architectures for mobile ad-hoc networks try to support QoS by managing the resources within each node. For example, the INSIGNIA architecture, an in-band signaling architecture for QoS support in mobile ad-hoc networks [9], the resources within the mobile nodes along a communication path are reserved in a request/reserve manner before communication starts. Rerouting packets due to the mobility of nodes results in a flow restoration along the new path. Although those approaches support QoS for communication, they do not utilize the fact that a sending mobile node can choose between different transmission technologies if the mobile node is able to switch between them seamlessly.

In the area of the IETF many drafts deal with the mobility support for IP and the challenges in MANETs. Most relevant for our work is the work in the MANET working group of the IETF, which basically tries to cope the heterogeneity with routing protocols for mobile ad-hoc networks. However, end-to-end quality of service aspects are currently not addressed in their work.

The Intentional Naming System [10] developed by a research group at the MIT also allows to seamlessly handle node or service mobility. There, a late binding mechanism allows finding the appropriate service at runtime. An overlay network of resolvers uses this binding for routing. The emphasis in this approach lies on the naming system, whereas we do not change naming and therefore need not change applications.

4 Implementation and Evaluation

In order to evaluate our ME, we implemented a prototype using Linux. We used the Netfilter mechanism of the Linux networking architecture to receive, examine and re-inject IP packets. The implementation of the part that controls the ME is completely operating in userspace, which leads to a maximum of independence from future kernel reorganizations. To receive the packets with a userspace process we used the queue target of Netfilter. A so called acting part of the userspace process listens to the state of MIB, where it gathers important information about link quality. It initially decides to change to another networking device by setting up a tunnel over the new link and adding a new rule to the routing table, which sends all packets to the communication partner over the new tunnel device. The so called reacting part listens for incoming tunneled packets (i.e., packets with 0x04 in the protocol field of the IP header). If such a packet arrives, the reacting part immediately creates a “reverse tunnel” and updates the routing table as well. All information necessary for this procedure can be extracted from incoming tunneled packets.

Of course, the implementation in the userspace is very time sensitive. Therefore, we implemented very efficient functions to establish tunnels, to set up links, to add IP addresses and to update routing tables. In order to increase performance, it was important as well to infiltrate packets into the kernel as fast as possible to avoid buffer overflows. Although the implementation works in userspace, it is efficient enough to manage even high data rates.

In order to evaluate our approach, we used two laptops, equipped with IEEE 802.11b PC Cards running in 2 Mbit/s ad-hoc mode. Unfortunately, we had no Bluetooth PC Cards at our disposal, so we used a twisted pair cross-over cable instead. In our tests, we examined the impact of our Multiplexer on the behavior of UDP and TCP packet streams with different data rates. At the beginning, a stream was sent from one laptop to the other using the Ethernet link. After 4.170 seconds, we requested the multiplexing entity to tunnel the stream via the wireless link. In order to get exact performance measurements of the switching process to an alternative networking technology, we avoided a detection of overflowing buffers to identify a broken link. We also ensured the correctness by unplugging the cable. After 8.5 s, the stream was stopped.

Figure 3 and 4 represent two basic aspects of our results. Figure 3 shows a spot of the behavior of a TCP stream with 1 Mbit/s and a packet size of 512 byte in the relevant time range of 4 s and 5 s (switching was initiated at 4170 ms). Firstly, it shows that the TCP connection does not break when the Ethernet cable is unplugged. Secondly, the switching to the wireless link has little impact on the TCP stream itself. Notice that we avoided TCP’s slow start by using traffic shaping within the end system. After sending a TCP packet, the ACK returned before the next TCP packet is sent.

Figure 4 shows our examinations of the delay that is generated by the Multiplexer in the interesting spot of 4.1 s and 4.3 s. We used a UDP data stream with 1 Mbit/s and a packet size of 512 byte. We measured the time when a packet arrived at the receiving laptop by tapping the processor time (i.e., the CPU cycles); thus, our measurements should be very precise. There is a only very short delay when the Multiplexer switches from the wireline to the wireless communication link. Of course, the measurements presented here are only a brief glance. Basically, the results show that the Multiplexer has hardly any impact on the packet jitter: Data rates with less than 512 kbit/s (512 byte packet size) and packet sizes of more than 1408 byte result an no significant delay, as the tunnel is set up between the transmission of two packets. Hence, the measurements show that our Multiplexer is able to switch quickly between different network interfaces, which might be even faster than discovering new routes in ad-hoc networks after a network reconfiguration.

However, a much more significant delay occurs when a broken link should be detected, e.g., by overflowing transmission buffers.

