SFB 637 Autonomous Logistics

Timm-Giel, A.; Görg, C.: Self-configuring communication service module supporting autonomous control of logistic goods. In: 3rd International Workshop on Management of Ubiquitous Communication Systems (MUCS) 2006. Cork, Irland, 2006, pp. 2-9

Self-Configuring Communication Service Module Supporting Autonomous Control of Logistic Goods

Andreas Timm-Giel and Carmelita Görg

University of Bremen, tzi – ikom, Otto-Hahn-Allee NW1, 28359 Bremen, Germany
{timm-giel, goerg@tzi.de}
http://www.tzi.de/ikom, http://www.comnets.uni-bremen.de

Abstract. This paper presents a concept of self-configuring mobile communication devices, such as sensor cluster heads, car mounted communication gateways etc. supporting the autonomous control of logistic goods. The main idea is to select and configure the communication network, service and protocol in a self-organised way at the mobile device, rather than initiating the networks, protocol and service selection by the network. In order not to force application developer to configure the complex communication devices and protocols, a communication service module (CSM) with a simple and well defined interface has been developed selecting and configuring the communication network and protocol. With this approach the application specifies the requirements by selecting a communication profile only. The Communication Service Module selects and configures the most suitable communication network technology and protocol. This enables easy improvement of the communication services and protocols without having to change the applications or user interfaces. The paper describes the concept and the first implementation on different computing platforms in detail.

1 Introduction

In logistic systems and processes complexity has dramatically increased over the last years. For example in transport logistics one sees an increased granularity of goods partly caused by Internet sales, a strong competition between carriers, new Value Chains and virtual companies, increased traffic congestions on the roads, and environmental concerns.

On the other side technology has become more mature and cost expectations are promising: RFID, power saving communication technology, sensor networks, powerful, small and cheap embedded computers are key technologies having the potential to support and enable autonomous control of logistic processes, which is anticipated to be one feasible way to substantially improve the performance and in particular the stability of logistic processes.

The evaluation of this assumption is the main topic of the Collaborative Research Center (CRC) “Autonomous Cooperating Logistics Processes – A Paradigm Shift and its Limitations” funded by the German Research Foundation at the University of Bremen [1].

Timm-Giel, A.; Görg, C.: Self-configuring communication service module supporting autonomous control of logistic goods. In: 3rd International Workshop on Management of Ubiquitous Communication Systems (MUCS) 2006. Cork, Irland, 2006, pp. 2-9

The CRC researches different aspects of autonomous cooperating logistic processes, from basic definition of autonomous control, impact of autonomous control on management structures within the enterprises, to risk management, decentralized autonomous routing mechanisms for logistic entities, sensor networks and required communications.

The Communication Network Group of tzi University of Bremen focuses on routing mechanisms and the communication enabling autonomous control, the latter being topic of this paper.

For autonomous control devices need to be able to communicate with the devices and sensors in the vicinity and with the outside world, e.g. to inquire on traffic conditions. Devices considered for autonomous control are transport vehicles (e.g. lorries), but also the transport goods (individual or palettes) itself. Both travel through different, heterogeneous communication networks on their way from source to destination.

Different scenarios for communication capabilities of the different devices exist. For limitations in costs and power the transport goods usually have limited communication and processing capabilities. This is only different for high value goods. Transport goods might have a simple IEEE802.15.4 enabled sensor node or a future RFID tag storing an agent code or a reference to the agent ID attached. The agent is then executed in the next agent environment, e.g. the container or lorry, and represents the transport good and takes decisions on the routing for it.

In case of the IEEE802.15.4 enabled devices the computing and communication platform on the transport vehicle or container acts as a Communication Gateway for the communication to the outside world, actually same as for the agents on the agent platform. The agents are representing goods having no or very limited communication and processing capabilities.

The Communication Gateway needs to be able to connect to the Internet and other logistic devices with any available technology and protocol. Typical network and protocols considered are GSM/GPRS, UMTS, WLAN, Bluetooth and IEEE802.15.4. Besides of infrastructure modes also infrastructure-less, direct or ad-hoc communication needs to be possible and supported by the communication gateways. Additionally the gateways should support a seamless roaming across the heterogeneous networks. Solutions and concepts to support these applications and the seamless roaming exist, Mobile IP may be mentioned as one example [2,3,4].

The configuration of networking protocols, service and systems is complex and the application developer is typically no communication network expert. For this reason and in order to realize a high flexibility also allowing for an easy extension to new and different communication the concept of the Communication Service Module has been developed in the framework of the wearit@work project, an EC funded IP in the IST 6 Framework Programme on wearable computing [5]. This concept and implementation is generic and has also been applied for the logistic applications of the CRC.

