Concepts for Service Provision in Disaster Environments†

Kamill Panitzek
Technische Universität Darmstadt,
panitzek@tk.informatik.tu-darmstadt.de

***Abstract*—Mobile devices like smartphones and tablets gain in power as well as in popularity each day and introduce new opportunities in life saving missions. When interconnecting mobile devices of first responders an ad-hoc peer-to-peer network can be created providing important services like voice communication, for instance. Realizing such a system in the field poses different challenges which are investigated in this work. Finally, a short insight into my current research is given.**

I. Introduction

My vision for the next generation of peer-to-per (P2P) systems is the distribution of applications and services. Over the past decade different P2P systems have been proposed to share and distribute data in a decentralized fashion. Examples are file sharing and VoIP communication. Discovering and distributing services and applications will enable a new family of P2P systems so-called P2P service overlays. One application for such overlays are first response missions where important services like an alert buoy, a voice chat service, or a map service can be provided.

Previously, Bradler et al. evaluated different P2P overlays to use in first response on top of a mobile ad-hoc network (MANET) [1]. In 2011 we proposed to use publicly available wireless access points (in shops, bars, etc.) to make such a network more resilient [2]. More than that, this work also was motivated by the idea of so-called cloudlets [3]. Cloudlets are nearby resource-rich computing nodes (e.g., in cafés) which could be harnessed by mobile devices over high-bandwidth, low-latency, one-hop wireless connections. P2P service overlays in combination with such an infrastructure would enable swapping of heavy tasks away from mobile nodes into more powerful cloudlets, thus improving not only battery life but also overall system performance.

Such P2P service overlays pose a number of interesting challenges. Services need to be discovered at changing locations, execution states need to be transferred consistently, and due to device heterogeneity services are not executable on every device. Also, robustness and load balancing, as well as other related problems, need to be addressed. Therefore, in this work I first define the main components of P2P service overlays (Sec. II). Afterwards I discuss requirements, challenges, and questions that arise in disaster environments and sketch preliminary solutions to tackle them (Sec. III).

†This work was funded by the German Research Foundation, Research Group 733, "QuaP2P: Quality Improvement of Peer-to-Peer Systems".

II. P2P Service Overlay

P2P service overlays are composed using three major building blocks. *Peers* are devices connected to each other, thus creating the P2P overlay. Peers are best described by their given software and hardware resources including, for example, operating system, runtime environment, CPU, memory, storage capacity, network connectivity, and also energy supply. Users interact with the entire P2P system over their respective peer including all user behavior such as real-world movement, service requests, and other interactions.

Mobile objects describe any object stored and distributed in the P2P system. These mobile objects are categorized into stateless data items or streams, e.g., as known from file sharing systems, and stateful executable items like services or applications. Services provide a basic functionality and can be composed to more powerful applications. P2P service overlays concentrate on the distribution, placement, and discovery of these stateful mobile objects.

Every peer contains a *runtime environment* to store and execute mobile objects. To protect the peer from malicious code the runtime environment should be isolated from peer's hardware and software. For instance, a virtual machine is a possible solution to realize such an environment.

III. Requirements and Preliminary Solutions

I identified basic functional and non-functional requirements of P2P service overlays as well as possible components and solutions covering these requirements. In the following I will focus on service distribution since stateless mobile objects are covered by most P2P systems today.

As mentioned before, peers must be able to distribute and discover mobile objects inside the system. A **service registry** provides this functionality and could be realized using a classical distributed hash table (DHT) approach like Chord [4], for instance. Furthermore, the service registry can be segmented into the problems of *service placement*, *matching*, and *service discovery*. In mobile environments services might be needed at changing locations because peers are mobile while interacting with a service. Especially in MANETs the physical distance between service and service consumer introduces a large number of hops and thus a high latency among other problems. In 2008, Wittenburg et al. published a survey about different solutions to the service placement problem in MANETs [5]. Their conclusion was that literature provides answers to the question of *how many*

978-1-4244-9528-3/12/$26.00 ©2012 IEEE

service instances *where* to place. But the question of *when* this should be done was not addressed in depth.

On the other hand, due to device heterogeneity peers' resources have to be *matched* to the resource requirements of mobile objects. Devices cannot host services requiring other resources than are available. To provide this functionality a **matching algorithm** is needed. By investigating the problem using flow graphs I am currently working on deriving a distributed matching algorithm.

