Chapter 4

DESIGN CHALLENGES AND OVERVIEW OF PROPOSED FRAMEWORK

Introduction

Recent advancements in the computational capabilities of portable devices with their increased wireless connectivity have favoured the emergence of service provisioning. Service provisioning in mobile ad hoc networks faces special difficulties due to the constraints of the ad hoc environment—such as lack of central infrastructure, high level of device heterogeneity, degree of mobility, limited device and network resources. As the mobility of users may lead to different contexts, users can increasingly benefit from services whose results can be adapted to the changing context such as variations in users’ position, preferences and requirements and locally available resources (Bellavista et al (2003)). In these context-aware service provisioning scenarios, it is crucial to enable the dynamic retrieval of available services in the nearby of the user’s current location and environment, while minimizing user involvement in service selection, configuration and binding.

Two main features of the proposed service provision system are (1) the ability of the service entities to tolerate the disconnections of services and (2) the ability to handle changes. This can be achieved by making the different entities of the architecture less dependent by loosening their strong ties with other entities. Being less dependent means that each service entity is more autonomous, this in turn means less needed interactions. This is essential in mobile environments. Additionally loosening the ties means more tolerance to change and failure. The ability to handle change can help improve performance of the service provision. This notion of loose coupling is currently used in Service oriented paradigm. It has effectively influenced the
late-binding of consumers to provide. In order to design a viable service provision system, loose coupling is highly desirable on many challenging points. The other aspect is selecting the best matching service provider node which meets the user’s need by taking the current context into consideration. Providing services even in the condition of mobility is also the major issue to be considered. In 4.1 elements in service provisioning are discussed. Section 4.2 focuses on the requirement to be satisfied by a service provisioning framework for mobile ad hoc networks. Section 4.3 focuses on the design challenges to design a service provisioning framework. Finally, section 4.4 provides an overview of the proposed framework followed by the chapter summary.

### 4.1 Elements in Service Provisioning

A framework for service provisioning should have the following salient elements, with each element fulfilling a very specific role in the framework.

- The *service description* element is responsible for describing a service. The service to be provisioned has to be specified and named. This phase is named as service specification phase. The services have to be specified with the service description, preferences.
- The *service advertisement* element is responsible for advertising a given service description on a directory service or directly to other hosts in the mobile ad hoc network.
- The *service discovery* element is the keystone of the framework and carries out three main functions. It formulates a request, which is a description of the needs of a user. This request is formatted similar to the service description. The element also provides a matching function that selects the best matching service provider for the request. Finally, it provides a mechanism for the user to communicate with the service provider.
- The *service invocation* element is responsible for facilitating the use of a service. Its functions include transmitting commands from the user to the service provider and receiving the results. A good invocation mechanism abstracts the communication details from the user and, in case of network failure, redirects requests to another provider or gracefully terminates.
- The *service maintenance* element is responsible for maintaining the service throughout its entire time. Its functions include upgrading the functionality offered to the clients,
upgrading the quality of assistance already offered. In the course of service maintenance the dynamic adaptation of the service to the resource variations must be assured. The performance degradation of the service due to dynamic change in network condition or other resources has to be considered. A good service maintenance mechanism performs its tasks with no or little interaction with the client.

- **Service termination** is a special kind of adaptation needed when the participating node moves. The node on the move can be either the consumer or the provider. Because of the mobility the status about the network has to be change. Dynamic adaptation to the resource variation and adaptation requires continuous monitoring of resources and the service context.

Since all these elements are to be implemented in case of every applications, it is a reasonable design choice to gather and implement them at one place in the form of framework, so that the application developers do not have to deal with these common functions rather can focus on only application specific issues.

### 4.2 Requirements to be Fulfilled by the Service Provisioning Framework for MANET

There is a broad spectrum of mobile applications which require feasible environment for deployment. A framework that supports for the development of applications for the mobile ad hoc network is a novel approach that will offer information access and sharing by considering the challenges and constraints of the mobile ad hoc networks. Though the framework for mobile ad hoc networks is tightly coupled with the applications one point framework may not be designed. Utmost care has been taken to develop a framework to adhere to the design constraint of the framework.