Instead of configuring all details of the network connection the application or application developer respectively should only specify the QoS, cost, and security requirements he/she has at a certain moment by the selection of a predefined profile. Based on the selected profile(s) the most suitable networking connection is automatically selected and configured.

Timm-Giel, A.; Görg, C.: Self-configuring communication service module supporting autonomous control of logistic goods. In: 3rd International Workshop on Management of Ubiquitous Communication Systems (MUCS) 2006. Cork, Irland, 2006, pp. 2-9

The concept of the Communication Service Module has been implemented for different mobile and wearable computing platforms and operating systems, such as WinXP, WinCE, and Linux.

The detailed concept of the Communication Service Module (CSM) is described in the following section. The third section details the implementation of the Communication Service Module for different computing platforms with different Operating Systems. The fourth section gives a conclusion and an outlook.

2 CSM Concept

As mentioned above and also presented for the application of wearable computing [6] the key point of the concept of the communication service module is that the user or his application only selects a profile detailing his QoS, security and monetary cost requirements.

The CSM on the Communication Gateway operates autonomously and selects based on the requirements (profiles) and the available services the most suitable network and service itself. If the network availability changes over time the selection process based on the specified profiles is repeated and as long as one service and network is available meeting the minimum requirements, the application is not involved, although the Communication Gateway might seamlessly roam from one network technology to another.

The functions, the user or his application calls are simplified to:

- start_communication (process_ID, profileID)
- update_communicatoin(sessionID, profileID)
- stop_communication (sessionID)
- get_network_status(sessionID)

If the application for example only needs a slow Internet connectivity to send from time to time positioning information, it can select a suitable profile which stays connected with low minimum required bandwidth and minimal costs and power consumption. Therefore it connects using the “start_communication” command with the respective profile ID. In order to allow for several applications (or agents) accessing the Internet, the Communication Service Module assigns a process ID of the application.

All requested profiles, possibly of different applications (agents), are aggregated by the CSM and the latter then selects the most suitable network(s) and service(s) to provide the required QoS under the given cost and security constraints. The CSM can also log all relevant communication details for each agent allowing for charging in case goods of different owners are transported. Generally several profiles can be selected by different applications and the requested connectivity can be provided by one or several network connections. The CSM therefore needs to provide means of con-

Timm-Giel, A.; Görg, C.: Self-configuring communication service module supporting autonomous control of logistic goods. In: 3rd International Workshop on Management of Ubiquitous Communication Systems (MUCS) 2006. Cork, Irland, 2006, pp. 2-9

figuring the network drivers, provides routing capability and also needs to be able to configure communication protocols, like Mobile IP [4] or Ad hoc protocols [7, 8].

The CSM finally answers the start_communication request with an acknowledgement, which includes a session ID for further reference to this communication session and an error code. The error code indicates if the start_communication request with the selected profile was successful or not.

As soon as the agent e.g. needs to access several databases, e.g. for calculating a new route, he can update the profile of the session, for example changing to a profile with high data rate and a higher price. This operation can also be done by the application directly if it is adapted for the use with the CSM. Important is that the Communication Gateway keeps its IP address even when changing its network connection, e.g. by the use of MobileIP, and so can continue the application with improved QoS.

When no data needs to be sent, the application can terminate the session using the "stop_communication" command with the session ID as parameter.

Regularly the application or the application can check the network status using the get_network_status command. This can be used by the user or application to adapt the application or behaviour to the actual network status.

As illustrated by the example the application only needs to start the communication with a certain profile, to update the profile, whenever his requirements change, and to stop the profile if there is no need to communicate any more. Additionally it can poll the network status.

The Fig. 1 below illustrates the protocol between user and CSM.

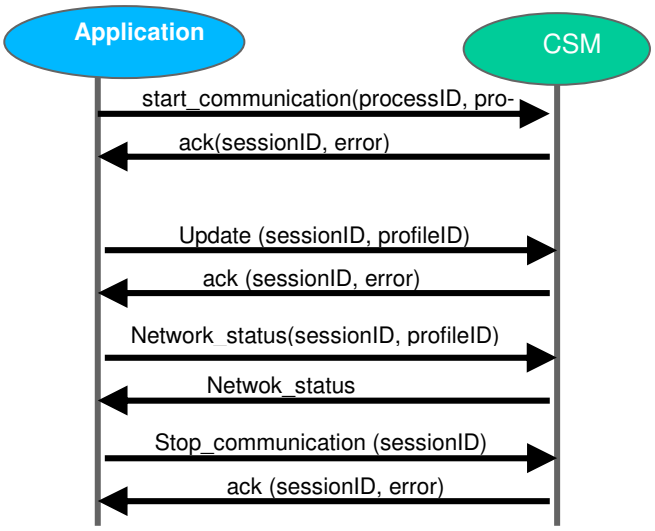

Fig. 1: Protocol between user and CSM

2.1 Profiles

The profiles comprise information on the required QoS, monetary cost, power consumption, security constraints, required connectivity and a possible abort reason. For all parameters a mandatory value and a weight are given. The mandatory value needs to be met by a suitable service. If several services are suitable the selection is done by calculating the most suitable using the weight factor to each parameter.

