PRISM, an IP-based architecture for broadband access to TV and other streaming media*

- a work-in-progress report -

A. Basso, C. D. Cranor, R. Gopalakrishnan, M. Green, C. Kalmanek, D. Shur, S. Sibal, C. Sreenan† and J.E. van der Merwe

AT&T Labs - Research
Florham Park, NJ, USA

† *University College Cork*
Cork, Ireland

1 Introduction

IP-based networks such as the Internet have been the focal point for innovative technologies, services, and applications. These networks have enabled hugely successful services like the World Wide Web, and they are even attracting traditional services such as telephony. The service model for accessing Web content offers tremendous control and choice to users. In particular the on-demand nature of this model provides users instant access to content of their choice.

Another class of services enabled by IP networks is the delivery of streaming audio and video content. While the Web service model also extends to IP-based streaming content, the user experience today does not compare favorably with the quality of broadcast TV. Current broadcasting technologies such as TV, cable, and satellite offer high quality but provide limited choice and do not allow users to access content at arbitrary times (i.e. no on-demand access). Broadcasting is geared towards a "push" model where content distributors control what content is made available to viewers and the time at which it is broadcast. Furthermore, broadcasting systems are essentially closed to all but major media companies. This is in contrast to the Web model where anyone can be a content creator for a global audience.

An IP-based model allows specialized producers of content to target arbitrarily small audiences. This in turn fosters the production of content that is rich in variety. Multimedia content generated in this environment will be much larger in volume than the "classical" entertainment content available today. This explosion in content makes the current broadcast service model infeasible.

In this paper we present the Portal Infrastructure for Streaming Media (PRISM) architecture that offers a new service model combining the high quality of the broadcasting world with the immediacy of the Web. PRISM uses IP as the fundamental infrastructure for realizing this vision. Basing PRISM on IP paves the way for future convergence and interaction of video, data, and voice services.

The PRISM service model emulates the current schedule driven TV broadcast system as a base level of service. As is the case in current systems, users can access this content based on the scheduled "airing" time, and the content is delivered using multicast over the IP network. However, in the PRISM architecture content is also stored within the network thereby enabling a variety of sophisticated on-demand services. In particular, this architecture has an inherent archiving ability that allows users to view content based on both the name of the content as well as the time at which it was "aired." For example, users can request the CNN Headline News that aired at 1PM on January 1st, 2000. Content stored inside the network is made accessible throughout the whole PRISM infrastructure enabling another unique feature of our architecture, namely, global access to TV content. For example, a user based in the U.S. can access European sporting events or other TV content either live or on-demand. The architecture is also able to support more conventional streaming service models such as "movie-on-demand." We envision this architecture co-existing with existing cable TV services and serving as a stepping stone towards a future of personalized full-on-demand streaming content.

A key feature of our architecture is that content traverses the access network only if there are active viewers. This holds true for both schedule driven and on-demand access and is a radical departure from the current broadcasting paradigm. The capacity freed up in this way facilitates the on-demand access component of our architecture. In the final version of this paper we will show the feasibility of our approach by analyzing the available capacity in a hybrid-fiber-coax (HFC) cable plant as an example broadband access network.

The components of the PRISM architecture are introduced in Section 2. In this paper we concentrate on the content discovery aspect of the architecture. We introduce a naming scheme suitable for our needs and the framework of our content discovery approach. In Section 3 we present

*Contact author: Kobus van der Merwe, AT&T Labs - Research, 180 Park Avenue, Florham Park, NJ 07932, USA, kobus@research.att.com.

a sampling of the considerable body of related work. The paper concludes with a description of the current status of our work as well as future plans.

2 PRISM Architecture

The PRISM architecture is built around three basic elements, as shown in Figure 1. These elements are:

Live sources: receive streaming content from a content provider, encode and packetize the content, and then transmit it into the PRISM IP network infrastructure. Live sources are typically directly connected to a high capacity backbone network.

Replay portals: receive multimedia content from live sources and other replay portals and retransmit it to PRISM clients. Replay portals can store and archive live content, thus allowing content to be viewed on-demand. Replay portals also provide VCR-like functions such as fast-forward and rewind to clients. Portals are positioned between clients and servers, typically at a bandwidth discontinuity such as cable head-ends.

