Design, Implementation and Operation of a Large Enterprise Content Distribution Network

J.E. van der Merwe, P. Gausman, C.D. Cranor and R. Akhmarov
AT&T Labs - Research
Florham Park, NJ, USA

Abstract

Content distribution networks (CDNs) are becoming an important resource in enterprise networks. They are being used in applications ranging from broadcasted townhall-style meetings to the distribution of training and educational material. Building an effective CDN in an diverse enterprise environment is a challenge. In this paper, we present the design, implementation and operation of a large enterprise CDN and describe some of the lessons we learned from our experiences building such a CDN.

1 Introduction

The use of streaming media is gaining popularity in many environments where it can be used for both live Webcasts and on-demand access to streaming content. This is especially true in corporate environments where streaming is used for employee town-hall meetings and training courses. One of the main reasons for this gain in popularity is advances in encoding technology that enable reasonably good quality video to be obtained at fairly modest encoding rates. While its popularity is increasing, large scale distribution of streaming content remains a challenging problem in networks that includes a wide-area-network (WAN) component where link speeds can be a limiting factor.

In this paper we describe the design, implementation, and operation of a large scale corporate enterprise content distribution network (CDN). The CDN provides streaming services to more than 400 distributed sites connected by a private IP backbone in a large corporate environment with a total user population in excess of 50000 users. The CDN enables Webcasts throughout this environment scaling from departmental meetings with tens of users to all-employee broadcasts involving potentially all users.

Creating a streaming CDN in this environment was technically challenging for the following reasons:

- The scale of the target environment is quite large, not only in terms of the number of sites to be covered, but also in terms of the potential number of simultaneous users that a Webcast such as a live all-employee broadcast can attract.
- Target sites are connected to the corporate backbone through WAN links that have widely varying link speeds. Additionally, WAN links are shared with other enterprise applications, so the streaming service bandwidth usage cannot be allowed to fully monopolize a link's bandwidth.
- Different parts of the target network have differing capabilities in terms of the transport options available for content distribution. The IP backbone is multicast enabled as are all of the sites connected to the backbone. However, approximately one third of the sites, while multicast capable internally, have no multicast connectivity to the backbone. Also, in our environment users who connect to the corporate intranet via VPN only have unicast transport available.

A key design goal in addressing these issues is to create a solution that minimizes and simplifies human operational involvement. The system should be fully automated with zero human intervention necessary during normal operation. Tools necessary to aid in fault management when problems occur should be available in a self-serve form on the Web so that capable users can quickly diagnose their own problems without having to wait for someone from a "help desk" to help them. We describe our approach and the system that was deployed in detail in this paper.

It is important to note that the challenges in running a successful streaming service in an enterprise environment are not all purely technical. As with any system deployed throughout a large organization, many players are involved. For a streaming service such as the one described here, that includes end-users, the network provider, a support organization (help desk), content owners and the content service provider. These roles will typically be present in any enterprise deploying streaming services. Some of the roles could be combined and performed by the same organization, or some of the roles might be performed by third parties in an outsourcing scenario. In this paper we address

the problem mainly from the point of view of the CDN which might be operated by the network provider, the content service provider or a third party. However, to provide a holistic view we also describe the Content Service Provider requirements in some amount of detail and touch on interactions with other role players.

The outline of the paper is as follows. Section 2 describes the higher level services that the CDN accommodates and the underlying network infrastructure in which the CDN was realized. The design of the CDN and in particular the redirection system is described in Section 3. In Section 4 we describe operational issues and relate some of the lessons learned in "operationalizing" an enterprise CDN. We cover related work in Section 5 and end the paper with a conclusion and future work section.

2 Service Perspective and Network Infrastructure

The work presented in this paper describes the realization of a set of services that attempts to capture a common set of streaming requirements:

Presentation service: provides tools and functions for streaming presentations (by reservation), from pre-existing encoder rooms. Users can optionally select to use various rich-media components within the presentation browser window, like synchronized slides, Q&A applications and surveys. Presentations are typically live and can be archived for later On-Demand or scheduled Channel viewing (see below).

