



# Delivering Live Multimedia Streams to Mobile Hosts in a Wireless Internet with Multiple Content Aggregators

CRISTIAN HESSELMAN

*Telematica Instituut, P.O. Box 589, 7500 AN Enschede, The Netherlands; University of Twente, Faculty of Electrical Engineering, Mathematics, and Computer Science, P.O. Box 217, 7500 AE Enschede, The Netherlands*

HENK EERTINK

*Telematica Instituut, P.O. Box 589, 7500 AN Enschede, The Netherlands*

ING WIDYA and ERIK HUIZER

*University of Twente, Faculty of Electrical Engineering, Mathematics, and Computer Science, P.O. Box 217, 7500 AE Enschede, The Netherlands*

**Abstract.** We consider the distribution of channels of live multimedia content (e.g., radio or TV broadcasts) via multiple content aggregators. In our work, an aggregator receives channels from content sources and redistributes them to a potentially large number of mobile hosts. Each aggregator can offer a channel in various configurations to cater for different wireless links, mobile hosts, and user preferences. As a result, a mobile host can generally choose from different configurations of the same channel offered by multiple alternative aggregators, which may be available through different interfaces (e.g., in a hotspot). A mobile host may need to handoff to another aggregator once it receives a channel. To prevent service disruption, a mobile host may for instance need to handoff to another aggregator when it leaves the subnets that make up its current aggregator's service area (e.g., a hotspot or a cellular network).

In this paper, we present the design of a system that enables (multi-homed) mobile hosts to seamlessly handoff from one aggregator to another so that they can continue to receive a channel wherever they go. We concentrate on handoffs between aggregators as a result of a mobile host crossing a subnet boundary. As part of the system, we discuss a lightweight application-level protocol that enables mobile hosts to select the aggregator that provides the 'best' configuration of a channel. The protocol comes into play when a mobile host begins to receive a channel and when it crosses a subnet boundary while receiving the channel. We show how our protocol can be implemented using the standard IETF session control and description protocols SIP and SDP. The implementation combines SIP and SDP's offer-answer model in a novel way.

**Keywords:** multimedia content distribution, wireless networking, mobility

## 1. Introduction

Wireless Internet enables mobile users to use an application-level service from any location at any time [7,12]. As suggested by [9,17,51] and others, such a service may be provided by multiple service providers. In general, mobile hosts can therefore choose from multiple alternative service providers to deliver the service. The service providers may be available through different interfaces (e.g., in a hotspot).

In a rich environment of multiple alternative service providers, mobile hosts first need to select the 'best' provider of a service. Once the mobile host receives the service, it may decide to handoff to another provider of the service for a number of reasons. To prevent service disruption, a mobile host may for instance switch to another provider when it roams out of the subnets that make up the service area of its current provider. A service area can for instance consist of a single hotspot cell, the subnets of an operator's cellular network, or the subnets of multiple (cellular) networks of multiple operators. Alternatively, the mobile host may switch to a provider that can offer the same service at a 'better' quality, for instance

because the mobile host has moved into a subnet with more available bandwidth.

We consider a specific application-level service, namely the distribution of channels of live multimedia content (e.g., radio or TV broadcasts). Our service providers are aggregators that receive channels from content sources (e.g., cnn.com) and redistribute them to a potentially large number of mobile hosts. Each aggregator can offer a channel in various configurations to cater for different wireless links, mobile hosts, and user preferences. As a result, a mobile host can generally choose from different configurations of the same channel offered by multiple alternative aggregators, which may be available through different interfaces (e.g., in a hotspot).

In this paper, we present the design of a system that enables (multi-homed) mobile hosts to seamlessly handoff from one aggregator to another so that they can continue to receive a channel wherever they go. We concentrate on handoffs between aggregators as a result of a mobile host crossing a subnet boundary and do not consider other events that may lead to handoffs (e.g., an aggregator taking a configuration out of service). As part of the system, we discuss a lightweight

application-level protocol that enables mobile hosts to select the aggregator that provides the ‘best’ configuration of a channel. The protocol comes into play when a mobile host begins to receive a channel and when it crosses a subnet boundary while receiving the channel.

The contributions of this paper consist of the design of the system and a realization of the application-level protocol using the Session Initiation Protocol (SIP) [32] and the Session Description Protocol (SDP) [31], which are standard IETF protocols for session control and description, respectively. The realization uses SIP with SDP’s offer-answer capabilities [31] in a multiparty fashion to determine which aggregator provides the ‘best’ configuration. This differs from [31] where SDP’s offer-answer capabilities are used between two parties to select a codec.

The rest of this paper is organized as follows. We first detail our approach in Section 2. In Section 3, we discuss the architecture of our system, followed by a description of our protocol in Section 4. We discuss related work in Section 5 and close with conclusions and future work in Section 6.

## 2. Approach

We use *roles* to model and understand the distribution and reception of live multimedia in an environment with multiple alternative service providers and hotspots. A role describes a number of typical functions that a *domain* playing that role fulfils. A domain can have multiple roles, possibly simultaneously. The assignment of roles to domains describes a specific business *constellation*. Examples of roles in hotspot environments are service providers, access providers, location owners, and infrastructure owners [44]. We formalize the business relationships that can exist between roles in *agreements* (e.g., a subscription of a user with a service provider). Roles and agreements are a common way to get a handle on a complex problem (e.g., [43,45]).

### 2.1. Roles

We distinguish two application-level roles for handling multimedia channels: content sources and content aggregators [16]. A *content source* (e.g., cnn.com) is the original source of one or more channels. Sources send channels of live multimedia content to mobile hosts directly or via *content aggregators*. A content aggregator collects channels from content sources and offers them to mobile hosts. Sources and aggregators primarily deal with forwarding multimedia channels in the form of streams of RTP packets [36] (as opposed to IP packets). However, some aggregators only act as a broker and do not participate in the actual forwarding of channels. Such aggregators merely enable users to find a channel and rely on sources to send channels to mobile hosts. Mobile hosts consume channels in that they depacketize and decompress the channels’ streams and render them to the user.

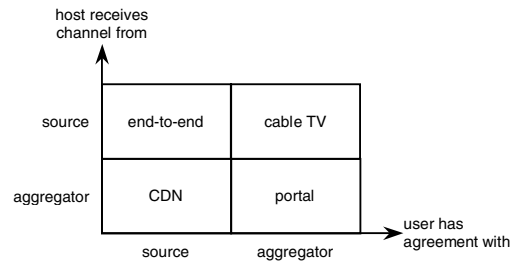


Figure 1. Business constellations.

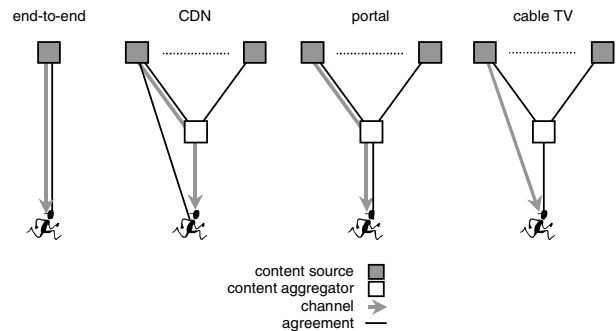


Figure 2. Examples of business constellations.

In our model, users need to subscribe to the channel delivery service. They can accomplish this by setting up an agreement with a source or with an aggregator. In general, their mobile hosts receive channels either directly from sources (in an end-to-end manner) or via an aggregator (in proxy-based manner). Figure 1 plots these possibilities. We assume that each user logs onto one mobile host.

Figure 1 shows that we can thus describe four important types of business constellations for distributing channels of live multimedia content, specifically end-to-end, Content Distribution Network (CDN), portal, and cable TV constellations. Figure 2 shows an example of each type of constellation.

The main difference between portal and cable TV constellations is that portal constellations redistribute channels. Redistribution means that a channel travels through two end-to-end paths: one from the source to the aggregator, and one from the aggregator to the mobile host. In cable TV constellations the aggregator is not involved in the actual transmission of channels. Instead, the aggregator only provides an Electronic Program Guide (EPG) that allows users to select the channels from contracted sources.

