Accelerating Service Creation and Deployment in Mobile Networks

Michael E. Kounavis, Andrew T. Campbell, Gen Ito, and Giuseppe Bianchi

Abstract—**Future mobile networks should be built on a foundation of open programmable networking and software radio technology that is capable of responding to the specific radio, mobility and quality of service needs of wireless service providers. We describe the design, implementation and evaluation of a programmable architecture capable of profiling, composing and deploying wireless and mobile network services. We introduce a number of new mobile services as proof of concept using network programmability and service creation. First, we describe 'multi-handoff access networks', which are capable of simultaneously supporting multiple styles of handoff over the same physical access network infrastructure. Next, we discuss 'reflective handoff services', which allow mobile devices to roam across heterogeneous access networks. Finally, we describe the notion of a programmable MAC, which supports the composition of new MAC services at run-time. We believe that programmable mobile networks with their capability to dynamically create new mobile services on-demand will accelerate the pace of innovation in wireless and mobile networking and computing.**

***Index terms*—programmable mobile networks, distributed systems, service creation, reflective handoff**

I. INTRODUCTION

Programmable mobile networks are motivated by the limitations of existing network architectures. First, existing mobile network services and protocols cannot be easily extended or modified because they are typically implemented using dedicated firmware or as part of low level operating system support. Next, incompatibility of signaling systems and physical radio layer technologies prevents mobile devices from roaming between heterogeneous wireless systems. Third, access network protocols make specific assumptions about the capability of mobile devices (e.g., Mobile IP [13] and mobile ATM [15] approaches assume that handoff control is located at the mobile device). Such mobile-controlled handoff schemes may not be suitable for many low-power mobile devices that are incapable of continuously monitoring channel quality measurements. Finally, no software tools are available that automate the process of architecting mobile networks.

In response to these limitations, we propose an alternative way of building signaling systems and physical/data link layer support into mobile networks and terminals. We argue that mobile network services and protocols can be composed of fundamental building blocks from which one can program a wide range of new services. Wireless physical layers can be created by introducing appropriate code into software radio base stations with wide-band tunable front-ends [5-7]. MAC layers can be programmed on top of open programmable radio environments [8]. Communication algorithms that implement mobility control and management can be introduced into the network infrastructure as a set of distributed and programmable objects [3]. In addition, network architects can observe, analyze and modify the structure and building blocks of programmable mobile networks at run time [24].

Programmable mobile networks represent an emerging area of research [3, 5-11, 21]. While the community has mostly addressed the programmability of the physical layer and signaling plane as separate network components, little work has been done on programmability and service creation spanning all layers and planes that characterize mobile networks. This paper is structured as follows. In Section II we describe a programmable architecture that enables mobile service creation in a unified manner. Following this, in Sections III and IV we discuss the implementation and evaluation of our architecture. Finally in Section V we provide some concluding remarks.

II. ARCHITECTURE

An architecture that supports the profiling, composition and deployment of mobile network services is illustrated in Figure 1. Our architecture comprises a binding model and a service creation environment. The binding model describes how collections of distributed algorithmic components can be combined with each other in order to compose mobile network services. The service creation environment allows network architects to design and dynamically deploy mobile network services, taking into account user, radio, and environmental factors.

Michael E. Kounavis and Andrew T. Campbell are affiliated with the Center for Telecommunications Research, Columbia University, {mk, campbell}@comet.columbia.edu. Gen Ito is affiliated with NEC Japan, itogen@sns.abk.nec.co.jp. Giuseppe Bianchi is affiliated with the University of Palermo, Italy, bianchi@morgana.elet.polimi.it.