Channel service: provides a content channel for related programming. (Like a cable TV station, only over IP). For example, the public relations organization within a company might "own" a channel to distribute all of their content. Content is live or replays of pre-recorded content.

On-demand service: provides all functions needed for the publishing, management, storage and distribution of pre-recorded presentations that can be accessed in an on-demand fashion.

Venue service: provides high bandwidth streaming distribution to sites with large displays for group audiences. Typically used as a substitute for expensive satellite distribution mechanisms where practical.

To provide these streaming services, two functional components are required:

The publishing platform: provides content owner and end-user user interfaces and serves as a central publishing repository. The publishing platform also provides presentation tools, billing, and content management functions. The publishing platform thus serves as the main source of content for the CDN. It further interacts with the CDN by selecting distribution regions, setting content aging parameters, and turning encoder feeds on and off, automatically, as per content owner and service administrator configurations. As the focus of our work is on the distribution platform, the publishing platform will not be considered further in this paper.

The distribution platform (CDN): distributes Web and streaming content and is described in detail in the next section.

The corporate network in which the CDN is deployed consist of a private IP backbone distributed across the US. The backbone interconnects more than 400 distributed edge sites housing from tens to thousands of employees per site. Depending on its size and function, edge sites connect to the backbone network at line speeds ranging from 1.544 Mbps or less (T1 or fractional T1) to 45 Mbps or higher (full T3 or OC3). Within each site the LAN is typically a switched fast-Ethernet which also connects some legacy 10Mbps segments. Several data centers connect to the backbone network. The backbone runs a single multicast domain using Protocol Independent Multicast - Sparse Mode (PIM-SM) [1] routing which extends into most of the edge sites. Approximately 20% of the edge sites form Distance Vector Multicast Routing Protocol (DVMRP) [2] multicast islands isolated from the backbone multicast domain. Virtual-office users access the corporate network through a number of virtual private network (VPN) gateways that are also located in different data centers.

3 CDN Architecture

Figure 1 shows the architectural components of our CDN network. The CDN receives its content from encoders that are located in meeting rooms distributed throughout the organization. For the *Presentation Service* we are currently offering a fixed number of content *channels*. Based on a reservation mechanism in the publishing platform, at different times different encoders provide the actual content for each channel. Thus, content providers essentially "rent" time for their presentation on a fixed channel. To accommodate the varying amounts of bandwidth that is available to different sites, each channel is provided at multiple encoding rates. Our intelligent redirection system (described in detail in Section 3.1 below) is used to connect viewers to the appropriately encoded stream based on various parameters such as IP subnet and the bandwidth allocated to the site. In our current deployment we only use Microsoft Windows Media Technologies.

Each channel's streams are sent from its encoder to a set of one or more media servers located in central data centers. As shown in Figure 1, the data center also houses the publishing platform, the redirection system, and the caches that

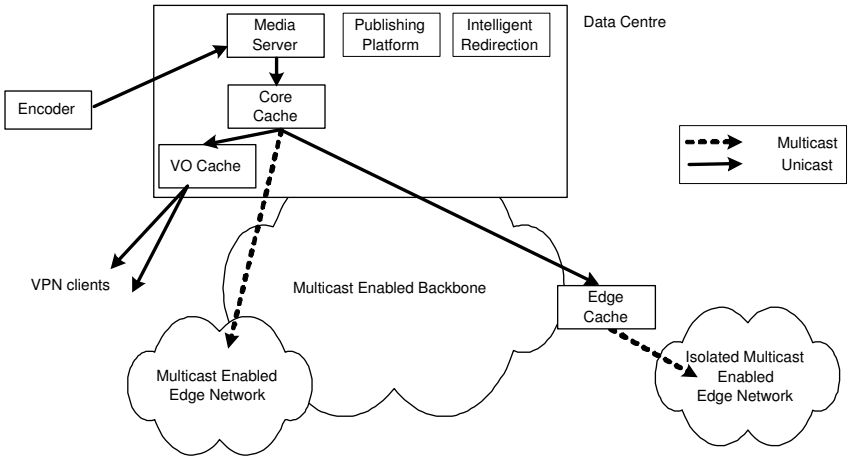

Figure 1: CDN architecture

make up the distribution network. The media servers serve as the content origin for both live and on-demand content.

