Application-oriented Routing for Mobile Ad-hoc Networks

Michael Matthes, Felix Apitzsch, Michael Lauer, and Oswald Drobnik[1]

J.W. Goethe-University, Department of Computer Science, Frankfurt/Main, Germany.
e-mail: {*matthes*, apitzsch, lauer, drobnik}@tm.cs.uni-frankfurt.de

***Abstract*: In this paper several routing protocol extensions for mobile ad-hoc networks (MANETs) are proposed, which are designed to support a typical class of ad-hoc application scenarios. In the considered scenarios the reachability of essential services, resources, and users is of special interest from an users resp. applications point of view. Mobile terminals gather information about currently offered services and resources. This information is filtered according to individual interests of users. Only communication paths to "interesting" services and resources need to be managed. This paper presents corresponding extensions to the *Temporally Ordered Routing Algorithm (TORA)*, which have been implemented and evaluated in a MANET emulation environment developed by our group. The interaction between the application-oriented routing protocol and an E-learning application for ad-hoc networks is also shown.**

1 Introduction

Over the recent years, new protocols have been developed to support networking operations for mobile ad-hoc networks (MANETs), e.g., routing. Routing mechanisms for MANETs can be classified in two categories: *pro-active* and *re-active* protocols [3]. Pro-active protocols provide routing facilities to *any* mobile terminal within the current sphere of influence of a MANET. Dynamic changes of the MANET topology most probably cause route maintenance procedures on all mobile terminals. Re-active protocols only provide routing facilities to mobile terminals with active communication paths based on specific communication requests. Hence, this kind of protocols are also referred to as "on-demand" protocols. If the MANET topology changes, maintenance procedures are only required for established communication paths.

Most of the proposed protocols for mobile ad-hoc networks only consider characteristics and requirements of the network layer. Thus, what is achieved with those protocols is to establish communication links with specified mobile terminals. In a MANET it is more applicable to address a service and a service type, or a kind of resource than to address a specific terminal. This extends the view of the responsibilities of a communication infrastructure of a MANET. Besides providing routes to specific mobile terminals, it should provide routes to specific services (e.g., see content-based routing in [2, 8]). That implies, that in addition to the reachability of mobile terminals their offered services should be announced, which triggers the interest in communication requests in the first place.

In this paper several enhancements and application-oriented extensions for an on-demand routing protocol for MANETs are proposed, which realize support for application-oriented routing as outlined above. The following sections 2 and 3 give an overview of current routing protocols, present a typical class for ad-hoc applications scenarios, and derive specific requirements for application-oriented routing facilities. As outlined in [4], ad-hoc routing protocols should be specifically tuned to the characteristics of the scenario they will be deployed in. The resulting extensions for an on-demand ad-hoc routing protocol are defined in section 4, which meet the previously identified requirements. In section 5 an evaluation environment for MANET protocols and applications is presented. Section 6 shows the interaction between the developed routing protocol and an E-learning application, which also has been developed by our group. Finally, the results are discussed.

2 Routing for Ad-hoc Networks

Route management in mobile environments is facing several challenges which do not exist in wired network infrastructures. Other than in fixed networks, where routing and forwarding uses a hierarchy of domains and subdomains, the dynamic changes of ad-hoc network topologies prevent such hierarchical configurations. Hence, for each terminal separate routing efforts needs to be applied. Even if routes have been established for every terminal in a MANET, moving terminals might cause link losses and multi-hop communication paths might break. It is the task of route mechanisms to provide alternative paths or to find new ones. Another issue with mobile ad-hoc networks is the integration of new mobile terminals in existing MANET infrastructures. A new mobile terminal should be informed about the currently setup and maintained routes. Then it should check if it can provide new alternative paths.