In a Content Distribution Network (CDN) constellation, a mobile host sends a request for a channel to a source (e.g., cnn.com). The source forwards the request to a CDN (e.g., akamai.com), which routes the request [2] to a media server that is ‘close’ to the user (e.g., http://nearest-server.akamai.com/CO231234), possibly in a resource-aware manner [51].

In this paper, we concentrate on portal constellations. We therefore only consider aggregators that actually participate in

forwarding channels to mobile hosts. The aggregators in our model are solely responsible for serving mobile hosts (see Section 2.2 for an explanation). We can thus also describe end-to-end constellations by co-locating the aggregator role with the role of a source (i.e., a domain serving mobile hosts in an end-to-end manner is both a source and an aggregator). In our model, end-to-end constellations are therefore considered a special case of portal constellations.

At the network-level, we distinguish two roles: access providers and backbone providers. An *access provider* provides first-hop IP connectivity to mobile hosts, aggregators, and sources (cf. [30]) at various bandwidth levels. Mobile hosts typically use wireless access providers. We do not require any special features (e.g., QoS assurances or mobility handling) from access providers. To limit the complexity of our model, we assume that access providers provide a best-effort service. Access providers are interconnected by backbone providers.

Using the above four roles, we decouple application-level responsibilities (aggregators and sources) from network-level responsibilities (access and backbone providers). In addition, we separate roles that serve mobile hosts (aggregators and access providers) from those that do not (sources and backbone providers). Aggregators can furthermore be responsible for application-level multicasting in an environment where the end-to-end availability of IP multicast is limited [5].

In this paper, we concentrate on aggregators whose service area is limited to a number of access providers. In our model, we accomplish this by setting up an agreement between an aggregator and a number of access providers [16]. The subnets of the access providers form the service area of the aggregator.

Figure 3 shows an example of such a constellation. We will use this example throughout this paper. It involves a user (Bob), three aggregators (stream-it.com, streams-to-go.com, and multimedia-forward.nl), and two access providers (connect-you.nl and hotspot.nl) that each operate a different type of network. The agreements between the aggregators and the access providers limit the service area of stream-it.com to the 802.11 network of hotspot.nl, the service area of media-forward.nl to the UMTS network of connect-

you.nl, and the service area of streams-to-go.com to both networks.

Bob uses a multi-homed mobile host with an 802.11 and a UMTS interface. He begins to receive channel CNN Radio at point A from stream-it.com over the 802.11 network. Between B and C, Bob's mobile host receives CNN Radio from media-forward.nl. At point C, it switches back to stream-it.com.

The example in figure 3 is the most elaborate type of constellation that our model can describe: it has all the agreements that our model covers (see next two sections) and involves domains that each play one role. Without introducing additional complexity, we can cover other constellations by considering fewer agreements (e.g., without agreements between aggregators and access providers) or by assigning the roles of a source and an aggregator to one domain.

We assume that bandwidth is only constrained at wireless access providers. We do therefore not consider backbone providers and do not show them in figure 3. Figure 3 also does not show the access providers to which the sources and the aggregators connect. We will explain the agreements that appear in figure 3 in the next sections.

### 2.2. Mobility

In our model, mobile hosts have to communicate with an aggregator (e.g., a channel directory, see Section 3.2) via subnets that are part of the aggregator's service area. For example, Bob's mobile host has to communicate with stream-it.com via the 802.11 network of hotspot.nl. Communication with an aggregator through a subnet that is not part of the aggregator's service area (e.g., Bob's host communicating with stream-it.com via the UMTS network) may fail, for instance because the aggregator only accepts traffic from subnets (access providers) that are part of the same corporate infrastructure (a similar firewalled setting is discussed in [17]).

As a result, mobile hosts need to be able to route application-level packets to specific aggregators via specific network interfaces. This makes mobility handling (i.e., dealing with the changing IP addresses of mobile hosts) at the application-level more suitable than transparent network-level mobility handling (in particular using Mobile IP [40]), in which traffic can typically only be sent through one interface.

The consequence of handling mobility at the application-level is that aggregators might need to keep track of the IP addresses of mobile hosts, for instance to send events to them (e.g., when a particular configuration has been taken out of service).

In our model, mobile users typically set up an agreement with one aggregator (e.g., Bob with media-forward.nl) to be able to receive channels. Such a *home aggregator* sets up *application-level roaming agreements* [16] with other aggregators to enable its users to receive streams from these *foreign aggregators* as well. For example, Bob's home aggregator media-forward.nl has an agreement with foreign aggregator stream-you.com so that Bob can receive channels from stream-you.com in the hotspot.

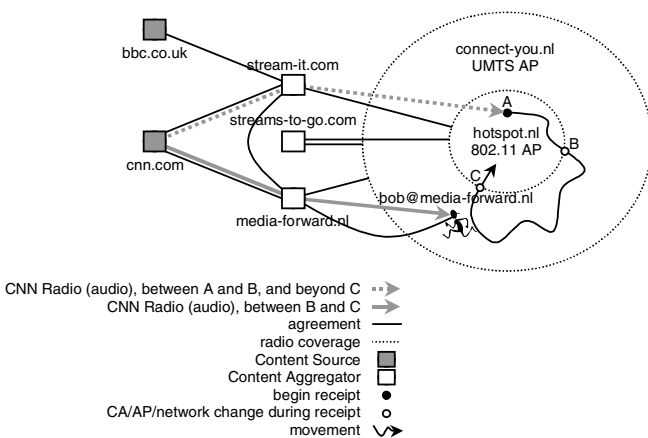


Figure 3. Example with multiple alternative aggregators.

Application-level roaming agreements are similar to network-level roaming agreements (e.g., [10,21]), but they only address application-level issues (see Section 2.3). A roaming agreement does not include user-specific information.

### 2.3. Configurations

Sources and aggregators need to be able to serve different types of (mobile) hosts that connect to the Internet through different types of (wireless) networks. In addition, they need to be able to deal with different user requirements regarding aspects such as cost and perceived perceptual quality. To accomplish this, sources and aggregators support a number of different *configurations* of a channel (e.g., [28,51]). Mobile hosts can potentially use these configurations to receive a channel from an aggregator, while an aggregator can use them to receive a channel from a source.

We define a configuration as a bundle of streams of well-formatted RTP packets [36] of a certain bandwidth (e.g., a 64 kbps MP3 audio configuration, a 32 kbps G722.1 audio configuration, a 24 kbps G722.1 audio configuration, and so on). Some configurations are suitable for wireless links and mobile hosts ('mobile friendly' configurations), while others are not (e.g., configurations for HDTV). Configurations that are suitable for wireless links and mobile hosts typically provide in a lower perceptual quality level.

In our model, sources and aggregators have their own set of configurations at which they can send out a channel. The set can be relatively small, which keeps sources and aggregator relatively simple and thus helps to increase scalability. This approach is furthermore more suitable for channels that are being multicast (e.g., [23,50]). Scalability would be harder to achieve if aggregators would generate streams that were, say, fine-tuned to the currently available bandwidth of individual mobile hosts.

The function of transmitting channels at 'mobile friendly' configurations can be performed by a source, by an aggregator, or by both. If a source is 'mobile friendly', then aggregators can simply reuse the source's configurations. If it is not, then an aggregator might manipulate the 'mobile unfriendly' configurations of a source (e.g., configurations meant for HDTVs) to create 'mobile friendly' ones, for instance by transcoding high-bandwidth streams to a low bandwidth format (e.g., [1,51]). Such an aggregator provides value-added services by increasing the available set of configurations.

