Moversight: A Flexible P2P Group Communication Protocol for Mobile Collaborative Applications

Jan Gaebler and Michael Kirsche
Computer Networks and Communication Systems Group, Brandenburg University of Technology Cottbus, Germany
{jgaebler, mkirsche} @ informatik.tu-cottbus.de

***Abstract*—We introduce Moversight, an early stage group communication protocol, specifically designed for collaborative Peer-to-Peer (P2P) applications in mobile environments. The protocol especially deals with problems encountered in mobile scenarios. It provides basic properties like virtual synchrony of cooperating entities, support for stationary and mobile participants, self-organization and scalability, as well as support for narrow-band communication links without additional central infrastructure.**

I. Introduction

Mobility is a key aspect of today's communication usage. 3G networks and smartphones enable an almost world-wide mobile access to information and applications. This anywhere/anytime access to communication and information, often associated with the term *ubiquitous communication*, leads to new opportunities for collaborative applications in mobile and ubiquitous scenarios, including the chance to use traditional infrastructure-based cooperative appliances in mobile environments. Computer supported cooperative work (CSCW) applications or other collaborative examples like video conferencing systems can thus be freed from their infrastructure boundaries and used in an ubiquitous manner. On the other hand, wireless networks like UMTS and portable devices introduce new problems for collaborative applications. Future applications should be usable from a wide range of devices and access networks. Application developers are hence confronted with the task of dealing with various access technologies, runtime environments, and device characteristics while keeping the problems of wireless networks, mobile users, and collaborative applications in mind.

The *uBeeMe* project (refer to [1], [2]) deals with these topics and the task of creating a framework that supports application developers with a solution for problems related to ubiquitous network access as well as application and user mobility. Within the *uBeeMe* project, the group communication protocol *Moversight* (*Mobile Virtual Synchronous Group Communication*) is developed. Through *Moversight*, collaborative applications are supported in their tasks of managing participating group members, administering join and leave processes, providing a membership list and an information service for participants, and finally securing the closeness, security, and virtual synchrony of all group members.

Moversight and the *uBeeMe* project specifically consider collaborative applications like video conferencing systems or cooperation supportive applications in Peer-to-Peer (P2P) contexts. Other examples for current and next generation collaborative P2P applications are: mobile auctions, mobile multi-player games, and the coordination of relief units in emergency scenarios. All these applications need a consistent and reliable group management. These partially open or closed groups usually consist of a mix of mobile and stationary participants, whereas the mobile participants are characterized by varying connection and mobility criteria.

Currently available group communication protocols like *DCCP* [3], *GCP* [4] or *IGMP* [5] do not support the mobile aspects of group members in their full extent. These protocols do not consider typical problems of mobile systems like high churn rates, unstable wireless connections, varying Quality-of-Service (QoS) parameters or a non-availability of infrastructure. Central goal of the *Moversight* group communication protocol is therefore the consideration of mixed closed groups and their group management through a light-weight P2P approach. With the abandonment of traditional central infrastructure elements (e.g. group access or management server, central sequencers), a flexible, scalable and self-organizing structure can hence be defined, which itself can be used without any additional means in ad-hoc environment and thus in the considered ubiquitous scenarios.

II. The Moversight Protocol

The *Moversight* protocol focuses on the management of the group. Other services are hence needed to support *Moversight* in localizing users and transmitting messages. These services are provided by other modules of the *uBeeMe* platform. Information regarding these modules can be found in [2].

Moversight enables a group communication service for closed peer groups. It supports mobile participants and provides a message transfer service, which itself follows the principle of virtual synchrony. Through this, all peers in a virtual group are guaranteed to have the same state. The peers are classified as master or slave in a cluster-based approach. Figure 1 illustrates this with an example topology and a mapping between actual access network connections and virtual *Moversight* group connections. Initially, every peer can potentially fulfill the role of a slave or master node. *Moversight* enables the use of different placing strategies to define how peers are selected as master or slave and how they are placed inside a group. Current placement strategies implement a simple first come - first served approach. More complex strategies are currently work in progress. We think about extensive strategies based on additional information on

the peers, e.g. peer capacity utilization or geographic positions compared to other group members. Through the placement strategy, energy consumption and work load can be optimized and equally distributed inside the groups.