In the following, relevant re-active routing protocols for MANETs and their characteristics are addressed. Re-active routing protocols are in general source-initiated and on-demand driven [3]. Routes are created and maintained only by explicit route requests to specific destinations. This might cause some inital delay in the communication flow of an application, when setting up a new connection. However, re-active protocols have the advantage that route maintenance is only necessary for needed routes. The protocols can be differentiated by their route maintenance mechansims: MANET-wide and local recovery mechanism. MANET-wide recovery mechanisms implicate route deletion due to (single) link losses and – if re-requested by the source – a new route discovery mechanism has to be started. The *Ad-hoc On-Demand Distance Vector Routing (AODV)* determines during route discovery a single path from the source to the destination. If the route fails, a new discovery mechanism has to be initialized. The *Dynamic Source Routing*

[1]This work has been supported by the Deutsche Forschungsgesellschaft (DFG) as part of SPP 1140.

(DSR) protocol caches alternative routes to destinations. If a single route fails, the next one is chosen. If no alternative routes exist, a new route discovery mechanism is started. In contrary to those protocols, the *Temporally Ordered Routing Algorithm (TORA)* protocol [7] needs only recovery mechanisms close to the location of the link loss. During route discovery TORA also detects alternative paths from a source to a destination. If a link loss occurs, all routes over that lost link are re-routed via an alternative link. If no alternative link is available, re-routing is directed to the predecessors on each route. This mechanism is called *link-reversal*. If a link-reversal takes place, it is the responsibility of the predecessor to identify alternative routes. This might lead to a "cascade" of link reversals. If no alternative path can be found, the route is deleted and no longer maintained.

3 Requirements Analysis

In this section a special type of application scenarios for ad-hoc networks is identified. In each scenario a common pattern of communication flow can be observed. Based on the scenarios several requirements for application support on the network layer are derived.

3.1 Application Scenarios for Ad-hoc Networks

The following application scenarios are reflecting the spontaneous characteristics of ad-hoc networking [6].

Instant Messaging: Internet messengers like ICQ, IRC or USENET are used for synchronous and asynchronous discussion forums. Generally, communication takes place in a homogeneous group where all participants share common topics of interest. In ad-hoc environments users with common interests have to be found dynamically and have to be made aware of each other.

Study Groups: In schools or universities, students often come together in order to discuss specific topics of current teaching materials or to prepare for the next exam. In such contexts it is desirable to find students with similar interests and to cooperate with them, for instance, grant each other access to adequate information resources. For this purpose students should provide a profile of their learning interests.

Accomodation Agency: In an application for room renting there are two kinds of participants: hirer and renter. The base for a successful transaction is a detailed specification of the available rooms, for example of a hotel or for subletting, and a perhaps less concrete specification of wanted rooms.

Car Pooling: Two kinds of participants can be distinguished: the drivers and the passengers. A specification is needed for describing the circumstances of the drive: destination, departure time and location, number of seats, and more.

Flea Market: Two roles for participants can be identified: seller and buyer. A flea market especially is a very flexible application because of the wide range of specification possibilities for offered or wanted products.

In these ad-hoc scenarios there exist a context-specific accumulation of interest groups as it might occur in reality, e.g. study groups are most probably found in the vicinity of universities. This context-specific accumulation is useful in defining requirements for application-oriented routing mechanisms.

3.2 Application-oriented Route Requirements

The above mentioned application scenarios have a common procedure for communication requests, which comprises interest-, role- and content-based patterns. The reason for a communication request consists of a notification of applications resp. users about "interesting" services, resources and other users with similar interests or complementary roles. The flow of information is determined by specific interests rather than by explicit well-known mobile terminals.

Consequently the following requirements need to be addressed when implementing application-initiated route request mechanisms:

1. Each participant of a MANET only requires access to context-oriented resources and services. The context is defined by the individual interests of a participant.

2. Offered resources and services need to be propagated within the current MANET to discover correlation between interests.

3. Access to the services offered from an application-oriented routing protocol should be provided for middleware and application components as well as for users.

From requirement 1 follows, that there is no need to manage routes to each terminal of a MANET. Only routing support among mobile terminals with correlated interests is required. The current correlation of interests, which implies the required routing configuration, is given by the interests, offered resources and services of the current participants of the MANET, which may change over time. Hence, for routing in such MANETs an on-demand routing protocol is appropriate.

