Multicast Caching: Efficient Distribution of Encrypted Content to Mobile Clients

JANNE LUNDBERG, CATHARINA CANDOLIN
Laboratory for Theoretical Computer Science
Helsinki University of Technology
P.O. Box 9201, FIN-02015 HUT
FINLAND
jlu@tcs.hut.fi, candolin@tcs.hut.fi

Abstract: - Multicast caching is an efficient way to reduce traffic in the core network, and for distributing information to a large group of clients. Problems that are are associated with caching multicast data for mobile clients are different from, for example, HTTP-caching. Reliable transport protocols cannot usually be used, and varying wireless link quality increases the probability of packet loss. In this paper, we describe packet level multicast caching, its possibilities, and show how it can be used to efficiently distribute information to paying customers. We present the system which we are building, and also discuss what kind of information can efficiently be cached and distributed using multicast.

Key-Words: - multicast, caching, mobility, requirements, wireless

1 Introduction

Multicast is a well understood and efficient way of sending data simultaneously to multiple receivers while conserving network capacity. However, it requires that all clients are available and listening when the data is being sent. Such assumptions are usually not possible when data is meant to be received by hosts which are connected to the Internet through some wireless technologies and the hosts are assumed to be mobile. Similarly, when a host roams between access points in the network, it may lose packets while it is contacting the new router. In both cases the multicast receiver needs some alternative method of recovering from packet loss. Also, the system needs to be able to send encrypted data, if it is used by content providers to send data for which users are expected to pay. While using the basic service of IP multicast [1] for mobile hosts can be done, some changes are needed for multicast to be fully usable.

In this paper, we provide a solution for the problem of distributing multicast data to mobile hosts which are connected to the network using wireless links. We describe caching related problems which are unique to multicast in an environment with mobile hosts, where cryptographically protected data needs to be distributed. We also present a solution to these problems, and discuss a business model that this architecture makes possible.

The rest of this paper is organized as follows. In Section 2 we present some relevant technologies and problems which are typical to multicast caching. Section 3 presents the we architecture which we propose to solve the associated problems in IPv6 [5]. Finally, Section 4 presents our conclusions.

2 Background and related work

A lot of work has been done on using multicast distribution to improve scalability of Web services. In [2], this is done by transmitting the most popular pages repeatedly through a predefined multicast group. Reliability is achieved by dividing the data into chunks, and waiting until all chunks that build a page are received.

In [3], the authors consider the benefits between repeatedly transmitting data through multicast and distributing it through a hierarchy of Web caches. The paper concentrates on fre-

quently changing documents, and the authors conclude, that the best overall performance is achieved when both multicast distribution and hierarchical caching is used.

Both of the above papers, as the rest of the multicast research, focus mostly on distributing Web pages, which are sent in clear text to clients which are connected to fixed networks. Caching of streaming data which does not consist of fixed length files has been mostly ignored. An important reason for this may be, that in the traditional form of multicast with many simultaneous senders, this would have required the caches to store the timings of received packets to create a valid picture of the multicast session. However, the inter-host timing problem is removed when only one sender is allowed for each group. Also, adopting the Source Specific Multicast (SSM) [6] model, which is described further in Section 2.3, simplifies the problem.

2.1 Multicast caching

Multicast data differs significantly from data that is transferred by, for example, HTTP. Thus, the principles of cache operation cannot be directly copied from Web caching. The most important differences are:

- Data loss is frequent and allowed. Data that is sent using multicast cannot be guaranteed to be received in its entirety be all recipients. Work is being done by the IETF to standardize protocols for reliable multicast [7] [8], but such schemes usually do not scale to very large groups and some information may still be lost.

- Data can be stream oriented and may not have a natural beginning or an end. Such data cannot be cached by traditional methods which store and retrieve data as simple files.

- Since data can be stream oriented, earlier parts of data may need to be deleted before the entire data is received. For example, a cache that stores the latest 24 hours of programs from a TV-channel, will need to start removing old data while it is receiving new data from the same stream.

- Data may need to be encrypted. Since multicast data may be transmitted to a large audience at one time, it is difficult to provide access control to the data by limiting who can receive it. If a content provider wishes to gain money from the information it is distributing, it must be able to protect the data using cryptographic methods. Also, since data can be lost during transport, the data must be decryptable even if a number of packets are lost from the stream.

2.2 Mobility needs

Supporting multicast for wireless and mobile hosts places a number of additional requirements to the architecture.