0-7803-6268-3/00/$10.00 (c) 2000 IEEE

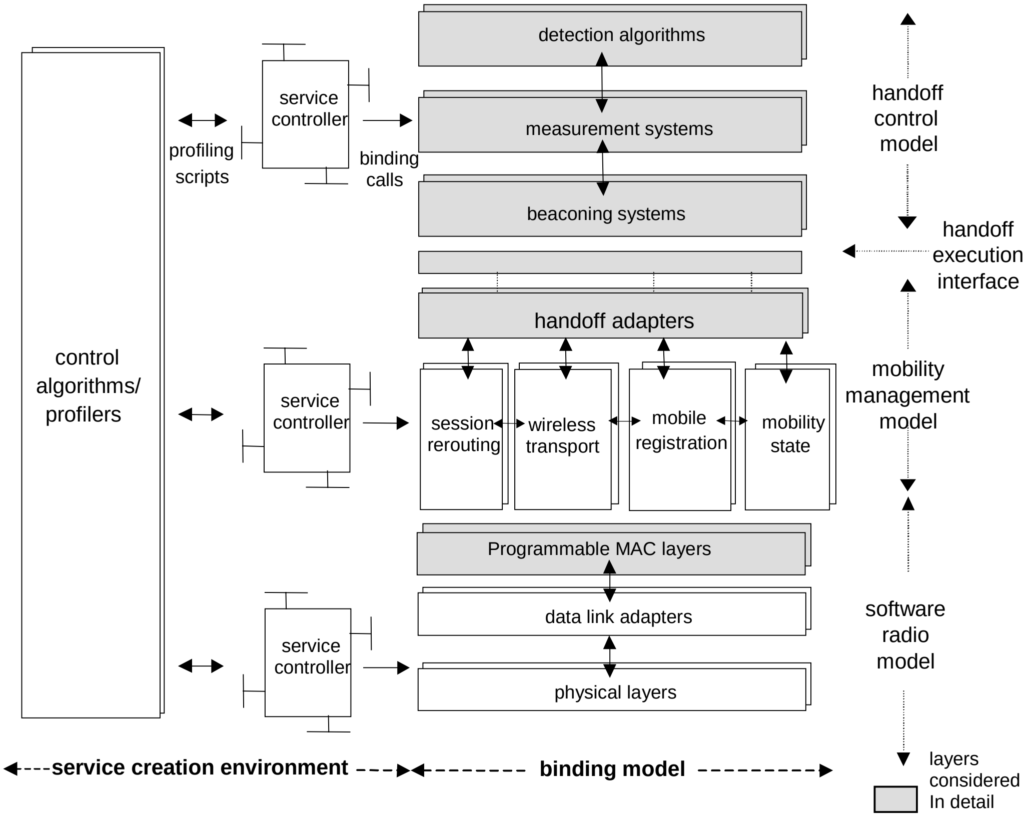


Figure 1: Programmable Architecture for Service Creation

A. *Service Creation Environment*

A service creation environment offers interfaces for the dynamic introduction or modification of mobile network services. Mobile network control and management services are implemented as collections of open distributed objects. Middleware technologies (e.g., CORBA, Java RMI) are leveraged for enabling network-wide system interoperability, separating the definition of programmable objects from their implementation. Network objects expose control interfaces that allow them to create bindings at run time. Physical and data-link layers can be built from components as well. The service creation environment comprises a set of service controllers. Service controllers create, modify, or delete handoff control, mobility management, and physical/data link layer systems. Service controllers can create new programmable mobile services or modify existing ones. Objects are selected from component repositories. Modification of programmable mobile services takes place after some bindings are removed or objects are deleted. Service controllers may run in wireless access networks or mobile devices.

Service controllers interact with profilers and control algorithms. Profilers are design tools that drive the service creation process. Profilers illustrate the topology of access networks and are used for creating graphical representations of mobile network services. Graphical representations of mobile network services are translated into profiling scripts. Control algorithms query for the architectural description and state of mobile network services, making decisions whether mobile network services should be introduced, modified, or removed.

B. *Binding Model*

A binding model, illustrated in Figure 1, reflects the composition of mobile network services. The binding model comprises a handoff control model, a mobility management model, a handoff execution interface, and a software radio model.

1) *The Handoff Control Model*

The handoff control model separates the transmission of beacons from the collection of wireless channel quality measurements, and from the handoff detection algorithm. Typically, these functions are supported in a single ‘monolithic’ structure. By separating the handoff detection algorithm from the collection of wireless channel quality measurements, we allow for new detection algorithms to be dynamically introduced in access networks or mobile devices. For example, detection algorithms specific to overlay networks can be introduced to mobile devices allowing them to perform vertical handoffs or detection algorithms specific to micro-cellular networks can be selected for compensating against the street-corner effect.

By separating the collection of wireless channel quality measurements from the beaconing system, mobile networks can support different styles of handoff control over the same wireless infrastructure. For example a wireless Internet Service Provider may want to offer a network-controlled handoff service (e.g., supported in AMPS cellular systems) for simple mobile devices, a mobile-assisted handoff service (e.g., as in GSM) for more sophisticated mobile devices, involved in the process of measuring channel quality, or a mobile-controlled handoff service for laptop computers or palmtops.

The handoff control model illustrated in Figure 1 comprises the following services:

- *detection algorithms*, which determine the most suitable access points where a mobile device should be attached on the basis of a number of different factors including channel quality measurements, resource availability and user-specific policies. A mobile device can be attached to a single or multiple access points simultaneously.
- *measurement systems*, which produce and update handoff detection state. By handoff detection state we mean the data used by detection algorithms to make decisions about handoff. Detection algorithms and measurement systems use the same representation for handoff detection state.
- *beaconing systems*, which assist in the process of measuring wireless channel quality. Programmable beacons can be customized to support service-specific protocols like QOS-aware beaconing [3] or reflective handoff as discussed in Sections III and IV.

2) *The Handoff Execution Interface*

A handoff execution interface separates handoff control from mobility management**.** We argue that implementation details of mobility management algorithms can be hidden from handoff control systems, allowing handoff detection state (e.g., the best candidate access point for a mobile device) to be managed separately from handoff execution state (e.g., mobile registration information). This software approach can be used for enabling inter-system handoffs between different types of access networks.

The basic idea behind realizing inter-system handoffs is that the same detection mechanisms operating in mobile devices, or access networks can interface with multiple types of mobility management architectures, operating in heterogeneous access networks. Handoff control systems issue a number of generic service requests, which mobility management systems execute according to their own programmable implementation. In one extreme case where the location of the handoff control system is at the mobile device, different mobility management protocols can be loaded into mobile devices dynamically, allowing them to roam between heterogeneous wireless environments in a seamless manner.

