

FRAMEWORK FOR SERVICE PROVISIONING IN MOBILE AD HOC NETWORKS

Károly Farkas, Lukas Ruf, Martin May, Bernhard Plattner

Computer Engineering and Networks Laboratory (TIK)

Swiss Federal Institute of Technology (ETH)

CH-8092 Zurich, Switzerland

{farkas, ruf, may, plattner}@tik.ee.ethz.ch

Abstract

Mobile ad hoc networking, as a typical example of self-organized networks, is an emerging and promising communication paradigm. Not only the variety of devices but also the diversity of services is continuously increasing. Such services must be provisioned in a flexible and distributed way without central infrastructure. Thus, service deployment and management for such mobile devices are extremely difficult since a provisioning framework must cope with the high level of device heterogeneity, degree of mobility, and take limited device resources into account.

In this paper, we propose SIRAMON, a generic, decentralized service provisioning framework for self-organized networks. SIRAMON integrates the functions required to deal with the full life-cycle of services. SIRAMON offers sufficient capabilities to specify, lookup, deploy, instantiate and manage not only trivial but also complex services like mobile ad hoc group applications.

Keywords

Service Provisioning, Self-Organization, Mobile Ad Hoc Networks, SIRAMON

1. INTRODUCTION

Ad hoc networks have been receiving much attention recently due to their immense field of application. A mobile ad hoc network is built of a collection of diverse wireless nodes (users). These nodes form a multi-hop network communicating spontaneously without relying on any pre-existing infrastructure or central administration. In order to make an ad hoc network functional, the nodes must organize themselves. They not only provide terminal but also relaying functionality for distant nodes. In this mobile ad hoc environment, efficient service provisioning requires flexible and distributed mechanisms. While the different applications and the great number of mobile devices make ad hoc networking interesting, the lack of central infrastructure, the high level of device heterogeneity, the degree of mobility and the resource constraints of devices make it hard to provide ad hoc services.

So far, applications of Mobile Ad hoc NETWORKS (MANETs) [1] have been envisioned mainly in the field of emergency and military situations. However, MANETs offer many more possibilities. We assume that mobile ad hoc services will be introduced in the future into everyday life, and not just supporting work or daily activities but also giving pleasure in spare time¹. Hence, complex Mobile Ad hoc Group (MAG) Services² (such as distributed group games) will appear in MANETs soon.

While new applications in MANETs will provide powerful environments for group services, the complexity of service deployment and management in ad hoc networks call for the support of a service provisioning framework. Traditional techniques for service provisioning used in communication and data networks (Jini [2],

¹ This assumption is supported by the lately published report on mobile entertainment (<http://www.mgain.org/MGAIN-wp3-d311-revised-final.pdf>) and by the recently announced cooperation between SUN and NOKIA on new online multiplayer gaming solutions using Java technology (the latest game console of NOKIA, called N-Gage, is now furnished with a networking middleware and mobile Java technologies providing a Java-based, cross-platform approach to mobile multiplayer gaming)

² We use the term MAG or 'Mobile Ad hoc Group' services referring to complex, mobile, device independent, distributed, software-based services (e.g., to support collaboration or to provide service for mobile, online group games)

Bluetooth SDP [3], SLP [4], Chameleon [5], etc.) are not well suited for MANETs. First, they focus on a subset of service provisioning functionality (e.g., resource discovery) only. And second, they are often based on the client-server model using central infrastructure which is not available in mobile ad hoc networks.

The lack of a solution motivated us to propose a new service provisioning framework for self-organized environments, called SIRAMON (Service provisioning fRAMework for self-Organized Networks) which is appropriate also for mobile ad hoc networks. In this paper, we identify and discuss the necessary functions to provide mobile ad hoc services, and introduce the architecture of SIRAMON. SIRAMON is based on a modular and distributed design. Its components can be replaced according to current demands. The proposed modules are the following: (i) Service Specification; (ii) Service Indication; (iii) Service Deployment; (iv) Service Management; (v) Environment Observer. Service Specification contains the used Service Model which describes the role of the device in the service, the functions and connections of service elements to build the service. Service Indication is responsible for service advertisement if the node hosts a service, or service lookup if the node intends to use a service. By the Service Deployment module, creation, installation and configuration of services are carried out. The Service Management component controls the service maintenance, reconfiguration and termination functions. And the Environment Observer module deals with the monitoring of the node resources and the service context.