In our model, the agreement between a user and his home aggregator lists the aggregator's configurations at which the user can receive channels [16]. Bob's agreement with media-forward.nl could for instance specify that Bob can use the three audio configurations of media-forward.nl shown in figure 4. Since a foreign aggregator may use another set of configurations, application-level roaming agreements need to state which configurations a user has access to at a foreign aggregator. The roaming agreement between media-forward.nl and stream-it.com (see figure 3) could for instance state that

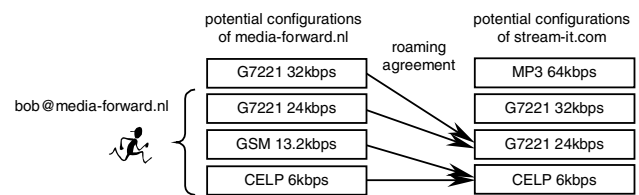


Figure 4. Roaming agreement.

users of media-forward.nl can only use two of stream-it.com's audio configurations (see figure 4). The mapping from configurations of a home to those of a foreign aggregator may be based on factors such as bandwidth, cost, or perceptual quality. The bottom line is that a mobile host can generally receive the same channel from multiple alternative aggregators, but in different configurations.

The configuration information in the agreements is typically combined with a pricing model, but that topic is outside the scope of our work.

To be able to serve a large number of users, sources and aggregators use a pool of media servers to send out channels (cf. the clusters of [1] and [4]). When a host switches to another aggregator, it will also have to receive the streams of the target configuration from a media server of the target aggregator. A switch between aggregators may involve transferring application-level context information (e.g., the state of a predictive encoder) between media servers [33]. In addition, it may also involve a policy-controlled handoff (e.g., [15,41]) between subnets (e.g., between the UMTS and 802.11b subnets of figure 3). Application-level context transfers and handoff policies are however outside the scope of this paper.

Our model also covers agreements between other roles (e.g., between sources and aggregators to describe if and how aggregators are allowed to manipulate streams) [16], but we will not discuss them in this paper.

## 3. Architecture

Figure 5 shows the architecture of our system. It enables mobile hosts to receive multimedia channels wherever they go in an environment with multiple alternative aggregators (see Section 2).

The system's design is based on three requirements:

- The system should enable mobile hosts to quickly switch between aggregators;
- It should minimize the amount of control information transferred to mobile host; and
- It should be scalable. In our work, scalable means that the amount of inter-aggregator traffic and the authentication load on home aggregators should be minimized (cf. [30]). Following Internet design principles [6,34], it also means that aggregators should maintain the minimum possible amount of state.

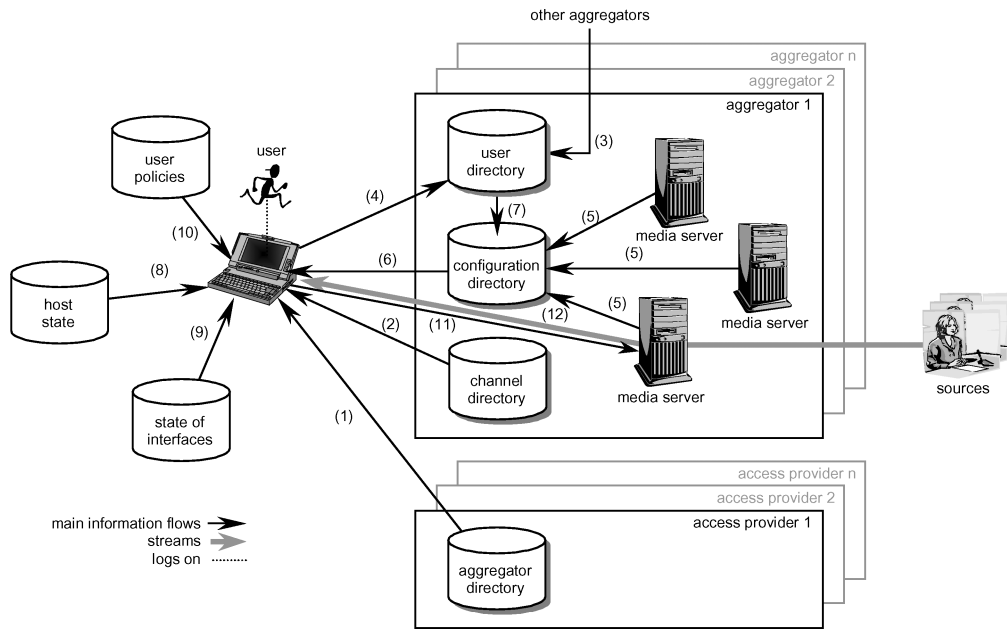


Figure 5. System architecture.

In this section, we discuss the architecture of figure 5 in terms of the responsibilities of access providers (Section 3.1), aggregators (Section 3.2), and mobile hosts (Section 3.3). In Section 3.4, we provide a high-level description of the interactions between mobile hosts and access providers and between mobile hosts and aggregators. The numbers in the text (e.g., (10)) correspond to the numbers in figure 5.

### 3.1. Access providers

An access provider maintains an *aggregator directory* that contains pointers (typically URIs) to the aggregators the mobile host can reach through the access provider. Mobile hosts can for instance retrieve the pointers (cf., [11,30]) through a DHCP option (e.g., [35] for SIP URIs) when the mobile host receives an IP address from the access provider (1).

Access providers do not maintain an aggregator directory if they do not have an agreement with aggregators. In this case, mobile hosts can only use aggregators known to them (e.g., those set when the user subscribed to the channel delivery service) or they will need to revert to other discovery techniques to find out which aggregators they can use (e.g., using a well-known multicast group to which aggregators listen [11]).

### 3.2. Aggregators

Besides media servers (see Section 2.3), an aggregator maintains a channel directory, a configuration directory, and a user directory.

#### 3.2.1. Channel directory

A *channel directory* contains the names of the channels that an aggregator can offer. A channel directory is publicly available, which means that users do not need to be authenticated to access it. Mobile hosts that access the directory (2) do thus not put a load on their users' home aggregators. This helps to increase scalability when a mobile host queries the directory but then decides not to use the aggregator (e.g., because the user selects a channel that the aggregator does not offer). The channel directory can for instance be based on a SAP cache [14].

#### 3.2.2. User directory

An entry in the user directory consists of a user identity and the set of configurations at which that user can receive channels from the local aggregator. Each user that has an agreement with the aggregator (see Section 2.3) has an entry in the user directory, which is created when the user subscribes to the aggregator's services. The allowed configurations are those that are specified in the user's agreement with the aggregator. The user directory is also responsible for authenticating these users. For foreign users, the user directory delegates authentication to the user directory of the user's home aggregator.

When the user directory has successfully authenticated a user, it places the identity of the user and the local configurations he is allowed to use in an authentication cache (part of the user directory, not shown in figure 5). The locally allowed configurations for a foreign user are based on the roaming agreement with the user's home aggregator and the configurations he can access there (cf. the example of figure 4). Aggregators retrieve the latter type of information from home aggregators (3), typically as part of the authentication process. Inter-aggregator communication typically takes place through

a AAA protocol like Diameter [3]. AAA protocols are however outside the scope of this paper.

When a user has an entry in the authentication cache, the user directory considers a user authenticated and refrains from reauthenticating him (at his home aggregator). This reduces the load on home aggregators, which aids the scalability of the system. It also reduces the interaction delay between the mobile host and the configuration directory, which enables mobile hosts to switch between aggregators more quickly.

The entries in the authentication cache are softstate, which means that a mobile host must refresh its user's entry regularly (4). The refresh interval is aggregator-specific. To prevent malicious users from imitating authenticated users, the user directory hands out admission tokens that the mobile host must use in further communications with the aggregator (e.g., when querying the configuration directory, see below). The user directory deletes entries from the cache that have not been refreshed in time.

### 3.2.3. Configuration directory

An entry in a configuration directory consists of a channel name (see Channel Directory) and a number of configurations at which the media servers can currently deliver the channel. The set of configurations is based on the available resources on the media servers (5), which may change dynamically. Each configuration comes with a small number of URIs (e.g., SIP and RTSP URIs) that point to the media servers of the aggregator that can offer the channel at that configuration.