As shown in Figure 1 the media server is front-ended with a core streaming cache. These core caches are typically special purpose streaming appliances whose sole role is to stream data. In addition to being able to handle many more streams than the media server, the core caches are also the root of the distribution tree. In our implementation only the core caches request content from the media servers. All other requests (from other caches or end-users) are handled by the core cache.

Once the media reaches the core cache, it can be streamed out in three ways depending on the location of the end-user. First, if the end-user is at an edge network that is part of the backbone multicast domain, then the core cache can use multicast through the backbone to stream it directly to the client. Second, if the end-user is located in an isolated multicast enabled edge network, then the core cache can unicast the media to an edge cache placed on the end-user's edge network. The client can then receive the media via multicast from the edge cache. Finally, for end-users residing in VPN-based virtual offices we have a special set of virtual office (VO) caches. The core cache streams the media to the VO caches via unicast, and then the VO caches stream the media onward to the VPN clients also using unicast

3.1 Redirection System

In order for our CDN architecture to function properly, it is essential that end-users are directed to the appropriate cache for their location, use the correct type of URL (i.e. a unicast URL or a multicast URL) and request the appropriate stream quality based on the bandwidth allocation for the site. Note that in the case of the isolated multicast islands, this is critical for correct operation as end users in those domains will not be able to receive multicast from the backbone multicast domain. We have developed a "content-aware," "location-aware" application-layer redirection system to seamlessly glue the system together while hiding all the technical details from users of the system. The architecture of our redirection system is shown in Figure 2.

To properly redirect clients, the redirection system needs to understand the topology of the intranet that it is operating in. The redirection system uses as input a set of configuration files obtained from the different role players in the company. These files contain a list of bandwidths allocated for each site, a specification of what streams a site with a specific bandwidth allocation should get, and the subnet information for each site as well as the VO subnet ranges. As far as the CDN is concerned this process is fully automated as these files are posted on a Web site from where it is pulled on a daily basis (or on-demand as needed). The organizations responsible for generating the files are currently using systems with varying levels of automation. There are only a small number of VO subnets which change infrequently and this file is updated manually as needed. Assigning allowable bandwidths to each site is a semi-automated process that involves running a script (currently on at least a quarterly basis) that look at utilization trends on the WAN links. Finally, the process to create the per site subnets is fully automated and harvests subnet information directly from operational routers. This is crucial in a large network as subnets change on a daily basis and attempting to maintain this by means of a manual process would be error prone and therefore problematic.

Figure 2 also shows the "override files." These are a set of files that a system administrator can use to override settings in any of the regular feeds. For example, if an omission in the subnet file is detected, it can immediately be corrected through these files without waiting for the actual subnet discovery process to be corrected and/or debugged.

The configuration files are parsed and manipulated to translate the information they contain into a set of Intel-