The rest of the paper is organized as follows. In Section 2, we identify and discuss the different phases of service provisioning. We present SIRAMON, our service provisioning framework in Section 3. Following this, we compare our approach with previous work in Section 4 and finally we conclude the paper by a brief summary in Section 5.

2. SERVICE PROVISIONING IN AD HOC NETWORKS

Service provisioning in mobile ad hoc networks is faced special difficulties due to the constraints of MANETs – such as lack of central infrastructure, high level of device heterogeneity, degree of mobility, limited device and network resources. In this section, we survey these difficulties and identify the different service provisioning phases using a sample mobile ad hoc application scenario.

2.1 Sample Application Scenario

We use an imaginable scenario for an online multiplayer game in a mobile ad hoc network as our sample application scenario. A tourist, travelling by train, wants to spend his travel time playing a multiplayer game. To do so, he has a mobile device (e.g., a laptop) with a game software installed on it. Some other passengers on the train are supposed to possess also mobile devices that are able to communicate among each other and, at least, a couple of them are willing to play. While only some devices are capable to actively participate in the game, all may act as a relay to forward data. Since they have come close together, a mobile ad hoc network has been setup spontaneously.

Upon establishment of the ad hoc network, the initiator device advertises the availability of a new application/service on the network. When each of the connected devices is informed they can decide whether to join the game or not. Thereafter, the service will be automatically deployed on the selected devices (and in the network) and the game can be started. Further passengers are allowed to join (if the game allows for late arrivals) and leave the ongoing group game at any time. Even the initiator must be able to leave without interrupting the game session for the remaining players.

2.2 Phases of Service Provisioning

Analyzing this sample application scenario we can identify the following phases of service provisioning.

In the first phase, the service to be provisioned has to be specified and named. We call this phase *Service Description*. The service specification must be able to describe the *role* of the node in the service, the *functions* of the service elements and the *connections* among them. For example, in our game service the specification has to describe the participating device's role (e.g., active player or auxiliary node which just aids the service to run), the device-level local view (the game is composed from moving pictures with music, sounds and voice requiring real-time interaction from the player, etc.), and the network-wide global view (the device has to communicate real time with all other player devices, etc.) of the game.

When the ad hoc network is established, the initiator device (the host) has to *advertise* the game service such that new participants can join. On the other hand, the devices of the other passengers must be able to *lookup* the service before deploying and using it. This phase is called *Service Indication*. Mobile networks have special constraints which must be addressed to develop appropriate service discovery mechanisms. Usually, MANETs cannot provide a permanent, central directory where the announced services can be registered and from where the available services can be read out. Moreover, the service hosting role can change dynamically in a mobile ad hoc group service. For example, in our scenario the initiator device hosted the game service but when it quits the game some other device has to take this role over or this role has to be implemented in a distributed manner.

When the devices in the ad hoc network are informed of the game service the next step is the *Service Deployment* phase when the game software has to be *requested, downloaded, installed, configured* and *activated*. Every device has to accomplish these procedures separately but in a synchronized manner to get a consistent state by when the game starts. This phase introduces some interesting problems. In general, heterogeneous devices form the mobile networks representing several different platforms. This requires the support of all the ported versions of the service software to these platforms from the service hosting device or the service has to be implemented in a platform independent way. Moreover, the lack of central infrastructure and the mobile behavior of the devices make it impossible to discover network resources beforehand and to use them permanently during the service session.