Following Parameters have been identified as part of the profile:

QoS Parameters
- Bandwidth
- Delay
- Error Rate
- Signal Quality

Costs
- Cost Limit
- Cost per Byte
- Cost per second
- Cost per Session

Power Consumption
- Power Consumption Class

Timm-Giel, A.; Görg, C.: Self-configuring communication service module supporting autonomous control of logistic goods. In: 3rd International Workshop on Management of Ubiquitous Communication Systems (MUCS) 2006. Cork, Irland, 2006, pp. 2-9

Security
  Security Level
Connectivity
  Direct Communication
  Internet Access
  Seamless Internet Access
Abort Reason
  Cost Limit
  Duration

The QoS parameters can be different for up- and downlink.

Due to the complexity of the topic the power consumption is simplified to power consumption classes.

Different security levels are defined, such as “open”, “encrypted on radio interface”, “cellular”, “Secure IP”.

For the connectivity it is differed between a direct communication only in a local ad hoc network, a simple Internet access and an Internet access with seamless roaming.

Finally the connection can be aborted automatically after reaching a certain duration or cost limit.

2.2 Architecture

The architecture of the Communication Service Module needs to be very modular allowing on one side to support different computing hardware platforms and on the other side to allow support of new communication technologies like IEEE 802.16 (WiMax) [9] or Ultra Wide Band (UWB) without major changes in the architecture. Ideally only a new network interface with new capabilities is added, as well as some scripts for the configuration of the new network interface.

One key advantage of the concept of the CSM is that neither the user interface nor the application selecting the profile directly needs to be changed, when a new networking technology or a new communication protocol, e.g. an extended ad hoc protocol is added to the CSM. The mechanism of selecting the profile, when starting a communication connection remains unchanged, same as the specific profiles, which do not need to be changed.

The basic architecture is in following Fig. 2:

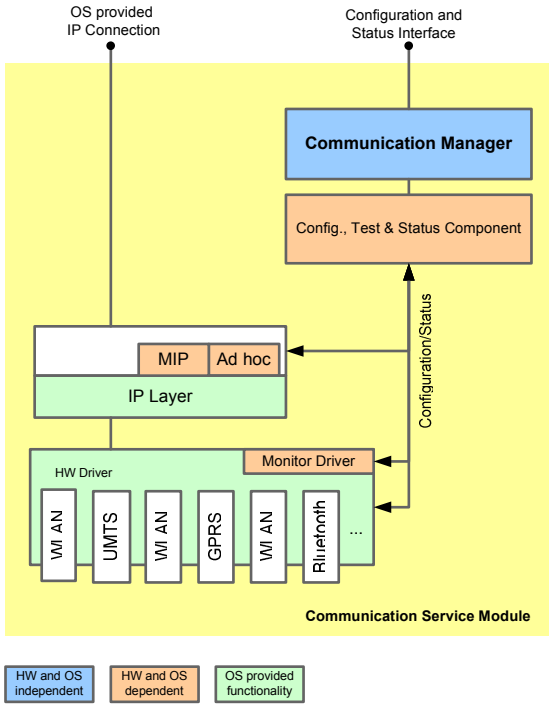

Fig. 2 Architecture Communication Service Module

It is important to note, that as far as possible hardware and operating system (OS) independent functions are separated from hardware and OS dependent functions to ease the deployment on the different mentioned platforms and to allow easier integration of new networking technology.

3 CSM Implementation

The current implementation of CSM is tested on 3 different operating systems of WinXP, Linux and embedded Linux. The testing has been done using a Laptops, different PDAs and in the framework of the wearit@work project different wearable devices, such as QBIC [10]. The implementation adopts two different architectures for Windows and Linux.

The process sockets based architecture is implemented on Linux. Here the CSM is a memory resident component (daemon) that comes up when the operating system is booted. All applications that need services of the CSM must first statically link with an API that provides all functionality of the CSM. This API opens a process socket to the daemon through which it communicates to avail the services of the CSM. The Daemon maintains the current state of CSM.

On Windows a DLL (Dynamic Link Library) based architecture has been implemented. In this case the CSM is a DLL that is loaded when the applications call CSM related functionality. The state is maintained in a common data area accessible to the DLL.

