A Policy-based System for Handoffs between Intermediary Content Providers in the Wireless Internet

Malohat Ibrohimovna Kamilova[1,2], Cristian Hesselman[1,2], Ing Widya[2], and Erik Huizer[2]
[1]Telematica Instituut, P.O. Box 589, 7500 AN, Enschede, The Netherlands
E-mail: malohat.kamilova@telin.nl, cristian.hesselman@telin.nl
[2] University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands
E-mail: i.widya@utwente.nl, e.huizer@utwente.nl

ABSTRACT

We consider the distribution of real-time multimedia content (e.g., radio or TV broadcasts) through multiple aggregators. An aggregator is an intermediary content provider that operates a pool of proxy servers to aggregate content from sources and forward it to mobile hosts. Aggregators package content into channels (e.g., CNN or ABC) and offer them in various versions (e.g., using different encodings) that differ in quality or price. Mobile hosts receive channels via the wireless Internet, which consists of multiple types of wireless networks (e.g. 802.11 and UMTS). At specific locations, mobile hosts can connect to multiple networks simultaneously (e.g., in a hotspot) and can thus potentially receive different alternative versions of a channel from different aggregators through different interfaces.

In this paper, we propose a control system that enables mobile hosts to automatically deal with these (changing) alternatives in a manner transparent to the mobile user. The system's novelty lies in the use of application-level policies. They for instance define when to look for a 'better' version of a channel (e.g., if packet loss increases to a certain threshold) and what constitutes 'best' based on the user's preferences. The policies thus define when and how to adapt the reception of a channel to changes in available resources or user's preferences.

I. INTRODUCTION

In the near future, the fringes of the Internet will consist of different types of wired and wireless networks that are operated by different administrative authorities [1]. As a result, mobile hosts will generally be able to receive service from multiple networks of different operators, for instance when they roam into a hotspot [2, 3].

At the application-level, the same real-time multimedia content (e.g., radio or TV broadcasts) can be streamed through multiple proxy servers, with mobile hosts handing off from one server to another as a result of mobility (e.g., because different proxy servers serve different networks) [4-7]. This idea can be extended to the distribution of channels through multiple *aggregators* [8, 2, 3]. An aggregator is an intermediary content provider that operates a pool of proxy servers to aggregate content from sources and forward it to mobile hosts [2, 3]. Aggregators package content into channels (e.g., CNN or ABC) and offer them in various versions (e.g., using different encodings) that differ in quality or price. As a result, mobile hosts can potentially receive different versions of a channel from different aggregators, possibly through different network interfaces (e.g., in a hotspot).

A research challenge is to develop a control system, which enables mobile hosts to automatically deal with such a (changing) set of alternatives in a manner invisible to the user [9]. We are designing such a control system based on policies (i.e. "if-condition-then-action" rules). Policies are rules that can be used by a controlling entity to constrain the behavior of a controlled entity in a way that the behavior of the controlled entity becomes aligned to the goal of the policy [10]. Policies are commonly used in network management, for instance to configure an RSVP router [11]. The advantage of policies is that they can be maintained in a central repository and then rolled out, which enables policy-controlled entities (e.g., routers) to be reconfigured with new policies (i.e., behavior) in a flexible manner.

The novelty of our control system is that it uses well-defined application-level policies. This means that the actions of the policy are enforced at the application-level. An application-level policy could for instance read: if the number of lost packets of a channel increases to a certain threshold (the condition), then invoke an application-level protocol (the action) to look for another aggregator that can offer the channel, possibly on another interface. Other application-level policies define when to handoff to another aggregator, and what constitutes the 'best' version of a channel based on the predefined user preferences. Using these policies, the system can adapt the reception of a channel to the capabilities of the Internet environment in the vicinity of the mobile host (e.g., in terms of available bandwidth), to the available resources of the mobile host (e.g., available battery power), and so on.