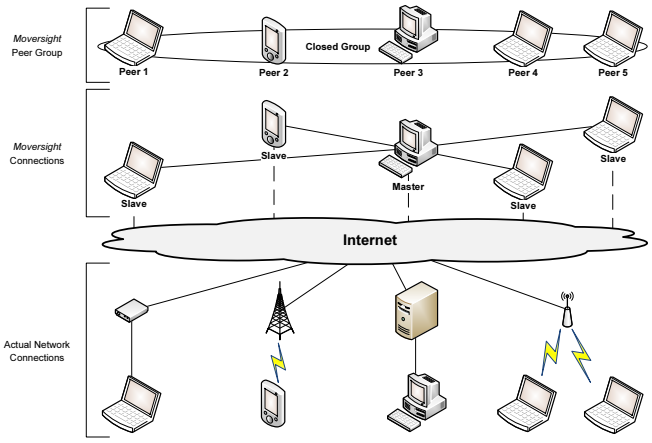

Fig. 1. Moversight topology and mapping between virtual group connections and actual access network connections

While group communication protocols like *GCP* [4] use a token ring concept to synchronize participants, *Moversight* enables the virtual synchrony of groups through a Lamport-based atomic send approach, comparable to the *ISIS* system [6] but extended with our clustering approach. Information about local clocks is distributed in the group and globally aligned by the group's master. Clock information about send and receive operations is thus used on a global scale to arrange and align messages and events. Through this, virtual synchrony can be guaranteed on local and global level. The atomic send operation is split in the distribution of information between slaves and between masters. Information from slaves is only sent to the master, while master peers exchange information globally and later on distribute the synchronized information again in their local group. This is a consequence of the introduced mobile and ubiquitous P2P scenarios, where traffic should be reduced and work load should be equally distributed.

A peer itself maintains important information locally. He stores a *location descriptor*, which consists of a peer ID, a group ID, and the public transport address of the peer (namely IP address and port number). In addition, peers locally maintain a list of all global group members with their individual status, location descriptors, cluster number, and roles (master or slave). If a peer temporarily cannot reach his group, he can use these information to perform a fast rejoin with minimal message exchange. Reachability of peers is detected through a two-stage heartbeat mechanism. Inside a cluster, all members inform each other through regular heartbeat messages about their status. If a master drops out of the cluster, any other slave can instantly become the new master since he possesses all needed information through his local information list. If a slave does not respond, a `ViewChange` event is generated, through which all members are informed of the absence of the slave peer. To reduce the number of absence notifications, peers first assume that the absent peer will come back in a defined time `t` and thus set his status to pending. If the absent peer does not respond after `t`, he is then dropped from the group. This is necessary because peers connected through wireless networks might encounter regular short connection losses, which would lead to excessive exchange of absence notifications.

The current state of the *Moversight* protocol architecture is depicted in a simplified way in Figure 2. The higher protocol services reflect typical group management operations: join and leave processes for peers as well as group view updates. The message transfer provides a dependable data transfer, which follows the described principle of virtual synchrony. Local (`LT`) and global (`GT`) clock times as well as necessary timers are hence maintained and used. The network failure detector monitors the reachability of participants through the introduced heartbeat mechanism. The network module provides a simple interface to the *uBeeMe* platform transport interface.

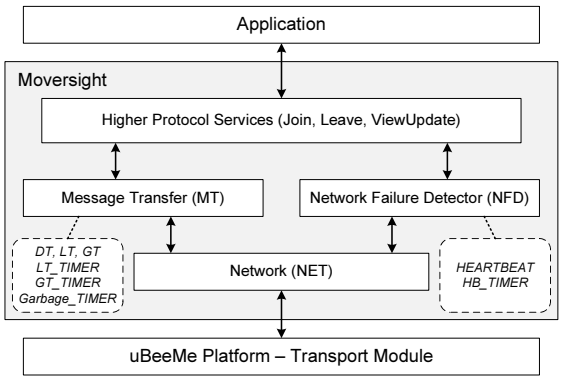

Fig. 2. Moversight Architecture Overview - Status Quo

III. Conclusion and Future Work

In its current state, *Moversight* is developed in the network simulator OMNeT++. This enables us to instantly test new approaches and alterations of strategies and concepts. Future tasks include the consideration of security functions for authentication of peers and the inclusion of a split and merge mechanism, to support separation and consolidation of groups in ad hoc scenarios. With this, *Moversight* will provide all necessary properties to support collaborative applications in the depicted ubiquitous P2P scenarios.

References

[1] “uBeeMe Project,” [online] http://www.ubeeme.org/.
[2] M. Kirsche, M. Dreissig, R. Kopsch, J. Gaebler, R. Klauck, M. Pink, F. Liu, and H. Koenig, “uBeeMe - A Platform Supporting Mobile Collaborative Applications,” *in Journal PIK (Practice of Information Processing and Communication)*, accepted for publication, in press.
[3] E. Kohler, M. Handley, and S. Floyd, “Datagram congestion control protocol (DCCP),” IETF, RFC 4340, März 2006.
[4] M. Zuehlke and H. Koenig, “GCP - A Group Communication Protocol for Supporting Closed Groups in the Internet,” in *Proc. of IFIP TC6 WG 6.7 7th International Conference (SMARTNET 2002)*, 2002, pp. 211–227.
[5] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, “Internet Group Management Protocol, Version 3,” IETF, RFC 3376, Oktober 2002.
[6] “The ISIS Project,” [online] http://www.cs.cornell.edu/Info/Projects/ISIS/.