Requirement 2 implies a distribution mechanism for profiles which describe the kind of resources, services and the interest in those. Actually, the distribution of application layer information is not a function of routing mechanisms. But in general, routing mechanisms exchange and distribute routing information in order to operate. Hence, it would be efficient to combine different distribution mechanisms, if there is a need on several protocol layers for information distribution[2]. In this way, it would be possible to reduce the usage of communiation resources for management functions.

Requirement 3 refers to an application programming interface (API), which provides as set of functions for using the services of the routing mechansims. Resulting from the requirements above the following functions are necessary: initiating route setups, deleting routes, adjusting level of importance for maintenance of specific routes[3], passing *piggyback* distribution information, and

[2] For instance, a common set of multipoint relays as used in the Optimized Link State Routing (OLSR) [1].

[3] Level of importance should indicate the effort to apply for maintainig a route. This should correlate with the interest in that route, for instance, given by the utilization/usage of the route.

setting the distribution degree, e.g., time-to-live (TTL) for bounding the range of propagation, or distribution frequency.

4 TORA Enhancements

Following requirement 1 of section 3.2 it is appropriate to deploy an on-demand routing protocol. As a first approach we have chosen to implement and extend the MANET routing protocol *Temporally Ordered Routing Algorithm (TORA)* because it implies low costs for route management. Compared to other re-active ad-hoc routing protocols (e.g., DSR, AODV) it requires only one communication phase and no periodic status messages to recover from broken links. Furthermore, it has the capability to resolve broken routes locally to the point of occurrence in trying to forward via alternative paths already found during route request procedures.

For efficient support of the described ad-hoc scenarios several enhancements were needed to implement. In the following sections the features of TORA are presented. Furthermore, we identify cases, in which TORA does not provide efficient support, and propose appropriate mechanisms to enhance TORA.

4.1 Dynamic Neighborhood Awareness

TORA is a highly adaptive, loop-free, distributed routing algorithm based on the concept of link reversal. During route request procedures, TORA gathers information about multiple routes between the source and the destination. Despite link losses there might be no communication needed between MANET stations in order to maintain a route. First, alternative paths via still existing links are used. Only if no further link providing an alternative path is available, TORA tries to resolve the broken route. This is done local to the point of failure by informing the predecessor on the broken route and instructing it to select another path via its alternative links. This is accomplished with so called *link-reversals*, that means, the link designating the current terminal as the next hop on the path is reversed. Depending on the network topology and its degree of dynamics most broken routes can be handled locally for quite some time. Compared to other on-demand protocols like DSR and AODV the need for re-requesting a lost route can be postponed.

Furthermore, TORA is able to detect network partitioning and in that case to erase invalid routes. A route is only valid in the partition, which contains the destination terminal of the route.

The problem of route establishment in TORA is based on generating and maintaining a destination-oriented graph. A graph is called destination-oriented if each terminal knows a path to a given destination, that means in particular the next hop in direction to the destination. Such a graph is modelled as a so called *directed acyclic graph (DAG)*.

A. Unawareness of alternative paths: After route establishment TORA does not exchange any route management information other than link reversals. This behaviour saves communication bandwidth. However, in some cases more information exchange is desirable. The problem is, that after the initial route construction phase

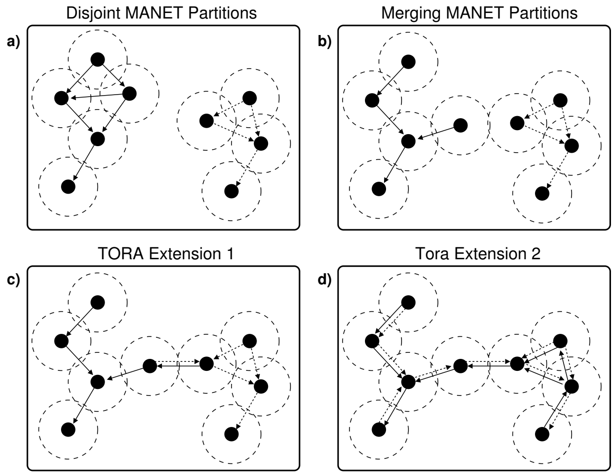

Figure 1: *Merging* MANET *partitions: a) Two seperate* MANET *partitions. b) Merging partitions with* TORA*. c) Route exchange of new neighbors. d) Merging partitions after gradual propagation.*

no further alternative paths are searched. New mobile terminals entering the sphere of influence of a MANET could provide alternative paths. But if a mobile terminal arrives after a route query procedure, it has no knowledge about that route and TORA does not announce the existence of an already established route (see (a) and (b) in figure 1).

