Video Broadcasting using Overlay Multicast

Dragan Milić, Marc Brogle and Torsten Braun
Institute of Informatics and Applied Mathematics
University of Bern, Neubrückstrasse 10, 3012 Bern, Switzerland
email: {milic|brogle|braun}@iam.unibe.ch
phone: +41 31 631 {5309|8668|4994}

Abstract

Despite the availability of high bandwidth Internet access for end-users, video broadcasting over the Internet is not widely spread. Multicast communication decreases the network load by eliminating redundancy of the data transfer. However IP Multicast was never widely accepted by commercial Internet service providers (ISP). Existing solutions solving this problem, like MBONE tunneling, are not available for end-users accessing the Internet via xDSL or TV cable. Application Layer Multicast using peer-to-peer (P2P) overlay networks could solve the problem of sparse IP Multicast support in the Internet. A limitation of this approach is the lack of standardized interfaces for existing IP Multicast applications. We propose a solution, which bridges Application Layer Multicast and IP Multicast and uses a P2P (Overlay) Network to transport multicast data. Our solution – including a "proof-of-concept" prototype – enables video broadcasting over the Internet using existing IP Multicast applications without requiring additional service deployment.

1. Introduction

Despite the availability of high bandwidth Internet access for end-users, video broadcasting over the Internet is not widely spread. The reason for this is the lack of IP Multicast support by commercial Internet service providers (ISPs). The Internet Protocol (IP) is designed with built-in support for multicast [9, 12, 7] communication. Multicast communication enables delivery of data (such as audio or video streams) from one sender to multiple receivers (a group of hosts interested in receiving the data) with minimal network overhead. With unicast communication the sender must send the data separately to each receiver. Using the multicast communication paradigm enables replication of data packets by the transporting routers only when needed. In this way, the network load is minimized – the data flow traverses each link at most only once. The weakness of IP Multicast is the required support for IP Multicast by the transporting routers on the whole path from the sender to the receivers.

Although multicast communication reduces the network load by eliminating the redundancy of the data transfer, IP Multicast was never accepted by commercial ISPs. The reason for the lack of acceptance, is mainly of "political" nature. For example, there are issues regarding inter-ISP cooperation. Routers must be additionally configured and additional security considerations for transporting multicast traffic have to be taken into account. As a result of the reluctance of commercial ISPs deploying native IP Multicast, the Multicast Backbone MBONE [11] was deployed.

The MBONE consists of "islands" of multicast enabled networks in the Internet. These islands are interconnected through different type of tunnels across the Internet. Essentially, tunnels are unicast data flows that encapsulate IP Multicast packets. The MBONE tunnels are set-up manually and require stable endpoint addresses on both sides of the tunnel. The MBONE solves the problem of providing IP Multicast for networks with constant Internet connectivity and stable IP addresses. However the administrative overhead and coordination requirements are not feasible for most of the typical Internet users, which are connected to the Internet using a cable, modem or xDSL connection.

For providing audio and video broadcasts for typical Internet users, content providers use content distribution networks (CDNs) [24]. A CDN consists of numerous hierarchically organized "reflector" hosts, which are receiving the content (audio and/or video stream) from hosts that are one level below in the hierarchy. Those "reflector" hosts redistribute the content to hosts one level lower in the hierarchy. The sending host is the highest host in the hierarchy and receiver hosts are located on the lowest hierarchy level. The "reflector" hosts are usually geographically dispersed all over the Internet to enable the end-user hosts to connect to reflectors, which are near them (in terms of the network latency). The use of CDNs requires a substantial investment in infrastructure with costs rising according to the number of receivers. The communication protocol in CDNs is usually a proprietary protocol based on TCP or UDP. It requires special servers and clients for distributing and receiving the video stream.

An alternative to CDNs is the use of peer-to-peer (P2P) networks for content distribution. P2P networks rely only on unicast communication between end-systems. There is no distinction in P2P networks between servers and clients – every peer can be server, client or both at the same time. P2P networks are usually self-organizing (do not require an infrastructure) and adapt to changing conditions (leaving/joining of peers, network failure etc.). There are many proposals for using P2P networks as overlay networks to provide multicast functionality for the peers (hosts) [28, 34, 4, 5, 8, 35, 3, 19]. All of those solutions provide multicast communication, which resembles to IP Multicast. However none of them offer a standard IP Multicast interface for the applications. The lack of the standard interface prevents the endorsement of those solutions by software vendors and end-users.

