Broadcasting Multimedia Channels in Future Mobile Systems

Cristian Hesselman and Henk Eertink

Telematica Instituut, P.O. Box 589, 7500 AN Enschede, The Netherlands
{hesselman, eertink}@telin.nl

Abstract. In this paper, we present ongoing work on an application-level platform that assists in broadcasting multimedia streams in heterogeneous mobile systems. Our work revolves around the notion of a technology domain, which is a set of resources that serve mobile clients connecting to the mobile system through a specific wireless technology. We consider the policies that technology domains use to (a) manage the availability of content channels and their perceptual quality levels, and (b) to manage the adaptation strategy that clients must follow in switching between quality levels. We discuss our future work in these areas, which will focus on adaptation as a result of clients roaming between different technology domains.

1. Introduction

In a Future Mobile System (FMS), applications will run on top of an IP-based network infrastructure that uses a wide variety of wireless network technologies and serves many different mobile devices [1]. An FMS will be operated by various administrative authorities [2] that enable mobile users to roam in an unrestricted manner [3-7].

In this paper, we present ongoing work on an application-level platform [8] that assists in broadcasting multimedia streams in an FMS (cf. [9]). Our model of an FMS revolves around the notion of a technology domain. A technology domain consists of a set of resources at the edge of an FMS that serve mobile clients connecting to the FMS through a specific wireless technology [2]. Technology domains manage the availability of content channels (e.g., the eight o'clock news of a certain television network) and their perceptual quality levels (e.g., videophone or TV quality [10]) through channel policies. Similarly, they manage the strategy that clients must follow in switching from one quality level to another through adaptation policies [11]. Our work concentrates on adaptation as a result of clients roaming between different technology domains [3-6].

The rest of this paper is organized as follows. In Section 2, we introduce the high-level notion of a broadcast channel that we use in our work. In Section 3, we present our view of an FMS in terms of technology domains. We consider the channel and adaptation policies that technology domains use in Sections 4 and 5, respectively. We conclude with a summary and a brief outlook on our future work in Section 6.

M.J. van Sinderen and L.J.M. Nieuwenhuis (Eds.): PROMS 2001, LNCS 2213, pp. 35-43, 2001.

2. Channels

A *channel* broadcasts multimedia content (e.g., a soccer match) from one server to a number of (mobile) clients. A channel consists of a small number (up to 3 or 4) of *differentiated channels* (diff channels) that the server simulcasts [12]. The diff channels of a channel contain the same content, but are optimized for different coarse-grained classes of clients (e.g., clients with small displays). At the application-level, a server compresses and packetizes each diff channel using various types of encoders (e.g., MPEG-4 [13]) and packetizers (e.g., RTP [14]) to attain good client coverage. The output of these two operations consists of a set of *streams* for each encoder-packetizer combination. The number of streams depends on whether the server packetizes the audio and video parts of a diff channel together, and on whether or not the involved encoder is a layered one [15-17]. A client can receive several streams of a channel at a time, but we assume that they must belong to the same diff channel.

As an example, consider the channel of Figure 1. The content that it carries is a class on Object Oriented Programming (OOP). The server uses two diff channels for the OOP channel, one of which has additional contrast to better suit (mobile) clients with small screens. The server encodes and packetizes the high-contrast stream into four streams using two encoder-packetizer combinations. One of these combinations involves MPEG-4 (encoding) and RTP (packetization) and results in three layered MPEG-4/RTP streams. Client 2 in Figure 1 is a PDA that subscribes to one or more of these streams to receive the high-contrast diff channel.

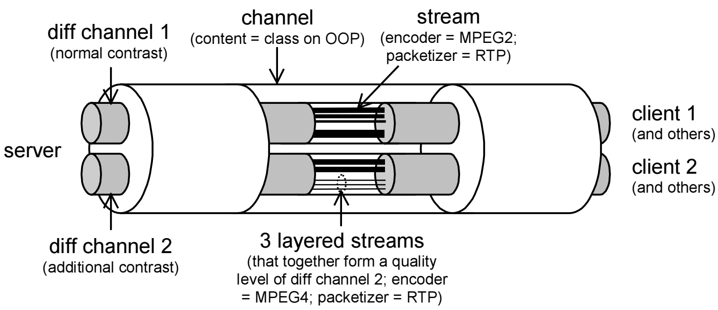

Fig. 1. Example of a channel.