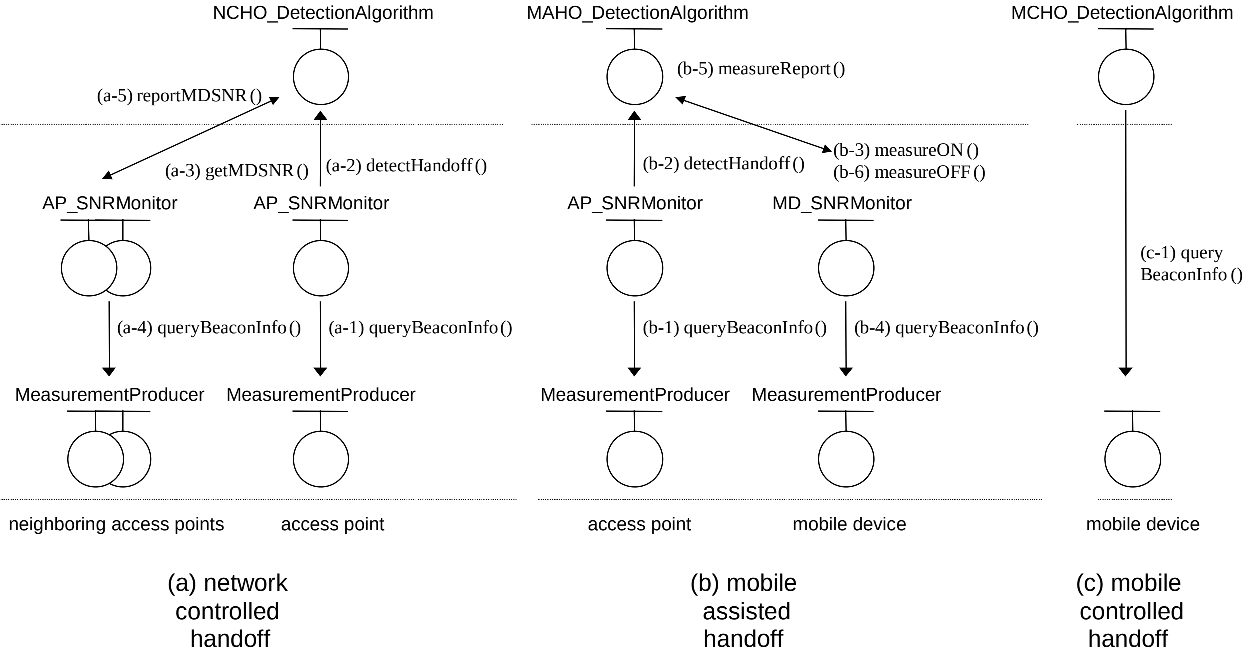
The separation of handoff control from mobility management can be realized through ‘facade’ objects, which expose the handoff execution interface. Facade objects interact with handoff adapters. Handoff adapters are distributed systems that can be deployed in mobile devices, access points and mobile capable routers/switches. The role of adapters is to convert the handoff execution interface to interfaces supported by specific mobility management architectures. For example, a ‘Mobile IP’ adapter would support connectivity for mobile devices using some Mobile IP scheme (e.g., acquiring a care-of address through DHCP and registering it with a home agent).

3) *The Mobility Management Model*

The mobility management model, illustrated in Figure 1, reflects the composition of services that execute handoff. We use a neutral architectural model, supporting the design space of different mobile networking technologies. The mobility management model represents handoff services, and does not encompass the whole range of mobility management functions (e.g., location management, fault, and accounting management) that are typically supported in mobile networks.

We identify the following services as part of the handoff execution process:

- *session rerouting* mechanisms, which control the datapath in access networks in order to forward data to/from mobile devices through new points of attachment. Rerouting services may include admission control and QOS adaptation for the management of wireless bandwidth resources.
- *wireless transport* objects, which interact with the physical and data link layers in mobile devices and access points to transfer active sessions between different wireless channels. A channel change may be realized through a new time slot, frequency band, code word or logical identifier. In addition, transport objects can provide valued-added QOS support (e.g., TCP snooping [22]).
- *mobile registration*, which is associated with the state information a mobile device exchanges with an access network when changing points of attachment.
- *mobility state*, which determines the status of the connectivity of mobile devices with an access network. Mobility state can be expressed as addressing and routing information, bandwidth and name-space allocations or mobile user preferences.



(a) network controlled handoff

(b) mobile assisted handoff

(c) mobile controlled handoff

Figure 2: Programmable Handoff Styles

4) *The Software Radio Model*

A software radio model defines the composition of physical and data link layer services. Software radios allow access points and mobile terminals to implement physical layer functionality (e.g., modulation, equalization, channel coding) in software. The degree of programmability software radios offer depends on the type of technology used. Software radios can be implemented using Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), Digital Signal Processors (DSPs), or general-purpose processors.

The software radio model illustrated in Figure 1, supports functions such as the dynamic assignment of channel locations and widths and the selection of modulation and coding techniques used on each channel. Software radios allow mobile devices to dynamically 'tune' to the appropriate air interface used by their serving access network, while roaming between heterogeneous wireless environments. In addition, MAC protocols can be made programmable, allowing for services with different QOS requirements to be supported. Physical and data link layer modules can be implemented in various ways (e.g., as shared libraries in a Unix environment). Data link adapters separate data link layer modules from the lower physical layer components. For example, data link adapters allow programmable MAC protocols to operate on top of any type of channel coding, or modulation scheme.

III. IMPLEMENTATION

A. *Service Creation Environment*

A service creation environment has been built in our testbed using CORBA [19]. CORBA is a widely accepted standard for the development of distributed applications. CORBA does not support transportable software required by our programmable handoff architecture. To allow for the dynamic introduction of new handoff control and mobility management services, the service creation environment provides explicit support for transportable code by selecting, deploying and dynamically binding distributed objects. A network architect can create new objects through inheritance from abstract classes (e.g., an abstract detection algorithm class). Service controllers activate objects, invoking binding calls on their control interfaces. MAC layers are composed in user space from shared libraries. During the profiling process objects are customized by the network architect. Environment, user or service specific parameters characterizing the operation of mobile network services can be passed into objects at run-time through the profiler and service controllers.

Our scripting language is simple, and supports command, assignment, and exception handling statements. Service creation requires the specification of the names and locations of all the objects involved. Such an approach works well for small access networks and simple network services. An alternative approach would be to separate the 'binding rules' defining handoff control and mobility management architectures (e.g., a rule for placing handoff detection objects inside the network) from the 'binding data' (e.g., network topology, user preferences). In this case, a service would be the instantiation of a set of binding rules over some binding data. We are currently investigating the formulation of such a scripting language.

Using the service creation environment we have designed, deployed and evaluated new mobile services that include programmable handoff control and MAC layer services. In a companion paper [3], we describe the implementation of programmable mobility management services, supporting adaptive QOS control and active wireless transport.

B. Programmable Handoff Control

We have designed and implemented a 'multi-handoff access network', which simultaneously supports multiple styles of handoff service over the same physical wireless infrastructure. Another service that we have deployed using our service creation environment is a 'reflective handoff service' that enables mobile devices to dynamically load signaling systems on-the-fly. In this manner, reflective handoff enables seamless roaming between heterogeneous access networks.