Mobile hosts query the configuration directory to check which configurations of a channel are available to the user (6). To accomplish this, a configuration directory checks the user directory (7) and returns those available configurations of a channel that a user is allowed to use and that can be handled by the media servers. A mobile host may inform the channel directory that it is interested in a limited number of configurations, for instance because the mobile host cannot handle stereo configurations. Since a mobile host generally does not know which configurations it can receive from a foreign aggregator, it specifies its interest in terms of the configurations the user can receive at home. A channel directory does not serve users that the user directory cannot authenticate (e.g., because there is no roaming agreement with their home aggregator).

The configuration directory also conveys the media server URIs of the configurations that are available to a user (6). This enables mobile hosts to directly connect to the media server that provides the most suitable configuration. As a result, media servers need to be able to interact with a AAA entity in the back-end of the aggregator (e.g., to signal that a user has begun to receive a channel). An alternative configuration directory would be one that does convey URIs to mobile hosts. In this case, the configuration directory acts as a proxy for the aggregator's media servers. Mobile hosts would have to inform the configuration directory of the configuration they want to use, after which the configuration directory would select an appropriate media server (URI) and connect the mobile host to it. The downside of this approach is that it requires more in-

telligent configuration directories. The advantage is that such configuration directories can act as signaling gateways (e.g., translating SIP messages to RTSP messages) and that this will enable aggregators to serve a more heterogeneous population of mobile hosts with a homogeneous pool of servers. This is also the disadvantage of the direct signaling paths between mobile hosts and media servers: it requires aggregators to operate a heterogeneous pool of media servers (e.g., with RTSP, SIP, Real, and WindowsMedia servers) to be able to serve different types of mobile hosts.

Aggregators use centralized configuration directories. The disadvantage of using a central configuration directory per aggregator is that the directory forms a single point of failure and that it can potentially become an aggregator's bottleneck. An alternative approach is to distribute the configuration directory of an aggregator across its media servers. In this case, each media server keeps track of the configurations it can offer (cf. [1]) and mobile hosts communicate with the media servers to determine at which configurations they can receive a channel from the aggregator. However, this approach is not very efficient in a unicast environment. Mobile hosts might for instance need to consult multiple media servers, which increases the number of round trips. In addition, different media servers may report availability of the same configuration, which increases bandwidth usage. Each media server might furthermore request the home aggregator to authenticate the user, which would increase the load on home aggregators. These problems can be alleviated in a multicast environment where the media servers of an aggregator and the mobile host share a multicast group. A technique like multicast damping [1] can for instance suppress responses reporting availability of the same configuration. However, IP multicast is not very widespread at this point [5]. In addition, the user-specific lists of configurations that media servers would have to produce do not match the one-to-many character of IP multicast very well.

### 3.3. Mobile hosts

A mobile host keeps track of its own state (e.g., which codecs it supports, its battery level) (8) and the state of its network interfaces (e.g., if they are up or down and the available bandwidth levels) (9) to determine if it has the resources to actually receive a channel at a certain configuration. A configuration can for instance be useless because the bandwidth on an interface is insufficient to carry the configuration's streams (e.g., the UMTS interface of Bob's mobile host may not have enough bandwidth to receive a TV broadcast at several hundreds of kbps). A configuration can also be useless because the mobile host cannot or chooses not to decode the streams (e.g., when battery power is low).

A mobile host uses the policies of the user (10) to determine which of the useful configurations is the 'best' one. One user can for instance consider the cheapest configuration of a channel the best one, while another considers the one with the highest perceptual quality the best one. Another user-level aspect that can play a role in the definition of 'best' is power consumption [47].

The user's policies enable a mobile host to automatically select the 'best' configuration without user intervention. This hides the complexity of the infrastructure in terms of aggregators and access providers from the user, thus facilitating user-friendly roaming in a heterogeneous environment [19].

### 3.4. High-level behavior

We distinguish two sequential phases with respect to the interactions between mobile hosts and the infrastructure (aggregators and access providers): an initiation phase and a roaming phase. The *initiation* phase starts when a user selects a channel. The *roaming* phase begins when the user begins to receive that channel and ends when the user decides that he no longer wants to receive it. During roaming, our design focuses on enabling the system to operate quickly (e.g., during a handoff from one aggregator to another). During initiation, the design focuses on minimizing bandwidth consumption and the load on home aggregators.

#### 3.4.1. Initiation

A mobile host first discovers the aggregators that are available on each of the networks it can connect to (1). It communicates with an aggregator via the interface on which it has discovered the aggregator (see Section 2.2). For example, Bob's mobile host communicates with aggregator stream-it.com via its 802.11 interface and with media-forward.nl via its UMTS interface.

Next, the mobile host accesses the aggregators' channel directories (2) and presents the names in those directories to the user. The public nature of channel directories ensures that aggregators that do not support the channel the user selects do not put a load on home aggregators. The protocol between the mobile host and the channel directory can for instance be HTTP. It is however outside the scope of our work.

When the user has selected a channel, the mobile host consults the configuration directories (6) of the aggregators (home or foreign) that support the channel to determine at which configurations the user can receive the channel from which aggregator. The information from the channel directories ensures that the mobile host only queries configuration directories of aggregators that can offer the channel the user has selected. The advantage is that less information needs to be sent to the mobile host, which reduces bandwidth usage. The downside is that the two rounds add delay, but this is less important during initiation.

To speed up operations, the mobile host should preferably consult the configuration directories of multiple aggregators simultaneously. For example, at point A Bob's mobile host should at the same time consult the configuration directories of stream-it.com (via the host's 802.11 interface) and media-forward.nl (via the host's UMTS interface).

Based on the configurations provided by the configuration directories, the mobile host determines which of the available configurations it can use (8, 9) and selects the 'best' one (10). It then uses a URI and the associated session control protocol

(e.g., SIP or RTSP) to connect to a media server (11) and receive the corresponding configuration of the channel (12).

#### 3.4.2. Roaming

In this paper, we only consider handoffs between aggregators as a result of a mobile host moving into or out of a subnet (e.g., at points B and C in figure 3). Mobile hosts detect that they have moved in or out of a subnet when the IP address of one of their interfaces changes. An example of another reason for a mobile host to switch to another aggregator is that its current aggregator takes a configuration out of service.

There are several reasons to handoff to another aggregator when a mobile host crosses a subnet boundary. For example, a new aggregator can appear (i.e., the host roams into a service area) that provides a 'better' configuration of the channel that the mobile host is receiving. Another possibility is that the mobile host's current aggregator disappears because it moves into a subnet where this aggregator is unavailable (i.e., the host roams out of a service area). When no aggregators (dis)appear, the mobile host might for instance handoff to another aggregator because the available bandwidth on the new network can accommodate a configuration of another aggregator that is 'better' than the current one.

For these reasons, a mobile host consults the configuration directories (6) of the aggregators it can reach (old and new ones) when it crosses a subnet boundary. The mobile host should preferably access the directories simultaneously.

Mobile hosts do not query the channel directories of aggregators that have just appeared to check if they provide the channel the mobile host is receiving. This saves one round-trip and thus speeds up operations. To reduce the load on home aggregators, configuration directories do not consult their user directories when they detect that a mobile host is looking for a channel the aggregator does not support. To further reduce bandwidth usage, the mobile host could keep track of such information so that it will not query the aggregator for the channel again at a later stage.

If the new 'best' configuration (8, 9, 10) belongs to another aggregator, the mobile host will connect to one of the media servers of the target aggregator (11) using the URIs it received from that aggregator's configuration directory. As a result, the mobile host will switch to another set of streams (configuration) (12), but it remains connected to the same logical channel. The new 'best' configuration may also belong to the aggregator from which the mobile host was receiving the stream. In this case, the mobile host may need to switch to another media server of that same aggregator. Alternatively, the target configuration can be provided by the same media server. This does not require any server switching at all, but it does require mobile hosts to inform the media server of the mobile host's new IP address. In all cases, the target configuration may provide a different quality level than the old configuration.

During handoff, the mobile host will attempt to disconnect from the old aggregator. We assume that an aggregator can detect a disconnected mobile host (e.g., using RTCP [36])

that fail to disconnect in an orderly manner (e.g., because their network connection went down suddenly). The media server can then clean up the resources the mobile host was using (e.g., transcoders, if any) and inform a AAA entity in the aggregator (not shown in figure 5) that the user is no longer receiving the channel.