In our model, servers associate *perceptual quality levels* [10] with each diff channel they transmit. These quality levels can for instance be based on perceived quality assessments [11, 18]. For each combination of encoder and packetizer, a server also defines which streams a client must receive to attain which quality level [15-17].

Servers provide information on the channels they support so that clients can determine which diff channels and which of their quality levels they can receive. This information pertains to the content a channel carries, the properties and perceptual quality levels of its diff channels, the encoding and packetization properties of its streams, their relation to the quality levels of a diff channel (e.g., the number of streams a client needs to receive to attain a certain quality level), and the IP-level QoS requirements of streams (e.g., in terms of bandwidth and loss characteristics). A server can for instance describe the perceptual quality levels of a stream in the form of parameters such as frame rate and pixel count [19]. Alternatively, they may be expressed in the form of well-known labels such as 'videophone' or 'TV' quality [10]. For ease of notation, we will use the latter option in this paper.

Channels may be aggregated into *multi-channels* if there exist multiple servers that simultaneously transmit the same content. For instance, there could be multiple servers that transmit a soccer match. We assume however that clients only receive streams of one channel of a multi-channel (e.g., the soccer match transmitted by CNN) at a time. Observe that multi-channels are identified by the content they carry. Individual channels are identified by the content they carry and the server that transmits them. We will not consider multi-channels any further in this paper.

3. Domains

We model the infrastructure of an FMS in terms of *administrative domains*. An administrative domain consists of a set of resources (hosts, networks, routers, and gateways) that is owned and governed by a single administrative authority (based on [20, 21]). We think of an administrative domain as an autonomous system that operates its resources according to a set of policies [21]. Our work concentrates on edge domains that (at a minimum) provide client devices with wireless IP-level connectivity to an FMS (e.g., the campus, Computer Science and Electrical Engineering domains of Figure 2). We focus on domain policies that manage application-level resources such as channels, diff channels and streams. In this context, we particularly zoom in on policies that deal with Quality of Service (QoS) and mobility issues (see Sections 4 and 5).

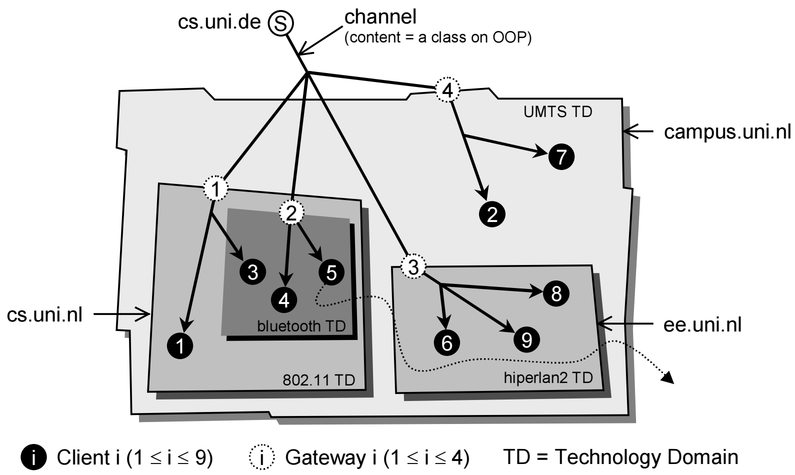

Fig. 2. Broadcasting an audio-video channel in a multi-domain FMS.

Our model distinguishes one or more *technology domains* within each administrative domain. A technology domain consists of the subset of resources and policies of an administrative domain that serve client devices connecting to the FMS through a specific link-level technology. In our work, these are typically wireless technologies such as UMTS, 802.11, and HIPERLAN2.

