Active Networks for 4G Mobile Communication: Motivation, Architecture, and Application Scenarios

Christian Prehofer, Qing Wei

DoCoMo Communications Laboratoires Europe,
Landsberger Str. 308 – 312, 80687 Munich, Germany
{prehofer, wei}@docomolab-euro.com

Abstract. In this paper, we examine the application of active networking technology to future mobile networks. We first introduce an architecture for programmable 4th generation (4G) mobile networking, including all system layers on the network and terminal side. Based on this architecture, we discuss programmability in future mobile networks. We investigate the main driving forces and obstacles for the application of active networks. In particular, we show that flexible component installation and cross-layer interfaces are a main motivation for programmable mobile networks. This is illustrated by a number of applications for future mobile networks, including context-aware mobility management and paging, where flexibility is a key requirement for future mobile services.

1 Introduction

In this paper, we present the driving forces for applying active network technology to future mobile networks. Active networks introduce flexibility into network elements by adding programmable execution environments for deploying new software. For the successful adaptation of active networks in future mobile networks, we think that there has to be a clear common vision. For this purpose, we first introduce an architecture for programmable 4th generation (4G) mobile networking, addressing networking on both terminals and network elements. Based on this, we discuss requirements regarding flexibility and programmability for 4G.

In our view, a main impediment for the usage of active networks is that active networks are, to some degree, a disruptive technology [7]. Active networking forms a significant change in the software architecture of commercial network elements. In particular, active networks are likely to be more expensive in the beginning. This is mainly due to the overhead of execution environment regarding performance, memory usage and code deployment infrastructure. We hence examine the main driving forces and obstacles for the application of active networks. In particular, we show that flexible component installation and cross-layer interfaces are a main motivation for programmable networks. In order to enable this programmability, active networking has to support reconfiguration and changing interfaces in different layers.

In the following, we first introduce active networking and examine different classes of programmability. Then we present in Section 2 an architecture for mobile

J. Sterbenz et al. (Eds.): IWAN 2002, LNCS 2546, pp. 132-145, 2002.

networks with the focus on programmable technologies. This shows that programmability is a pervasive concept in future mobile networks. In Section 3 we discuss the driving forces for programmability, with the special focus on mobile networks. This is illustrated in Section 4 by a number of application areas for future mobile networks, including context-aware mobility management.

1.1 Active Networking and Programmability

Active networking is a promising technology, in which the network nodes are programmed to perform customized operations on the packets that pass through them. A programmable node will enable the fast deployment of new services by using open interfaces to the network resources. With this technology, new protocols can be installed on the network elements which use the lower layer resources for creating new services. Active network technology typically provides an execution environment which is operating system and hardware independent. There are several approaches on active networks, which include AMNet [22], ANTs [1], ANN [23], CANEs [24], Mobiware [4]. They focus on different architectural domains (for instance quality of service (QoS) control, signalling, management, transport, and applications). Some approaches aim for high performance needed for flexible, per-packet processing, while others only aim at flexibility in the control path.

Another, related technology for programmability is software defined radio in reconfigurable terminals [3] or base stations. This technology will add flexibility to the lower layers, including RF, base band and networking protocols. For instance, terminals will not only have platforms for installing new applications, but also the radio hardware and the stacks will be programmable.

To classify these technologies, we consider the following different classes of programmability:

1. Parameterization of software or hardware modules is the classical way to introduce flexibility without the need for software updates.
2. Complete SW update or exchange. This is typically a firmware update of a device, e.g., a mobile phone or hardware module software update. This kind of update often requires that the device has to be disabled for some time period, which is clearly not desirable.
3. Partial SW update of a component or a software module in a complex software system. In this case, there are no open interfaces and the correct functioning of the complete software has to be reconsidered. The update may also lead to service interruptions.
4. Installation of a software module in an open, preferably virtual, execution environment. In this case, the environment and the interface design should ensure proper functionality, if the component behaves properly. In this case, a software update should not lead to service disruptions.
5. In the capsule approach (e.g., [1]), the program code is contained in packets. This approach is possible in some active networks environments where the packets include the code needed to handle these packets.

For active networks, we consider typically the classes 4-5 above, while software radio [3] also employs update of classes 1-3 for internal hardware modules. Currently, network equipment and mobile terminals are mostly capable of class 1 and 2 updates, and may in future be capable of some class 3 or 4 updates.