1) Multi-handoff Access Networks

Figure 2 illustrates the objects and APIs used when programming different styles of handoff control services. In what follows we discuss the programmability of three well-known services:

- The *network controlled handoff (NCHO)* scheme uses the access point to measure the signal strength (as indicated by (a-1) in Figure 2) of the mobile device it is serving. SNR represents only one of the many measurements, which may be relevant to the handoff decision. When the channel quality drops below a certain threshold, the access point invokes a detectHandoff() method on a detection algorithm, running inside an access network (a-2). The detection algorithm dispatches a thread which queries and compares mobile device measured signal strength from all neighboring access points (a-3, a-5). Network controlled handoff moves most of the complexity for controlling handoff from the mobile to the network. This style of handoff simplifies the mobile device software design.

- The *mobile assisted handoff (MAHO)* scheme moves some of the functional support, and therefore complexity, to the mobile device. In this scheme (as in the case of network controlled handoff) a serving access point measures the signal strength of a mobile device continuously. If the signal level drops below a certain threshold, a detection algorithm invokes a MeasureON() method at the mobile device to initiate measurements from neighboring access points (b-3). These measurements are returned back to the detection algorithm, which makes the handoff decision. The detection algorithm does not need to collect any additional measurements but bases its decision for handoff on the data collected by the mobile device.

- The *mobile controlled handoff (MCHO)* scheme moves most of the complexity for managing handoff to the mobile device alleviating the network from centralized control of handoff. In this respect mobile controlled handoff is very scalable and more distributed than other schemes. However, it assumes that the mobile can monitor signal strength measurements continuously (e.g., a wireless laptop device).

In our experiments, all handoff styles use the Mobiware architecture for mobility management support. Mobiware [3] is programmable, promoting the separation between signaling, transport, and state management. In Mobiware, all sessions between a mobile device and a gateway to the core network are abstracted as a single entity called flow bundle. Flow bundles switch IP flows providing general encapsulating and routing services similar to ATM virtual paths or IP tunnels. Using flow bundles a 'mobility agent' object only has to discover a single crossover switch and reroute all sessions to/from the new access point. Open programmable switches allow the establishment, removal, rerouting, or adaptation of user flows. An active transport environment supports end-to-end transport adaptation. Object bindings (e.g., bindings between adapters and mobility management objects) are setup and cached prior to handoff execution in the best candidate access points of a mobile device.

2) Reflective Handoff

In reflective handoff, mobile terminals reprogram their protocol stacks in order to seamlessly roam across heterogeneous wireless environments. We call the service 'reflective' in this context because mobile devices can ideally identify the mobility management architectures and radios supported in the neighborhood of an access network, and customize their signaling systems and air-interfaces in order to interact with these access networks[1].

A mobile networking environment supporting reflective handoff is shown in Figure 3. A Mobile IP enabled internetwork is connected with wireless access networks via gateways. Mobile IP is used for managing macro-level mobility, whereas access networks support fast local handoffs. Mobile devices attached to access networks use the IP addresses of gateways as their care-of addresses. Access networks provide mechanisms for initiating Mobile

---

[1] The concept of reflection was first introduced in the context of programming languages. We have identified many parallels between the behavior of reflective interpreters and programmable mobile terminals. For example both can 'reason' about themselves and manipulate their own operation and structure.

IP-based inter-gateway handoffs and establishing datapaths between gateways and the access points, where mobile devices are attached.

We have implemented reflective handoff as a mobile controlled handoff scheme. Access points transmit beacons carrying globally unique identifiers that designate specific access networks. A reflective detection algorithm uses access network identifiers to determine whether a mobile device is likely to move to the coverage area of a new access network. Each mobile device maintains a local cache of signaling system modules. When a mobile device is likely to perform a handoff to a new access network, it checks whether a signaling module, associated with the candidate access network is cached. If a module is not cached, it is loaded dynamically. Access points support module loaders, deployed as part of the service creation process. Modules are loaded from the old access network. A two-way handshake mechanism is used for loading signaling system support into mobile devices. Modules are loaded before reflective handoffs occur. Access networks schedule the transmission of signaling modules over the air interface to avoid flooding the wireless network

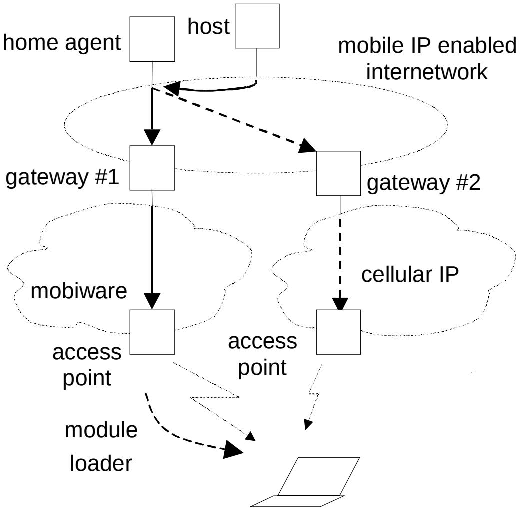


Figure 3: Reflective Handoff Environment

Two distinct types of access networks support reflective handoff in our testbed: a Mobiware and a Cellular IP access network. Cellular IP [4] supports fast local handoff control in datagram oriented access networks. Cellular IP supports per-mobile host state, paging, routing and handoff control in a set of access networks that are interconnected to the Internet through gateways. In Cellular IP, packets sent from mobile hosts create routing caches pointing to the downlink path so that packets destined to a mobile device can be routed using these caches. Mobiware and Cellular IP access networks support the same wireless data link and physical layers (WaveLAN), but use different mobility management systems. Future work will include extending reflective handoff to allow access networks load physical/data link layer support into mobile devices.