- The mobile host may need to contact a new access router frequently and unexpectedly. Many future wireless high speed tehnologies will unavoidably have an even smaller cell radii, and access router changes will become even more frequent in the future.

- The cache with which the mobile client is communicating may change whenever the client changes its access router. While one cache may span several access routers, cache changes due to mobility can be frequent. Therefore, creating new associations with caches must be done quickly and efficiently.

- The available network capacity can vary by several orders of magnitude. Less than 10 kilobits per second capacities can be obtained with, for example, GPRS in congested areas, whereas other systems such as IEEE 802.11b may be able to provide over 10 megabits of bandwidth. The amount of available capacity may also change very rapidly, especially with vertical handovers.

- A high percentage of data can be lost during the wireless last hop. Since multicast is usually assumed to lose packets even within the fixed network, the higher protocol layers are usually already able to adapt to some level of packet loss, but missing packets will always result in worse user experience.

It is worthwhile to note that the requirements do not need to contain the ability to maintain a permanent IP address. Multicast is inherently a connectionless technology and does not therefore require functionality such as Mobile IP to operate.

2.3 SSM Multicast

Source Specific Multicast (SSM) [6] is a simplification of the traditional type of multicast. Usually, a multicast group is identified by the destination address in the IP-header. In SSM, a group is not defined only by its destination address, but by the source address and the destination address in the packet. The major benefit of Source Specific Multicast is that hosts can generate new global multicast groups without any inter-host coordination [4]. This results in a great simplification over the traditional multicasting model which requires complex negotiations to allocate an address for transient multicast groups.

2.4 IPsec

IPsec [10] provides a standard way of encrypting data with little overhead in packet headers. The Encapsulating Security Payload (ESP) [9] provides connectionless authentication, and encryption. Also, packets are given an order with the sequence number field. When ESP is used in transport mode, the placement of the header is illustrated in Figure 1. Each packet can be decrypted and authenticated separately from other packets, so if packets are lost, the remaining packets can still be decrypted.

Figure 1: ESP-packet

3 Solution

In this section we give a high level view of the system without going into details of the needed protocols.

3.1 Architecture

The solution is based on the concept of storing multicast data as packets in the caches. When a server transmits a multicast packet into the network, it includes an IPsec ESP header [9] into the packet. The ESP-header performs multiple tasks in the packet:

1. ESP provides protection for the data against disclosure threats while the packet is forwarded through the network. This is the traditional use, for which ESP was originally designed.

2. Content encryption. The data is stored in its encrypted form on the caches in the network as well as on the local cache on the users hard drive until the user has received the necessary keys to decrypt the data. This is different from the previous item, since in this case the data is being protected against disclosure to the recipient.

3. Packet identification and ordering. While the source and destination addresses in SSM multicast together identify a multicast packet stream, the sequence number field in the ESP header gives each packet its correct place within the packet stream.

4. It provides the ability to decrypt each received piece of information separately from other data. Even if neighboring packets in the stream are lost, all packets can be decrypted and used.

When a cache receives a multicast packet from the network, it stores the data on disk as the exact packet format in which it was received. The goal in using ESP is to give each packet a globally unique identity that can be used to request the packet from any available cache in the network. The cache is built so, that it can easily find any packet which it has stored, by only using the source and destination addresses, and the sequence number in the packet. The packets that are stored in the cache can be encrypted using IPsec, and the caches do no need to know the key to the data. A simple protocol is needed for clients to request missing packets from caches.

Figure 2 shows the overall view of the system. This illustration gives a high level view on the operation of the architecture. The phases of the operations are marked in the picture beside the arrows.

In phase 1, the multicast server sends data to the network. The server may send packets at a rate which enables clients to utilize the data without caching, or at any rate which can be handled by most of the paths in the multicast tree. When a multicast packet reaches a cache which is configured to store this data stream, the cache writes the data on disk. If the cache is located at an access point to the wireless network, the access point may also simultaneously send the data through its wireless interface if interested clients are nearby. The data can simultaneously be stored in multiple caches, which do not need to know the keys with which the data is encrypted. Neither do the caches need to coordinate their operation with other caches.