The radio front-end of a technology domain typically forms a separate subnet that covers a certain geographical area. This area is determined by the location of the administrative domain (e.g., a university campus in the Netherlands) and the coverage characteristics of the associated network technology (e.g., short-range for Bluethooth, medium-range for 802.11, and long-range for a UMTS technology domain) [2]. The back-end resources of different technology domains in the same administrative domain may overlap. For instance, an 802.11 and a Bluetooth technology domain may share routers, fixed network links and gateways. Figure 2 shows several examples of technology domains (e.g., the 802.11 technology domain in the Computer Science administrative domain). In this example, we assume that the campus domain (campus.uni.nl) forms an overlay [1, 4] for the Computer Science and Electrical Engineer domains (cs.uni.nl and ee.uni.nl, respectively). We will explain the other objects that appear in Figure 2 in the next section.

4. Channel Policies

A technology domain uses a *channel policy* to define which channels (e.g., the class on OOP from server S in Figure 2) and diff channels (e.g., the high-contrast one only) are available to clients in a technology domain, at which perceptual quality levels (e.g., videophone and VCR quality), and under which conditions. The latter may for instance depend on the time of day (e.g., resource hungry quality levels are only available off rush hour) or on the types of clients that are present in the technology domain at a particular point in time (e.g., there is a majority of devices with a small display). A channel policy also prescribes which types of streams (in terms of encoding and packetization formats) must be used under which conditions.

We use an *application-level gateway* [10, 11, 17, 22-26] to *enforce* the channel policies of a technology domain. A gateway (logically) belongs to one technology domain and intercepts a channel before it reaches any of the client devices in the technology domain. This means that clients always receive a channel from a gateway in one of the technology domains they can connect to and do not directly connect to a server [15, 16, 27]. Our approach is therefore breaks the Internet's end-to-end paradigm. Unfortunately, technology domains would be unable to locally enforce their channel policies without an intermediate gateway. The exception to this rule are clients that reside in the same technology domain as the server. In this case, we expect the server to enforce the channel policy. Figure 2 shows four gateways (one for each technology domain) that intercept the OOP channel.

A gateway *may* modify a channel if it does not conform to the channel policy of a technology domain. A gateway does not make any modifications to a channel if the channel is already in line with the policy. In this case, the gateway simply relays the channel to the clients in the technology domain.

A gateway may change a channel in several ways [10] to have it meet a certain channel policy. It may for instance reduce the perceptual quality of a diff channel (e.g., by dropping frames) to reduce the bandwidth it consumes. In this case, the gateway essentially defines and creates its own quality levels. A gateway may also change the encoding of a stream [22, 25] if the clients in the technology domain do not support the encoding that the server uses. Although technology domains are free to support as many domain-specific quality levels and streams as they see fit, we envision that they will only make a small number of them available simultaneously for reasons of scalability.

We expect that servers will indicate to what degree they allow gateways in the FMS to alter their streams. Some servers may for instance be very strict (e.g., a server of a television network) and will not allow gateways to make any changes to their (copyrighted) content. Other servers (e.g., the tele-learning server S of Figures 1 and 2) might allow this, but may want to put restrictions on the perceptual quality levels to which a stream may be lowered. They may also want to limit the set of encoding formats to which streams may be transcoded, how many times a stream may be modified, and so on. Yet other servers might not care about modifications at all and allow gateways to change their streams as they see fit. Observe that in this paper we only allow gateways to alter streams. We do not consider clients that modify a stream other than in terms of depacketization and decompression operations.

The effects of a channel policy (e.g., the availability of certain high-end perceptual quality levels) should be restricted to the corresponding technology domain. To accomplish this, we have proposed that gateways transmit streams onto IP multicast groups that are scoped [28] to the gateway's technology domain [29]. Clients in the technology domain can subscribe to one or more of these multicast groups (streams) to attain a quality level that is appropriate for them. As we will see in Section 5, this approach also allows us to combine QoS adaptation and handoff control.

Clients use a service discovery mechanism (e.g., a well-known multicast group [22, 23]) to learn about the technology domains they can use to connect to an FSM. For instance, client 5 (Figure 2) can use the Bluetooth and 802.11 domains of cs.uni.nl, and the UMTS domain of campus.uni.nl. Clients interact with these technology domains to determine which channels, diff channels, streams, and perceptual quality levels they can receive from each technology domain. The inputs of this negotiation process consist of the channel policies of the involved technology domains (e.g., of the 802.11 technology domain), the capabilities of the client (e.g., the properties of client 5's display and its encoder support), and the available resources on the paths from the client to the various gateways (e.g., gateways 1, 2 and 4 for client 5). The result consists of a set of scoped multicast groups from one of the technology domains that the client must join to attain a quality level associated with a channel of its choice.