The mobile host continually refreshes the authentication softstate (4) in the user directories of the aggregators it can use (i.e., the aggregators that support the channel and that the host has access to). The refresh interval is aggregator-specific.

Each mobile host that receives a channel is thus involved in  $N$  ‘refresh sessions’ (where  $N \geq 1$  is the number of aggregators the mobile host can use) and one multimedia session through which it receives the ‘best’ configuration of the channel.

#### 4. Protocol

In this section, we discuss a signaling protocol that realizes interfaces (4) and (6) in figure 5. We first explain why we realized the protocol with SIP and SDP (Section 4.1) and then discuss the protocol’s behavior during initiation (Section 4.2) and when the mobile host crosses a subnet boundary (Section 4.3). We also discuss the implementation of the protocol (Section 4.4).

##### 4.1. SIP and SDP

We realize both interfaces with the Session Initiation Protocol (SIP) [32]. The four main reasons are that: (1) SIP is typically used to convey descriptions of multimedia sessions (i.e., the streams of a configuration); (2) SIP provides hooks for shared secret user authentication; (3) SIP can be used to refresh softstate [8]; and (4) one of the main purposes of SIP is to set up multimedia sessions. The latter enables us to use SIP for the interface between mobile hosts and media servers as well, which reduces the number of protocols in the system. In turn, this reduces the complexity of the system. We do however stress that aggregators can also use other session control protocols on their media servers such as RTSP [37] or WindowsMedia. As a result, mobile hosts may also need to handoff between different types of media servers, for instance from a SIP media server of one aggregator to an RTSP media server of another aggregator.

Other reasons for using SIP is that it is reasonably bandwidth efficient compared to the relatively high bandwidth levels that streaming applications usually require and that it uses textual messages, which are easier to extend, process, and debug than binary encoded messages. SIP can furthermore be run on top of UDP, which is a natural choice for a signaling protocol that does not require long-lived connections. UDP messages can furthermore be sent right away without having to wait for a TCP connection to be established, which benefits the speed of operation. The disadvantage of using UDP is that it limits the number of configuration descriptions that can be transferred to the maximum size of a UDP packet. In this paper, we assume that the configuration descriptions that

```
s=CNN Radio
...
m=audio 0 RTP/AVP 96 97 98
a=rtpmap:96 G7221/16000
a=fmtp:96 bitrate=32000
a=fmtp:96 bitrate=24000
a=rtpmap:97 GSM/8000
a=fmtp:97 bitrate=13200
a=rtpmap:98 MP4A/LATM/8000
a=fmtp:98 bitrate=6000
```

configuration

Figure 6. Configuration descriptions in SDP.

a mobile host receives from an aggregator fit into one UDP packet.

SIP can also be used to extend our system beyond what we have presented in this paper. SIP enables aggregators to push events (e.g., a signal that reports a configuration going out of service) to interested hosts [29] and will enable us to use public key authentication [27]. Finally, we consider SIP to be a generic application-level signaling protocol that will eventually become ubiquitously available. SIP has for instance been adopted as the signaling protocol for UMTS multimedia sessions [49].

The disadvantage of using SIP is that we need to use another protocol for the interaction between the media servers and the configuration directory (protocols such as SLP [11] cover both interfaces in one protocol). However, from a roaming perspective it is irrelevant which protocols aggregators use between their configuration directories and their media servers.

To keep as close to the existing state of the art as possible, we use the Session Description Protocol (SDP) [13] to describe configurations. In this paper, we do not make any assumptions on how aggregators describe configurations internally or in roaming agreements. Another candidate for describing configurations is SDPng [20], but this is not a standard yet.

Figure 6 illustrates how we describe the configurations of a channel (CNN Radio) in SDP.

The  $s =$  line contains the name of the channel. Each combination of an rtpmap and an fmtp line together define a configuration. We note that the bitrate parameters in the fmtp lines in the above example are for illustrative purposes only. In reality, these parameters are codec-specific. Also note that the payload types in the media line ( $m =$ ) merely form alternatives and do not express an ordering as is normally the case in SDP. For simplicity, we omitted all other SDP lines other than  $s =$  and  $m =$ .

##### 4.2. Initiation

After the user has selected a channel, a mobile host queries the aggregators that support the channel by sending an INVITE message to them (query requests). Aggregators run our protocol on top of a SIP user agent server because aggregators are SIP end points. Similarly, mobile hosts run our protocol on top of a SIP user agent client.

Figure 7 shows the behavior of the protocol at point A of figure 3 immediately after the user has selected a channel.



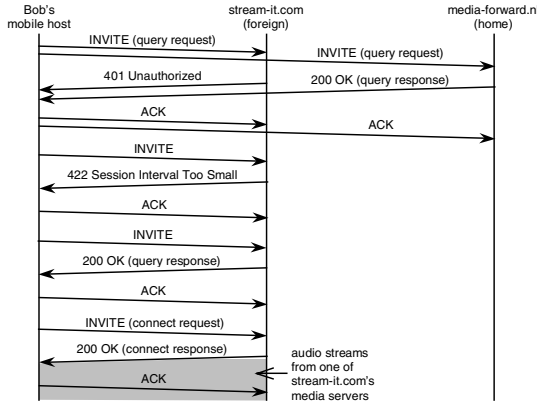


Figure 7. Example initiation behavior at point A.

In this example, Bob's mobile host transmits two INVITEs simultaneously: one to media-forward.nl via its UMTS interface and one to stream-it.com via its 802.11 interface.

Each INVITE contains the user's ID in the From header, the user's credentials in the Authorization header [32], and the mobile host's proposal for a refresh interval in the Session-Expires header [8]. An INVITE also contains SDP with the name of the channel the user wants to receive in the *s* = field. The SDP may contain configuration descriptions if the mobile host is interested in a limited number of configurations (e.g., because it does not have the capabilities to deal with configurations that provide stereo audio) [25]. The SDP is 'inactive', which means that the last line in the SDP payload is an *a* = inactive line. The *a* = inactive line informs an aggregator that the SDP describes capabilities and that it should not begin to stream [31]. Unlike [31], we use this mechanism in a multi-party fashion because a mobile host generally queries multiple aggregators.

An aggregator returns a 200 OK (query response) if its configuration directory admits the user and accepts the proposed refresh interval (see the interaction with media-forward.nl in figure 7). A 200 OK establishes a SIP signaling association (a dialog) between the mobile host and the aggregator. A 200 OK contains the final refresh interval in the Min-SE header [8] and an admission token in the Admission-Authorization-Token header (cf. the P-Media-Authorization-Token of [22]). The 200 OK's SDP is also 'inactive' and contains the name of the channel, the configurations from the aggregator at which the user can receive the channel, and URIs to media servers. Figure 8 shows an example.

The request-response procedure is similar to that of SDP's offer-answer model [31]. Unlike [31] the configurations in a response may differ from those in the corresponding request (if any). This is because a foreign aggregator might support a different set of configurations than a user's home aggregator.

The protocol interaction with stream-it.com in figure 7 illustrates that the mobile host may need to resubmit an INVITE a number of times before it receives a 200 OK. In this case, querying a configuration directory results in multiple SIP transactions. In the example, the first INVITE to stream-it.com

```
s=CNN Radio
...
m=audio 0 RTP/AVP 96 98
a=rtpmap:96 G7221/16000
a=fmtp:96 bitrate=24000
a=sip:server1.stream-it.com
a=rtsp://server2.stream-it.com
a=rtpmap:98 MP4A/LATM/8000
a=fmtp:98 bitrate=6000
a=sip:server1.stream-it.com
a=inactive
```

configuration plus URIs

mark SDP as 'inactive'

Figure 8. Available configurations.