Clients: receive content from a replay portal and display it to end-users. Clients are IP capable set-top boxes or personal computers and are assumed to connect to the backbone network via a broadband access network. A client normally interacts with a replay portal that is close to it in the network. This portal is called the *local portal*. Note that a client's local portal may act as a proxy when it receives a request for content that it does not have locally stored. This allows the local portal to provide VCR-like controls to the client even when the content being viewed is coming from a remote portal. It also allows the local portal to cache the recently viewed content in case users wish to review it later.

The Real Time Streaming Protocol (RTSP) [1] is used between clients and portals, and portals and live sources to control delivery of streaming content.

As shown in Figure 1, the three types of elements are connected together by the following components to form the PRISM architecture:

Content distribution network (CDN): transfers content from live sources to the replay portals and between replay portals. The main features required of a streaming CDN are efficient distribution and semi-reliable transfer of content.

Content delivery network: streams content from replay portals to clients on the access network.

Content discovery infrastructure: discovers the existence and location of streaming content within the PRISM infrastructure.

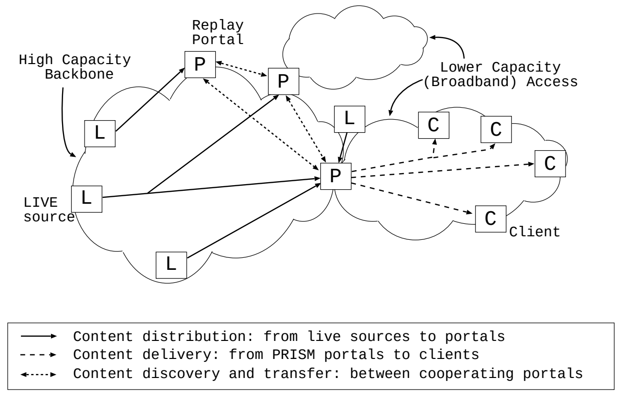

Figure 1: PRISM architecture

In the remainder of this section we will focus on aspects relating to content discovery. In particular we will consider:

Local portal discovery: used to determine which replay portal should act as the client's local portal.

Content naming: used to name content. PRISM uses a generic content naming scheme that is independent of its architecture. This allows content directory servers such as TV program guides to inter-operate with PRISM without having to know architectural details.

Mapping service: used to map PRISM-independent content names to one or more PRISM servers (typically in preparation for playback).

2.1 PRISM Interactions

Figure 2 illustrates how PRISM services a request for content. A user starts by choosing the content to be viewed. Note that the PRISM architecture does not dictate how content should be selected. Example content selection methods include program guides accessed through web browsers, channel names, and video search systems [2]. The result of content selection is a Uniform Resource Identifier (URI) [3] that names the content. The naming URI only specifies what the content is, it does not specify where or how to get it.

Once a content naming URI has been selected, it is passed to a PRISM client. For example a client on a personal computer could use a browser for content selection. The browser will start up a player passing it the requested URI. The PRISM client uses (or has already used) the local portal discovery mechanism to determine which portal to access PRISM through. The selection of a local portal takes into account factors such as locality and portal load. The PRISM client then requests the content by sending the content naming URI in an RTSP request to its local portal.

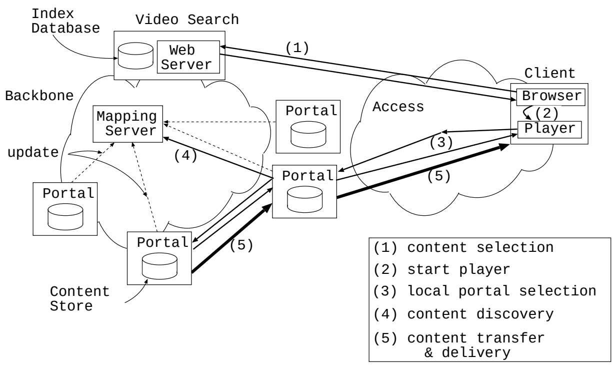

Figure 2: PRISM interaction

The local portal first determines if it is already receiving the requested content (for live content) or if it is available from its local store (for on-demand content). If either of these conditions are true, then the local portal can immediately start streaming the content to the client. If, on the other hand, the local portal does not have the requested content, it attempts to find it elsewhere on the PRISM network.