C. Programmable MAC

We have developed and deployed a programmable MAC layer [8], as a proof-of-concept introduction of a new MAC layer service. Our implementation of programmable MAC, shown in Figure 4, operates in user space over commercially available wireless LAN environments supporting adaptive mobile services with QOS. We have deployed a programmable MAC over Lucent's WaveLAN. WaveLAN utilizes a collision avoidance mechanism to coordinate access to the wireless medium among competing traffic from mobile devices and access points. WaveLAN provides no explicit support for sophisticated bandwidth sharing and service adaptation.

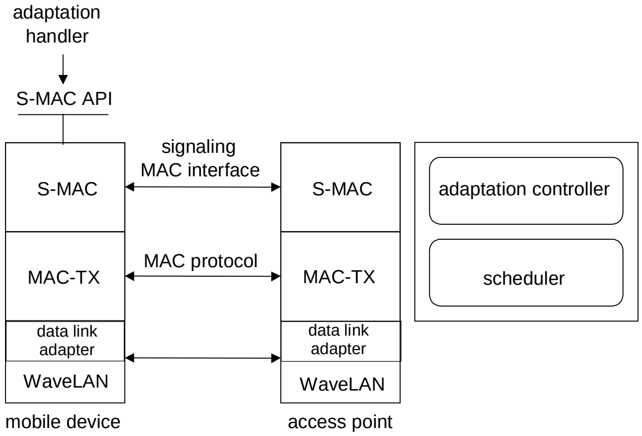


Figure 4: Programmable MAC

In contrast, the programmable MAC, allows mobile applications to map their service requirements into groups of 'bearer' service profiles supported by a programmable scheduler. Profiles are traffic classes supported by the scheduler. The programmable MAC is based on decoupling the individual QOS requirements of each mobile application from the QOS provided by the data link controller. Instead of supporting service specific traffic classes in the scheduling system of the access point, programmable MAC allows mobile applications to program their own adaptive services from a finite set of core traffic classes.

Mobile applications specify their service requirements in terms of:

- *utility curves*, which quantitatively model the degree of adaptability of each application to bandwidth changes;
- *adaptation time scales*, which govern the speed of reaction of adaptation mechanisms to bandwidth variations; and
- *adaptation policies,* which account for adaptation constraints.

Application requirements are mapped to one or more bearer service profiles, which are supported by the centralized scheduler. A centralized adaptation controller running at wireless access points and a set of distributed adaptation handlers drive adaptation. The adaptation handlers interact with the adaptation controller using a special signaling protocol called S-MAC. The adaptation controller uses the interfaces of the scheduler to create delete or modify service profiles. Whenever there is a change in the bandwidth allocated to a profile, or when new profiles are created, the values of bandwidth allocated to each profile are passed to a programmable MAC scheduler. A programmable MAC scheduler uses this bandwidth information to perform packet scheduling for each profile accordingly.

A data link adapter interfaces programmable MAC modules with the lower data link control and physical layer components. Through the data link adapter, the programmable MAC can operate on top of any type of channel coding, or modulation scheme. The deployment of the programmable MAC architecture over WaveLAN results in the creation of a data link controller that is different from the contention-based WaveLAN protocol. The programmable MAC offers QOS support to mobile multimedia applications without requiring any specialized hardware or modifications to the operating system kernel.

IV. EVALUATION

A. Handoff Evaluation

1) Programmable Handoff Services

To evaluate programmable handoff services, we have extended the Mobiware testbed [3]. The experimental environment provides wireless access to the Internet and comprises four ATM switches and four wireless access points. The access points and switches run the service creation environment, programmable handoff control, and mobility management (Mobiware) code. Our radios are based on WaveLAN[2]. For the results provided in this paper the mobile devices and access points use IONA's Orbix v2.0 CORBA running under Windows NT and UNIX operating systems.

We measured the handoff latency observed by three mobile devices supporting different styles of handoff services, when the system was lightly loaded. We also ported our code to an emulation mode, operating in the same experimental environment, in order to evaluate programmable handoff taking into account load conditions[3]. We measured the average handoff latency observed by each emulated mobile device and varied the average values of intervals between successive handoffs, resulting in different cell crossing rates and signaling loads.

	measurement collection latency ***(msec)***	*wireless connection latency* ***(msec)***	*wireline connection latency* ***(msec)***	*total latency*
Mobile Controlled Handoff	-	19 ± 1	22 ± 1	41 ± 1
Mobile Assisted Handoff	709 ± 65	20 ± 1	21 ± 1	750 ± 65
Network Controlled Handoff	641 ± 59	20 ± 1	22 ± 2	683 ± 59

Table 1: Handoff Latency Measurement Results

Table 1 summarizes the results of our handoff latency measurements for a lightly loaded testbed. We took 20 latency measurements for each handoff scheme. The average handoff latency for the mobile controlled handoff scheme was measured to be 41 msec. This measurement broke down into 22 msec for wireline connection setup and 19 msec for wireless connection setup between the access point and mobile device. Mobile controlled handoff is associated with the least amount of latency because in this scheme a mobile device takes signal strength measurements continuously. The handoff execution process for this scheme does not include channel quality measurements.

The average handoff latency for the mobile assisted handoff scheme was measured to be 750 msec. The greatest portion of this latency (709 msec) was absorbed by the process of measuring neighboring access point signal strength at the mobile device. The 'hunt' period for collecting measurements at the mobile device was set to 300 msec, whereas beacons were transmitted by access points and mobile devices every 100 msec. Our detection algorithm initiated handoff if a candidate access point with better SNR was indicated over two successive hunt periods. The average network controlled handoff latency was measured to be 683 msec. The measurement collection component was measured to be 641 msec.