The efficiency of TORA is also affected, if two neighboring terminals diverge from each other and loose their connection for some time. As long as they were neighbors during route establishments they are aware of their relations regarding each route. When they drift apart these relations are lost. Route information regarding those lost relations is not exchanged when they approach each other again, although they might provide alternative paths.

These aspects are not covered by TORA. But they are relevant in scenarios as described in section 3. In order to integrate new mobile terminals into established routes, we extended the original TORA algorithm. When new links between approaching mobile terminals are detected, they update their routing information. The information exchange mechanism causes additional load on the communication resources for route management. It also adds a kind of pro-active behaviour regarding requested routes. However, it ensures the consideration of each MANET terminal – also new ones – for route maintenance procedures and improve route stability (see (c) in figure 1). Furthermore, adding new alternatives decreases the probability of link reversals.

B. Merging network partitions: The situation of new mobile terminals also affects merging network partitions. If disjoint partitions get in contact with each other, the routes established in each partition are not propagated into the other partition (see (b) in figure 1).

Mobile terminals moving into the overlapping zone of two partitions get route updates from the "edge" terminals of the foreign partition (cf. the mechanism outlined in A). In addition, a mechanism is needed to propagate that route information into each partition respectivley.

To anticipate uncontrolled flooding, gradual propagation has been implemented (see (d) in figure 1). The rate of propagation can be specified and should depend on the mobility rate within a partition: the higher the mobility rate the higher the probability of terminals moving into the "foreign" partition. In this way, the foreign partition is requested to provide routes for a new mobile terminal in advance, local to its entering point and to the region it might move to.

4.2 Application Interface

Our implementation of TORA provides two interface types for external access: route management and application-oriented management. The access is implemented with an RPC mechanism. The route management interface includes the following methods:

- requestRoute(*MobileTerminal*): request a route to a specified address of a mobile terminal.

- clearRoute(*MobileTerminal*): remove a route to a specified terminal.

These methods allow applications to set up routes to known mobile terminals. But in order to learn about currently reachable mobile terminals within the MANET resp. currently offered services the following additional notification mechanisms have been implemented in the application-oriented management interface:

- fuzzy presence awareness of mobile terminals,

- service- resp. interest description propagation.

For a fuzzy presence awareness, information about currently reachable mobile terminals is gathered passively. This information can be provided to the upper protocol layers. The information is collected via directly neighboring terminals, whose existence is known by TORA anyway, and through recording the destinations of successful route queries. If a link to a neighbor is broken or a route to a mobile terminal is cleared, a timeout is started, which marks the probably remaining lifetime of the mobile terminal in the MANET. The timeout depends on the mobility rate of the MANET, which must be estimated separately. This presence awareness does not add any communication costs, but on the other hand it does also not necessarily cover all mobile terminals which could be reached.

The propagation mechanism distributes application-specific data containing descriptions of services, resources, or interests (cf. section 3). Route management messages are extended whith an extra *cargo data* field. Application-specific data can be distributed in a so called *piggybacking* manner. In that way duplication of lower layer functionality is prevented. In general, the *cargo data* field extension is not big enough to propagate application-specific information at once. Therefore a fragmentation mechanism is implemented, which splits the information packets and propagates them sequentially. To span multiple hops, received information will be forwarded.