In the following section, we present an abstract model for mobile network elements and devices. While this model presents a very homogenous view of the different layers, the platforms on these layers may use different classes of programmability.

2 4G Programmable Mobile Network Architecture

In this section, we introduce a high-level architecture for future mobile systems with the focus on programmability. To address the creation and provisioning of unanticipated services, the whole system has to be designed to be as flexible as possible. Openness and configurability, not only at the service level but also with respect to the whole system architecture, will invite third party vendors to evolve the system as it unfolds and is therefore the key to viable solutions.

There exist many visions on fourth generation mobile networks (4G) [1] and the WWRF [6] is currently working on a combined vision and research agenda. 4G systems are expected to provide an integration platform to seamlessly combine heterogeneous environments, e.g., see [2]. Core abstraction layers cover hardware platform, network platform, middleware platform, and applications

The abstraction layers can interact with each other using well-defined interfaces. Besides their regular cooperation in an operational setting, each layer can be configured separately and independently via configuration interfaces.

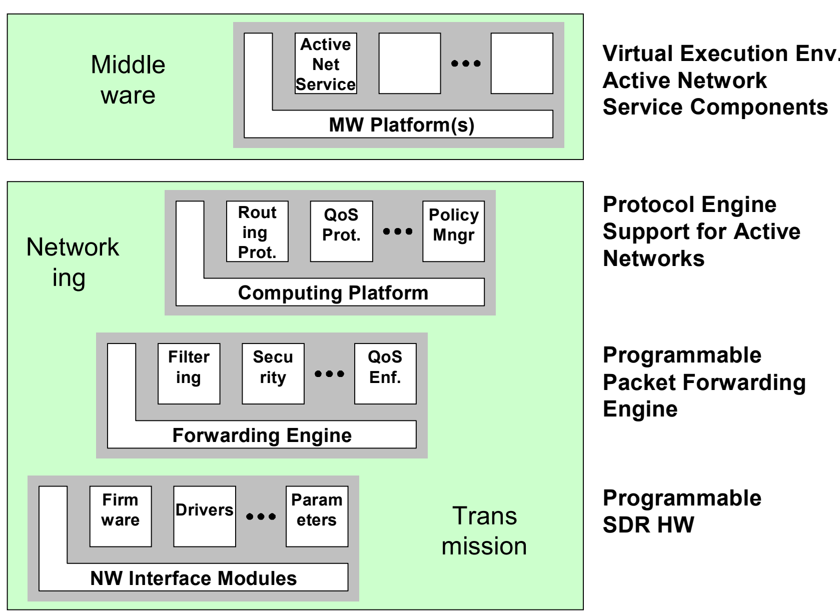

Fig. 1. Programmable Network Element Model

2.1 Network Element Architecture

Our generic architecture for the network elements of a mobile network is shown in **Fig. 1** (excluding applications). In this architecture, we consider the following abstraction layers, each programmable with configurable components:

- Middleware platform, typically with a virtual execution environment for platform independent distributed processing.
- The computing platform serves as a general purpose platform for processing stateful protocols, e.g., routing, QoS signaling or connection management.
- The forwarding engine is in the data path of a network node and it connects the interface modules, e.g., by a switch matrix. This engine can be implemented as dedicated hardware or as a kernel module of common operating systems. The forwarding engine is programmable for performance-critical tasks which are performed on a per-packet basis.
- The interface modules are medium specific for different wireless or wired standards. They can be configured or programmed to adapt to new physical layer protocols or for triggering events in higher layers.

It is instructive to discuss the different approaches to active networking in this architecture. Some approaches offer a virtual execution environment, e.g., a Java virtual machine, on the middleware layer. Some approaches also include native code in the computing platform, e.g., for flexible signaling. Others employ programmable hardware for forwarding [22]. A key ingredient in these approaches is interfaces to the lower layers and programmable filters for identifying the packets to be processed in the active networking environment.

2.2 Mobile Terminal Architecture

In the following, we discuss an architecture for mobile terminals. Although active networking traditionally refers to network elements and less on terminals, we think that the terminals have to be included as well for end-to-end services. Furthermore, with upcoming ad-hoc or multi-hop networks, the (conceptual) distinction between network elements is blurred. The main components of the terminal architecture shown in **Fig. 2** are

- Middleware platform, typically with a limited virtual execution environment.
- Smart Card, e.g., USIM for UMTS, which includes subscriber identities and also a small, but highly secure execution environment. This can be used ideally for personal services like electronic wallet.
- Programmable hardware, which is designed for one or more radio standards.
- Native operating system which provides real time support, needed for stacks and certain critical applications, e.g., multimedia codecs.