From the point of view, mobile ad hoc networks are created by interconnection of individual nodes and are formed independently i.e. not based on any kind of external infrastructure. This implies that the service discovery process must follow network formation
stage, service and resource discovery, which can take place by advertising the service available in them.

In order to propose a solution for service discovery protocol for mobile ad hoc networks, the requirements of such a network should first be analysed and classified as a realisable requirements. The requirements are classified into functional, technical and performance requirements.

**Functional requirements**

Functional requirements identify what functionalities must be provided by discovery protocol. Following are the functionalities required.

*Decentralised directory:* the service directory must be accessed by the node always regard less of the nodes mobility. The discovery should not depend on any particular node.

*Support for service mobility:* the service discovery process must support service mobility. Due to mobility the services can be handed over from one server to another. Service discovery must identify equivalent/adaptable services to support the aforementioned handover.

Making the type of handover depends on the nature of the service used. Sometimes a service may be offered to the user on an ‘online basis’ i.e. the service provision point must move when the user and terminal move. For example if a user watches a real-time video, the stream must also be provided at the new location. Sometimes, the nature of a service does not allow handover e.g. an uncompleted printer job. In this case, the service cannot be handed over to another printer, but the original printer must either complete the job in unconnected mode or terminate the printing process.

*Support for service provisioning:* the service discovery process must provide proper interfaces to other service elements using the discovered service specifications. The elements include service provision, service selection, service control, service user notification, service adaptation etc.

Service discovery is the initial phase of service provisioning. It must support following phases that are expected to occur after discovery. Support must be offered providing proper
interfaces to the other service elements such as: service provision (how to benefit from the service), service selection (what is the most appropriate service), service control (how to control service provisioning), service user notifications (how to notify user about the status of the service) and service adaptation (how the service is to be adapted to the user context).

_Support for multicast_: the transport layer that the CASP resides on must support multicasting along with unicast and broadcast, because this is the only efficient way to address multiple devices with a service request.

_Technical requirements_

Technical requirements describe what kind of support should be provided to meet the functional requirements

_Browsing capability_: the service discovery protocol must provide the ability to browse the available services within the network. The entire local i.e. one hop and to a minimum of 3 hop neighbour nodes’ services can be discovered & listed and the size of the list has to be small. The user needs to benefit from all the services available. The browsing enables the user to be aware of all the available services in the network. The user must be able to browse at least the one hop neighbour nodes’ services.

_Proper specification of services_: the service discovery protocol must specify the services in a proper manner. The specification of the service must be machine-readable. The network entities must be able to recognise the service and be able to select it and use it without any user interaction, because user likes to automatically use the most appropriate service for his/her requirements.

_Proper service search mechanism_: a service discovery protocol must provide a proper search capability. Sometimes users search for a specific service, or a particular service supporting specific attributes. The user must be able to search for their required services. The service search must support for various search operations (i) exact or approximate search i.e. finding an available type of service, (ii) attribute search i.e. finding a service which contains a particular attribute, or (iii) attribute value search i.e. finding a service which contains a specific attribute with a particular value or range of values.
Proper mechanism to handle wireless network dynamics: A service discovery protocol must be able to handle the issues of wireless network’s dynamics and failures. The specific problems to wireless networks like link failure; node failure etc. in such situations, the service discovery protocol must not become trapped in a deadlock situation.

Performance requirements

The performance requirements set the targets for success and cost of a solution.

Reasonable discovery time: the discovery time in mobile ad hoc networks must be acceptable and of the same order as the network formation time.

Reasonable network load: the service discovery process must use the network capacity efficiently and not slow down normal network traffic. The incurred overhead should be of the order of other discovery processes in the network, such as route discovery, address assignment.

Proper usage of computing resources: the computing resources in mobile ad hoc network nodes’ are very limited. The Service provisioning framework must be designed to be as light as possible, using minimal amount of CPU time, memory, battery lifetime. These limitations dictate simplicity of the code and applications that can be executed on such devices. Some service discovery protocols are lightweight but do not provide support for provision of service and deal only with discovery while others deliver both discovery and provision with a higher expense.