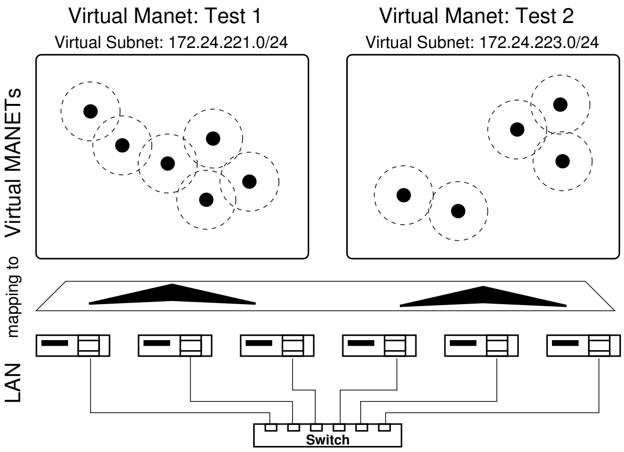

Figure 2: *Mapping LAN stations to Virtual Terminals of several Virtual MANETs.*

5 Evaluation Environment

Protocols and their interaction with applications cannot be evaluated very conveniently in real mobile ad-hoc networks with several mobile terminals moving around. Using simulation environments like *ns2* or *GloMoSim* helps in analyzing protocol behaviour. However, those tools do not feature any real application interaction. Our research focus is to study how applications are affected by the underlying protocols. For testing protocols and applications with respect to mobility characteristics of MANETS our group has designed and implemented a MANET emulation environment for regular switched LANs, which is called *Manet Server SImulating Virtual Environments (Massive)*.

On each station of the wired LAN infrastructure a *Massive*-Server emulates several virtual MANET terminals – one virtual terminal for each MANET. A MANET consists of a specified subset of virtual terminals (see figure 2). A virtual terminal communicates with other virtual terminals over a virtual device, which operates over the stations' real network device. Several virtual devices can be configured for a real network device. Thus, several virtual MANETs can be emulated concurrently with *Massive*. A MANET environment is modelled by controling the reachability among each virtual terminal of a virtual MANET. This is accomplished through packet filtering. Packets received from virtual terminals, which are not in the specified immediate vicinity, are discarded. In this way, a mobile ad-hoc network can be emulated with terminals moving in resp. out of each other's vicinity.

Our system has been implemented on Linux using its packet filtering mechanism *IPTables*. The reachability of LAN stations regarding a virtual device is controlled by deploying filters in the kernel. Activating and deactivating filters in the system is carried out by the *Massive*-Server. A similar approach for MANET emulations has been carried out in [9] and the so called *MobiEmu* emulator. For users without root priviliges *MobiEmu* provides two different concepts: user-space packet filtering and virtual machines running on top of an operating system. In our approach a password protected access to the privileged mechanisms of the kernel-space packet filtering is provided. The interface is based on RPC calls. Hence, manipulation of the reachability of virtual terminals with the speed of kernel-space packet filtering is accessable for non-root users. The access to the privileged filter functions is provided by the following RPC calls:

- addNeighbor(ManetName, Password, MobileTerminal): checking access privileges to the MANET and configure IPTables in order to receive packets from the mobile terminal. Incoming packets are passed through to upper protocol layers.

- delNeighbor(ManetName, Password, MobileTerminal): checking access privileges to the MANET and configure IPTables in order to reject incoming packets from the mobile terminal. Incoming packets are discarded.

For configuring virtual MANETs, a management tool has been developed. It provides a graphical user interface to setup virtual MANETs, to assign stations to them and to choose the routing protocol (see figure 3). For

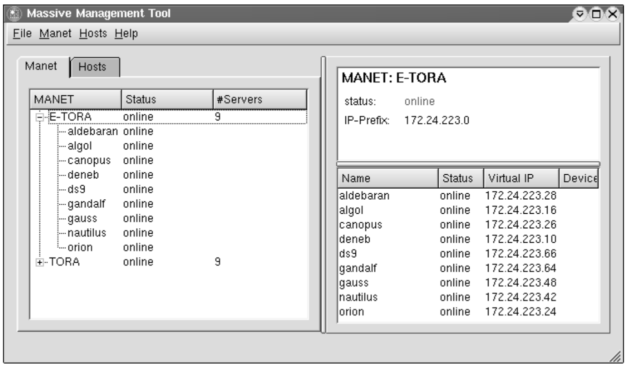

Figure 3: *Manet Management Tool.*

modelling the reachability and the dynamics within a MANET a simulation tool has been developed. It provides a graphical user interface for arranging virtual terminals within a MANET and for defining direction and speed of movements for each terminal (see figure 4). Several mobility patterns are integrated (e.g., random, directed). The range of the radio transmitters is modelled by circles. For visibility reasons the circles indicate half of the radio range. An overlapping of two different circles denotes that the corresponding mobile terminals are within direct radio range. The simulator tool then calls the neighborhood management RPCs of the appropriate *Massive*-Servers.