Compared to network elements, the SmartCard is a new, programmable component. Due to resource limitations, the Forwarding Engine and Computation Platform just collapse to one operating system platform. Also, the middleware layers are typically quite restricted.

Service deployment and control of reconfigurations are complex since there is a split of responsibility between operator and manufacturer. For instance, the manufacturer has to provide or digitally sign appropriate low-level code for reconfigurations. On the other hand, the operator is interested in controlling the configuration to fit the user and network needs.

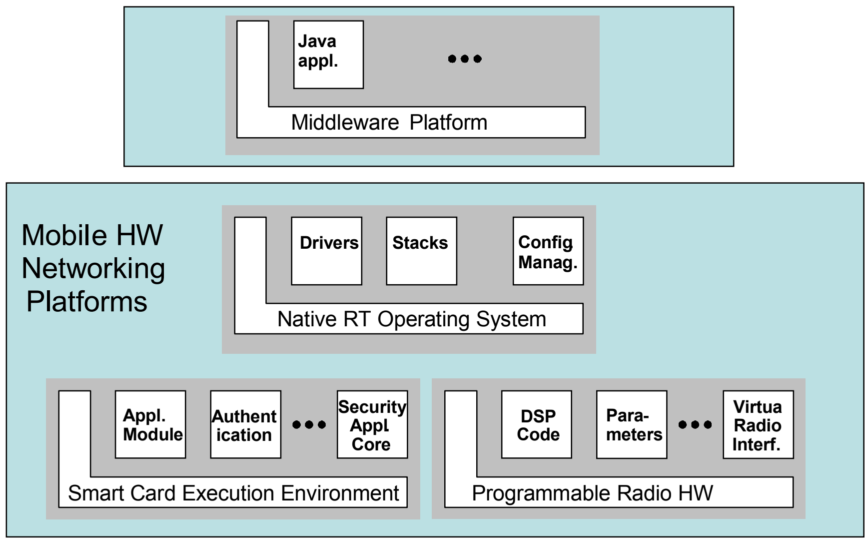

Fig. 2. Mobile Terminal Architecture

3 Driving Forces for Active Network Technology

Active networking technology is a major step towards flexibility in networking. This follows a long-term technology trend towards open, virtual execution platforms in many application areas. While there has been extensive research on active networks, their commercial application is still in its infancy. The vision of a fully programmable network is difficult to achieve, since one has to balance the additional costs for active networks with their benefits. The main cost factors are more expensive hardware, development of a reliable active networking platform, integration with existing software etc. The main benefits are increased flexibility for new software and more efficient software development, including maintenance.

Since this increased flexibility is the main stronghold of active networks, we focus in the following on this. Typically, one assumes that more flexibility is needed on the application and middleware layers or higher, and less on the networking layer. For mobile networks, we argue that flexibility in networking is needed as well for the following reasons:

– Mobile networks are very fragile, due to wireless links and mobility
– Mobile networks are more expensive than wired networks, hence optimizations pay off more easily
– Considerable innovation in air interfaces will require adaptation for new technologies

We show in the following that flexible component installation and cross-layer interfaces are a key ingredient for wireless network and can be ideally supported by active networks.

3.1 Cross-Layer Interfaces and Active Networks

Networking protocols are traditionally classified in layers, based on the ISO/OSI classification of layers. This is an important abstraction needed for modularity and for managing the complexity of current networking systems. However, in mobile networking, one often needs specific and possibly medium-dependent interfaces between the layers, as e.g., discussed in [8]. Cross layer interfaces define APIs between modules in different layers which exchange information beyond the standard interfaces between the layers.

In the classical view, cross-layer interfaces may break modularity and the important concept of abstraction. The main benefit of this abstraction is that layers can be exchanged modularly, e.g., a different air interface should not impact the higher layers. However, in wireless networks, the layer 3 and above often need direct interfaces to the layer 2, e.g., for hand-over support. More examples are discussed in [10][11]. These cross-layer interfaces can be classified as follows:

- Additional information about the network, e.g., bandwidth or delay variations
- Notification of imminent changes in the network services, e.g., hand-over or QoS degradation
- Notification of actual changes in the network service
- Configuration APIs, e.g., setting parameters for wireless interfaces and also the updating of the components (reconfiguration)
- Information from the applications including user context

The problem is that standardizing cross-layer interfaces is inherently difficult. Furthermore, once standards have been established, these interfaces have to be supported once and for all, even if they are outdated or not suitable any more. In summary, cross-layer interfaces are difficult to maintain because cross-layer interfaces are often media or application specific and evolve fast.

