Communication Context for Adaptive Mobile Applications

Arjan Peddemors Henk Eertink
Telematica Instituut, The Netherlands
{Arjan.Peddemors, Henk.Eertink}@telin.nl

Ignas Niemegeers
EWI Faculty, TU Delft, The Netherlands
I.Niemegeers@ewi.tudelft.nl

Abstract

This paper describes our ongoing work on a system that provides information about network resources and related communication facilities to applications running on a mobile device. It defines and elaborates on the concept of communication context and describes the architecture of this core low-level enabling system for ubiquitous and pervasive applications operating in a mobile environment.

1. Introduction

For many entities described in pervasive computing scenarios, the availability of communication resources is an important aspect of their overall mode of operation. Especially for applications on mobile devices, typically being in a continuously changing network environment, the ability to adapt to changing network resources is key to providing a pervasive service to their users.

The variations in available network resources have a number of causes. First, wireless networks have more dynamic characteristics than their fixed counterparts. Second, mobile hosts may dynamically and concurrently connect to multiple access networks, as they become increasingly more equipped with multiple network interfaces. And third, these multiple access networks are based on a wide variety of different network technologies, each with their own network characteristics. The variety in network technologies is likely to increase, because new technologies become available on a regular basis; currently for instance IEEE 802.15.4 (ZigBee), Ultra Wideband (UWB), and IEEE 802.16 (WiMAX).

In this paper, we elaborate on the concept of *communication context* and explore the kind of middleware functionality needed to provide communication context to adaptive pervasive applications running on a mobile host. We introduce a system level architecture, called CoSphere, responsible for generating communication context information and allowing applications to influence this context. The main CoSphere contribution is the definition of a coherent information model, in the form of communication context, that captures the elements used by various kinds of mobile applications for their decisions on network resource usage in heterogeneous mobile environments. Furthermore, the architecture brings together, in our opinion, a number of existing system elements in a novel fashion. We regard the system presented here as a core low-level service for pervasive and ubiquitous computing applications that operate in a mobile environment. CoSphere builds on the mobility management and application awareness work presented in [5]. We plan to provide a full CoSphere reference implementation in the future.

Many existing approaches formulate means to deal with context information in general terms. These approaches look at such aspects as collection, distribution, transformation and inference of generic contextual information, which then may be applied to more specific forms of context information. Here, we limit our scope to communication context information, and focus on how to allow applications running on mobile devices to be aware of their network and related resources so that they can adapt and provide a pervasive service. We feel that this limited domain provides, for real-world systems, many challenges to overcome. Also, by learning from the solutions and mechanisms found for a specific domain, we may be able to better understand and recognize the principles valid for generalized forms of context handling. We focus on the small and powerful mobile device as prime target computing platform because we think it will be pivotal for many real-world applications and it is truly pervasive on a global scale.

This paper is organized as follows. Section 2 introduces the concept of communication context and section 3 describes the CoSphere architecture. Section 4 discusses application interaction, followed by section 5 on related work. We wrap up with a summary and future work in section 6.

2. Communication Context

For applications to adapt, they need to have information about the current state and characteristics of the available networks on a mobile host. This information can be referred to as *network context* information. The network context can, for example, be used to select the path that outgoing packets must take for a connection with another host; the application may prefer a high rate access network path to a low rate one, when both are available at the same time. An application that communicates with other end-points over IP normally uses a transport protocol such as UDP or TCP to send and

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

receive data. In a mobile situation, where networks get activated and deactivated in a dynamic fashion, these transport layer connections must be either 1) under control of a mobility management facility such as Mobile IP, or 2) must be replaced by a transport layer protocol that can deal with changes in attachment to the network, or 3) must be handled by the application itself in case of interruption. Even in those cases where the mobile host takes care of connection management at the system level in the event of handover, the application may need to adapt to the capabilities of the physical network actually used to transfer the packets for the connection. In other words, the state of other communication facilities such as mobility management protocols is important to adaptive mobile applications. The state of the network context and the mobility management protocols can be captured with a stack view, where cross-layer information is flowing towards the application.