For evaluating routing protocols, appropriate MANET scenarios can be configured. The movement of the virtual terminals can be managed manually or by using predefined mobility patterns. Our initial interest covers on-demand protocols only. Hence, the simulator provides methods to setup routes between virtual terminals. The following methods are provided as an interface to the routing module:

- requestRoute(Source, Target): initiate a route request from *Source* terminal to *Target* terminal.

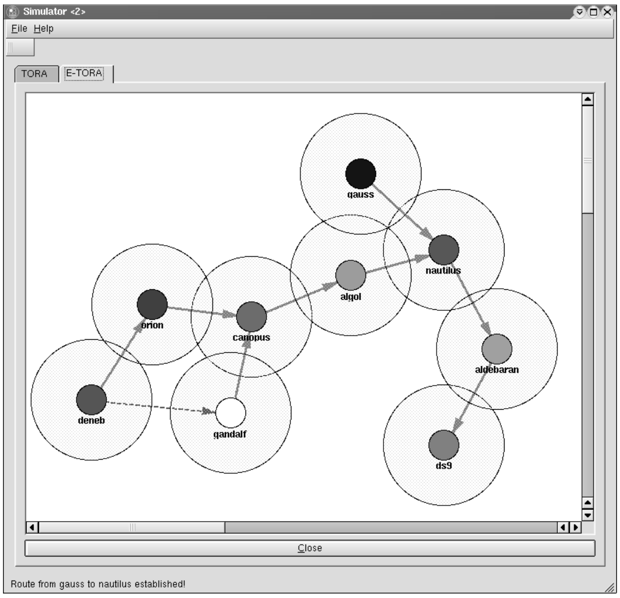

Figure 4: *Manet Simulation GUI.*

- clearRoute(Source, Target): remove the route from *Source* terminal to *Target* terminal.

Modifications to the routing table on each station regarding the virtual MANET are reported to the simulator. The simulator visualizes added routes with an arrow, where the color of the arrow complies with the color of the destination. The direction of the arrow marks the next hop to the destination. Thus, the operating characteristics of routing protocols can be validated for general and for specific ad-hoc scenarios.

The simulator may also visualize the routing mechansim for pro-active routing protocols. As those protocols calculate routes to every terminal in a MANET visualization filters have been implemented in order to observe the maintenance mechansim for specific routes.

Further advantages of a MANET emulation in a switched LAN are the evaluation facilities for applications designed for ad-hoc networks. Applications can be tested as if running in mobile ad-hoc networks[4] by configuring the communication properties of the application such that the virtual device of the appropriate MANET is used.

6 Application Layer Interaction

In this section the interaction between applications and the extended TORA protocol is described. Our group currently developes an E-learning application for mobile ad-hoc networks. The considered learning environment complies with the requirements of the ad-hoc scenarios described in section 3. A detailed description about the application scenario and the proposed objectives as well as a framework for developing applications for this special scenario are outlined in [5] and [6]. In the following, a short overview is given over the E-learning application and the development framework. The interaction of the E-learning application and the extended

[4]So far, the characteristics of the radio layer are not considered in the *Massive*-System.

TORA protocol is shown with our MANET emulator.

6.1 ELAN Project

The goal of the project *E-learning in Ad-hoc Networks (ELAN)* is to develop an integrated component-based collaboration environment for E-learning in mobile ad-hoc networks. The work is motivated by a scenario often found at universities. Several students use their laptops or PDAs to gather information, to carry out tests, to exchange information with other students or to discuss the currently presented contents of courses. Collaboration in such an environment is most probably motivated by similar educational objectives. Hence, the first step for initiating a collaboration is to notify students about common interests, common research areas, commonly attended lectures, etc.