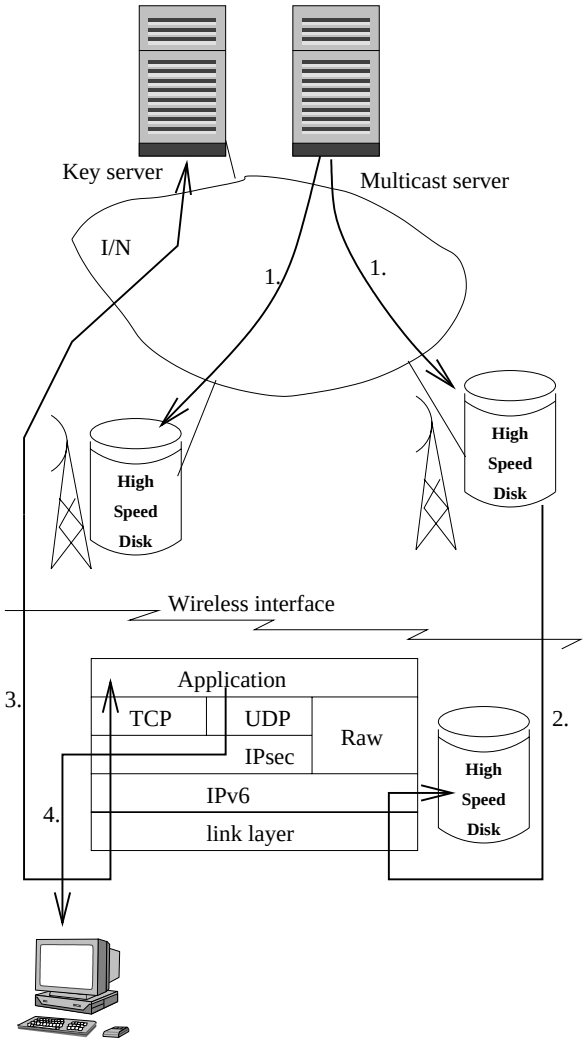

Figure 2: Overall architecture

In phase 2, the client device has arrived at a location where good network coverage and affordable transmission costs allow it to download the data it needs. The client requests the local cache to transmit the data which the client wishes. The client can request individual packets from the multicast cache, or it can request the cache to send some known range of packets, which are identified by the source, destination and sequence number fields in the packet headers.

The client does not need to receive all of the packets from the same cache. If the client moves away from the network coverage of the current cache, the client can search for a new cache. Since packets have a global identity which is based on the source, destination, sequence number fields, all other caches have the same notion of packet identity and can deliver the needed packets to the client. When a client receives a packet, the packet is written to disk without processing the ESP header. That is, the client can receive data that it cannot decrypt because it does not have the necessary keys. The disk in the client works as the lowest level of cache hierarchy, within the client.

In the optional phase 3, the client requests the decryption key it needs to decrypt the ESP-protected packets. This phase can also occur during packet reception or even before it. If the ESP header is not used to encrypt the packet, but only to provide the sequence number information, this step is skipped. The content provider can use any type of key distribution scheme, and bill the client in any way it wants. The client also benefits from the system, since it is necessary to pay only for the data that has been decrypted for use, not for the data that is received. That is, the system allows the client to receive data when it it the most convenient, but to pay only once the data is used. Since the key distribution scheme can use a very low bandwidth connection, it can be performed over wireless technology which is available almost everywhere. One alternative is to use GPRS to perform the key distribution.

In phase 4, the client has received the data and the keys it needs to decrypt the packets. The client can now rebuild the stream and use it as if it was being received from the network.

3.2 Multicast server

Only minor modifications in the operation of the multicast server are needed to support the architec-

ture. The server only needs to add an ESP-header into each packet it sends. It also needs to allocate a new source and destination address pair to each separate stream, to enable clients to distinguish between streams.

The server can either send data at the rate which clients can consume it if the data is stream oriented, or it can attempt to maximize throughput if the data is file type information such as newspapers.

3.3 Multicast cache

When a cache receives a multicast packet, it determines whether or not the packet belongs to a stream which the cache is maintaining. The decision is based on the source and destination addresses in the packet. If the group which is identified by the addresses does not match any cached stream, the packet is dropped.

The cache is not required to do any further processing to the packet. The cache stores the packet on disk in any way that it can use the source, destination, sequence number identifier to quickly locate the packet, when it is requested by clients. Associated to the packet is also an approximate time when the packet was received to enable the cache to destroy obsolete data.

The cache implements a protocol which the client can use to request packets of a given stream. The client also indicates the maximum speed at which it wishes to receive the data.

3.4 Multicast client

The client also acts as a multicast cache. The difference in comparison to the the caches which are located in the fixed network, is that caches which are located in the clients, are usually located behind a network interface which uses some wireless technology to transmit data. Whereas the caches which are located in the network, attempt to store all the information that it receives, the cache which resides in the client device attempts to minimize the amount of wireless network capacity which is required by the device.