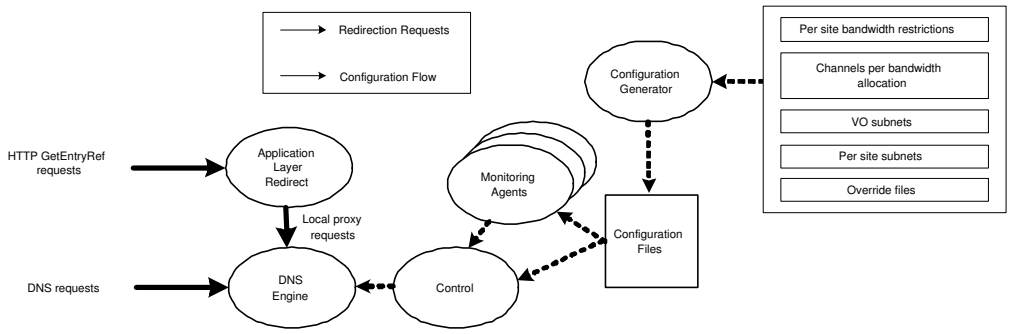

Figure 2: Redirection System

ligent Domain Name Services (IDNS) configuration files. IDNS [3] is a special purpose, intelligent DNS server. IDNS consists of three main components: a DNS engine that processes queries based on the IP address of the client and current network conditions, a control process that dynamically reconfigures the DNS engine as network and configuration conditions change, and a set of monitoring agents that track server health. When configured with a set of subnets, proximity information, and current status, IDNS will resolve DNS requests into an IP address or CNAME that is close to the client requesting the resolution.

For our CDN application this basic DNS-based redirection have been extended to enable application level redirection which works as follows. First, end-users request streaming content, typically by selecting a URL on a Web page hosted by the publishing platform. As is often the case with streaming, this Web object is a streaming meta-file (e.g. an ASX file) which contains the streaming URL. In our system the meta-file does not contain a static streaming URL, but rather contains an HTTP URL that points to one of our redirection server machines. The redirection server is then contacted to obtain the actual streaming URL.

Each redirection server runs a special purpose Web server that receives the Web request for the URL embedded in the meta-file. The Web server uses the IP address of the end-user and the channel being requested and encapsulates it into a DNS request to the locally running IDNS server. The IDNS DNS engine receives the request, extracts out the client's IP address and uses that to generate an answer to the query. In our case the answer is an encoded CNAME record. The CNAME record is sent back to the Web server, which decodes the record and uses it to dynamically produce a streaming meta-file (or a portion of a streaming meta-file) that contains the actual streaming URL. Note that as indicated in Figure 2, the redirection system also serves as a "normal" DNS server at the same time.

The CDN was designed to have no single point of failure. The CDN data center equipment is duplicated in two different data centers. Monitoring agents that are part of the redirection system (see Figure 2) monitor the health of the equipment in each data center, and feed this into the redirection system in real time. The media server/core cache combination in the active data center carries all of the load. Should either of these devices go down or connectivity to the data center as a whole be lost, the redirection system will move all traffic to the media server/core cache in the other data center. The redirection system itself is fully redundant with two separate servers in each data center. Redundancy between VO-caches is handled in similar fashion by the redirection system.

4 CDN Operation and Lessons Learned

The long term success of a service such as the one described here largely depends on its operational costs. As stated in the introduction, one of our main goals when designing the service was to reduce human involvement during normal operation of the system. The fully automated redirection system presented in the previous section is aligned with this goal. In this section we consider operational aspects of the system where human involvement can not totally be eliminated. None the less, we attempt to provide tools and systems to also reduce or simplify human involvement with these aspects.

Day-to-day operations involving the CDN fall in two categories:

Pro-active management and monitoring: involves monitoring all equipment via a network management tools, generating alarms when certain conditions occur on the CDN devices, and monitoring the redirection system for error conditions.

Reactive management: involves determining where the problem is when an end-user is unable to receive a Webcast. These type of problems (described in detail below) are somewhat unique in a streaming scenario.

In our environment, when an end-user selects a channel from the publishing platform and fails to receive the stream the cause can be any of a number of reasons. The user

might be experiencing a **desktop** problem such as miss-configured software. Or there could be a **network** problem which could either be local to the user's location or somewhere else in the network (e.g. between the user's location and the backbone). The problem could be due to a **redirection** error. For example incorrect subnet information might have been entered into the configuration feeds resulting in users being redirected incorrectly. Finally there could be a problem with one of the **CDN** devices or the with the media **source** (the encoder).