In such a dynamic environment mobile objects must be discoverable. Ververidis et al. published a detailed survey about the *service discovery* in MANETs [6]. Incorporating the findings of Bradler et al. [1] I believe that a **superpeer overlay without indirection schemes** known from typical DHTs today would be a good service discovery solution for P2P service overlays. In most DHTs today, to access an object the object's location must be obtained from the DHT first. In contrast, such a mechanism would allow for direct invocation during the service look up process if the service is not yet running. Otherwise the already invoked service instance would be retrieved.

Availability of mobile objects is very important and can be compromised with high churn rates (peers connecting to and disconnecting from the system) as known from today's P2P systems. Adding peer mobility further increases this problem due to possible connection loss resulting from limited range of wireless hardware. Therefore, a **replication mechanism** is needed to automatically replicate popular services, thus increasing the availability of mobile objects. Recently a detailed survey on replication strategies was published by Amjad et al. [7]. In P2P service overlays the replicated objects also include execution states which have to be *synchronized* continuously. Additionally, a replicated service must take over if the primary service fails. This is also called *failover* [8].

Reaching high *robustness* to dynamic changes in the environment and to peer mobility is of great importance to P2P service overlays in general, but especially in disaster environments. Therefore, not only replication is important but also *service mobility*. Code and execution states must be transferred efficiently between peers. From a users' point of view this transfer has to be unnoticeable requiring *seamless service provision*. For instance, if users chat over a voice chat server the communication should not interrupt or stutter during the migration process of the voice chat server from one peer to another. This can be achieved by using **code migration** and **persistent socket connections**.

To further improve robustness of the system, peers must be prevented from overloading, so that hosted services (mobile objects) can be provided continuously. A **proactive load balancing mechanism** must identify possibly overloaded peers and replicate or migrate hosted objects onto other peers. More precisely, this must happen before peers overload. Thus **prediction mechanisms** of the future peer load and network state are needed. Therefore, the current state of the network and the peers must be *monitored*.

IV. Concluding Remarks

In the presented work I focused on the main building blocks for P2P service overlays. I gave an overview on the requirements and challenges of such a system and proposed solutions known from literature as well as own preliminary ideas to tackle them. As a feasibility study, we already implemented a Java-based framework called PeerMoS providing some of the features and components mentioned above. We used Chord as service registry and realized code migration and persistent socket connections using parts of the MundoCore framework [9] as well as own components. The idea is to have a platform where modules like the service registry, for instance, can be easily exchanged. Using this framework we created a voice chat service prototype that provides seamless voice transmission during the migration process. Currently we are porting PeerMoS to the Android platform to experiment with P2P service overlays in realistic mobile environments. I am also investigating approaches in the fields of matching, service placement, and service discovery. In parallel I am working on simulation models for first responders in disaster environments.

References

[1] D. Bradler, J. Kangasharju, and M. Mühlhäuser, "Evaluation of Peer-to-Peer Overlays for First Response," in *IEEE PerCom*, 2008.

[2] K. Panitzek, D. Bradler, I. Schweizer, and M. Mühlhäuser, "City Mesh - Resilient First Responder Communication," in *ISCRAM*, 2011.

[3] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The Case for VM-based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, 2009.

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," in *ACM SIGCOMM*, 2001.

[5] G. Wittenburg and J. Schiller, "A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks," *IEEE PerCom*, 2008.

[6] C. Ververidis and G. Polyzos, "Service discovery for mobile Ad Hoc networks: a survey of issues and techniques," *IEEE Communications Surveys & Tutorials*, 2008.

[7] T. Amjad, M. Sher, and A. Daud, "A survey of dynamic replication strategies for improving data availability in data grids," *Future Generation Computer Systems*, 2012.

[8] J. Balasubramanian, S. Tambe, C. Lu, A. Gokhale, C. Gill, and D. C. Schmidt, "Adaptive Failover for Real-Time Middleware with Passive Replication," in *IEEE RTAS*, 2009.

[9] E. Aitenbichler, J. Kangasharju, and M. Mühlhäuser, "MundoCore: A light-weight infrastructure for pervasive computing," *Pervasive and Mobile Computing*, 2007.