To find non-local content, a PRISM portal sends the content naming URI to a PRISM mapping server. PRISM mapping servers maintain a distributed database of available content. If the mapping server finds a source for the requested content, then it returns this information to the requesting local portal as a content location URI. The local portal can then use the content location URI to fetch the requested content and stream it to the client. Streaming through the local portal allows the client to perform VCR-like functions on the stream in a timely manner. However, there are cases in which it will not make sense for content to be streamed via the local portal. Therefore, as an option, the local portal might redirect the client directly to the remote portal after it has received the location URI from the mapping server. This is performed by means of the RTSP redirect method.

2.2 Content Naming and Discovery

Being able to identify and access unique segments of content is a critical component of the PRISM architecture. Identification is achieved through the use of PRISM's content naming scheme, and access is achieved by mapping the content's name to a file or location on a PRISM portal.

Content location URIs are significantly different from content naming URIs[1]. In the context of the Web, the distinction between what content a user is looking for and where it might be best obtained from has historically not been made. However, recent content distribution approaches are changing this through the use of a combination of proprietary naming and DNS mechanisms [4]. Explicitly separating out naming from location in the context of PRISM is important for a number of reasons. First, unlike the Web, there is no "origin server" for content. Live sources are not assumed to archive content. Second, timely distribution of high quality streaming content will depend more on it being served from "the most appropriate server" than would be the case for Web content. A third aspect concerns access to "historical" content. In the Web world, it is generally infeasible to ask for CNN's entry page on June 30th because it is constantly overwritten. Web content distribution is mainly interested in distributing the "current" version of the page rather than presenting the current and all "old" versions of a page. In PRISM, it makes sense to say "show me ABC TV aired on June 30th, from 9AM to 10AM."

Content naming URIs specify a high-level name for content. For example, a content naming URI could consist of a TV network name and time ("ABC at 10PM"). Content naming URIs are independent of PRISM. On the other hand, content location URIs specify a content file on a specific PRISM portal ("file 5 on `prism4.att.com`"). Separating content naming from PRISM content location allows content to be accessed in a variety of ways (e.g. schedule based or via content aware search engines) without having to make the portal structure of PRISM visible to the world.

2.2.1 Naming

There are three key design goals for PRISM's content naming scheme. First, the naming scheme should be independent of the PRISM architecture. This allows services to make use of PRISM's infrastructure without having to know the details of the architecture and similarly allows the naming scheme to be adopted by other systems dealing with broadcast content. For example, our scheme is consistent with the TV Broadcast URI Requirements document [5] which provides a framework for the naming of broadcast resources. Second, the naming scheme should be compatible with the Web. All content names in our scheme are in the form of URIs that can be embedded within Web documents. Third, the naming scheme should allow content to be identifiable to a fine level of granularity while at the same time not requiring such detail when it is unnecessary. For example, to properly capture localized content from a cable system it may be necessary to specify what cable head-end the content was sent over in order to allow for local content insertion/modification.

The syntax of our naming scheme is shown in Figure 3. The channel name (described in detail below) identifies a unique stream of content. The start and stop times are expressed as UTC times [1] (e.g. "utc:20000215T2200"). The program name is a text string identifying content within a particular channel, and the time offset is an offset relative to the start of a program in seconds (e.g. "pro-

[1] A naming URI is a Uniform Resource Name (URN) and a location URI is a Uniform Resource Locator (URL). In this paper we use the more descriptive naming URI and content URI respectively.

```
name      := "stv:" channel_name
            | "stv:" channel_name "?" spec
spec      := "start=" utctime
            | "start=" utctime ";end=" utctime
            | "program=" name
            | "program=" name ";offset=" time

channel_name      := "<" brand ";" channel ";"
                     distributor ";" location ">"
```

Figure 3: Naming scheme URI syntax

gram=nypd_blue" or "program=nypd_blue;offset=60"). A naming URI with only a channel name specified implies the current live version of that content.

The format of the channel name allows content to be identified at a fine level of granularity, but does not require it in the general case. For example, it may be sufficient to identify the TV network the content is on. However, to properly capture local programming or local advertising we need to distinguish between TV stations that are affiliated with the same TV network but are broadcasting from different locations (even if a significant portion of the content is the same for both stations). The same logic applies to cable-based networks that allow cable operators to insert local content into their feed.