(the query request) does not contain the user's credentials. As a result, the aggregator responds with a 401 Unauthorized [32] asking the mobile host to resubmit the INVITE. The second INVITE in figure 7 contains the user's credentials, but the mobile host proposes a refresh interval that stream-it.com considers too small. Stream-it.com therefore returns a 422 Session Interval Too Small to the mobile host containing its minimum acceptable refresh interval in the 422's Min-SE header [8]. The third INVITE contains all the necessary information and results in a 200 OK (query response). Observe that the response to the second INVITE may have been a 403 Forbidden if the aggregator could not authenticate the user (e.g., because it has no roaming agreement with the user's home aggregator). This would have ended the query as well, but with a negative outcome (negative query response). Notice that figure 7 does not show the AAA interactions between stream-it.com and media-forward.nl to authenticate Bob.

In the example, the best configuration of CNN Radio at point A is provided by stream-it.com. In this example, we assume that the corresponding streams are available from a SIP media server. To receive the streams, the mobile host therefore sends an INVITE to the media server (connect request). The INVITE contains the Admission-Authorization-Token so that the media server can verify that Bob has been authenticated. The SDP payload contains the name of the channel (CNN Radio) and a description of the selected 'best' configuration of the channel. The SDP must not contain an *a* = inactive line.

A 200 OK from the media server (connect response) establishes a dialog between the mobile host and the media server and indicates that the media server is transmitting the channel to the mobile host.

### 4.3. Roaming

When a mobile host moves into or out of a subnet, it sends an INVITE (query request) to any new aggregators that it discovered on its interfaces and a re-INVITE (status request) to aggregators that it already knew. Re-INVITEs are sent over an existing dialog.

Figure 9 shows the behavior of our protocol at point C of figure 3. In this example, Bob's mobile host rediscovers stream-it.com and therefore transmits an INVITE to this aggregator via its 802.11 interface. At the same time, it also sends a re-INVITE to media-forward.nl via its UMTS interface.

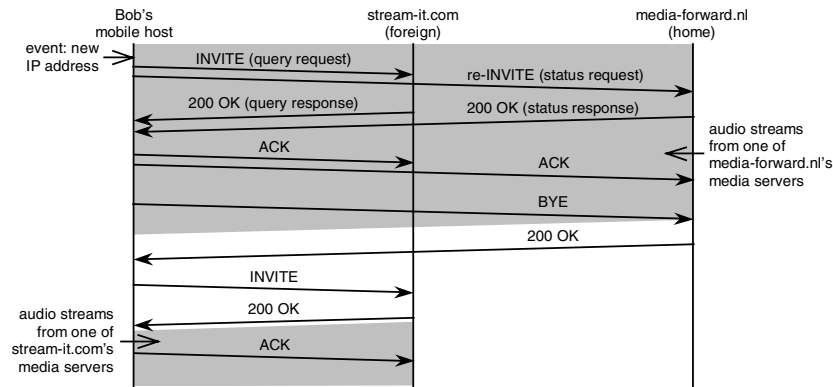


Figure 9. Example roaming behavior at point C.

Re-INVITES contain the same type of information as INVITES, except that a user's credentials are replaced by an admission token. A re-INVITE also contains 'inactive' SDP. Mobile hosts also transmit re-INVITES at regular intervals (refresh requests) to refresh the authentication state in an aggregator's user directory. For simplicity, we have omitted the regular re-INVITES from figure 9.

Aggregators usually respond to a re-INVITE with a 200 OK (status or refresh response). The 200 OK contains the same sort of information as a 200 OK to an INVITE (query response, cf. figure 8). The 200 OKs that aggregators transmit in response to regular re-INVITES also enable mobile hosts to detect changes in the availability of configurations. A more efficient and timely approach would be to have configuration directories use an announcement protocol [29] to signal such events. The use of such a protocol in our system is an item of future work.

The reaction of an aggregator to an INVITE (query request) are the same as during initiation (see figure 7), which means that a query can again result in multiple SIP transactions. During roaming, an aggregator also needs to be able to indicate that a channel does not exist since mobile hosts do not check an aggregator's channel directory first. An aggregator uses a 404 Not Found to signal such an event.

In the example, the mobile host considers one of stream-it.com's configurations 'best' and therefore connects to one of this aggregator's media servers.

#### 4.4. Implementation

We have implemented the protocol in C on top of open SIP [26]. At this point, the protocol runs in a fixed environment consisting of three subnets, one server machine (Linux), and a laptop (Linux) equipped with two Ethernet interfaces. Each of the laptop's interfaces connects to a different subnet, which in turn connect to a third subnet on which the server machine resides. The software setup is such that we emulate the examples of figures 7 and 9. The server machine 'hosts' two aggregator domains in the form of two processes that run the aggregator-side software of our system (protocol, configuration directory, and user directory).

The software on the laptop and on the server machine consists of a SIP user agent with our protocol running on top of it. The laptop furthermore runs a control component on top of our protocol that invokes our protocol in reaction to an event (e.g., an address change). The control component also collects the responses from the configuration directories, selects the 'best' configuration, and initiates a connection.

One of the problems we encountered in developing our protocol is that our version of Linux (Redhat) did not support end-host routing (i.e., enabling an application-level program to send and receive packets over a specific interface). If our protocol attempted to send out two INVITES (one to each process on the server), Linux sent both messages out through one interface instead of one INVITE through each interface. To bypass this problem, we created an additional routing table on the laptop (using the tool `ip`). We then associated the standard routing table with one of the laptop's interfaces and the additional table with the other, and set the default routes of both tables to the gateway of their local subnet. We had no problems with end-host routing on Windows 2000.

We are currently extending our implementation with two additional software components. One is a mobility manager that keeps track of the status of the interfaces of a mobile host. It can for instance report events such as the assignment of a new IP address to interested applications. Such an event is one of the triggers of our protocol (cf. the example in figure 9). The other component we are adding is a modified version of VIC [46] that can deal with address changes of mobile hosts.

## 5. Related work

Several papers in the literature consider hosts that hand off from one server machine to another [9,17,33,38,42,51]. They do however not investigate handoffs between the servers of access-controlled domains (aggregators) that operate in an environment based on agreements (see Section 3).

Dutta et al. [9], Xu and Nahrstedt [51], Hsieh et al. [17] and Roy et al. [33] consider handoffs between server machines as a result of mobility. The MarconiNet system of Dutta et al. [9] consists of affiliate domains that receive streams from radio

or TV broadcasters and forward them to mobile hosts using a set of media servers. Each media server is responsible for one subnet and delivers streams to mobile hosts through locally scoped IP multicast groups. Mobile hosts switch from one server to another by switching to another multicast group. In [9], the authors discuss a signaling protocol that realizes such handoffs in combination with DiffServ-based QoS control. Like our system, MarconiNet runs in a managed environment with agreements between affiliates and radio or TV broadcasters. Their affiliates are similar to our aggregators. One of the main differences with our work is that Marconinet only considers handoffs between servers of the same affiliate and that they do not explicitly distinguish application and network-level roles. As a result, they miss most of the agreements of our model (e.g., application-level roaming agreements). Other differences are that we explicitly consider multi-homed mobile hosts and that their approach does not include the notion of a configuration. Our system therefore differs considerably from theirs. We do however not cover security issues, which Dutta et al. do.

Like in our work, Xu et al. [51] also consider the delivery of multimedia services in multiple configurations (they call this feature service polymorphism). In their ServiPoly system, clients request a multimedia service from a server. Clients include their available resources in the request (e.g., battery power and available bandwidth), which servers use to select a service configuration. Servers then deliver such a service configuration directly or through an intermediary proxy server. Proxy servers may be part of different domains and are therefore comparable to our media servers. While it is not the main focus of their paper, Xu et al. suggest that mobile hosts can handoff to a proxy server of another intermediary domain by resubmitting their request for a multimedia service. The original server (or a replica thereof) would then select a new configuration and a new proxy server for the client. This is similar to our system, except that their servers possess most of the intelligence that we put on clients (cf. the CDN constellation of figure 2). ServiPloy furthermore delivers configurations tailored to individual clients, which is probably less scalable for live multimedia channels with a large number of receivers.