Given this array of potential problems and our desire for a fully automated environment, we have developed a set of Web based "debugging" tools that allows users and support personal to quickly rule out and/or identify problems. In particular the tool for end-users tell them what their IP address is and whether the redirection system had any problems dealing with that address. The tool then tries to stream a small clip from the CDN infrastructure, which, if it fails, will point to some sort of a network or CDN problem, and, if successful, points to a probable multicast problem. Similarly, the tool for the support personnel allows them to see where the redirection system will redirect based on the user's IP address. Through the tool they can also verify that all CDN equipment involved in this distribution is functioning correctly and that network connectivity exists between the CDN equipment and the end-user.

Since the streaming service has been operational for only a few months, the webcasts hosted so far have been fairly modest. Audiences ranged from 50 to 300 persons for the Presentation Service and 100 to over 8000 persons for the Channel Service. Both services have distributed to audiences located in from three to over a hundred sites at one time, while simultaneously distributing to Virtual Office viewers.

In deploying the CDN infrastructure described in this paper we learned some lessons and indeed confirmed some of our suspicions. For the most part realizing the service happened without any major issues. That said, there were the normal difficulties involved with deploying a large distributed infrastructure. Configuration of all equipment was performed at a single location. This process itself was highly automated which significantly reduced the possibility of miss-configuration. The logistics of having to ship a large number of boxes first to the place where they are configured and then to where they are installed is not very efficient or cost effective though. If the ability existed to ship an un-configured device and separate media with per device configuration (e.g. a CD-rom or flash memory), thereby allowing simple remote configuration, that would have greatly simplified this part of the process. By necessity the physical installation of equipment at remote sites were performed by technicians who did not necessarily have training and/or experience with the specific equipment used. Having mostly appliances that could simply be connected and switched on allowed this to happen. (Although we did have our fair share of "remote debugging by phone" during this phase.)

Once deployed the streaming caches are "centrally" configured by having the caches periodically poll a configuration server to check for new configuration files. Since most of the caches share the same basic configuration with only a small delta that is cache-specific, this approach proved very effective in updating the configuration of the complete CDN infrastructure in a timely fashion. (We have out-of-band access to all equipment to enable us to recover from any miss-configurations or to remotely administer network connectivity changes.)

The use of multicast enabled our streaming CDN service to be highly network efficient and is therefore highly desirable. However, most of the initial technical issues we encountered were due to the use of multicast. This included initial problems with multicast in the CDN vendor products. This is probably because those functions have not been used as extensively as their unicast counterparts. This was also true for the multicasting networking equipment (i.e. routers and switches). Multicast deployments of this scale are not that common in enterprise networks, with the resulting lack of real world operating experience for those products.

We mentioned in the introduction that many players are involved in realizing an enterprise streaming service. This means that when things go wrong it is not always clear who is responsible for fixing it. The debugging tools described earlier in this section not only help to identify the problem but also help to identify which group has responsibility for it, thus making the fault management process much more efficient.

Finally, in testing and operating the system we soon discovered that for technical or business reasons, some subnets and/or locations were not able to receive the streaming content. Not wanting to re-discover these problem areas every time and in keeping with our desire for automation, we have dealt with this problem by having a set of "known-problem-subnets." The IP address of an end-user experiencing problems is automatically checked against these subnets when she accesses the debugging tools, informing her whether the service is expected to work for her subnet.

5 Related work

While the use of CDN technology in enterprise environments are fairly common these days, we are not aware of published work that report on the operational aspects of an enterprise CDN. Our own work on the optimal placement of streaming caches that form an overlay distribution infrastructure in a VPN environment is clearly related [4]. However, that work was intentionally limited to a unicast-only environment and did not report on an actual CDN realization.