The mobile host may provide several *system level facilities* to support the communication tasks of its applications. The availability and state of these facilities are relevant to the adaptation of the applications. An example of such a facility is the entity that decides on the activation and configuration of access networks, through the network interfaces of the device. The state of activation as well as the intentions for future activation provides applications with useful hints on current and future network resources.

An application that needs to communicate with other end-points may need various *network infrastructure facilities*, such as Domain Name System (DNS), proxy services or service discovery services. For instance, in a multi-homed situation, the mobile device may have multiple entry points to the DNS, and applications may want to decide or influence, as with path selection, which entry is used to execute DNS lookups.

We call the information describing the actual state and characteristics of all technological facilities relevant to a mobile application that needs to initiate and sustain communication with one or more other entities the application's *communication context*. Summarizing the discussion above, communication context consist of three different elements: *network context* in the form of cross-layer information, the *state of the (operating) system facilities* relevant to the communication with other end-points, and the *state of facilities available in the network infrastructure* that support this communication.

Even in the case a mobile device has one or more network interfaces and access networks can be made available through these interfaces in a certain surrounding, the device does not necessarily have to attach to these networks. In fact, in many mobile scenarios where power conservation is taken into account, the access network activation should be precisely tuned to the needs of the applications, because most network access, even in an idle state, consumes a certain amount of power. Therefore, the applications should be able to indicate their interest in the discovery and activation of access networks or the mobile host operating system should be able to estimate the needs of the current applications. So, the applications on the mobile host directly or indirectly influence their own network and communication context. In other words, an interesting relation exists between the network and communication context provided to applications and their own influence on this information.

3. CoSphere[1] Architecture

One big challenge is how communication context can be expressed and supplied to the applications so that they can make balanced decisions about their usage of network resources, facilities for mobility management, and other communication facilities. It is clear that the cross-layer information must provide an abstraction of the entities at the link layer, network layer, and transport layer; if an application were to receive all link layer details (e.g. technology specific signal and noise parameters for indications of network quality) it would be flooded by information and, more seriously, would have to have knowledge of all aspects at all layers. The hiding of details is the great benefit of the layered protocol stack and is at the heart of the success of the Internet [3]. An application using a transport layer protocol only knows of the end-to-end pipe abstraction offered by this layer. However, in order to choose suitable network resources, the application must know about the available paths (network layer abstraction) and characteristics of links (link layer abstraction). Another challenge is that every type of application requires a different level of abstraction from the communication context. A third challenge is how to let applications influence their own communication context.

The CoSphere architecture proposed here deals with these challenges. It comprises a wide variety of information sources that together supply the primitives for the communication context. The central component in the architecture, the communication context provider, collects the data from these sources and takes care of aggregation and inference to a level of abstraction and detail that matches the needs of the communication context consumers: the applications (see also figure 1). The architecture discerns three different types of context information sources: the active system entity, a specialized form of an active system entity in the form of the mobility management protocol handler, and the active

[1] CoSphere is short for *Communication Sphere*, and expresses those network and communication facilities that are in the direct surroundings (sphere) of a user's mobile device.

state entity. The components and their responsibilities are described below.