Known policy-based systems for Internet service control typically use network level policies rather than application-level policies and focus on determining which network (operator) provides the best service [12-15]. Wang et al. [12] do however not use well-defined policies like we do (i.e. rules with goals). Murray et al. [15] discuss the selection of a best network for a mobile host according to the current load on the networks. The selection in their system is controlled by policy decision logic that sits in the infrastructure, while ours only sits on mobile hosts. Clark et al. [13] and Lee et al. [14] take a different approach to determine the best service, which uses algorithms rather than policies.

The rest of this paper is organized into four sections. In Section II, we describe the environment for which our policy-based control system is designed. In Section III, we present the system's architecture. Thereafter, we discuss some of the policies that our system uses in Section IV. Finally, Section V summarizes the state of our research and explains our future work.

II. ENVIRONMENT

We consider an environment that consists of application-level service providers that deliver real-time multimedia content (e.g., radio or TV broadcasts) to mobile hosts in the form of channels (e.g., CNN Radio or BBC Television). We distinguish two types of providers: content sources and content aggregators [8, 2-3]. A *content source* is the origin of one or more channels and transmits them in a mobile agnostic manner (e.g., unaware of the changing IP addresses of mobile hosts). A *content aggregator*, on the other hand, is specifically designed to serve mobile hosts. It receives channels from sources and forwards them to mobile hosts in a mobile and wireless aware manner (e.g., it forwards channels in a way suitable for the limited capabilities of mobile hosts). The proxy-like distribution scheme via aggregators increases scalability in the absence of IP multicast [16], which is important when channels need to be distributed to a potentially large number of receivers. Sources and aggregators primarily process and forward application-level data units, typically in the form of RTP packets [17].

Figure 1 shows an example in which source cnn.com[1] distributes audio channel CNN Radio via aggregators stream-it.com and multimedia-forward.nl. User Bob receives CNN Radio either from multimedia-forward.nl through the UMTS network of network operator connect-you.nl, or from stream-it.com through the 802.11 network of hotspot.nl. The solid line between stream-it.com and hotspot.nl indicates that stream-it.com is only available through the 802.11 network. Similarly, media-forward.nl is only available through the UMTS network.

An aggregator can deliver its channels in different *versions* (cf., [4, 18]). This enables it to deal with different user requirements (e.g., pertaining to cost or quality) and to serve different types of hosts that connect to the Internet through different types of wireless links. We refer to the description of a channel version as a *configuration* (e.g., using SDP [19]). Each aggregator supports its own set of configurations of a channel. For example, stream-it.com could support various high-quality configurations of CNN Radio (e.g., in 'studio' quality), while media-forward.nl could only support medium-quality configurations of the same channel (e.g., in 'FM radio' quality). Mobile hosts can thus receive the same channel from different aggregators at different configurations, possibly through different interfaces (e.g., at point A in Figure 1).

[1] The domain names in this paper are for illustrative purposes only.

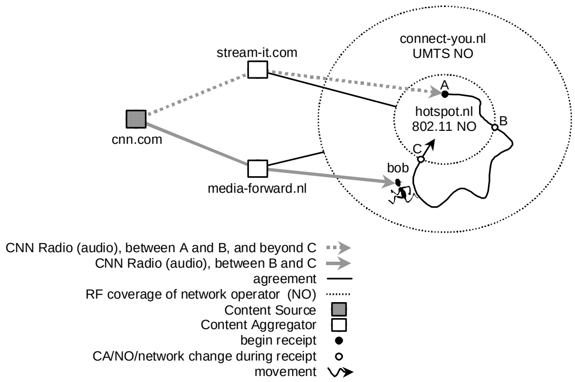

Figure 1. Streaming via multiple alternative aggregators

An application-level protocol [2, 3] enables mobile hosts to request which versions of a channel are available from the aggregators it can reach. A mobile host invokes the protocol when it is looking for a 'better' configuration of the channel it is receiving, for example when it moves into a subnet (an aggregator with a better configuration may appear) or moves out of one (aggregators may disappear, which may result in a new best aggregator). The assignment of a (new) IP address to one of the host's network interfaces (e.g., to Bob's 802.11 interface at point C), and the loss of packets or a decreasing signal strength (e.g., of the 802.11 network at point B) could signal these two events, respectively.