Our proposed solution bridges P2P (Application Layer) Multicast and IP Multicast. We provide a standard IP Multicast interface for the applications and use P2P (Application Layer) Multicast for transporting the data. The proposed solution can be used for broadcasting video over the Internet without the need of deploying any kind of additional network or server infrastructure.

In the next Section we describe the work related to our research. In Section 3 follows a description of the challenges of Application Layer Multicast. This includes locality awareness and efficient multicast distribution tree building. We also propose a solution for providing a native IP Multicast interface to applications. Closing this Section is a description of how our solution can be used to enable the Internet wide video broadcasting for existing multicast aware applications. In Section 4 we present a prototype implementation of our proposal, which is developed within the EUQoS [30] project. In Section 5 we show the different kinds of problems that arise when taking security and privacy issues into account. Quality of Service related problems and solutions are discussed in Section 6. In the last Section we summarize the results and give an outlook for the further research.

2. Related Work

Different protocols for Application Layer Multicast with overlay networks have been proposed [28, 34, 5, 8, 35, 4, 3, 19]. Androutsellis and Spinellis give a survey of peer-to-peer content distribution technologies [1]. A framework for analyzing peer-to-peer content distributing technologies is presented, which focuses on nonfunctional characteristics like security, scalability, performance, fairness and resource management. Studies about routing mechanisms, applied distributed object location mechanisms, content replication, caching, migration and security related issues like encryption support, access control, authentication, etc are also performed in that paper. In the authors perspective there exist two defining characteristics of peer-to-peer architectures: sharing computer resources by direct exchange and treating instability and variable connectivity as the norm. They propose a classification of peer-to-peer applications in regards of communication and collaboration, distributed computation, Internet service support, database systems and also content distribution. Furthermore they investigate the overlay network centralization and classify it into purely or partially centralized and hybrid decentralized architectures. Also the different overlay network structures are classified into structured or unstructured overlay networks.

In [34] an effective passive replication scheme designed to provide a reliable multicast service is presented. *PeerCast* is an efficient and self-configurable peer-to-peer *End System Multicast* (ESM) framework. Peers in the PeerCast overlay network act as clients and servers. The PeerCast Middleware is divided into two tiers: “P2P Network Management” and “End System Multicast Management”. The authors focus on the development of an analytical model to discuss the fault tolerance and to present an effective node clustering technique based on landmarks. This way PeerCast can cluster end-system nodes by using physical network proximity information for fast multicast group subscription and efficient data dissemination.

As stated in [21] the main bottleneck of Video on Demand (VoD) services is the server’s storage I/O and network I/O bandwidth. Using multicast improves the distribution of a video program to multiple clients, hence leading to better performance of a VoD service. The authors critically evaluate and discuss the progress in developing multicast VoD systems. They also present a concept and architecture for multicast VoD and then introduce advanced techniques that can be used in multicast VoD systems. Problems related to multicast VoD services are also analyzed and evaluated.

3. Multicast Middleware

We propose to provide a standard IP Multicast interface for common video streaming applications like VLC [32] using an Application Layer Multicast P2P network to transport the multicast data.

3.1. IP Multicast and Application Layer Multicast

IP Multicast [9, 12, 7] is a concept for efficient n-to-m data dissemination over IP networks. Basically in IP Multicast, the sender sends an IP datagram to a so-called group address (see also Section 3.3). An IP datagram is forwarded to all receivers interested in receiving the data by replicating the IP datagram on the path to the receivers only when needed. In this way the network load is minimized compared to unicast transmission. With unicast the sender would send the same IP datagram once for each receiver. IP Multicast has been proposed and specified almost two decades ago and numerous applications [22] exist. However it has not been widely supported by major commercial Internet service providers (ISPs).