Our results indicate that the performance of a multi-service access network is satisfactory when the network is lightly

---

[2] We used the first generation 2 Mbps WaveLAN cards (not compliant with IEEE 802.11). Lucent technologies provided us with an open API for these cards, in order to program the beacon. We augmented beacons with a 6-byte field for implementing service specific protocols.

[3] The only modification made to the source code relates to the objects that measure the channel quality. These objects, which execute at wireless access points, have been modified to suppress the generation of real signal strength measurements. Rather, they receive measurement data from the mobility emulator. Emulated mobile devices have been programmed to remain inside the same cell for an exponentially distributed interval and to move with the same probability to any neighboring cell. The propagation model used by the mobility emulator is based on the path loss component of signal fading only.

loaded. Our system performs well because binding latencies are eliminated. We experienced higher latencies when bindings between objects were not setup or cached prior to handoff.

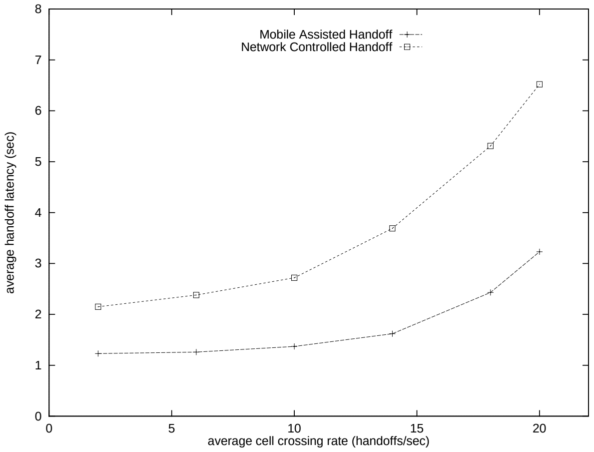
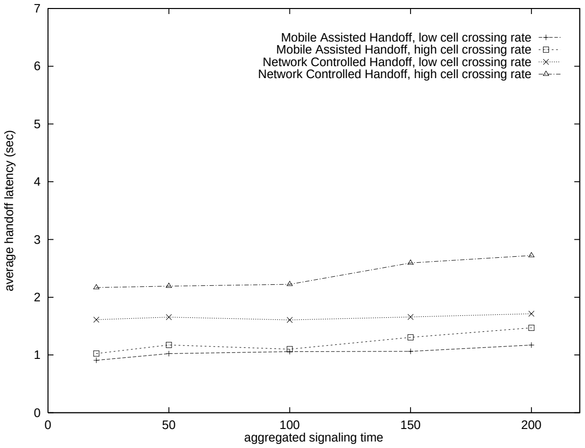


Figure 5-a: Handoff Latency as a Function of the Average Cell Crossing Rate



5-b: Handoff Latency as a Function of the Aggregated Signaling Time

The average handoff latency experienced by a single mobile device as a function of the cell crossing rate characterizing the mobile environment is presented in Figure 5-a. The results shown in Figures 5-a and 5-b were produced using the mobility emulator. We emulated the movement of 1000 mobile devices and took measurements over a period of 60 min. Half the emulated mobile devices were using mobile assisted handoff, whereas the other half networked controlled handoff. The average handoff latency experienced by a mobile device when performing mobile assisted handoff was measured to be between 1.23 and 3.23 sec, when the average cell crossing rate was ranging between 2 handoffs/sec and 20 handoff /sec. In the case of a network controlled handoff the average handoff latency was measured to be between 2.13 and 6.52 sec. The increase in average handoff latency observed when the system operates under signaling load is partially caused by the fact that signaling functions are implemented as interactions between distributed objects. A common feature of many implementations of CORBA is that remote method invocations are queued until they are served in a first come-first served basis. However, in many distributed systems, such as programmable signaling platforms, service requests occur in bursts. In these cases, some significant amount latency can be introduced between the time a request is issued and the time it is served. In the case of programmable handoff control, latency is introduced in the handoff process between the time a handoff execution call is issued by a detection algorithm, and the time the call is served by a mobility agent. In the case of a network controlled handoff, the latency is exasperated due to the fact that access points transmit measurements reporting on mobile device up-link signal strength.

To overcome this problem, we enhanced handoff adapters and signal strength monitoring objects to transmit aggregated service requests or signal strength reports with a single remote method invocation. Aggregated calls carry all service requests that have been issued in the interval between two successive invocations. We call the interval between two successive aggregated calls aggregated signaling time We experimented with different values of aggregated signaling time and measured the average handoff latency experienced by mobile devices using mobile assisted and network controlled handoff. In addition, we took care that access points were not synchronized when transmitting aggregated calls, avoiding further increase in handoff latency.

We performed two sets of experiments. In the first set of experiments the average cell crossing rate of the mobile environment was 2 handoffs/sec. In the second set of experiments, the average cell crossing rate of the mobile environment was 20 handoffs/sec corresponding to more frequent mobility. The results from these experiments are illustrated in Figure 5-b. Aggregation of CORBA calls at access points results in significant reduction of the average handoff latency experienced by mobile devices. Optimal values for the aggregated signaling time are between 20 and 100 msec.

2) *Reflective Handoff*

We have evaluated reflective handoff across a Mobiware and a Cellular IP access network. The Cellular IP testbed consists of three base stations. One of the base stations serves as a gateway router. Details about the Cellular IP testbed can be found in [4]. Table 2 summarizes the characteristics of the signaling modules used in our experiments. Both modules use IP to communicate with access networks. The Mobiware signaling module has been implemented using CORBA technology (Iona’s ORBIX), while the Cellular IP module does not use CORBA.

Signaling modules can have varying sizes, and memory footprints, which affect their downloading time and performance.