5. Adaptation Policies

Technology domains use an *adaptation policy* to define at which points a client in a technology domain should initiate and complete a change to which other perceptual quality level (e.g., when to drop back from VCR to videophone quality) [11], and under which conditions (e.g., depending on the time of day). This may be a quality level of the same diff channel that the client currently receives, or a quality level of one of the other diff channels in a channel (cf. Figure 1).

Adaptation may be required for various reasons, including roaming, RF interference, increased traffic load [30, 31], network congestion [15, 16, 27], or a channel policy evaluating to false (which could for instance trigger the removal of a quality level that some clients are receiving at). Our work concentrates on adaptation as a result of a client roaming between different technology domains, either within the same or across different administrative domains [7]. Application-level adaptation policies are also discussed in [11], but for the purpose of configuring per-stream rate controllers.

Since channels are scoped to technology domains at the FMS' edges (see Section 4), a roaming client (e.g., client 5 in Figure 2) may have to hand off to a scoped channel in another technology domain when it crosses a technology domain boundary. A *handoff policy* is a special type of adaptation policy that defines when a mobile client should initiate and complete such an inter-tech handoff. We have outlined several application-level handoff policies in [29] that are based on the packet loss characteristics of the paths between a client and the gateways it can reach. These policies can be used for inter-tech handoffs in an overlay situation. Work on IP-level handoff policies for similar purposes can be found in [32].

Before a client can execute a handoff, it must first discover if the channel (i.e., content) it is currently receiving is available in the technology domains that are potential targets for handoff. For each domain where this is the case, the client must also figure out which diff channels, quality levels, and streams it can use. This procedure is similar to the one we discussed at the end of Section 4. The result consists of a set of diff channels and quality levels for each technology domain the client can handoff to. In case of a tie, policies of the end-user should ensure that a handoff can commence automatically ("user involvement with minimal user interaction" [32]). As an example, consider client 5 (Figure 2) roaming from the Bluetooth domain into the 802.11 domain. Before the client can execute a handoff, it must first determine if the OOP channel it is receiving is available in the 802.11 domain. Assuming this is the case, the client has to figure out which diff channels, quality levels, and streams it can use. If there are multiple possibilities, the end-user policies must resolve the tie. The end-user could for instance favor the videophone quality level of the 802.11 domain over its TV quality level because the former consumes less energy [32]. Alternatively, client 5 may consider subscribing to OOP channels other than the one in the 802.11 domain (e.g., the UMTS technology domain of campus.uni.nl). This situation for instance occurs when the OOP channel is not available in the 802.11 domain. Another possibility would be that client 5 considers the OOP diff channels and quality levels that are available in the 802.11 domain unacceptable.

The handoff targets in our model (a diff channel, a quality level, and its associated streams) map to a set of scoped multicast groups in the target technology domain (see Section 4). The client therefore subscribes to the multicast groups in the target technology domain to execute the handoff. To smooth the handoff [11] and to handle ping-pongs [3] between the current and the target technology domains, the client may buffer the streams it receives from the gateway in the target domain. When it is done buffering, it puts the streams in the target domain on screen and leaves the multicast groups in the old technology domain.

Finally, it is important to observe that our solution operates at the application-level and combines QoS adaptation and handoff control (cf. [11]). Similar application-level approaches exist for Internet Telephony [33, 34] and H.323 [35, 36]. They can be used for inter-technology and inter-domain roaming as well, but as far we know they only deal with mobility management (including handoff control) and do not consider QoS-related issues ([36] briefly mentions it, but gives no details). Also note that the indirection provided by IP multicast frees us from having an invariant IP address for mobile clients (UDP communications only). In fact, a client in our model gets a new IP address (e.g., through DHCP [37]) every time it crosses a boundary of a technology domain. We are thus not required to use Mobile IP [38]. Although IP multicast has been used to support IP-level mobility [4, 6, 39-41], we are not aware of any work that uses IP multicast for combined QoS adaptation and handoff control at the application-level.