We claim that active networks are an important technology to install components with cross-layer interfaces. Programmable networks can resolve the above problems by introducing “dynamic” cross-layer interfaces. Instead of static, built-in cross-layer interfaces, the components providing the interfaces are installed when needed.

By dynamically installing APIs, we can avoid many problems with static interfaces as described above. This concept is ideal for medium specific API, since the components can be installed in a medium specific way. For instance, different APIs can be installed for each medium, depending on the desired services. This notion of service deployment architecture has been investigated for active networks in [9], but not in the context of cross layer interface.

From this discussion, we argue that active networking and programmability on other layers has to be integrated. Lower layer reconfiguration, e.g., software radio, may not use an open execution platform but often use simpler forms of software deployment. However, this programmability has to be combined with active networking to offer a full solution.

4 Application Examples

In this section, we will sketch a number of examples to show the importance of flexible networking protocols and the need of dynamic cross layer interfaces in future mobile networks. Mobility management and hand-over optimization are central services of mobile networks. Theses services can often be optimized in many ways, for instance by using the context information available on other layers. Furthermore, applications can be optimized with mobility management. For instance, flexible quality of service and adaptation of services is a common example of active networking, as discussed below.

4.1 Context-aware Hand-Over

In this example we show that hand-over can be optimized by information about the user context. As an example, we consider the optimal selection of a new access point during hand-over, as shown in **Fig. 3**. In this scenario, the terminal moves into an area covered by both AP2 and AP3. The problem is to decide which Access Point (AP) to choose. State of the art are many algorithms based on signal strength analysis or on available radio resources. Even if one AP is slightly better regarding these local measurements, the decision may not be the best. For instance, if the terminal in **Fig. 3** is on the train, it is obviously better to hand-over to AP3, even if AP2 is better reachable for a short period of time.

In many cases, the hand-over can be optimized by knowledge of terminal movement and user preferences. For instance, if the terminal is in a car or train, its route may be constrained to certain areas. Also, the terminal profile may contain the information that the terminal is built in a car. Alternatively, the movement pattern of a terminal may suggest that the user is in a train.

A main problem is that hand-over decisions have to be executed fast. However, the terminal profile and location information is often available on a central server in the core network. Retrieving this information may be too slow for hand-over decisions. Furthermore, the radio conditions during hand-over may be poor and hence limit such information exchange.

The idea of the solution is to proactively deploy a context-aware decision algorithm onto the terminal which can be used to assist hand-over decisions. A typical example is the information about the current movement pattern, e.g., by knowledge of train or road routes. For implementation, different algorithms can be deployed by the network on the terminal, depending on the context information. This implementation needs a cross layer interface, which collects the context information from different layers and makes an optimized decision about deploying a decision algorithm.

Similar ideas have already been investigated in [12], where a middleware was presented which provided a hook for a function which determines the best next access point. In [13], algorithms based on a static user profile are presented which select among different networks for hand-over.

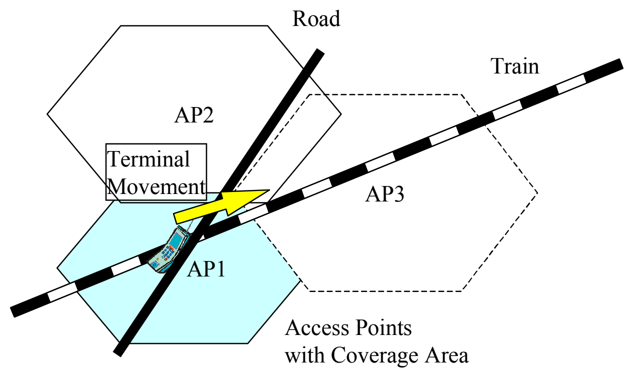

Fig. 3. Context-Aware Hand-Over Prediction

4.2 Customized Paging

In mobile networks, the network needs to know the location of the mobile terminal (MT) in order to keep the connection with it. This requires the mobile terminal to send the network a location update continuously. A fast moving MT has to send the location update frequently, which causes considerable signaling overhead. For this reason, paging is used instead to save the energy and decrease the signal overhead of location update when a MT is in idle state [15].