As explained in Fig. 2, the communication manager consists of independent functionality that is used by both of the above mentioned architectures. A key functionality in the communication manager is the Select_Network function. Select_Network is performing the network selection based on the selected profiles. For the first implementation, only the required bandwidth to select the best (Best Fit method) network attachment was used.

The CSM uses 3 informational entities. They are the Profiles, as explained earlier keeping the requirements for session, the Network Capabilities keeping the capabilities of the different networks and services and, and a list of the Active Sessions keeping a relationship between application, session, network, and profile. Profiles and Network Capabilities are persistent tables while the list of Active Sessions is a dynamic list.

During first tests, WLAN, UMTS and GPRS were used as networking capabilities, while selecting them based on requirements of profiles as shown in Fig. 3. Fig. 4 shows the GUI of the CSM, when running on a Xybernaut.

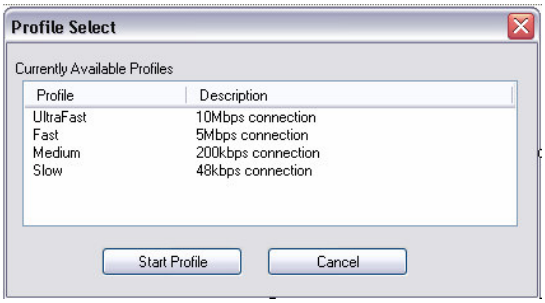

Fig. 3 Profiles of CSM used in the testing with WLAN, GPRS & UMTS

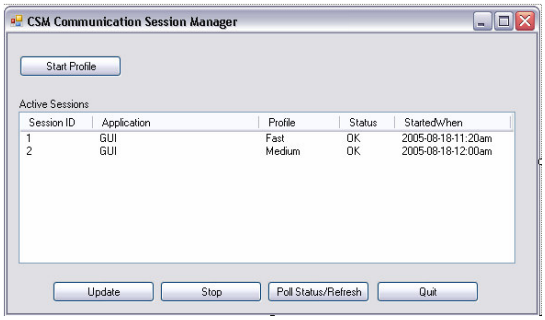

Fig. 4 CSM GUI on Xybernaut

4 Conclusion and Outlook

In this paper a decentralized approach for selecting the most suitable network, service and protocol has been introduced and the first implementation described. With this approach the network and service is selected from an isolated user point of view. Congestion in the network can only be reduced by the fact, that a user with a low performance in one network might change to another network.

One future improvement is to utilize information from the network resource management in addition. This means if the network management is aware of a congestion situation, it may "ask" the mobile devices for example to switch to a different networking technology or protocol.

Generally the approach of having a modular Communication Service Module at the user equipment is applicable to many different application areas. In this paper the example is given of a Communication Gateway in a container or transport vehicle utilizing the Communication Service Module in order to seamlessly roam across different networks, technologies, and protocols and providing most adequate services to all transport goods connected.

The technology is also being applied for the mobile worker in firefighting, aircraft maintenance, car production and hospital environment in the framework of the wearit@work project.

5 References

[1] Freitag, M.; Herzog, O.; Scholz-Reiter, B.: Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen. In: Industrie Management, 20(2004)1, GITO, Berlin, 2004, S. 23-2

[2] C.E. Perkins, IP Mobility Support for IPv4, RFC 3344. Aug 2002

[3] K. Kuladinithi, A. Könsgen, S. Aust, N. Fikouras, C. Görg, and I. Fikouras, "Mobility Management for an Integrated Network Platform", in Conf. 2002 Mobile and Wireless Communications Networks Conference (MWCN), Sweden

[4] A. Timm-Giel, Amadou, S. Aust, C. Görg, L. Ehrichs, M. Kus and M. B. Wischnewski, "UMTS Application Trials: Teleambulance in the IST project xMOTION", 2003 IST Mobile Summit, Aveiro Portugal, June 2003

[5] www.wearitatwork.com

[6] Timm-Giel, A; Kuladinithi, K; Görg, C.: Self-configuring Communication Service Module for Wearable Computers, WWRF15 Meeting, Paris, December 2005, pp. 1-7.

[7] ETF-Working-Group, MANET: Mobile Ad hoc NETworks, www.ietf.org/html.charters/manet-charter.html

[8] K. Kuladinithi, A. Udugama, C. Görg, On demand self organising ad hoc networks - Implementation architectures, 12th WWRF meeting 2004, Toronto, Canada, Nov 2004

[9] http://ieee802.org/16/

[10] O. Amft, M. Lauffer, S. Ossevoort, F. Macaluso, P. Lukowicz and G. Tröster,"Design of the QBIC Wearable Computing Platform" in IEEE Application-Specific Systems, Architectures and Processors, 15th IEEE International Conference on (ASAP'04) September 27 - 29, 2004 , Galveston, Texas, pp. 398-410.