6. Summary and Future Work

We have presented a model for broadcasting multimedia content in an FMS. The model revolves around the notion of a technology domain. A technology domain manages the channels, diff channels, streams, and perceptual quality levels it offers to clients through a channel policy. Similarly, it manages the adaptation behavior of clients in case of an inter-tech handoff by means of a handoff policy.

Our future work will be on discovering the channels, diff channels, perceptual quality levels and streams that are available in the technology domain into which a client roams. This will involve negotiation protocols that take channel and handoff policies, as well as capabilities and available resources of mobile clients and wireless networks into account.

Acknowledgements

The authors would like to thank Ing Widya and Erik Huizer for their valuable comments on this work.

7. References

1. M. Haardt, W. Mohr, "The Complete Solution for Third-Generation Wireless Communications: Two Modes on Air, One Winning Strategy", IEEE Personal Communications, Dec. 2000
2. DOLMEN Consortium, "Open Service Architecture for Mobile and fixed environments (OSAM)", Final Release, Version 4.0, July 1998
3. K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J. Makela, R. Pichna, J. Vallström, "Handoff in Hybrid Mobile Data Networks", IEEE Personal Communications, April 2000

4. M. Stemm, R. Katz, “Vertical Handoffs in Wireless Overlay Networks”, ACM Mobile Networking, Special Issue on Mobile Networking and Internet, Spring 1998
5. J. Meggers, A. Park, R. Ludwig, “Roaming between GSM and Wireless LAN”, ACTS Mobile Summit, Granada, Spain, Nov. 1996
6. E. Brewer, R. Katz, Y. Chawathe, S. Gribble, T. Hodes, G. Nguyen, M. Stemm, T. Henderson, E. Amir, H. Balakrishnan, A. Fox, V. Padmanabhan, S. Seshan, “A Network Architecture for Heterogeneous Mobile Computing”, IEEE Personal Communications, Oct. 1998
7. G. Markoulidakis, G. Lyberopoulos, D. Tsirkas, E. Sykas, “Inter-Operator Roaming Scenarios for Third Generation Mobile Telecommunication Systems”, 2nd IEEE Symposium on Computers and Communications (ISCC’97), Alexandria, Egypt, July 1997
8. B. Aiken, J. Strassner, B. Carpenter, I. Foster, C. Lynch, J. Mambretti, R. Moore, B. Teitelbaum, “Network Policy and Services: A Report of a Workshop on Middleware”, RFC 2768, February 2000
9. R. Keller, T. Lohmar, R. Tönjes, J. Thielecke, “Convergence of Cellular and Broadcast Networks from a Multi-Radio Perspective”, IEEE Personal Communications, April 2001
10. N. Yeadon, F. Garcia, D. Hutshison, D. Shepherd, “Filters: QoS Support Mechanisms for Multipeer Communications”, IEEE Journal on Selected Areas in Comm., Sept. 1996
11. R. Liao, A Campbell, “On Programmable Universal Mobile Channels in a Cellular Internet”, 4th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98), Dallas, October, 1998
12. X. Wan, H. Schulzrinne, “Comparison of Adaptive Internet Multimedia Applications”, IEICE Transactions on Communications, June 1999
13. R. Koenen, “MPEG-4 — Multimedia for Our Time”, IEEE Spectrum, Feb. 1999
14. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, RFC 1889, Jan. 1996
15. S. McCanne, V. Jacobson, M. Vetterli, “Receiver-driven Layered Multicast”, Proc. of ACM SIGCOMM, Stanford, USA, August 1996
16. L. Wu, R. Sharma, B. Smith, “Thin Streams: An Architecture for Multicast Layered Video”, 7th Intl. Workshop on Network an Operating Systems Support for Digital Audio and Video (NOSSDAV97), St. Louis, USA, May 1997
17. A. Balachandran, A. Campbell, M. Kounavis, “Active Filters: Delivering Scaled Media to Mobile Devices”, 7th Int. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97), St. Louis, USA, May 1997
18. A.Watson, M. Sasse, “Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications”, Proc. ACM Multimedia ‘98, Sept. 1998, Bristol, England
19. OMG Telecommunications Domain Task Force, “Control and Management of A/V Streams”, ftp://ftp.omg.org/pub/docs/formal/00-01-03.pdf
20. TINA Glossary, http://www.tinac.com
21. D. Comer, “Internetworking with TCP/IP”, Prentice Hall, 1995
22. E. Amir, S. McCanne, R. Katz, “An Active Service Framework and its Application to Real-time Multimedia Transcoding”, Proc. of ACM SIGCOMM'98, Vancouver, Canada, Sept. 1998
23. K. Jonas, M. Kretschmer, J. Moedeker, “Get a KISS – Communication Infrastructure for Streaming Services in a Heterogeneous Environment”, Proc. of ACM Multimedia'98, Bristol, UK, Sept. 1998
24. B. Zenel, D. Duchamp, “A General Purpose Proxy Filtering Mechanism Applied to the Mobile Environment”, Proc. 3rd ACM/IEEE International Conference on Mobile Computing and Networking, Budapest, Hungary, Sept. 1997
25. E. Amir, S. McCanne, H. Zhang, “An Application Level Video Gateway”, Proc. of