With paging, the complex location registration to the networks needs to be performed only when the idle MT moves to another paging area. One paging area consists of several cells with access points. In order to receive an incoming call, the paging process is used to find the exact location of MT in the paging area. The paging strategy can be "Blanket Polling" or "Sequential Paging". In the first strategy, a paging request is sent to all the wireless APs in the paging area simultaneously, then the AP in the cell where the MT located reply the paging request. In the second strategy, paging requests are sent to the APs sequentially in decreasing order of the likelyhood that the MT is located in that cell.

The size and shape of the paging area are essential for the paging efficiency. If large paging areas are chosen, the cost of the paging process will increase, otherwise the rate of registrations and battery consumption will increase. In current systems, a fixed paging area size is used. It is of course not optimal. Active network technology facilitates the customization of the paging area, thus improving the paging performance. Similar to the customized handover, a customized paging also employs the user profile and mobility information to dynamically adjust the paging area.

Fig. 4 gives an example of customized paging area. Suppose the MT is on the train, then the direction and speed of its movement can be determined. In this case, the paging area outside the train route is not necessary. The optimized paging area should be along the railway track as shown in **Fig. 4**. Because normally trains move very fast, a bigger paging size is preferred to avoid the frequent location registration. Otherwise, if the MT's movement is slow and unpredictable (e.g., pedestrian's mobile phone), the optimal paging area should be centered around the MT and has smaller size (because the registration cost is low). From the above examples, we can see that it will be more efficient to use an adaptive paging area according to the mobility information of the MT.

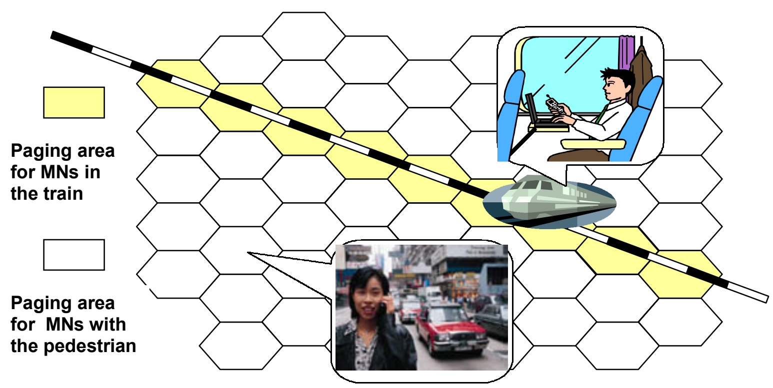

Fig. 4. Example of Different Paging Areas for Different MTs

Using active network technology, a program for the paging area calculation can be loaded to the paging area control node according to different scenarios (e.g., the algorithm and input parameters can be changed). The program can be loaded to the computing platform or execution environment in the networking platform. The local context of the MT decides which computation program to load. For example, the user profile may be obtained from the smart card.

There is some literature on the optimization of paging areas. For instance, [16] estimates the movement of the MT by sending samples. Parameters such as speed, direction are computed based on the samples received. Then the paging area is calculated from the movement parameters. In [17], a behavior-based strategy (BBS) is proposed to estimate the mobile's location by collecting the mobile's long term moving logs. A similar approach is discussed in [18]. With the active networks solution, we have the flexibility to load different algorithms. The input parameters can be changed dynamically according to different user profiles. Obviously, interaction between network layer and the application layer are needed in this application as well.

In the following, we use blanket polling as an example to describe an implementation of customized paging area using our architecture, as shown in **Fig. 5**.

1. The network side selects blanket polling as the paging protocol in the computing platform and informs the MT of the selected paging protocol.
2. The network side installs the appropriate interface between the paging protocol and customized paging area selection service.
3. The customized paging area selection service component in the MT side retrieves the user profile from Smart Card, and sends this information to the customized paging area selection service component in the network side. The user profile and other location information from the network side are used to decide which algorithm will be used to calculate the customized paging area. E.g., if the MT is installed in the car, the algorithm will be based on the road information and car speed.