After the game started the service *maintenance, reconfiguration* and *termination* functions come to the front. We call this phase *Service Management*. In the course of service maintenance the dynamic adaptation of the service to the resource variations must be assured. The players may experience frequently degraded performance of the service due to dynamic changes in network conditions or other resources. Service reconfiguration takes place fundamentally in two forms, locally and globally. We are talking about the former if a service user device modifies the configuration of its local service instance (e.g., selecting a better resolution for the pictures, etc.). In the latter the network-wide global view of the service session changes, for example, when a new player joins the game session. Local reconfiguration is mostly transparent for the other users while global reconfiguration is always perceptible for every participant. Service termination can be considered as a special reconfiguration if a service device stops running the service instance, such as the game initiator when it quits the game. This incurs also the release of the reserved resources. Moreover, dynamic adaptation to the resource variations and reconfiguration of the service session demand the continuous monitoring of the resources and the service context.

3. SIRAMON, OUR GENERIC SERVICE PROVISIONING FRAMEWORK

Our service provisioning framework is based on a decentralized and modular design.

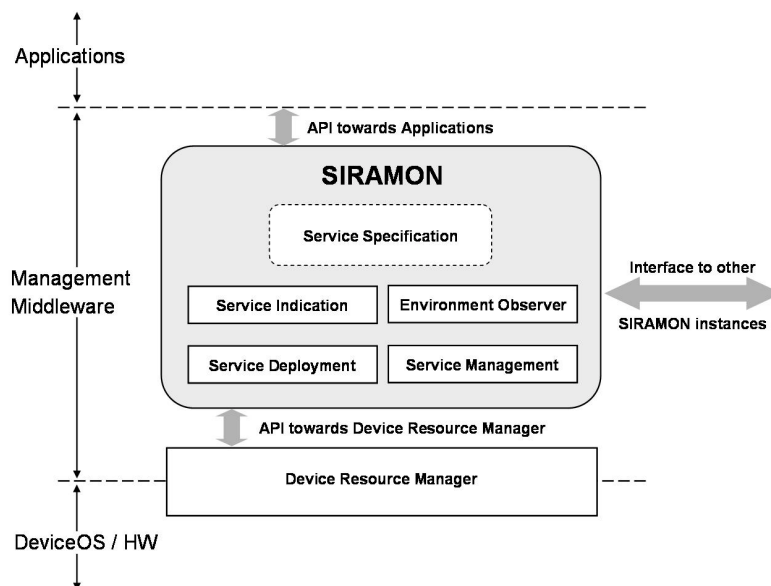


Fig. 1. Ad Hoc Device Model with SIRAMON

Every device runs an instance of SIRAMON which handles the control and synchronization among the devices, as well. We integrated the functions of the different service provisioning phases, as we discussed above, into separate modules. This makes the framework generic giving the possibility to replace the functions of a module with others which may fit better with the actual environment or the application's needs.

In our device model (see Fig. 1), we introduced a management middleware layer where SIRAMON integrating the service provisioning functions is located. This layer provides an interface between the device's resources and the applications. The Device Resource Manager is responsible for controlling these resources and mapping them to service elements. Note that in this paper, we do not cover this latter module. In the following, we describe the different modules of SIRAMON.

3.1 Service Specification

It defines the service model used in the given service. SIRAMON is not bounded to any specific service model. The only requirement is that the service model must be able to describe the role of the node in the service, the functions of the service elements and the connections among them. In case of different type of services, it is reasonable to use different service models if the other model suits better for describing the given service. However, the service model must be carefully selected for the given service because it constitutes the basis of service provisioning and it is used in the course of the whole provisioning activity.

3.2 Service Indication

This module covers service advertisement and service lookup procedures. As we discussed earlier, due to the lack of central infrastructure, for service advertisement and lookup the traditional central directory model cannot be applied in ad hoc environments. A solution could be to replicate this directory on every device connected to the ad hoc network. This introduces synchronization and propagation overhead to maintain a consistent database on all the devices but makes the service lookup very fast and efficient. Another solution is to establish a so-called virtual backbone in the ad hoc network selecting some devices which store a copy of the service directory. This approach slows down the service lookup and is less efficient compared to the previous one, but introduces less administrative overhead – however, it requires an extra procedure to select and maintain the set of virtual backbone nodes.

3.3 Service Deployment

Deploying a service requires requesting/downloading software according to the specification; discovering/gathering resources; mapping of this specification to resources; configuring the resources and installing/configuring the downloaded software; activating the service in a synchronized manner along with the other service participants; and handing the control on to the management module. The Service Deployment module includes these functions.