Roy et al. [33] discuss a system that enables mobile hosts to seamlessly switch between two transcoding servers. They accomplish this by migrating the state of a transcoding session (e.g., information to reconstruct the next frame from the source at the target transcoder) from one server to another. The authors discuss three types of inter-server protocols that can be used for this purpose. We consider this work complementary to ours.

Hsieh et al. [17] discuss a receiver-oriented TCP-clone that is able to hand a TCP connection off from one server to another. While they also consider multi-homed mobile hosts, their work is at the transport layer, which makes it quite different from ours.

Snoeren et al. [38] and Sultan et al. [42] consider handoffs between servers for other reasons than mobility, for instance to increase the availability of a service (e.g., handoff to another

server when the current server gets overloaded). They transfer TCP state (e.g., the sequence number of the last successfully acknowledged data segment) and some application-level state to resume a TCP connection (e.g., for HTTP applications) at the target server at exactly the same place where it left off at the original server. A similarity with our work is that the clients in [38] are responsible for selecting a target server just like our mobile hosts are responsible for selecting a target configuration, aggregator, and media server.

Our work can also be considered from an Internet Media Guide (IMG) perspective [25]. The MMUSIC group of the IETF is currently looking into a framework for the distribution of IMGs to a potentially large number of (mobile) users. They define an IMG as a structured set of descriptions of multimedia sessions (e.g., in SDP) and distinguish IMG senders, IMG transceivers, and IMG receivers. An IMG transceiver receives IMGs from senders, optionally modifies the IMGs, and forward them to IMG receivers. In our work, a multimedia session is a multimedia channel being transmitted at a certain configuration. Sources are IMG senders, aggregators are IMG transceivers, and mobile hosts are IMG receivers. Aggregators can be considered IMG transceivers because they bundle channels from sources and because they can offer channels to mobile users at other configurations than the sources from which they receive the channels (see Section 2.3). The work we presented in this paper addresses at least three of the requirements in [25]. First, Nomura et al. [25] requires that IMG receivers are allowed to communicate with multiple IMG senders simultaneously. Our (multi-homed) mobile hosts communicate with multiple IMG senders because aggregators are IMG transceivers and IMG transceivers are also IMG senders. A second requirement is that it must be possible to deliver customized IMGs to receivers. The configuration directories of our aggregators do exactly this (see Section 3.2). Third, Nomura et al. [25] states that IMGs may need to be protected at different levels. In our work, the channel names of IMGs (in the channel directories of aggregators) are publicly available, while the configurations at which a user can receive these channels are protected (in the configuration directories).

## 6. Conclusions and future work

We presented a role-based model of a wireless Internet in which multiple alternative content aggregators offer live multimedia channels in various configurations, possibly in specific subnets (e.g., in hotspots). We presented the design of a system that enables (multi-homed) mobile hosts to seamlessly handoff from one aggregator to another so that they can continue to receive a channel wherever they go. We concentrated on handoffs between aggregators as a result of a mobile host crossing a subnet boundary. As part of the system, we discussed an innovative application-level protocol that enables mobile hosts to select the aggregator that provides the ‘best’ configuration of a channel. The protocol comes into play when a mobile host begins to receive a channel and when it crosses a subnet boundary while receiving the channel. During

roaming, the protocol also refreshes authentication softstate at the aggregators that a mobile host can reach. The protocol is independent of the actual stream control protocol (e.g., RTSP) that aggregators might use and can therefore easily be deployed in addition to existing streaming services. We have shown that the protocol can be realized with existing Internet standards, notably SIP and SDP.

Future work includes the design and realization of an announcement protocol (cf. the announcement protocols of SIP [29], SLP [18], and UPnP [24]). The protocol should for instance be able to announce events such as a configuration becoming (un)available, or the beginning/end of an aggregator's service life-cycle (e.g., because the aggregator only provides service to access providers during rush hour). Other future work involves the specification, storage, and usage of the policies that mobile hosts use to determine which configuration is the 'best' one.

### Acknowledgments

The authors thank Hans Zandbelt, Remco Poortinga, and Arjan Peddemors (Telematica Instituut) for helping out with the implementation and the test environment.

### References

- [1] E. Amir, S. McCanne and R. Katz, An active service framework and its application to real-time multimedia transcoding, in: *ACM SIGCOMM'98*, Vancouver, Canada, (Sept. 1998).
- [2] B. Cain, A. Barbir, R. Nair and O. Spatscheck, Known CN request-routing mechanisms, Internet Draft, draft-ietf-cdi-known-request-routing-03.txt (April 2003).
- [3] P. Calhoun, J. Loughney, E. Guttman, G. Zorn and J. Arkko, Diameter base protocol, Internet Draft (April 2003) draft-ietf-aaa-diameter-17.txt
- [4] Y. Chawathe, Scattercast: An adaptable broadcast distribution framework, Special Issue of ACM Multimedia Distribution (2002).
- [5] J. Chennikara, W. Chen, A. Dutta and O. Altintas, Application-layer multicast for mobile users in diverse networks, in: *IEEE Globecom 2002* (Taipei, Taiwan, Nov. 2002).
- [6] D. Clark, The design philosophy of the DARPA internet protocols, in *ACM SIGCOMM* (Sept. 1988).
- [7] N. Drew and M. Dillinger, Evolution toward reconfigurable user equipment, *IEEE Communications Magazine* (Feb. 2001).
- [8] S. Donovan and J. Rosenberg, Session timers in the session initiation protocol (SIP), Internet Draft, (Nov. 2002). draft-ietf-sip-session-timer-10.txt.
- [9] A. Dutta, H. Schulzrinne, S. Das, A. McAuley, W. Chen and O. Altintas, MarconiNet supporting streaming media over localized wireless multicast, M-Commerce 2002 Workshop, Atlanta, USA (Sept. 2002).
- [10] 3GPP, Automatic establishment of roaming relationships, Technical Report 22.971, version 3.1.1 (April 1999).
- [11] E. Guttman, C. Perkins, J. Veizades and M. Day, Service location protocol, version 2, *RFC 2608* (June 1999).
- [12] M. Haardt and W. Mohr, The complete solution for third-generation wireless communications: Two modes on air, one winning strategy, *IEEE Personal Communications* (Dec. 2000).
- [13] M. Handley and V. Jacobson, SDP: Session description protocol, *RFC 2327* (April 1998).
- [14] M. Handley, C. Perkins and E. Whelan, Session announcement protocol, *RFC 2974* (Oct. 2000).
- [15] C. Hesselman, H. Eertink and A. Peddemors, Multimedia QoS adaptation for inter-tech roaming, in *6th IEEE Symposium on Computers and Communications (ISCC'01)* (Hammamet, Tunisia, July 2001).
- [16] C. Hesselman, I. Widya, H. Eertink and E. Huizer, A comprehensive framework for broadcasting multimedia content in the future mobile internet, in: *2nd IEEE Workshop on Applications and Services in Wireless Networks (ASWN'02)* (Paris, July 2002).
- [17] H.-Y. Hsieh, K.-H. Kim, Y. Zhu and R. Sivakumar, A receiver-centric transport protocol for mobile hosts with heterogeneous wireless interfaces, in *Proc. MobiCom 2003* (San Diego, USA, Sept. 2003).
- [18] J. Kempf and J. Goldschmidt, Mobile code for network service access, INET 2000, Yokohama, Japan (July 2000), [http://www.isoc.org/inet2000/cdproceedings/3c/3c\\_2.htm](http://www.isoc.org/inet2000/cdproceedings/3c/3c_2.htm)
- [19] L. Kleinrock, An internet vision: The invisible global infrastructure, *AdHoc Networks Journal* 1(1) (July 2003) 3–11.
- [20] Kutscher, Ott and Bormann, Session description and capability negotiation, Internet Draft, draft-ietf-mmusic-sdpng-06.txt (March 2003).
- [21] G. Markoulidakis, G. Lyberopoulos, D. Tsirkas, and E. Sykas, Inter-operator roaming scenarios for third generation mobile telecommunication systems, in: *2nd IEEE Symposium on Computers and Communications (ISCC'97)* (Alexandria, Egypt, July 1997).
- [22] W. Marshall, Private session initiation protocol (SIP) extensions for media authorization, *RFC 3313* (Jan. 2003).
- [23] S. McCanne, V. Jacobson and M. Vetterli, Receiver-driven layered multicast, in: *ACM SIGCOMM* (Stanford, USA, Aug. 1996).
- [24] Microsoft Corporation, Universal plug and play device architecture, Version 1.0 (June 2000), [http://www.upnp.org/download/UPnPDA10\\_20000613.htm](http://www.upnp.org/download/UPnPDA10_20000613.htm)
- [25] Y. Nomura, R. Walsh, J. Ott and H. Schulzrinne, Protocol requirements for internet media guides, Internet Draft, draft-ietf-mmusic-img-req-00.txt (Sept. 2003).
- [26] oSIP webpage, <http://www.gnu.org/software/osip/>
- [27] J. Peterson, Enhancements for authenticated identity management in the session initiation protocol (SIP), Internet Draft (Feb. 2003). draft-ietf-sip-identity-01.txt
- [28] T. Plogemann, V. Goebel, L. Mathy, N. Race and M. Zink, Towards scalable and affordable content distribution services, in: *Proc. 7th International Conference on Telecommunications (ConTEL 2003)*, (Zagreb, Croatia, June 2003).
- [29] A. Roach, Session initiation protocol (SIP)-Specific event notification, *RFC 3265* (June 2002).
- [30] J. Rosenberg and H. Schulzrinne, Internet telephony gateway location, in *Proc. IEEE Infocom'98* (March 1998).
- [31] J. Rosenberg and H. Schulzrinne, An Offer/Answer Model with the Session Description Protocol (SDP), *RFC 3264* (June 2002).
- [32] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, SIP: Session initiation protocol, *RFC 3261* (June 2002).
- [33] S. Roy, B. Shen, V. Sundaram and R. Kumar, Application level hand-off support for mobile media transcoding sessions, in *NOSSDAV'02* (Miami Beach, Florida, May 2002).
- [34] J. Saltzer, D. Reed and D. Clark, End-to-end arguments in system design, *ACM Transactions on Computer Systems* 2(4) (Nov. 1984) 277–288.
- [35] H. Schulzrinne, Dynamic host configuration protocol (DHCP-for-IPv4) option for session initiation protocol (SIP) servers, *RFC 3361* (Aug. 2002).
- [36] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, RTP: A transport protocol for real-time applications, *RFC 1889* (Jan. 1996). Issue of ACM Multimedia Distribution (2002).
- [37] H. Schulzrinne, A. Rao and R. Lanphier, Real time streaming protocol (RTSP), *RFC 2326*, (April 1998).
- [38] A. Snoeren, D. Andersen and H. Balakrishnan, Fine-grained failover using connection migration, in: *Proc. USENIX USITS* (San Francisco, USA, March 2001).