4. The algorithm selected above interacts with some location related applications, (e.g., train route service or road map service) to compute customized paging area.
5. The specifications of the customized paging area are sent to both the paging protocol in the network side and the paging protocol in the MT side. This information is used in the network for paging process and is used in MT for location update when the MT crosses the border of the customized paging area.

In addition, if the MT changes its profile (e.g., get off the train), the example will continue from step 3. When the MT leaves the customized paging area, the example will continue from step 4.

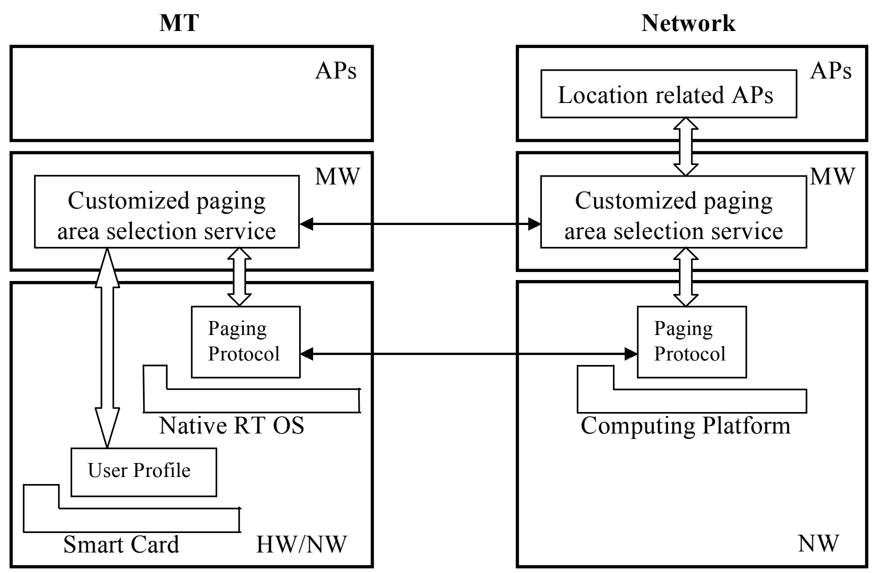

Fig. 5. Reference Architecture for Customized Paging Area

Similarly, we can also use sequential paging as the paging protocol. The use of different paging protocols does not only require the capability of loading different components into the computing platform, but also asks for a facility to adapt the interfaces between the MW and the NW layers.

4.3 Concatenated Location Management

Currently, location management is designed for individual mobile terminals, provided that each MT moves independently. Such independent location management will need a lot of signaling for handover and location registration. This signaling causes bandwidth waste in the case that MTs are moving in a group (in the same train or car). A novel concept was introduced in [2] to group individual MTs which are moving in the same manner into a given movement class.

The procedure of concatenated location registration is briefly described as follows. Each group of MTs performs a concatenated location registration as a group, and not each individual terminal, i.e. only one selected MT in this group needs to register to the network. When a new MT joins the group or when one MT leaves the group, e.g., when entering or leaving a train, the network is informed as well. This registration triggers the network to associate the mobility information of the new MT to the group. There are different ways to organize the group, e.g., by local communication within the group or by some extended node which is in charge of the group management e.g., which resides in the train. For this procedure, we argue to use active networks technology for the following reasons:

– Flexibility is needed for group management. When and how to form or to release a concatenation is a difficult question but quite essential for mobility management. Using active network technology, different algorithms can be loaded to make the decision according to different scenarios. With the emerging of new scenarios or new transportation methods, new algorithms can be offered. Active networks technology facilitates the flexible loading and update of the algorithm.
– Dynamic interfaces between the group management (typically middleware layer) and the location registration protocol are needed. Since we expect to have different group management solutions, their interfaces to the lower-layer protocols can also be different. The implementation of the algorithm may involve the programmable hardware interfaces, the OS (including drivers and stacks) and the middleware layer.
– Localized calculation. The group management functions can be loaded to appropriate access point or terminals when needed. Therefore, the active network approach may avoid communication overhead and has good scalability.

4.4 Customized QoS

Quality of Service is designed for the purpose of performance assurance and service differentiation. The networks should be able to offer different QoS according to the user requirements. For example, a television station is broadcasting some important news. For users with wired networks connections, they would like to receive both the video and audio streams with high quality. For users with their mobile device, only low quality video clips or only audio with text can be received in real time. Also the available wireless capability may change during the connection because of the mobility. E.g., after handover, only lower capacity is available. In this case, the user may prefer some lower quality media service to service interruption. This requires the networks to offer differentiated, dynamically adapted service. In this case, the networks should support multi-resolution scalable media.