Figure 2 shows Bob's mobile host querying media-forward.nl and stream-it.com at point C of Figure 1 to check which configurations of CNN Radio they support.

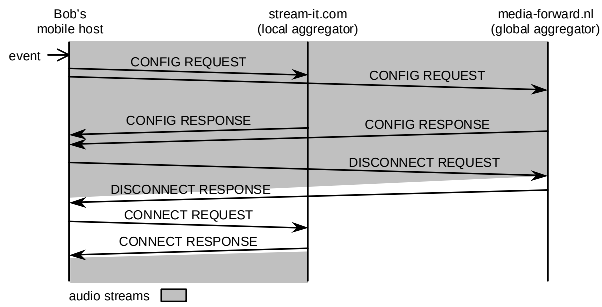

Figure 2. Typical protocol interactions for discovery and handoff.

Bob's host sends a configurations request to stream-it.com via its 802.11 interface (hotspot.nl), and a request to media-forward.nl through its UMTS interface (connect-you). Analyzing the responses of the aggregators, the host decides that stream-it.com provides a better version of CNN Radio than media-forward.nl. It therefore hands off to stream-it.com by sending a disconnect request to media-forward.nl and a connect request to stream-it.com (or the other way around). As a result, Bob's mobile host now receives the 'better' version of CNN Radio from stream-it.com via hotspot.nl's 802.11 network. The protocol's behavior is similar at points A and B, except that stream-it.com becomes unavailable around point B. We have implemented the protocol of Figure 2 using SDP [19] and SIP [20].

As we will see in Section III, the selection of the best aggregator and the trigger for querying aggregators is policy-driven. Examples of other occasions at which the mobile host could consult aggregators are when the host's battery power drops, when the available bandwidth on one of the host's network interfaces drops, when the user changes his preferences and so forth.

III. ARCHITECTURE

We use policies (i.e., "if-condition-then-action" rules) to flexibly define the behavior of mobile hosts roaming in the environment of Section II. We adopt the policy framework of the IETF [11, 21], which uses the concepts of a Policy Decision Point (PDP) and a Policy Enforcement Point (PEP).

A. Components

Figure 3 shows the high-level architecture of our control system. It consists of a PDP, a PEP, a policy repository, a resource manager, and a set of user preferences. In our current design, the PEP, the PDP and the resource manager are located on the mobile host.

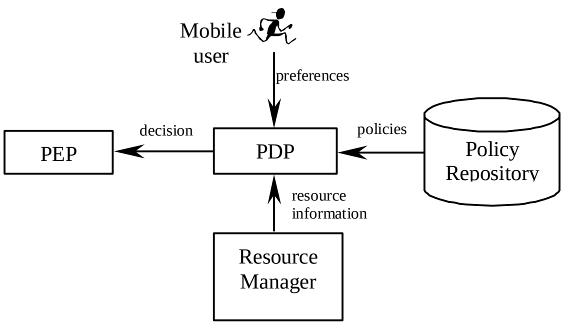

Figure 3. Architecture of the policy-based system.

A *PDP* represents a controlling entity that applies policies to control the behavior of a controlled entity (the PEP). In our control system, the PDP persistently monitors the state of the available resources and of the user's preferences and uses this information to evaluate the policies' conditions. If the circumstances are such that the "if" condition of a policy becomes true, then the PDP decides to enforce the actions defined in the "then" part of the policy. For example, if the if condition of a policy says "packet loss >= 20%", and the action reads "invoke protocol" (to discover new configurations, see Figure 2), then the PDP will enforce the discovery action if the number of lost packets of a channel exceeds 20% in a certain time interval. Our PDP is also responsible for selecting the best configuration according to the user preferences and the current available resources.