- [39] A. Snoeren, H. Balakrishnan and F. Kaashoek, Reconsidering internet mobility, in: *Proc. 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)* (Elmau/Oberbayern, Germany, May 2001).
- [40] J. Solomon, *Mobile IP – the Internet Unplugged* (Prentice Hall, 1998).
- [41] M. Stemm and R. Katz, Vertical handoffs in wireless overlay networks, ACM Mobile Networking, Special Issue on Mobile Networking and Internet (Spring 1998).
- [42] F. Sultan, K. Srinivasan, D. Iyer and L. Iftode, Migratory TCP: Connection migration for service continuity in the internet, in: *Proc. IEEE ICDCS*, Vienna, Austria (July 2002).
- [43] TINA Consortium, Business model and reference points (May 1997). [http://www.tinac.com/specifications/documents/bm\\_rp.pdf](http://www.tinac.com/specifications/documents/bm_rp.pdf)
- [44] J. Verhoosel, R. Stap and A. Salden, A generic business model for WLAN hotspots – A roaming business case in the netherlands, in: *Proceedings of the First ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH'03)* (San Diego, USA, Sept. 2003).
- [45] M. Vernick, S. Bryden, M. Condry, G. Disher, J. Straley and W. Walkoe, Requirements for end-To-end delivery of broadband content, Broadband Content Delivery Forum (Oct. 2001).
- [46] VIC at UCL, <http://www-mice.cs.ucl.ac.uk/multimedia/software/vic/>
- [47] H. Wang, R. Katz and J. Giese, Policy-enabled handoffs across heterogeneous wireless networks, in: *2nd IEEE Workshop on Mobile Computing and Applications (WMCSA 1999)* (New Orleans, USA, Feb. 1999).
- [48] E. Wedlund and H. Schulzrinne, Mobility support using SIP, in: *2nd ACM/IEEE Int. Conf. on Wireless and Mobile Multimedia (WoW-MoM'99)* (Seattle, USA, Aug. 1999).
- [49] K. Wong and V. Varma, Supporting real-time IP multimedia services in UMTS, IEEE Communications Magazine (Nov. 2003).
- [50] L. Wu, R. Sharma and B. Smith, Thin streams: An architecture for multicast layered video, in: *7th Intl. Workshop on Network an Operating Systems Support for Digital Audio and Video (NOSSDAV97)*, St. Louis, USA (May 1997).
- [51] D. Xu and K. Nahrstedt, Supporting multimedia service polymorphism in dynamic and heterogeneous environments, Technical Report UIUCDCS-R-2000-2159, University of Illinois at Urbana-Champaign, USA (Oct. 2000).



**Cristian Hesselman** is a research assistant at the Telematica Instituut and a Ph.D student at the University of Twente, both in the Netherlands. His research is on the distribution of live multimedia streams (e.g., for radio or TV broadcasts) to mobile devices that move around in a wireless Internet. He focuses on application-level support (e.g., signaling protocols) for unrestricted roaming in managed environments. Cristian was with Lucent Technologies from 1996 until 1998. He holds an M.Sc. in Computer Science from the University of Twente, the Netherlands (1996).

E-mail: cristian.hesselman@telin.nl



**dr.ir. E.H. (Henk) Eertink** is a senior member of scientific staff at Telematica Instituut, and leader of the 'infrastructures for context awareness' (INCA) group at Telematica Instituut. He holds a PhD from Twente University (1994) on protocol engineering and validation. At Telematica Instituut he has actively worked on software tools for process analysis, distributed system design, multimedia signalling, mobility management, service platform technology and security. His current research interests are mobility management and secure context information exchange. He is a member of several program committees of international conferences.

E-mail: henk.eertink@telin.nl



**Ing Widya** is Assistant Professor of the Faculty of Electrical Engineering, Mathematics & Computer Science at the University of Twente, the Netherlands. He is a member of the Architecture and Services of Network Applications group (ANSA) and his research is embedded in the projects of the Centre for Telematics and Information Technology (CTIT). He received his PhD in Signal Processing and his current research interests cover enterprise modelling of communication supports for networked applications, design and analysis of application-context aware services and protocols operating over large-scale infrastructures like the Internet, including QoS and standardised multimedia format encodings.

E-mail: i.widya@utwente.nl



**Erik Huizer** is professor Internet Applications at Twente University. Huizer is also Director Strategy, innovation and Business Development with NOB Cross Media Facilities, responsible for digital broadcasting via DVB, Internet and mobile.

Until June 2000 Huizer was managing director of the SURFnet Expertise Center (SURFnet Expertise-Centrum, SEC). Huizer has been strongly involved with the GigaPort project, the Dutch project for the next generation Internet.

Huizer used to work for SURFnet on projects that dealt with developing and introducing new services. He also initiated and participated in several EU projects, like the DESIRE 4th framework project. From December 1991 till April 1995 he was Area Director for the Applications area of the Internet Engineering Task Force (IETF) and as such a member of the Internet Engineering Steering Group (IESG). From 1995–1999 he was a member of the Internet Architecture Board. From 1999 till 2002 he was chairman of the Internet Research task Force (IRTF). Currently Erik Huizer is serving as a Trustee on the board of Trustees of the Internet Society (ISOC).

E-mail: e.huizer@utwente.nl