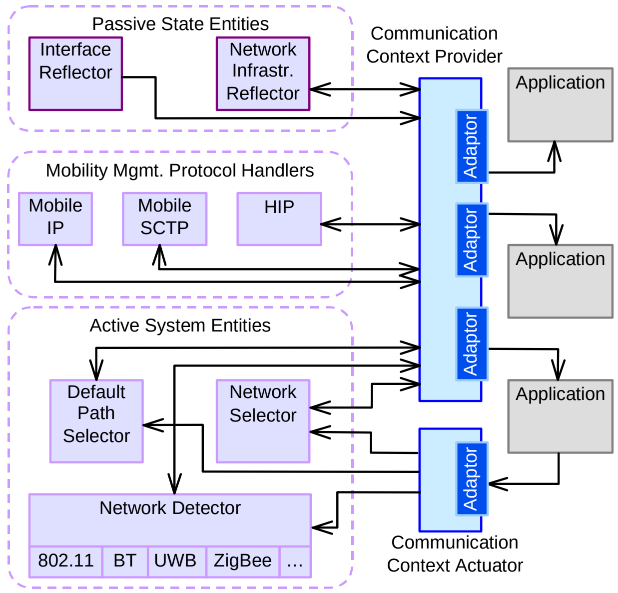

Figure 1: The CoSphere architecture

The *communication context provider* defines a number of different levels of abstraction and detail, each targeted at a specific type of application. When an application connects to the communication context provider, it chooses a type that best matches its own type. The context provider then deploys an adapter, capable of aggregating and inferring the raw data from the context sources, that is configured for this specific application type. Naturally, this means that analysis must 1) show a credible subdivision of mobile applications in specific types, and 2) reveal in plausible terms the level of abstraction and detail required for each type of application.

The context sources are in most cases also consumers of the communication context information. For instance, the network selector acts upon the events generated by the network detector about the availability or unavailability of networks. The context provider may offer specific adapters for these consumers (not depicted).

The *active system entities* influence the communication context with their actions. The network detector is capable of sensing the availability of networks (infrastructure or peer to peer) through each of the network interfaces available on the host. It knows how to deal with different kinds of network technologies. The network selector determines which networks will be made active through the available network interfaces; in many situations multiple networks are available, e.g. through an 802.11 interface, and only one can be connected to. The default path selector determines the default routing behavior of the host. The *mobility management protocol handlers* are in fact a special form of active entities. They support mobility management at different layers; the applications can choose between them for the management of their connections and sessions. Obviously, a mobility management protocol is only useful in situations where the other party can be reached through different network paths. As an example, figure 1 depicts Mobile IP (MIP), Mobile SCTP, and the Host Identity Protocol (HIP). The *passive state entities* only provide information on the current communication context. They do not directly influence or change this context themselves. The interface reflector indicates the current state of the network interfaces on the host (activation state, IP parameters). The network infrastructure reflector provides details on the availability of such facilities as DNS, proxies, and service discovery. The context actuator is the entity through which applications can actually influence their communication context. Again, this entity provides an adapter that matches the type of application; the means to influence the context is at the right semantic level for the application. This is useful for particular applications that really want to control or exploit network resources.

Many examples of CoSphere usage can be given. One straightforward scenario, but realistic in terms of usage of currently available technology, is an application that shows bursty network traffic depending on the availability of high-rate low-cost network resources. This could for instance be an application that caches large video news clips (always providing the latest news on the user's smart phone in video format) or intelligently downloads large email attachments. These larger chunks of data are retrieved at a moment with plenty and cheap network resources available, e.g. when the user is at home and his smart phone connects to the user's home WLAN network with ADSL uplink. Due to power-management issues the 802.11 network interface is not powered-on by default. The application, however, is interested in occasional high-speed connectivity and therefore requests the Communication Context Actuator to check for new networks with modest intensity. It registers at the Communication Context Provider for events that signal the availability of WLAN connectivity, and then sets-up its own download connection. When done obtaining the latest data feed, the application may signal to the Communication Context Actuator that it is no longer interested (at least not for the coming hour or so) in WLAN discovery and connectivity.

Another example is the Body Area Network (BAN) and personal sensor network gateway. The application, running on a small personal device, collects sensory information via the BAN and occasionally uploads large amounts of data for medical diagnosis when plenty of network capacity is available. Vital but low-rate data is continuously sent for monitoring (using Mobile SCTP).

4. Application Interaction

CoSphere should work in close cooperation with currently used mechanisms for the realization of connectivity with other end points, such as the Sockets API. Also, the mechanism for route selection (network path selection), and the application's influence on this process, must take into account the dynamic network configuration on the mobile host. A facility to easily use the available network and communication resources is therefore important for the development of mobile applications. Another interesting aspect is the capability of adding new entities to the system, in the form of support for new network technologies and mobility management facilities.