In order to find groups of students with similar educational objectives, the interests of the students have to be correlated. The ELAN system supports personalized profiles for knowledge, interests, and services. Each student can describe his own interests, objectives, and services he provides on his mobile terminal. Those profiles need to be propagated. If a correlation of profiles is detected, the corresponding mobile terminals resp. students should be notified.

6.2 ELAN Architecture

The ELAN system is based on a layered architecture (see figure 5). E-Learning components can be developed using the application framework introduced in [6]. With a *Profile Editor* users specify interests, available resources (for example, some self-provided report on a specific topic), and offered services. Then the profiles are passed to the middleware, which is responsible for the propagation and the correlation with incoming profiles. The middleware uses the services of the network

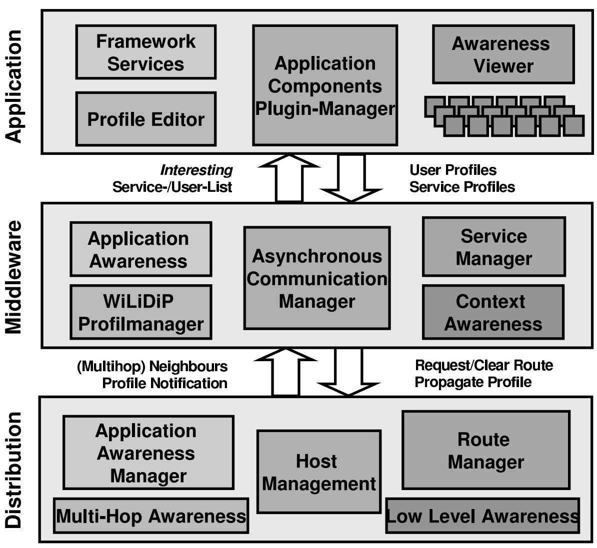

Figure 5: *ELAN Framework Architecture.*

layer for propagation. If the middleware finds overlapping interests, it notifies the user via the application layer and – depending on the degree of correlation – it initiates a route query to the corresponding mobile terminal in advance before any request from the upper layers is submitted. Thus, if an application resp. the user wants to initiate a communication, the route might has been setup already.

6.3 Interaction of TORA and ELAN

The extended TORA protocol as described in section 4 provides the required interface for the ELAN system. The middleware can use TORA's route management methods and the implemented propagation mechanisms. We have integrated the extended TORA protocol in the ELAN system and run tests within the emulation environment.

The propagation mechanisms are used to distribute the profiles of the ELAN system. The profiles of ELAN are organized hierarchically. Because the size of propagation packets is much less as the profile size (cf. section 4.2), the middleware segments each profile in hierarchical chunks (subtrees), which are passed successively to the routing module. In that way, a correlation can be conducted even if only a partial profile is available.

The notification of applications about overlapping interests of several students is shown in figure 6 and 7. Figure 6 shows the profile editor, where students can

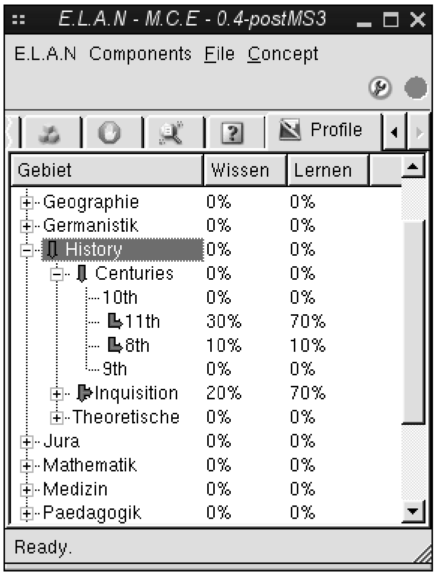

Figure 6: *ELAN Profile Specification.*

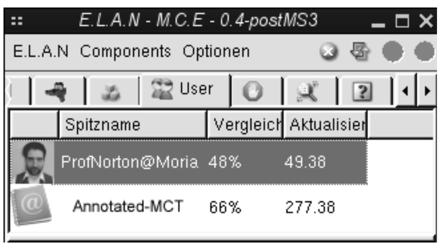

Figure 7: *ELAN User Notification.*

specify their interests, resources, and services. They can rate their knowledge and also their curiosity about serveral topics. A threshold for the correlation degree can be adjusted. Upper layers are informed about new events only if the correlation degree is above that threshold. The specified profiles are then given to the middleware, which is responsible for distribution and correlation.