Reasonable adaptation delay: because of the dynamic changes during formation of the network, the protocol must adapt itself to these changes. Many factors affect this delay, such as mobility of nodes, changes in the status of nodes e.g. on, off, sleep, change in availability or properties of the service i.e. while change is not reflected on the registry and change in availability of the network. Some of the protocols mentioned earlier are based on service registration with servers, which is time consuming process and the frequency of it adapts to the changes of the network. Another factor affecting the delays is the content of the transferred data. Using protocols based on the transfer of data instead of text based code can reduce the delay.

4.3 Design Challenges and Proposed Solutions
Due to the dynamic nature of the mobile Ad hoc networks, it exhibits frequent and unpredicted topology changes. Mobile ad hoc networks have to be able to adapt to the traffic and propagation conditions to the nodes mobility pattern. As mobility affect the neighbour nodes and the providers of the services may join or leave the network. The protocol should monitor and select a service based on the mobility pattern, battery life, distance between the provider and the consumer. This mobility and topology change can be addressed by the following mechanisms.

*Decentralized directory service.*

The service information in the network has to known by the participant of the network. In order to store them one cannot rely on central server because of the liveliness problem of either the server or the service provider which is depicted in Figure 4.1. This situation may lead to the problem of inconsistency because of the nodes mobility. So decentralized directory service is needed.
In order to alleviate this centralised directory, every node in the network has to maintain the available services which are reachable by it. The proposed structure of service directory in the networks is shown in Figure 4.2. In Chapter 5 service discovery issues with this proposed architecture are discussed.

**Figure 4.2 Decentralised Peer-Peer Directory Structure**

**Context awareness**

The wide spectrum of applications demonstrates that mobile ad hoc networks have some distinct advantages over wired networks, mainly due to their fault-tolerant and self-organizing characteristics. At the same time, mobile ad hoc networks present a number of complexities and design constraints that are not existent in wired networks. The most important factor
characterising a mobile ad hoc network is the high variability of the network state. The network state is affected by link connectivity, power control and mobility effect (Bisnik (2005)).

Link connectivity: In wired environment, the link connectivity is a binary decision, i.e. link exists between two nodes when they are connected by a physical medium like optical fibre or coaxial cable. In a mobile ad hoc network, the broadcast nature of the communication allows each node to be connected with multiple receiver nodes.

Power control: The broadcast nature of the wireless communication determines that each node may increase/reduce the number of neighbouring nodes by tuning its transmitting power. Thus, the topology of the network as perceived by each node is strongly dependent by the transmit power of each node.

Mobility effect: The nodes belonging to a mobile ad hoc network are free to move and organize themselves arbitrarily. The mobility effect affects the performance of the network protocols. At routing layer the mobility factor governs the performance of the routing protocols.

Meeting the requirements of the application despite variable link connectivity, network topology and power levels imply two issues in protocol design:

- Information sharing: each layer of the protocol stack should be able to access the information about the current network state.
- Protocol cooperation: performance gains may be obtained if joint solutions at multiple network layers are considered.

Unfortunately, the layered open system architecture (OSI) does not seem to support these requirements. The layered architecture is remarkably successful for networks which are made up of wired links, where the key assumptions and abstraction boundaries work well. The main drawback of the layered model is the lack of cooperation among non adjacent layers: each layer work in isolation with a very few information about the network. This cooperation among the layers can be achieved by cross layering. By using cross layering the network state or otherwise known as environment context can be accessed.

Context means every aspect that can impact the behaviour of an application. Therefore the framework should be context aware. The awareness can be broadly categorized as device
context and environment awareness. The environment context can be accessed using cross layering. Through cross layering information from one layer can be passed on to any layer in the stack. For example information in the MAC layer may be needed by PHY, Network, Transport and application. To have cross layering in effect adaptors are used. Adaptors are methods to access information from that layer. They are the interfaces between cross layer and layer N. These are set of functions which enable the cross layer to get and put information into the corresponding layer. For each context prediction predictors are used. These are methods used to process the information accessed from adaptors and decide the values to be taken by the parameters and put them through adaptors.