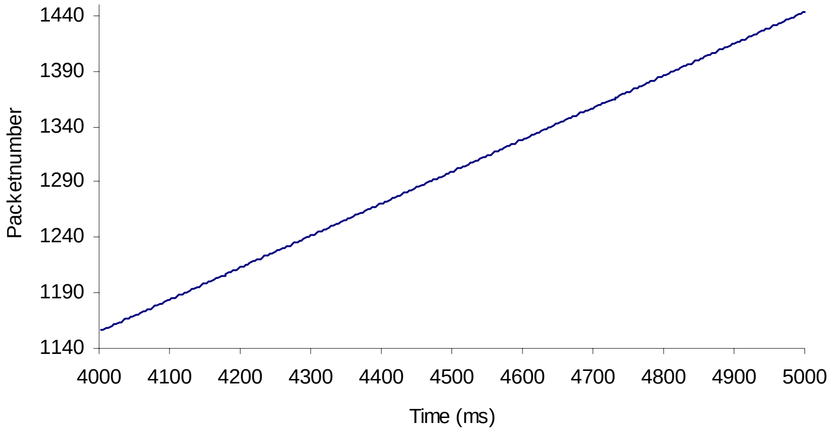

Figure 3. Behavior of a TCP Stream with 1 Mbit/s (512 byte Packetsize)

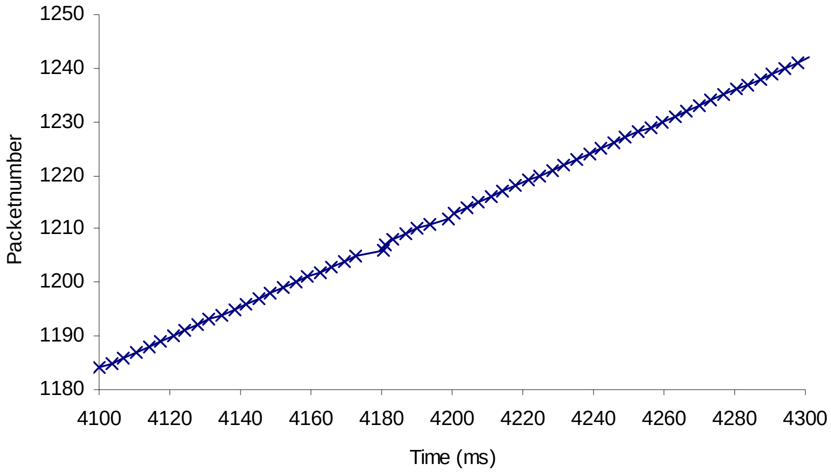

Figure 4. Performance of a 1 Mbit/s UDP Stream (512 byte Packetsize)

5 Conclusion

In this paper we presented a Management Entity for improving service quality in mobile ad-hoc networks. In contrast to other approaches such as QoS based routing, our approach is able to compensate link breakdowns by switching to alternative ad-hoc networking technologies in a seamless way – even if the communication partner is no further accessible via the current networking technology. In combination with information generated by a service location protocol, the QoS support can be significantly improved as the availability of services in ad-hoc networks also vary over time due to device mobility. Future work has to be done by defining common interfaces for our MIB, as well as to extend the MIB by various other parameters that allow a more exact prediction of the link characteristics. Also, we are currently improve our evaluation for the impact of service location protocols on the QoS support. Additionally, we have to consider upcoming communication technologies for ad-hoc networking (such as Bluetooth) as soon as they are available and affordable.

References

1. Sun Microsystems: *Jini Connection Technology*. http://www.sun.com/jini, 2001
2. Bluetooth Consortium: *Specification of the Bluetooth System Core Version 1.0 B: Part E, Service Discovery Protocol (SDP)*, November 1999
3. E. Guttman, C. Perkins, J. Veizades, and M. Day: *Service Location Protocol, Version 2*. RFC 2608, Internet Engineering Task Force, June 1999
4. M. Bechler, H. Ritter: *A flexible Multiplexing Mechanism for Supporting Quality of Service in Mobile Environments*. Proceedings of the 34th Annual Hawai'i International Conference on System Sciences (HICSS), Maui, January 2001
5. E. M. Royer, C. K. Toh: *A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks.* IEEE Personal Communications, April 1999
6. C. Bettstetter and C. Renner: *A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol.* In Proceedings of EUNICE 2000, 6th Open European Summer School, Twente, September 2000
7. R. H. Katz, E. A. Brewer, E. Amir, H. Balakrishnan, A. Fox, S. Gribble, T. Hodes, D. Jiang, G. T. Nguyen, V. N. Padmanabhan, M. Stemm, "The Bay Area Research Wireless Access Network (BARWAN)", *Proceedings of COMPCON Conference*, April 1996
8. J. Inouye, J. Binkley, J. Walpole, "Dynamic Network Reconfiguration Support for Mobile Computers", *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom '97)*, Budapest, September 1997
9. S.-B. Lee, A. T. Campbell: INSIGNIA: In-Band Signaling Support for QoS in Mobile Ad Hoc Networks. *Proceedings of the 5th International Workshop on Mobile Multimedia Communication (MoMUC'98)*, Berlin, October 1998
10. W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, J. Lilley, "The Design and Implementation of an Intentional Naming System", in *Operating Systems Review*, 34(5): pp 186 – 201, December 1999