In terms of redirection, some related work is summarized

in [5]. While it mentions streaming redirection, the focus is mainly on redirection for Web content. The fact that streaming normally involves the downloading of a metafile, offers a natural redirection opportunity. While this is widely recognized, we are not aware of published related work that follows the same approach as our redirection system. Other redirection related work involves the products of streaming vendors (e.g. Network Appliance and Cisco). The lack of flexibility of these products, for example, in terms of dealing with large numbers of subnets and taking different constraints into account, prompted us to develop the solution presented in this paper. There are also more forward looking approaches for scalable redirection that would form an integral part of the Internet suite of protocols [6]. Except for the fact that approaches such as these are not available yet, it is not clear that it would provide the flexibility needed to address a complex scenario such as the one presented in this paper.

Finally, the fact that our system takes into account the bandwidth available at a specific site has some related work in [7]. They developed routing algorithms for overlay multicast networks that optimized the bandwidth usage between the nodes forming the overlay network. Our flat distribution "tree" directly from the core caches to the edge caches was realized mainly for pragmatic reasons. However, it appears to be sensible in an environment (such as ours) where the bandwidth constraints are mainly in the WAN links connecting the edge sites to the backbone network, rather than in the backbone itself.

6 Conclusion and Future work

In this paper we presented the design and operation of a streaming CDN in a large enterprise environment. This CDN is novel in its use of multiple transports and multiple explicit bitrates to accommodate varying bandwidth conditions in a large enterprise network. We showed the critical importance of an intelligent and flexible redirection system in making this work. We stressed the importance of automated tools in operations and debugging and described some of the tools we use.

From a service perspective in the future we plan to extend the service suite to include secure distribution of content into and out of the enterprise environment. The presentation service will be extended to allow support for ad-hoc presenter owned encoders, e.g. laptops or PCs anywhere in the enterprise network. We plan an **event service** to enable presentations with multiple origination sites and multiple audience sites. Finally, we are in the process of increasing the footprint of the CDN to include several non-US networks that form part of the corporate enterprise.

For the content distribution platform we are currently investigating the use of a bandwidth probing tool to augment the per location bandwidth assignments. This is particularly relevant for virtual office users. Right now a single rule is assigned to all VO-users. However, VO-users might be using the VPN-access to the network from a variety of access technologies (e.g. dial-up or cable modem). This means that VO-users currently have to run at the lowest available bitrate per channel. Running a bandwidth probe to determine the actual available bandwidth and taking that into account when doing redirection will eliminate this problem.

The use of multicast is very desirable from a bandwidth efficiency point of view. However, once an error condition has been identified as a multicast problem finding the exact source of that problem remains difficult. We are investigating multicast debugging tools for this purpose.

Another issue related to our use of multicast is that, unlike with unicast streaming, there is no explicit quality feedback from the end-user players. This is due to the fact that when the media player is receiving multicast there is no control connection back to the streaming server/cache. This makes it difficult to determine the quality of the actual user experience unless users explicitly complain. We are investigating the use of tools that would provide explicit feedback about the quality of the stream even in a multicast environment.

References

[1] D. Estrin et.al., "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification." RFC2362, June 1998.

[2] D. Waitzman and C. Partridge and S. Deering, "Distance vector multicast routing protocol." RFC1075, Nov 1988.

[3] A. Biliris, C. Cranor, F. Douglis, M. Rabinovich, S. Sibal, O. Spatscheck, and W. Sturm, "CDN Brokering." Sixth International Workshop on Web Caching and Content Distribution, June 2001.

[4] Z. M. Mao, D. Johnson, O. Spatscheck, J. E. van der Merwe, and J. Wang, "Efficient and Robust Streaming Provisioning in VPNs." 12th International World Wide Web Conference (WWW'2003), May 2003.

[5] A. Barbir, B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck, "Known CN Request-Routing Mechanisms ." RFC 3568, July 2003.

[6] Mark Gritter and David R. Cheriton, "An Architecture for Content Routing Support in the Internet." USENIX USITS, March 2001.

[7] Sherlia Y. Shi and Jonathan S. Turner, "Multicast routing and bandwidth dimensioning in overlay networks." IEEE Journal on Selected Areas of Communication, Oct 2002.