Figure 7 shows the awareness notification. The system has found a student with correlating interests as well as a multiple choice test (mct), which covers a topic of interest. The degree of correlation with the other student and with the offered service is shown in percentage. Also the elapsed time since the last refresh of that information is shown. It is now possible to contact the student or to use the offered service.

7 Conclusion

The objective of this work was to design and implement application-oriented routing mechansim for specific ad-hoc scenarios. A typical class of ad-hoc application has been identified and a set of requirements for appropriate routing mechanisms has been determined. Based on the application scenarios the on-demand routing protocol *Temporally Ordered Routing Algorithm (TORA)* has been implemented and extended to meet the identified requirements. Beside mechanisms for detecting and handling network partitioning the extended routing algorithm also supports the merging of network partitions.

For interaction with upper protocol layers an interface for route management and application-oriented management has been integrated in the routing module. By gathering network information and propagating application-oriented information via *piggybacking* using route management packets the communication costs of serveral protocol layers have been combined and overall reduced. The proper operation of the enhanced TORA has been demonstrated by using a MANET emulation environment, that has been developed by our group. The interaction of the TORA extensions and applications have been presented using an E-learning application for ad-hoc networks. The contact facilitation of this ad-hoc application has taken advantage from the application-oriented management functions of the enhanced TORA.

Further work is currently carried out regarding the support for efficient group communication. In *Lightweight Adaptive Multicast (LAM)* a multicast extension for TORA is proposed. It needs further research if the proposed multicast mechanism is capable to support communication scenarios as outlined in section 3.

The distribution of profiles within an ad-hoc network could profit from an infrastructure usually built up by some pro-active routing protocols. For instance, the Optimized Link State Routing (OLSR) protocol defines so called *Multi-Point Relays (MPR)*, which optimize flooding procedures in MANETs. It has to be examined whether the communication costs for maintaining such an infrastructure are comparable to the piggybacking mechanism presented within this paper.

Another promising approach for application-oriented routing for ad-hoc networks are content-based routing mechanism used for peer-to-peer networks. Their usability for ad-hoc networks should be analyzed in more detail.

REFERENCES

[1] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol. *Work in Progress, IETF draft, MANET Working Group, INRIA Rocquencourt, France*, April 2003.

[2] A. Carzaniga and A.L. Wolf. Content-based Networking: A New Communication Infrastructure. *NSF Workshop on an Infrastructure for Mobile and Wireless Systems. In conjunction with the International Conference on Computer Communications and Networks (ICCCN'01), Scottsdale, USA*, October 2001.

[3] C.E. Perkins et al. Ad-hoc networking. *Neworking/Mobile Computing, Addison-Wesley, USA*, 2001.

[4] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. *In Proc. of International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, Washington, USA*, pages 195–206, August 1999.

[5] Michael Lauer and Michael Matthes. ELAN: An E-Learning Infrastructure for Ad-hoc Networks. *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (MobiCom 2002), Poster Presentation, Atlanta, Georgia, USA*, September 2002.

[6] Michael Lauer, Michael Matthes, and Oswald Drobnik. A Framework for developing applications for ad-hoc environments. *In Proc. of the 4th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, (SNPD'03), Luebeck, Germany.*, Oct 2003.

[7] V.D. Park and M.S. Corson. Temporally-ordered routing algorithm (tora). *Work in Progress, draft-ietf-manet-tora-spec-04.txt, IETF MANET Working Group*, Jul 2001.

[8] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. *In Proc. of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press, San Diego, California, USA*, 2001.

[9] Yongguang Zhang and Wei Li. An Integrated Environment for Testing Mobile Ad-Hoc Networks. *In Proc. of the third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02), Lausanne, Switzerland.*, June 2002.