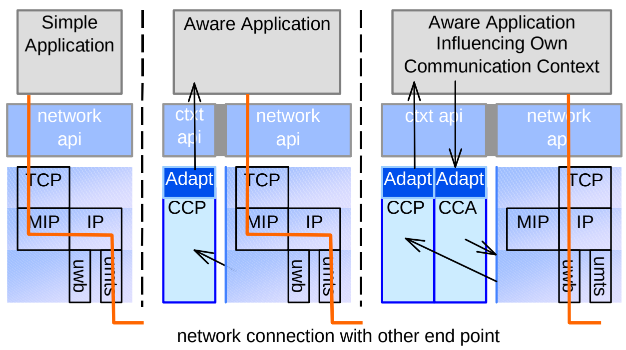

Figure 2: Different types of mobile applications using network resources; depending on their needs, they use the communication context provider (CCP) and actuator (CCA) through the communication context API, which may be coupled to the network API (e.g. the Sockets API).

Some typical configurations are depicted in figure 2. It shows three high-level distinct applications types and the way they maintain a network connection with another end point. The simple application does not use any information describing its communication context. It uses the traditional facilities for setting up a connection with another host. The CoSphere system service is, in this case, configured in such a way that by default all outgoing traffic is handled by Mobile IP. This allows the mobile host to maintain the TCP connection set up by the application also in the event of changing access networks. The simple application does not notice this functionality.

The aware application obtains communication context information through the context API offered by the Communication Context Provider. The information provided through this API is adapted to the level of abstraction needed by the application, and reflects, amongst others, the cross-layer information describing the current network context. For instance, the application receives information about the mobility management facilities available at the different layers and may get an indication of the link quality of the available networks. In this case, the application initiates a connection with another host and indicates that Mobile IP must handle it in order to support roaming in a dynamic mobile network environment. This interaction is executed using the network API in close cooperation with the context API.

The third application is also aware of its current communication context, but additionally influences its own context in an active manner. It activates, for example, through the context API and the Communication Context Actuator a network on the UWB interface. Subsequently, it sets up a TCP connection to another end-point, without using mobility management facilities available on the mobile host. The application receives updates on the quality of the UWB network and when this quality drops below a certain level, it will take action to reestablish the connection over another network interface.

At this point in time we have not made a decision on which entities have final control over network activation, routing behavior, etc. Resource contention most likely must be addressed, although on hosts with few applications this may be less of a problem.

The experience with the implementation of a mechanism allowing application on a mobile host to be aware of the mobility process, as presented in [5], shows that many aspects come together in a complex and integrated fashion. This most certainly will also be the case for the implementation of the CoSphere system.

5. Related Work

The work related to the topic of this paper covers a number of different areas: models for cross-layer information interchange, abstractions for network context, mechanisms for application and network adaptation, and mobility management in heterogeneous network environments. This section describes a non-exhaustive selection of related work coming from these areas.

The MobileMan project [2] defines a cross-layer reference architecture for the exchange of data between layers in a conventional protocol stack (network, transport, application, etc.). It uses a 'Network Status' vertical element to keep each layer informed about the state of other layers, with the explicit goal to support the dynamics in mobile ad-hoc networks and provide performance optimizations. Our work incorporates cross-layer information exchange with particular focus on the application layer for application adaptation, and does look at cross-layer performance enhancements. Our definition of communication context, however, comprises aspects that cannot be elegantly captured by a pure cross-layer descriptive model, such as the incorporation of the status of certain system facilities.

The work presented in [6] provides a formal definition of network context for ad-hoc mobile nodes. It primarily looks into the specification of the ad-hoc network topology within the vicinity of a mobile host and defines the cost for paths taken through this topology, but does not elaborate, as we do, on other aspects such as mobility management facilities.