In our naming scheme, the channel name consists of four elements, as shown in the bottom of Figure 3. The brand is the channel brand name users typically think of. This could be a simple identifying string such as "ABC," or it could be the fully qualified domain name associated with the brand. The channel is the call letters or channel number associated with the content. In some cases this field may be null. The distributor indicates the entity that is distributing the content. For example, the distributor could be the owner of a broadcast station, a cable company, a satellite company, or content distribution system on the Internet. The location is the source of the specified version of the content. The location can be used to indicate a specific broadcast tower or cable head-end. All the elements are optional. The meaning of unspecified elements in a channel name depends on the context in which the name was used. If the channel name is used in a channel listing query, then unspecified elements match all the available values for the given user. Otherwise, unspecified values take the default value for that user. In either case, the entity resolving the unspecified values has local knowledge of the user's profile so that meaningful values can be plugged in. Some examples of channel names are shown in Table 1. Note that end users need not be aware of the channel naming syntax as it can easily be hidden behind a higher-level user friendly Web interface.

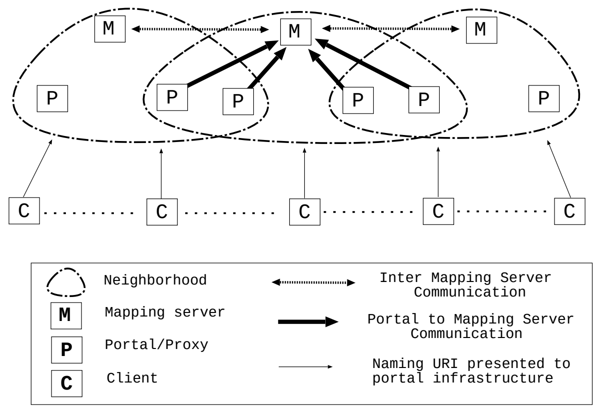

Figure 4: PRISM content location discovery

2.2.2 Discovery

When a local portal receives a content naming URI from a PRISM client it must determine where the requested content resides. If the local portal has the requested content in its local store then it responds immediately to the request. Otherwise, if the content is not available locally then the local portal must initiate a search for the content. PRISM's mechanism for handling content discovery is shown in Figure 4. Content discovery takes place by means of a distributed directory server called a *mapping server*. Each mapping server provides coverage for a neighborhood of portals. Neighborhoods overlap for redundancy purposes. Neighborhoods limit the scope of information distribution between portals and mapping servers. Detailed information is contained within a particular neighborhood.

Portals that have content available advertise this information to their local mapping server. The advertised content includes media information, time offsets, and some description of the current portal load. Some of the options under consideration for this detailed exchange include server and network load information which might dynamically change over short time scales. More static information, possibly limited to channel names, are exchanged between all mapping servers. Enough user and network profiling information is stored in the local portals and their mapping server to resolve unspecified elements in channel names (as discussed in the previous section).

3 Related Work

Our work consists of a system architecture made up of many components, and there is a substantial amount of related work. There is a large body of work in the area of video-on-demand [6] - our work includes a video-on-demand component, but is much more broad in its scope. There is substantial work in the area of interactive TV,

Channel name	Meaning (for listings)	Meaning (for requests)
<abc;;;>	list available ABC stations	the default ABC station
<abc;wabc;;>	list available source for ABC station WABC (broadcast, cable, satellite, etc.)	ABC station WABC from the default source
<;wabc;;>	list network and sources for station WABC	station WABC from the default source
<abc.net.au;;;>	list available Australian Broadcasting Co. sources	the default source for Australian Broadcasting Co.
<abc;;directv;>	list available ABC stations on DirectTV	the default ABC station on DirectTV
<;;comcast;>	list the available channels on Comcast Cable	the default channel on Comcast Cable (typically an informational channel)
<;;comcast;nj>	Same as above, but only in New Jersey	
<abc;wabc;comcast;orange_nj>	tests for ABC station WABC on Comcast's Orange, NJ system	ABC station WABC on Comcast's Orange, NJ system
<mtv;;akamai;>	list sources of the MTV network as distributed by the Akamai content distribution network	MTV as distributed by the Akamai content distribution network
<;;;>	list all available channels	the default channel

Table 1: Example channel names

which aims to use IP-style protocols for the control of, interaction with, and delivery of side information for TV channels delivered by conventional means [7]. We also consider this same capability, and in addition we consider the IP-based delivery of TV channels, which allows us to evolve away from the conventional broadcast TV model.