![Diagram of interactions between various layers](image)

*Figure 4.3 Interactions among Various layers for Context Accessing using Cross Layer*

Figure 4.3 shows the interactions between the units and layers used for cross layering. These predictors are the sources of network context to the framework. The predictor gets information from the lower layer PHY can identify the signal strength of the incoming signal. The use of signal strength is motivated by the communication hardware. To identify or predict the nodes movement signal strength can be used. A predictor for node movement has been designed based on this. Whenever the node movement is identified, it will be informed to the event handler which will raise the ‘node_movement’ event. It also notifies the happening of this event to all the event listeners. Any number of predictors can be placed for each of the event that needs to be monitored.
Though the location of the device can be had by using GPS systems, the usage of this cannot be taken as assumption because many a time the GPS devices will not be available. So rely on the lower layer information is the technique used in this thesis. The speed of the moving device, the direction of the device is available to determine the distance between the nodes (Lyes Dekar and Hamamache Kheddouci (2008)). Any number of predictors can be used on the available information by using the adaptors. The general architecture to be used while designing the cross layering is available in the work of Licia Carpa (2001). The framework should take advantage of this awareness with minimum effort and network overhead. Chapter 8 elaborates on this topic. Description on this has been presented in the paper of thesis author’s (Ponmozhi (2012)). The second challenge to be considered is Ability to tackle changes.

*Late binding:*

By hiding the implementation technology, a change in the implementation of a service provider will not require any change at the consumer side. But if the service contract changes that will affect the consuming client. The service provision system has to manage these changes to decrease the loss of service from a client’s point of view. Therefore the service-oriented paradigm enforces late-binding between consumers and providers in order to free their coupling until the last moment before invocation. Yet during the invocation, the service contract binds the consumer to the provider. As the devices move frequently the unavailability of the service may be a situation when the protocol should identify alternative services. The late binding to the service providers and switching to new providers based on the context is enabled by policy management. Rebinding to other provider when the already connected Service Provider cannot be accessed is shown in figure 4.4.

Policy-based management has been considered by two working groups namely IETF and Resource Allocation Protocol Working Group. The RAPWG has described a framework for policy-based admission control specifying the two main architectural elements:

- The Policy Enforcement Point (PEP) represents the component that always runs on the policy-aware node and it is the point where the policy decisions are actually enforced.

- The Policy Decision Point (PDP) is the point where the policy decisions are made.
Policy Framework WG describes four major functional elements namely:

- **A Policy Management Tool**, to enable an entity to define, update and optionally monitor the deployment of Policy Rules.
- **A Policy Consumer**, which is a convenient grouping of functions, responsible for acquiring, deploying and optionally translating Policy Rules into a form useable for Policy Targets.
- **A Policy Target**, which is an element whose behaviour is dictated by Policy Rules carrying out the action indicated by the Policy Rule.

This thesis applies policy management to implement the concept of rebinding. Challenges to be solved for the implementation are:

- What are the system artefacts that need to be monitored to enable realistic policies and reconfiguration rules? How fine grained the monitoring should be? Is this monitoring a periodic or continuous one?
• What are the system artefacts that can be adapted by reconfiguring rules and polices? How fine-grained this adaptation should be?
• How to define domain-specific policy languages that are easy to use and express concerns by a wide range of users? To what extend can the system’s behaviour be programmed as policies?

Effective use of the distributed systems requires an approach that enables them to be managed and configures easily and uniformly through, a policy based approach Nelson (Matthys et al (2008)). An application may, specify a policy that express that the most efficient service in the environment must be used on what conditions.

Such policy-based management approach consists of several steps (i) Gathering about the execution context (ii) Interpretation of policy files matching their conditions and triggering any related events (iii) The enforcement of their associated actions on the current execution context.

*Interoperability*

Service providers and clients should always be able to understand each other regardless of their implementation technology. The abstraction of the implementation can only be achieved using universal standards and specifications. A service provider implemented in Java should seamlessly interact with clients of other languages. This implementation abstraction is well done in service provision system like web services using XML as an interaction language and specifically using specifications like WSDL and SOAP for service description and invocation. The service platform proposed in the following chapters also uses WSDL and SOAP in the discovery and invocation interactions in a simple form.