A *PEP* represents the controlled entity upon which policy decisions are being enforced (by the PDP), yielding a constrained behavior of the PEP. A PEP therefore receives directives from a PDP. In our system, the application-level protocol of Figure 2 embodies the PEP because it executes policy decisions such as "invoke protocol" or "handoff smoothly" (also see Section IV).

A *Policy Repository* contains (inactive) policies written in a policy specification language such as IRML [22]. A policy repository allows policies to be flexibly downloaded into a PDP, possibly at run-time. Another advantage is that policies become platform independent.

In our system, the repository for instance contains discovery policies (they define when to invoke the protocol of Figure 2) and handoff policies (they determine how to execute a handoff). We will elaborate on these and additional policy classes in Section IV.

In our system, each policy has a goal (e.g., "high viewing smoothness"), which is part of the specification of a policy. To retrieve the appropriate policies, the PDP matches the preferences (i.e., goals) of the user with the goals of the policies in the repository. The PDP and the PEP together realize the goal of a policy the PDP retrieves.

We expect that the policy repository will typically reside in the fixed Internet, thus enabling a user to consistently apply the same policies to all of his devices.

A PDP can generally use external information sources to come to its decisions [11]. The external information source in our architecture is the *Resource Manager*. It is responsible for monitoring available resources, such as availability of networks, available bandwidth, signal strengths of networks, packet loss of a channel, and available aggregators and configurations. The PDP accesses this information by requesting it or by listening to events from the Resource Manager (e.g., appearance of an IP address of an interface).

B. Behavior

Figure 3 also shows the interaction between the components of the policy-based system. PDP receives user preferences from the user (arrow labeled "preferences"). On analysis of the new user preferences PDP may decide to retrieve new or additional policies from the Policy Repository (arrow "policies"), that match with the new goals of the user. PDP may consult the Resource Manager (arrow "resource information") for the available resources. Having all necessary information PDP for example makes a selection of the best configuration, which is inline with the preferences of the user in price and quality level. Finally, PDP sends it's decision to PEP (arrow "decision"). Figure 4 shows the system's behavior when the user moves towards and comes close to the point B, the figure 1. The Resource Manager informs the PDP by sending an *event* that, for example a packet loss is continuously increasing. By receipt of the event the PDP evaluates the condition of the discovery policy (1), if the condition is true, PDP *requests* new information from the Resource Manager on *available resources* at that moment (2) and makes a new selection of the best configuration (3). Once the selection is made, the *decision* is sent to PEP, which executes a handoff (4) connecting the mobile host to the selected aggregator (in this case media-forward.nl, see figure 1) using the selected configuration.

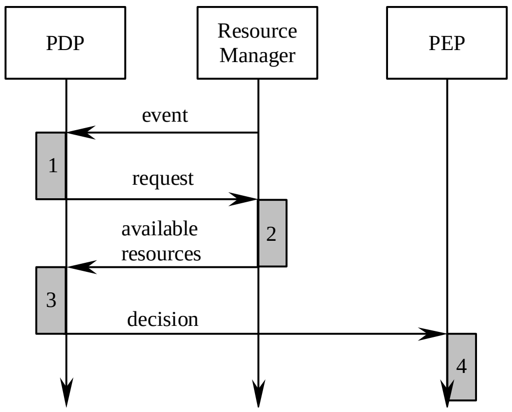

Figure 4. Example of system's behavior by receiving an event from Resource Manager

IV. EXAMPLES OF POLICIES

The entire system goes through three phases (see Figure 2): a *discovery phase* to send out *config* requests and collect the responses, a *selection phase* to determine which aggregator provides the ‘best’ configuration, and a *handoff phase* to handoff to a ‘better’ aggregator (if any).