3.4 Service Management

This module integrates the service maintenance, reconfiguration and termination functions.

Service maintenance is responsible for the dynamic adaptation of the service to the resource variations in order to optimize user's perceived service quality. We distinguish three ways to adapt to changes in resources:

- *Service-intelligent adaptation*: The application is powerful enough to do its own adaptation to resource changes. For instance, the audio subservice of a teleconference service can combine multiple streams encoded for different bit rates into a single clip. Thus during runtime, the audio stream dynamically adapts to changes in bandwidths.
- *Service-specific adaptation*: The application provides mechanisms for dynamic adjustment, but does not do so automatically. For instance, for bandwidth adaptation, there are different instances of the same codec intended for different bit rates. To select the appropriate one the application needs to be notified about the resource changes.
- *Service-independent adaptation*: If the management module doesn't have information about the underlying implementation of the application, it treats the application as a black box and tries to find other means to adapt to the changes.

Service reconfiguration, as we discussed above, has two fundamental forms, local and global reconfiguration. The Service Management module gives support to global reconfiguration while local reconfiguration is in charge of the application.

Service termination can be considered as a special form of reconfiguration when a device stops running the service instance. All the other service participants have to be informed about the termination and the reserved resources have to be released.

3.5 Environment Observer

This module is responsible for monitoring the device resources and the service context. Resource and context information have to be gathered and transformed into an appropriate form, which can serve as input for the deployment and management modules.

4. RELATED WORK

Several proposals have been developed to standardize the functions of service advertisement and service/resource lookup (e.g., Sun's Jini [2], Bluetooth's Service Discovery Protocol (SDP) [3], Service Location Protocol (SLP) [4] of IETF). However, these proposals are mainly based on the use of a central directory. Moreover, almost all of them include the client-server paradigm and mainly focus on resources/services provided by devices (e.g., printers, fax-machines) which restricts their applicability in ad hoc networks.

In contrast, some recently appeared new proposals directly target infrastructure-less networks. The Konark system [6] is a middleware designed specifically for service discovery and delivery of device independent services in ad hoc networks. Another proposal is the distributed service discovery architecture [7] which relies on a virtual backbone for locating and registering available services within a dynamic network topology. However, none of these proposals provide the full functionality of service provisioning.

Several research projects have recently focused on service specification, dynamic service creation and deployment in the context of active networks. One example is Chameleon [5], a platform independent, node level service deployment framework which facilitates the installation of a service on a single node. However, Chameleon targets only infrastructure-aware networks and doesn't offer comprehensive service provisioning functionality besides deploying service components on single nodes.

5. SUMMARY

In this paper, we identified the different phases of service provisioning in ad hoc networks and discussed the related difficulties. Based on this overview, we introduced SIRAMON, a new, generic service provisioning framework for self-organized networks. SIRAMON, with its decentralized, modular design suits well for ad hoc environments and is flexible enough to cope even with complex services, such as mobile ad hoc group applications.

References

- [1] Perkins, C. E., *Ad Hoc Networking*, Addison Wesley, 2001.
- [2] SUN Microsystems, JINI Architecture Specification, November 1999.
- [3] Specification of the Bluetooth System: <http://www.bluetooth.com>, December 1999.
- [4] Guttman, E., Perkins, C., Veizades, J., Day, M., "Service Location Protocol, Version 2", *IETF RFC 2608*, June 1999.
- [5] Bossardt, M., Ruf, L., Stadler, R., Plattner, B., "A Service Deployment Architecture for Heterogeneous Active Network Nodes", *In IFIP International Conference on Intelligence in Networks (SmartNet)*, Kluwer Academic Publishers, Saariselka, Finland, April 2002.
- [6] Helal, S., Desai, N., Verma, V., Lee, C., "Konark A Service Discovery and Delivery Protocol for Ad-hoc Networks", *In Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC)*, New Orleans, USA, March 2003.
- [7] Kozat, U. C., Tassiulas, L., "Network Layer Support for Service Discovery in Mobile Ad Hoc Networks", *In Proceedings of the IEEE INFOCOM*, San Francisco, USA, April 2003.