ACM Multimedia, San Fransisco, USA, Nov. 1995
26. I. Kouvelas, V. Hardman, J. Crowcroft, "Network Adaptive Continuous-Media Applications Through Self Organised Transcoding", Proc. Network and Operating Syst. Support for Digital Audio and Video (NOSSDAV'98), July 1998, Cambridge, UK
27. S. Cheung, M. Ammar, X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution", proceedings IEEE Infocom '96, San Francisco, USA, March 1996, pp. 553-560
28. D. Meyer, "Administratively Scoped IP Multicast", RFC 2365
29. C. Hesselman, H. Eertink, A. Peddemors, "Multimedia QoS Adaptation for Inter-tech Roaming", Proc. of the 6th IEEE Symosium on Computers and Communications (ISCC'01), Hammamet, Tunisia, July 2001
30. A. Campbell, "A Research Agenda for QOS-aware Mobile Multimedia Middleware", NSF Wireless and Mobile Communications Workshop, Virginia, USA, March 1997
31. D. Chalmers, M. Sloman, "A Survey of Quality of Service in Mobile Computing Environments", IEEE Communications Surveys, http://www.comsoc.org/pubs/surveys, 2nd Quarter 1999
32. 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
33. E. Wedlund, H. Schulzrinne, "Mobility Support Using SIP", 2nd ACM/IEEE Int. Conf. on Wireless and Mobile Multimedia (WoWMoM'99), Seattle, USA, Aug. 1999
34. W. Liao, "Mobile Internet Telephony Protocol (MITP): an Application-Layer Protocol for Mobile Internet Telephony Services", Proc. IEEE ICC'99, Vancouver, Canada, June 1999
35. W. Liao, "Mobile Internet Telephony: Mobile Extensions to H.323", Proc. IEEE INFOCOM'99, New York, USA, March 1999
36. D. Park, W. Yoon, D. Lee, "An Efficient Handoff Management for Mobility Support to H.323", International Workshop on Mobile Multimedia Communications (MoMuC'00), Tokyo, Japan, Oct. 2000
37. R. Droms, "Automated Configuration of TCP/IP with DHCP", IEEE Internet Computing, July-August 1999
38. J. Solomon, "Mobile IP — The Internet Unplugged", Prentice Hall, 1998
39. A. Helmy, "A Multicast-based Protocol for IP Mobility Support", ACM Second International Workshop on Networked Group Communication (NGC 2000), Palo Alto, USA, November 2000
40. C. Tan, S. Pink, K. Lye, "A Fast Handoff Scheme for Wireless Networks", Proc. 2nd ACM Workshop on Wireless Mobile Multimedia (WoWMoM'99), Seattle, Aug. 1999
41. J. Mysor, V. Bharghavan, "A new multicasting-based architecture for internet host mobility", Proc. ACM/IEEE Mobicom, Sept. 1997