There is significant ongoing work in the area of content distribution technology (for example, the work of companies like Akamai, Sightpath, FastForward, Inktomi) - our work assumes the use of a suitable content delivery mechanism for delivering content into portals. We also employ a detailed and general location independent content naming scheme, whereas most of the related content distribution work utilizes URL-based naming schemes.

There are various services emerging that utilize home based consumer equipment to manage the recording and time-shifting of TV delivered content (e.g., ReplayTV and TiVo). Our work differs from this in that it considers the primary storage medium to be located in the network. This has a number of advantages. In particular, the storage is shared by multiple users and the library of shared videos is potentially vast. A particular user can benefit from another's recordings. The recording service can be made very reliable, and the storage can be backed up. The techniques of [2] could be advantageously used to index the recorded content. Furthermore, there are multiple recorders within a PRISM location, so that an end-user could schedule simultaneous recordings.

There has also been research in the area of the caching of streaming media ([8, 9, 10]). This research could potentially be used in caching mechanisms in the PRISM proxies. The problem of maintaining a distributed global directory of objects stored in Web caching servers was considered in [11]. In that work, the name space was based on URLs. The problem of referencing television broadcasts via Web browsers was considered in [12] and [13]. Both these approaches are less general than our naming approach. In summary, our work builds on the work of many different areas, expands significantly on the assumptions typically made for network access, and combines these components to provide a novel architecture and capability of supporting conventional TV programming, video-on-demand, and replay services.

4 Concluding Remarks

In this paper we presented PRISM - an IP-based architecture for broadband access to TV and other streaming media. This is a work in progress report, in which we explained the technical challenges, system architecture, and our proposals for content naming and discovery. The current implementation of the PRISM project consists of software for a high-volume IP video caching portal, live video server, and client. The system is currently being tested on a small-scale cable testbed, and preparations are being made for a larger test using consumer cable networks and a widely distributed portal network. Our research in this area has given us insight into the advantages of IP-based content delivery over traditional TV delivery systems. There are several remaining issues which we intend to address as part of PRISM. These include designing protocols and media distribution mechanisms for use between portals, and dealing with scale as the system grows to accommodate globally-accessible content distribution and service reach.

References

[1] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)." RFC 2326, April 1998.

[2] D. Gibbon, A. Basso, R. Civanlar, Q. Huang, E. Levin, and R. Pieraccini, “Browsing and Retrieval of Full Broadcast-Quality Video.” Packet Video, New York, NY, April 1999. Available from: `http://mediabank.research.att.com/sbtv/`.

[3] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform Resource Identifiers (URI) : Generic Syntax.” RFC 2396, August 1998.

[4] Akamai. `http://www.akamai.com`.

[5] W. ten Kate, G. Thomas, and C. Finseth, “TV Broadcast URI Schemes Requirements.” `http://www.w3.org/TR/TVWeb-URI-Requirements`, October 1999.

[6] S.-F. Chang, “Video on Demand Systems: Technology, Interoperability and Trials,” *Kluwer Academic Publishers*, 1997.

[7] G. Blacker, J. Bennington, J. Boltax, K. Gage, M. Garcia, M. Richmond, M. Vickers, S. Watson, and D. Zigmond, “A Framework for Interactive Television based on Internet Standards.” `http://www.atvef.com/library/9806_framework.html`.

[8] J. Rexford, S. Sen, and D. Towsley, “Proxy Prefix caching for Multimedia Streams,” *IEEE Infocom*, April 1999.

[9] A. Dan and D. Sitaram, “Multimedia caching strategies for heterogenous applications and server environments,” *Multimedia Tools and Applications*, May 1997.

[10] R. Rejaie, M. Handley, H. Yu, and D. Estrin, “Proxy caching mechanism for multimedia playback streams in the Internet,” *Proc. International Web Caching Workshop*, March 1999.

[11] S. Gadde, J. Chase, and M. Rabinovich, “A Taste of Crispy Squid.” Workshop on Internet Server Performance, June 1998. Available from: `http://www.research.att.com/~misha/crisp/abstract.html`.

[12] D. Zigmond and M. Vickers, “Uniform Resource Identifiers for Television Broadcasts,” *draft-zigmond-tv-url-04.txt*, January 2000.

[13] D. J. Blackketter, M. A. Dolan, C. A. Finseth, G. Thomas, and D. Zigmond, “The Local Identifier (lid:) URI Scheme,” *draft-blackketter-lid-00.txt*, February 2000.