We distinguish policies for each of the above phases. *Discovery policies* define when to invoke the protocol of Figure 2; *selection policies* define which aggregator provides the ‘best’ configuration of a channel based on the user’s preferences; and *handoff policies* determine how to execute a handoff.

In this section, we discuss a few examples of discovery and handoff policies. We use “viewing smoothness” as a goal. In our system, when the user chooses high viewing smoothness as his preference, the system provides seamless roaming by means of early handoffs. If the user chooses moderate/low smoothness, then system allows some data loss and glitches during the handoffs.

To explain the effects of the policies, we consider the situation in which user Bob (see Figure 1) is at point B while receiving CNN Radio from stream-it.com through its 802.11 interface. We assume that the user has expressed high smoothness of viewing video. According to this, the policies with the corresponding goal have been downloaded into (i.e. activated on) the PDP. These policies could for instance look like this:

```
/* policy_type=discovery, exiting hotspot
 * policy_goal=high_viewing_smoothness
 */
if (packet_loss >= 20% &&
    receiving_interface == “802.11”) {
  /* Invoke discovery */
  run_protocol();
}
```

```
/* policy_type=handoff
 * policy_goal=high_viewing_smoothness
 */
if (handoff_flag &&
    receiving_interface == “802.11”)) {
  /* First connect, then disconnect */
  connect_to(new_aggregator);
  disconnect_from(old_aggregator); }
```

The discovery policy uses the degradation of the streams that the mobile host receives as an indication that the mobile host is moving out of the hotspot [23]. It provides high viewing smoothness because it causes the PDP to react proactively on packet loss: if the PDP detects that it has lost 20% of the packets it received on the host’s 802.11 interface during a certain period, then it will decide to enforce the discovery policy by ordering the PEP to run the protocol (cf. Figure 2). If the user would have selected low viewing smoothness, then the PDP would have downloaded another discovery policy, for instance one that behaves in a more reactive manner (e.g., using a packet loss threshold of 80%). The discovery policy could also have used the monotonic decrease of signal strength instead of increasing packet loss.

The handoff policy realizes high viewing smoothness by first connecting to a new aggregator on the overlay network (e.g., media-forward.nl on the UMTS network), and then disconnecting from the old aggregator (stream-it.com on the 802.11 network). A handoff policy that provides low smoothness could for instance do this the other way around.

The policy examples also show that policies with common goals can be combined to a more complex one (thus also be decomposed in more elementary ones). Policies may furthermore depend on another, in the sense that they are not commutative during processing. Independent policies may be processed in any order without influencing the result. These research issues are, however, beyond the focus of this paper [10].

V. SUMMARY AND FUTURE WORK

We have presented the design of a control system, which uses policies to automatically deal with different networks, aggregators, and channel configurations. The system takes the preferences of the user into account, thus allowing for automatic adaptation without user involvement.

We are currently implementing a prototype of the system in which the PDP, the PEP, the policy repository, and the user preferences are co-located on the mobile host. Next step is to design and implement the policy-based system for a distributed scenario, where the PDP and the policy repository are located on remote machines and the PEP is located on the mobile host. The motivation to put the PDP remotely is to reduce the complexity at the mobile host, since some mobile devices are very small and have limited capabilities. Furthermore, we plan to describe policies in a policy specification language (e.g., in XML [24, 25]) and to test our policy-based system in stationary and roaming scenarios.

REFERENCES

[1] M. Haardt and W. Mohr, "The Complete Solution for Third-Generation Wireless Communications: Two Modes on Air, One Winning Strategy", IEEE Personal Communications, December 2000.

[2] C. Hesselman, H. Eertink, I. Widya, E. Huizer, "A Mobility-aware Broadcasting Infrastructure for a Wireless Internet with Hotspots", Proceedings of the First ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH'03), San Diego, USA, Sept. 2003

[3] C. Hesselman, H. Eertink, I. Widya, and E. Huizer, "Delivering Live Multimedia Streams to Mobile Hosts in a Wireless Internet with Multiple Content Aggregators", to appear in Mobile Networks and Applications Journal (MONET), special issue on Wireless Mobile Applications and Services on WLAN Hotspots, Summer 2005.