An active architecture gives a flexible way to scale the media on demand. The MT sends the user scaling preference and radio profile to the access node. Based on this information, the corresponding access node loads suitable media filers or defines the parameters of the filers. A similar idea is implemented in [4], in which filters are dynamically uploaded to adapt to the capabilities of the visited networking environment.

In mobile and wireless environments, not only the capacity of the connection changes frequently, which make the performance assurance difficult, but also the handoff latency greatly affects the QoS. A lot of approaches are deployed to decrease the handover latency (e.g., fast handover, cellular IP, hierarchical IP, dynamically change the anchor point [2] [19]). Different handover strategies have different characteristics. Using active networks, the most suitable handover strategy can be chosen to fulfill the individual QoS requirement of the application.

4.5 Other Application Areas

There exist many other applications of active networks in the area of mobile networks. We list a number of them in the following:

Hand-Over Triggers

An interesting application area is hand-over optimization based on layer 2 triggers. In the hand-over procedure, a MT first needs to setup a new layer 2 connection, then gets a new address in a foreign network (often called Care-of-Address), then performs the layer 3 binding update. This procedure causes a hand-over latency which might be unacceptable for some delay sensitive applications.

Several handoff schemes have been suggested to eliminate the hand-over delay by using cross-layer interfaces between layer 2 and layer 3. One of them is proactive handoff, where link layer triggers assist the MT in determining that a handoff is imminent and establish packet flow to the target AP prior to the handoff event [20]. This hand-over scheme needs the coupling of the layer 2 and layer 3, which fits nicely with our architecture.

Software Radio

Another important application area is the control of terminal reconfiguration in software radio [3]. Note that software radio can be applied to both terminals and base stations. Since software radio often aims at simpler forms of software deployment, a local control module is important. This control of the software radio reconfiguration, in both base stations and terminals, can be provided by active networks. Since new software may implement un-anticipated new features, the software reconfiguration process should be as flexible as possible. Hence active networking can be an ideal implementation platform for software radio control mechanisms.

Ad-Hoc Networks

In ad-hoc networks, mobile terminals can be networking nodes as well. The connectivity among them may change frequently, depending on their mobility. Therefore, it is difficult to find the best ad-hoc routing protocol suitable for all circumstances. For instance, [21] uses active networks to support customization of routing protocols, where the most suitable routing protocol can be chosen from multiple routing protocols according to the QoS requirements, security concerns, link characteristics, speed of the ad hoc nodes and number of nodes in this ad hoc networks. Since there is potentially a large variety of ad-hoc technology and applications, it is important to devise a flexible architecture which is suitable for the integration of ad-hoc networks with cellular mobile networks.

5 Conclusions

We have discussed active networking and programmability in a complete architecture for future mobile networks. We have shown that a clear, common vision is needed to introduce active networking. In particular, we have shown that active networking must consider the configurability of different system components. Another issue is that well-accepted standards are needed for a smooth migration to active networks.

We think that future mobile networks can be the ideal target for the adoption of active networks. The main benefit of active networks is the added flexibility. This flexibility is mainly needed for two reasons. First, applications will evolve rapidly and adaptation of lower layer infrastructure will be needed to optimize these applications. Secondly, new wireless technologies and ad-hoc networks will require continuous adaptation of the networking layer.

We have shown that flexible cross-layer interfaces are needed for these applications. Active networks can be the key enabling technology for this. Furthermore, active networking also supports the separate evolution of different networking layers. This is in contrast to current mobile networks, where often the wireless technology sets the pace for introducing new networking infrastructure.

References

[1] Jamalipour, and S. Tekinay (eds.): Special issue on Fourth Generation Wireless networks and Interconnecting Standards. October 2001. IEEE Personal Communications Magazine

[2] H. Yumiba, K. Imai, and M. Yabusaki: IP-Based IMT Network Platform. IEEE Personal Communications Magazine, October 2001.

[3] N.J. Drew, M.M. Dillinger: Evolution toward reconfigurable user equipment. IEEE Communications Magazine, Vol. 39, Issue 2, Feb. 2001.

[4] T. Campbell, M. E. Kounavis, and R. R.-F. Liao: Programmable Mobile Networks. Computer Networks, Vol. 31, No. 7, pg. 741-765, April 1999.