Applications which wish to receive multicast data from the local cache, uses protocols such as IGMPv3 or MLDv2 to request data delivery from the cache.

3.5 Possible business model

One of the most difficult problems in distributing content such as movies over the Internet has been the sheer size of the data, which in case of movies can easily be hundreds of megabytes. Another problem is billing for the information that has been delivered to users.

The system provides a solution to both problems. Multicast technology saves bandwidth in the core network to the point that distribution costs play little or no role within the core network. On the client end of the network, the system enables the user to handle high bandwidth data transfers at a time when he is within the coverage of an affordable high capacity network, thus reducing the transfer costs to the users. Also, since the data can be encrypted even while on the hard disk of the user, the user does not not need to pay for the content until he decides to purchase keys to the data from a key server.

3.6 Types of data

The types of data that can be used to distribute with the system includes, for example.

Magazines of newspapers. Users can download their favorite newspaper from the cache at the local train station. When onboard the train, the user can choose whether or not to buy the keys which are needed to access the encrypted information.

Television can be adapted to use multicast technology. The system can replace or extend both VoD and ordinary unencrypted technology. In addition, the caches can be used to download data that was missed by the user at the time it was originally sent.

Teletext type of services. These services are very low bandwidth, and they are updated often. This can be, for example, a primitive type of newspaper, which is constantly being updated into the clients' hard disks. The client can view the information any time. When the client comes into the coverage of the access network, the client device automatically updates the information from the local cache.

Transmitting data through multicast is always a compromise between confidentiality, group size, and cost-efficiency. To save network capacity and transport costs it is beneficial to use multicast groups that are as large as possible. On the other hand, if multicast is used to transport encrypted data that is expected to create revenues to its producers, the content must not be easy to copy. Since it is very difficult to control who is receiving the data that was multicast, data needs to be encrypted. Also, obtaining illegal keys needs to be made difficult enough that it is more convenient to pay to the content providers than to obtain illegal keys from other users.

Effectively, data can be assumed to provide earnings to its creators if at least one of the following criteria holds.

- Cost of data is low. The data can be produced and distributed to the client in a way that discourages the client from making illegal copies of the keys because obtaining the keys legally is simpler and affordable enough.
- The data is such that is quickly loses its value if it is not used almost immediately.

An example of information that fulfills the former of the above criteria is TV-shows, while newspapers are an example of the latter.

4 Conclusions

Caching multicast data at packet level offers advantages for mobile clients which need to receive multicast data over wireless network interfaces. Since every packet is given a globally unique identity, a client can retrieve missing data from any cache connected to the network. Data that is distributed using the system can remain encrypted while it is stored in the caches and on the hard disk of the client.

The system offers an affordable way of distributing data to clients and enables a business model which is based on selling keys to data that is sent using multicast.

References

[1] Deering S. Host Extensions for IP Multicasting. Request For Comments 1112, IETF, August 1989.

[2] Almeroth K., Ammar M., Fei Z. Scalable Delivery of Web Pages Using Cyclic Best-Effort (UDP) Multicast. In proceedings of IEEE INFOCOM, San Francisco, USA, March 1998.

[3] Rodriguez P., Ross K, Biersack E. Improving the WWW: Caching or Multicast? In Computer Networks and ISDN Systems, 1998.

[4] Haberman B., Thaler L. Unicast-Prefix-based IPv6 Multicast Addresses. Work in progress, expires April 2002.

[5] Deering S., Hinden R. Internet Protocol, Version 6 (IPv6) Specification. Request For Comments 2460, IETF, December 1998.

[6] Holbrook H., Cain B. Source-Specific Multicast for IP. Internet draft, Work in progress, November 2002.

[7] Luby M., Gemmell J., Vicisano L., Rizzo L., Crowcroft J. Asynchronous Layered Coding protocol instantiation. Internet draft, Work in progress, April 2002.

[8] Adamson B., Bormann C., Handley M., Macker J. NACK-Oriented Reliable Multicast Protocol (NORM). Internet draft, Work in progress, March 2002.

[9] Kent S., Atkinson R. IP Encapsulating Security Payload (ESP). Request For Comments 2406, IETF, November 1998.

[10] Kent S., Atkinson R. Security Architecture for the Internet Protocol. Request For Comments 2401, IETF, November 1998.