The size of the Mobiware module is 1054 Kbytes, whereas the size Cellular IP module is 69 Kbytes. Mobiware implements a complex QOS-adaptive mobility management architecture, while Cellular IP implements a simple handoff scheme without QOS support. Our implementation of the Mobiware module has been based on a commercial ORB, not suitable for wireless environments, resulting in significant footprint and loading latency (7 sec). Clearly, this result indicates that reflective handoff is not a practical solution in our current testbed, where the air interface supports 2 Mbps throughput. If we use radios with speeds of 10 and 25 Mbps, loading times are expected to decrease to 1.4 sec and 0.56 sec, respectively. Future work will include porting the programmable handoff architecture to a ‘light-weight’ minimum ORB [18], which will further decrease the footprint of programmable signaling modules.

Signaling module	*size* *(**Kbytes**)*	*Loading time* ***(sec)***
Mobiware	1054	7 ± 1
Cellular IP	69	0.4 ± 0.1

Table 2: Signaling Modules

Typically, loading times do not affect handoff performance, because signaling modules are loaded prior to handoff execution. Activation latencies were measured to be at most 10 msec for the Mobiware and Cellular IP modules. A significant part of the overall handoff latency was absorbed by the Mobile IP signaling component (42 msec over a single hop). In our experiments reflective handoff latency ranged between 60 and 100 msec.

An illustrative example of the performance of a video application streamed to a mobile device during reflective handoff is shown in Figure 6. Cellular IP initiates care-of-address (i.e., gateway address) registration during entry handoffs only. Mobiware supports care-of address registration during both entry and exit handoffs. Packet loss during Cellular IP to Mobiware reflective handoff is higher because care-of-address registration occurs after the wireless data-link transfer takes place (i.e., a change in WaveLAN NWID). Mobiware to Cellular IP handoff is more efficient because care-of-address registration occurs first and is initiated by the Mobiware access network, during exit handoffs.

Some forwarding delay is introduced at the gateway that connects the Cellular IP access network with the Mobile IP enabled core. Forwarding delay is introduced temporarily for compensating against the time required to accomplish reflective handoff. Forwarding delay is 60 msec in the Mobiware to Cellular IP reflective handoff example as shown in Figure 6. In this case, the performance cost of reflective handoff experiences some minimum increase in interarrival jitter which networked applications are able to handle adequately.

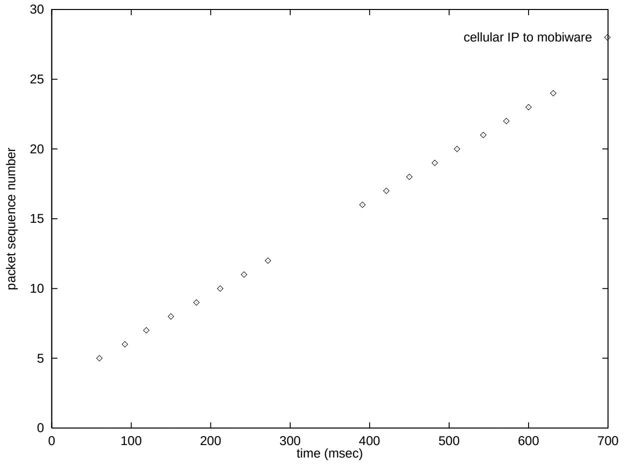


Figure 6-a: Cellular IP to Mobiware Reflective Handoff

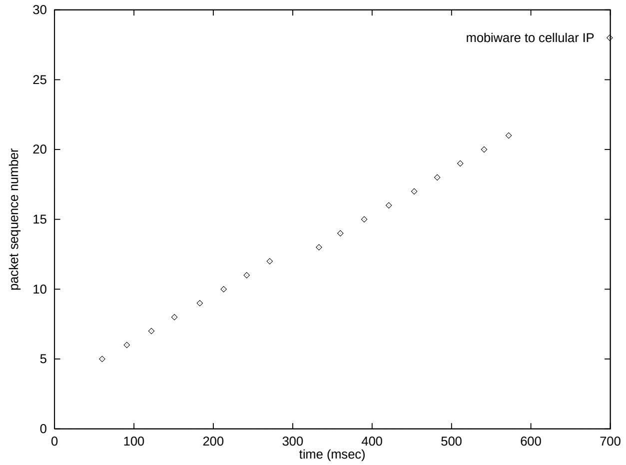


Figure 6-b: Mobiware to Cellular IP Reflective Handoff

B. MAC Evaluation

We have implemented and tested the programmable MAC architecture in the Cellular IP testbed, described above. Programmable MAC modules have been implemented as shared libraries. Programmable MAC modules exchange packets with the Cellular IP protocol stack via shared memory, which is the most efficient way of supporting inter-process communications. In addition, we have used our service creation environment to introduce the programmable MAC architecture on top of WaveLAN. The data link adapter module uses Berkeley’s PCAP packet filter interface to access the WaveLAN device driver. Our implementation choice results in some performance overhead associated with the collision avoidance mechanisms of the WaveLAN protocol. Collision avoidance mechanisms are redundant because the programmable MAC supports the centralized and coordinated access to the wireless medium. As future work we plan to deploy the programmable MAC architecture on top of software radio-based physical layer implementations.

The programmable MAC scheduler implements a constant bit rate scheduling discipline, while variable bit rate services are supported by renegotiating the declared rate of each profile. While a fair queuing algorithm should be optimally used to guarantee fairness, we utilized a Deficit Round Robin (DRR) scheduling discipline instead, to reduce the computational complexity of the scheduling algorithm. DRR provides fairness over long time scales and works well when packet sizes are small (packet sizes were at most 1500 bytes in our case). High performance scheduling is difficult when the scheduler runs in user space, because the scheduling algorithm is susceptible to context switching. Our user space implementation of programmable MAC can be easily deployed in wireless access points, while sacrificing some performance efficiency over firmware or in-kernel implementations. Detailed performance evaluation of the programmable MAC can be found in [8].