[5] R. Hirschfeld, W. Kellerer, C. Prehofer, H. Berndt, An Integrated System Concept for Fourth Generation Mobile Communication, EURESCOM Summit 2002

[6] WWRF, Wireless World Research Forum (WWRF), http://www.wireless-world-research.org/

[7] M. Christensen, The Innovator’s Dilemma, Harward Business Press, Boston, 1997

[8] M. Bossardt, L. Ruf, R. Stadler, B. Plattner: A Service Deployment Architecture for Heterogeneous Active Network Nodes. Kluwer Academic Publishers, 7th Conference on Intelligence in Networks (IFIP SmartNet 2002), Saariselkä, Finland, April 2002.

[9] Brewer, E.A.; Katz, R.H.; Chawathe, Y.; Gribble, S.D.; Hodes, T.; Giao Nguyen; Stemm, M.; Henderson, T.; Amir, E.; Balakrishnan, H.; Fox, A.; Padmanabhan, V.N.; Seshan, S., A network architecture for heterogeneous mobile computing, IEEE Personal Communications, Volume: 5 Issue: 5 , Oct. 1998

[10] Yue Fang; McDonald, A.B., Cross-layer performance effects of path coupling in wireless ad hoc networks: power and throughput implications of IEEE 802.11 MAC, Performance, Computing, and Communications Conference, 2002. 21st IEEE International , 2002

[11] Raman, B.; Bhagwat, P.; Seshan, S. Arguments for cross-layer optimizations in Bluetooth scatternets, Symposium on Applications and the Internet, 2001, Page(s): 176 -184

[12] Michael Kounavis and Andrew Campbell, Design, Implementation and Evaluation of Programmable Handoff in Mobile Networks Mobile Networks and Applications 6, 443-461, 2001 2001 Kluwer Academic.

[13] Helen J. Wang, Randy H. Katz, and Jochen Giese. Policy-Enabled Handoffs Across Heterogeneous Wireless Networks. In WMCSA 99, New Orleans, LA, 1999
[14] Konstantinos Psounis, Active Networks: Applications, Security, Safety, and Architectures, IEEE Communication surveys, http://www.comsoc.org/livepubs/surveys/index.html
[15] J. Kempf, Dormant Mode Host Alerting ("IP paging") problem statement, RFC3132, June 2001
[16] P. Mutaf C. Castelluccia, DPAC: Dynamic Paging Area Configuration, <draft-mutaf-dpac-00.txt>. September 2001.
[17] Hsiao-Kuang Wu, Ming-Hui Jin, Jorng-Tzong Horng, Cheng-Yi Ke, Personal Paging Area Based On Mobile's Moving Behaviours, IEEE INFOCOM, May 2001
[18] Z. Lei, C.U. Sarazdar and N.B. Mandayam, Mobility Parameter Estimation for the Optimization of Personal Paging Areas in PCS/Cellular Mobile Networks, Proc. 2nd IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC'99), May 9-12, 1999
[19] Atsushi Iwasaki, Takashi Koshimizu, Proactive Region Reconfiguration Handover, IPCN 2002, April 2002, Paris, France
[20] G. Dommety, A. Yegin, C. Perkins, G. Tsirtsis, K. El-Malki, M. Khalil, Fast Handovers for Mobile IPv6, draft-ietf-mobileip-fast-mipv6-04.txt, March 2002
[21] Christian Tschudin, Henrik Lundgren and Henrik Gulbrandsen, Active Routing for Ad Hoc Networks, IEEE Communications Magazine, April 2000
[22] R. Wittmann, M. Zitterbart. *AMnet: Active Multicasting Network*, International Conference on Communications (ICC '98), Atlanta, USA, June, 1998
[23] D. Decasper, G. Parulkar, S. Choi, J. DeHart, T. Wolf, B. Plattner, *A Scalable, High Performance Active Network Node*, In IEEE Network, January/February 1999
[24] S. Merugu, S. Bhattacharjee, E. Zegura and K. Calvert, *Bowman: A Node OS for Active Networks*. Proceedings of IEEE Infocom 2000, Tel Aviv, Israel, March 2000.
[25] David Wetherall, John Guttag and David Tennenhouse, ANTS: Network Services without the Red Tape, IEEE Computer, 32(4):42-9, April 1999.