### 4.4 Proposed Service Provisioning Framework Overview

Service provisioning in mobile ad hoc networks is challenging due to the characteristics of the network. In fixed networks service provisioning can be done based on the centralized entity called directory to store the information related to the services provider by different providers. And the applications developed need not worry about the bandwidth usage as they are
resource rich environments. The device capabilities of the mobile ad hoc networks vary and most of time limited.

This thesis defines Context Aware Service Provisioning (CASP) framework to provide basic modules needed for service provisioning in mobile ad hoc networks. This framework consists of service advertising and matching required services based on functional and non functional properties and maintenance of the binding between the service provider and the service requester. Service advertising and its related issues are discussed in chapter 5. Chapter 6 is dedicated to deal with the issues related to non functional properties management in CASP. Binding issues and maintenance phase is dealt in chapter 7.

As service discovery is application layer task, the framework need to be nearer to the application. The proposed framework sits between the application layer and the network layer. Figure 4.5 shows the placement of CASP framework. In order to cope with the changes in the underlying network it interacts with the network layer and other lower layers for information as cross layer interaction.

![Figure 4.5 Placement of CASP Framework in the Protocol Stack](image)

An overlay on top of the network layer has been created, (i.e.) at the application level and disseminate service advertisements, requests and replies. The placement of service discovery above the routing layer provides the advantage of (i) a modular and layered approach and one
can replace the protocols at any layer without affecting other layers. (ii) No assumption about any specific routing protocol or the underlying network is necessary. It is also possible to create pervasive service discovery across many domains.

In order to separate the concerns of communications and those of service interactions, separate modules for communication with other nodes have been designed. Other modules can consume the functions provided by this module. There are two primary agents service provider agent, user agents which uses the communication module to interact with each other.

Four main components of the frameworks were identified, each responsible for a particular aspect. Figure 4.6 shows the various components of CASP. The resource manager module is responsible for keeping track of installed components, implementing the core methods of managing the component framework in a device-specific language (resource manager). To hide the physical distribution of the environment to other levels and application a module for distribution has been added (advertisement and query manager), thirdly, for the adaptation process, environment monitoring framework have been defined, which collects information about the environment, provides a set of API to set monitoring triggers for single or multiple conditions, the monitoring framework may gather this information by using probes on resource availability. A policy parser and policy enforcer will interpret and enforce the policies. Policies are seen as a way to guide the behaviour of a network or distributed system through high-level, declarative directives. The proposed framework uses XML to represent Policies.
This proposed framework splits the service provisioning into four components, which can be tuned by every node according to their capabilities and policies. The framework will also break service discovery into programmable components and allow tuning depending upon application needs and the device capabilities. The various tunable components and facilitators in CASP are given in Figure 4.7. The modules are grouped into two the (i) self management facilitators are components which access the context and provides for decision making, policy management is the on which takes necessary adaptation based on the context value. (ii) Tuneable components they will working according to the context accessed. These are the target components which will be changed by the policy management modules.

The basic components are (i) facilitators which includes modules for Context-awareness and Policy management (ii) Tuneable components which includes Service advertisements (Query Vs Announcement), Service selection (based on user preferences and service provider’s capability) and Service maintenance (based on binding, reselection and rediscovery policies)
Given the above approach, a high level of autonomy can be introduced into the nodes so that they can automatically cope with the increased levels of heterogeneity and volatility, which present in a mobile ad hoc network environment.

Summary

Challenging points that must be addressed by the proposed framework were presented in this chapter. Proposed framework splits into four modules to take care of the different phases of service provisioning. It has been decided to use (i) peer-peer caching instead of central entity for directory service (ii) cross layer to access dynamic context (iii) policy management for adaptation. Supporting tunable and self-managements components were identified to increase user’s satisfaction by giving due considerations to the users centric data. Implementation of these modules aims to respect the design principle of independence and loose coupling. These design principles have been followed throughout the design process of communication, discovery & invocation and maintenance phases. From next chapter onwards the modules and their architecture in CASP will be elaborated.