V. CONCLUSION

In this paper we have presented a framework for the programmability of mobile networks. Our approach and platform is novel in that it allows systems designers to architect their own handoff and MAC architectures and program them using a set of well defined APIs and objects.

ACKNOWLEDGEMENT

The work is supported in part by the National Science Foundation (NSF) under CAREER Award ANI-9876299 and with support from Intel and Nortel Networks under grants for programmable mobile networking. We would like to thank Raymond Liao, Gahng-Seop Ahn, Hao Zhuang, and Chung-Ming Chan for their contribution to this work.

REFERENCES

[1] N. D. Tripathi, J. H. Reed, and H. F. Vanlandingham, “Handoff in Cellular Systems”, *IEEE Personal Communications Magazine,* December 1998.

[2] M. C. Chan, and A. A. Lazar, “Designing a CORBA-based High Performance Open Programmable Signaling System for ATM Switching Platforms”, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 9, September 1999.

[3] A. T. Campbell, M. E. Kounavis, and R. R.-F. Liao, "Programmable Mobile Networks", *Computer Networks and ISDN Systems*, April 1999.

[4] A. G. Valko, J. Gomez, S. Kim, and A. T. Campbell, "On the Analysis of Cellular IP Access Networks", *Sixth IFIP International Workshop on Protocols for High Speed Networks* , Salem, 25-27 August 1999.

[5] J. Mitola, “Technical Challenges in the Globalization of Software Radio”, *IEEE Communications Magazine*, February 1999.

[6] V. Bose, M. Ismert, W. Wellborn, and J. Guttag, “Virtual Radios”, *IEEE Journal on Selected Areas in Communications*, Special Issue on Software Radios, 1998.

[7] V. Bose, D. Wetherall, and J. Guttag, “Next Century Challenges: Radioactive Networks”, *Fifth ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM’99),* Seattle, Washington, 1999.

[8] G. Bianchi, and A. T. Campbell, "A Programmable Medium Access Controller for Adaptive Quality of Service Control", *IEEE Journal of Selected Areas in Communications (JSAC)*, Special Issue on Intelligent Techniques in High Speed Networks, 2000 (to be published).

[9] R. R.-F. Liao, and A.T. Campbell, "On Programmable Universal Mobile Channels in a Cellular Internet", *4th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)* , Dallas, October, 1998.

[10] M. E. Kounavis, A. T. Campbell, G. Ito, and G. Bianchi, “Supporting Programmable Handoff in Mobile Networks”, *Sixth International Workshop on Mobile Multimedia Communications (MoMuC’99),* San Diego, CA, November 1999.

[11] The ICEBERG Project, University of California at Berkeley, iceberg.cs.berkeley.edu.

[12] D. J. Goodman, “Wireless Personal Communications Systems”, *Addison-Wesley Wireless Communications Series*, 1997.

[13] P. Bagwat, C. Perkins, and S. Tripathi, “Network Layer Mobility: an Architecture, and Survey”, *IEEE Personal Communications Magazine*, June 1996.

[14] J. Ioanidis, and G. Maguire, “The Design and Implementaton of a Mobile Internetworking Architecture”. *Winter USENIX*, San Diego, CA, January 1993.

[15] D. Raychaudhuri, “Wireless ATM Networks: Architecture, System, Design and Prototyping”, *IEEE Personal Communications Magazine*, August 1996.

[16] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, “HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks”, *Seventh International Conference on Network Protocols*, Toronto, Canada, 1999.

[17] S. Seshan, H. Balakrishnan, and R. H. Katz, “Handoffs in Cellular Networks: The Daedalus Implementation and Experience", *Kluwer International Journal on Wireless Communication Systems*, 1996.

[18] R. Caceres, and V. Padmanabhan, “Fast and scalable wireless handoffs, in support of mobile Internet audio*”, Mobile Networks and Applications*, 1998.

[19] The Object Management Group, www.omg.org

[20] A. Noerpel, and Y-B. Lin, “Handover Management for a PCS Network”, *IEEE Personal Communications Magazine*, December 1997.

[21] R. Pillai, W. Wang, L. Seng, B. Jose, H. Sha, R. Agrawal and M. Ranganath, “Implementation and Performance Evaluation of an Open ControlArchitecture for Wireless ATM Networks”, *Second International Conference on Open Architectures, and Network Programming (OPENARCH’99),* New York, 1999.

[22] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, “Improving TCP/IP performance over wireless networks”, *First ACM/IEEE Conference on Mobile Computing and Networking (MOBICOM’95*), Berkeley, CA, November 1995.

[23] R. R.-F. Liao, P. Bocheck, A. T. Campbell, and S.-F. Chang, “Utility-based network adaptation in MPEG-4 systems”, *Ninth International Workshop on Network and Operating System*

Support for Digital Audio and Video (NOSSDAV'99), Basking Ridge, New Jersey, 1999.

[24] A. T. Campbell, M. E. Kounavis, D Villela, J. Vicente, H. De Meer, K. Miki, and K. Kalaichelvan, "Spawning Networks", *IEEE Network*, August 1999.

[25] The Mobiware Toolkit, source code distribution: comet.columbia.edu/mobiware.

[26] A. Tennehouse, and D. Wetherall, "Towards an Active Network Architecture", *Multimedia Computing and Networking*, San Jose, CA, 1996.

[27] A. A. Lazar, "Programming Telecommunications Networks", *IEEE Network*, October 1997.

[28] M. E. Kounavis, and A. T. Campbell "The Metabus: Breaking the Monolith of the Software Bus", *Technical Report*, Center for Telecommunications Research, Columbia University, April 1999.

[29] Object Management Group, "Minimum CORBA", Joint Revised Submission, *OMG Document*, orbos/98-08-04 ed., August 1998.

[30] J. Der Merwe, S. Rooney, I. Leslie, and S. Crosby, "The Tempest, A Practical Framework for Network Programmability", *IEEE Network,* May 1998.