[4] D. Xu, K. Nahrstedt, "Supporting Multimedia Service Polymorphism in Dynamic and Heterogeneous Environments", Technical Report UIUCDCS-R-2000-2159, University of Illinois at Urbana-Champaign, USA, October 2000.

[5] H-Y. Hsieh, K-H. Kim, Y. Zhu, R. Sivakumar, "A Receiver-Centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces", Proc. MobiCom 2003, San Diego, USA, September 2003.

[6] 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, 2002.

[7] S. Roy, B. Shen, V. Sundaram and R. Kumar, "Application Level Hand-off Support for Mobile Media Transcoding Sessions", NOSSDAV'02, Miami Beach, Florida, May 2002.

[8] C. Hesselman, I. Widya, H. Eertink, and E. Huizer, "A Comprehensive Framework for Broadcasting Multimedia Content in the Future Mobile Internet", Proceedings of the 2nd IEEE Workshop on Applications and Services in Wireless Networks (ASWN'02), Paris, France, July 2002.

[9] L. Kleinrock, "An Internet Vision: the Invisible Global Instrastructure", AdHoc Networks Journal, Vol. 1, No. 1, July 2003, pp. 3-11.

[10] M. Cox and R. Davison, "Concepts, Activities and Issues of Policy-based Communications Management", BT Technology Journal, Volume 17, Issue 3, July 1999, pp. 155-169.

[11] R.Yavatkar, D. Pendarakis and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

[12] H.Wang, R. Katz, J.Giese, "Policy-Enabled Handoffs Across Heterogeneous Wireless Networks", 2nd IEEE Workshop on Mobile Computing and Applications (WMCSA 1999), New Orleans, USA, February 1999.

[13] D.D. Clark, J. Wroslawski, "The Personal Router whitepaper", MIT Technical Report, March 2001.

[14] G. Lee, P. Faratin, S. Bauer, J. Wroslawski, "Automatic Service Selection in Dynamic Wireless Network Environments", a poster presentation at MobiCom (co-located with First ACM International Workshop WMASH'03), San Diego, USA, 2003.

[15] K. Murray, R. Mathur, D. Pesch, "Intelligent Access and Mobility Management in Heterogeneous Wireless Networks using Policy", Adaptive Wireless Systems Group, Department of Electronic Engineering, Cork Institute of Technology, Ireland.

[16] J. Chennikara, W. Chen, A. Dutta, O. Altintas, "Application-Layer Multicast for Mobile Users in Diverse Networks", IEEE Globecom 2002, Taipei, Taiwan, November 2002.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[18] T. Plagemann, V. Goebel, L. Mathy, N. Race, and M. Zink, "Towards Scalable and Affordable Content Distribution Services", Proc. 7th International Conference on Telecommunications (ConTEL 2003), Zagreb, Croatia, June 2003.

[19] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[20] H. Schulzrinne, "Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers", RFC 3361, 2002.

[21] A. Westerinen et al, "Terminology for Policy-Based Management", RFC 3198. November 2001.

[22] W. Ng et al, "Quality of Service Extension to IRML", Panasonic Singapore Labs, Internet-draft, July 2001.

[23] C. Hesselman, H. Eertink, and A. Peddemors, "Multimedia QoS Adaptation for Inter-tech Roaming", Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC'01), Hammamet, Tunisia, July 2001.

[24] I. Liabotis, O. Prnjat, L. Sacks, "Policy-based Resource Management for Application Level Active Networks", University College London, England, UK

[25] N. Damianou, A. Bandara, M. Sloman, E. Lupu, "A Survey of Policy Specification Approaches", Department of Computing, Imperial College of Science Technology and Medicine, London, 2002.