Many approaches exist that focus on ways mobile applications can adapt to changing resources. Although the work presented here proposes a means to supply information on communication resources so that applications may be adaptive and does not specify a mechanism for application layer adaptation itself, the actual strategies for adaptation or mechanisms forcing the application in a certain form of adaptation are important because they provide hints on what kind of information is relevant. The Odyssey platform for adaptive mobile data access models the adjustment of applications around the high level concepts of agility and fidelity. The adaptation functionality is divided between the application and the operating system. Experience with this system shows that adaptation must balance between agility and stability [4]; rapid changes in network resources must, for a more stable user perception, not always result in swift adaptation.

The AwareCon architecture described in [1] seeks to incorporate context information in the network itself, and supports adaptation at various layers in the network stack. The context information is distributed as a part of the payload of the communication packets set between the nodes that make up a network. It primarily focuses on low-power nodes and applications mostly operating in an ad-hoc fashion. Our focus is more on the adaptation of applications given a certain communication context, assuming standard network hardware and network stacks.

In our definition of communication context, a mobile host may provide multiple facilities for mobility management to handle the dynamically changing point(s) of attachment to the Internet. The work described in [7] introduces a mechanism that adapts the kind of mobility management depending on the behavior of the application, i.e. the kind of connection initiated by the application. The application itself is not adaptive.

6. Summary and Future Work

In this paper, we presented the concept of communication context, as a special form of the overall context of an application, and introduced the CoSphere architecture that deals with providing this communication context to applications running on a mobile host. We regard the possibility of a mobile application to receive, by using the CoSphere system, in a comprehensive and structured form all relevant information on the facilities available for its communication as the main contribution of this work. We feel that studying this particular form of context awareness provides valuable insight that helps with formulating systems that handle generalized forms of context.

However, there are still many challenges to overcome. We plan to work on the way communication context is expressed and want to analyze the information needs of different kinds of mobile applications in pervasive computing settings. Additionally, we want to define metrics that can measure the appropriateness of the abstraction offered to these types of applications, and metrics that measure the level of convenience of CoSphere usage. We plan to validate the CoSphere system by implementing and using representative mobile applications and deploy them in a non-trivial communication context, i.e. use with multiple types of network technologies, multiple facilities for mobility management, etc. Furthermore, we want to explore the possibilities for distributing the communication context information in a personal area network scope in such a way that the entities within this short-range network environment can share each other's (external) network resources.

7. Acknowledgement

This research has been supported by the Dutch Freeband Communication Research Programme (AWARENESS project) under contract BSIK 03025.

8. References

[1] M. Beigl, A. Krohn, T. Zimmer, C. Decker, and P. Robinson, "AwareCon: Situation Aware Context Communication", In *Proceedings of the 5th International Conference on Ubiquitous Computing (UbiComp 2003)*, Oct. 2003

[2] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-Layering in Mobile Ad Hoc Network Design", In *IEEE Computer, Special Issue on Ad Hoc Networks*, Feb. 2004

[3] V. Kawadia and P. Kumar, "A Cautionary Perspective on Cross Layer Design", In press for *IEEE Wireless Communication*, retrieved on 14/12/2004 from http://black.csl.uiuc.edu/~kawadia/papers/cross-layer-design.pdf

[4] B. Noble and M. Satyanarayanan, "Experience with adaptive mobile applications in Odyssey", In *Mobile Networks and Applications 4*, pp. 245-254, 1999

[5] A. Peddemors, H. Zandbelt, and M. Bargh, "A Mechanism for Host Mobility Management supporting Application Awareness", In *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys'04),* June 2004

[6] G.-C. Roman, C. Julien, and Q. Huang, "Network Abstractions for Context-Aware Mobile Computing", In *Proceedings of the 24th International Conference on Software Engineering*, pp. 363-373, May 2002

[7] X. Zhao, C. Castelluccia, and M. Baker, "Flexible Network Support for Mobility", In *Proceedings of the Fourth Annual International Conference on Mobile Computing and Networking (MobiCom'98),* Oct. 1998