With the raise of P2P networks [1] there has been a revival of multicast in the form of Application Layer Multicast. The Application Layer Multicast does not require any kind of multicast support by the host operating system or the network. Instead, Application Layer Multicast uses only unicast communication. The unicast links over the Internet construct an overlay network, which is then used for transporting the multicast data. As shown on the left side of the Figure 1, in the IP Multicast, the multicast traffic is replicated by the routers of the transporting network. In this way, the sender must send the data only once and there is no redundancy of the data transported through physical links. As shown on the right side of the same Figure, the Application Layer Multicast (Overlay Multicast) replicates the data only on the end-systems, which are interconnected using unicast (P2P) links. The efficiency of the Application Layer Multicast depends on the overlay network construction and routing. With an optimal overlay topology Application Layer Multicast can approximate efficiency of native IP Multicast.

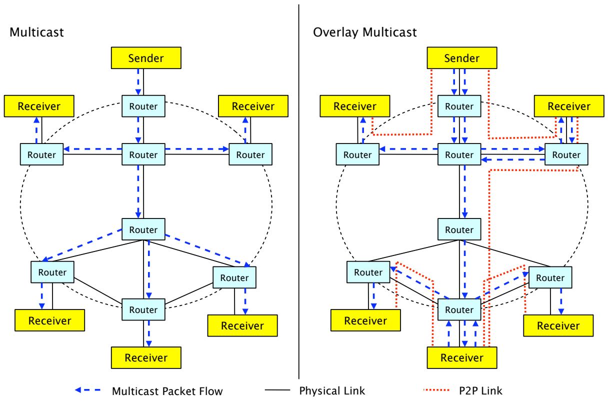

Figure 1. IP multicast vs. overlay multicast

The solution we propose can be used with any Application Layer Multicast network, which offers the standard multicast operations (subscription to a multicast group, receiving and sending multicast data). The typical P2P Application Layer Multicast network tries to approximate the efficiency of IP Multicast communication regarding link stress by using unicast communication. As seen in Figure 1, Application Layer Multicast is not able to totally avoid sending redundant data over the same physical link as IP Multicast can. However it can reduce the number of redundant data flows in the whole network. The overlay network is usually built in a topology aware manner. Therefore peers, which are "near" to each other in terms of communication latency, are directly connected. The P2P links are constantly monitored, which allows to react to failures in network communication or to failures of neighbor peers.

Eliminating the requirement for multicast support by the operating system and the underlying network makes the use of Application Layer Multicast very appealing for any kind of Internet users. The disadvantage of the Application Layer Multicast is the lack of standardization – each Application Layer Multicast has it's own API and addressing scheme. This prohibits already existing Multicast-aware applications from using the Application Layer Multicast.

We propose a solution called Multicast Middleware [6], which uses Application Layer Multicast for transporting multicast traffic. However it also offers a standard multicast interface for the applications. The Multicast Middleware is also aware of limitations regarding connectivity on the network layer [13].

3.2. Providing an IP Multicast Interface for Standard Applications

The IP Multicast interface for the applications is usually offered by the operating system. The operating system on the other side communicates with a multicast-enabled router in the local network using IGMP [9, 12, 7] for signaling. Sending IP multicast traffic is not different from sending IP unicast traffic. The only difference is the reserved source/address range, which denotes different multicast groups (groups of multicast traffic receivers). On the network layer, multicast traffic is handled differently. For example, in Ethernet the IP packets with destination a multicast group as a destination address get an Ethernet multicast address assigned.

To provide an IP Multicast interface for the whole system (including services integrated in the operating system's kernel), we propose to use a virtual Ethernet device (also known as TAP device [31] – a software analogy of a wiretap). The TAP interface is a special kind of network interface, which is seen by the operating system as a normal Ethernet device. However instead of forwarding the Ethernet frames to a hardware device, the TAP interface forwards the received Ethernet frames to a user-space process. On the other side, the TAP interface forwards all Ethernet frames received from the user-space process as incoming frames to the operating system's kernel. TAP support exists for all major operating systems such as UNIX/Linux, MacOS X and WIN32.

Using a TAP interface and a Multicast Middleware makes processing of multicast traffic transparent to all applications. This includes the multicast functionality integrated in the operating system's kernel. This approach does also not require any modification of application code. Multicast traffic originating from an end-user host can be routed through the TAP device. This device forwards the packets (encapsulated in Ethernet frames) to a user-space process (which we call the Multicast Middleware) for processing. The Multicast Middleware acts as a multicast router by implementing IGMP and transporting the multicast data.

IP Multicast enabled applications must subscribe to different multicast groups to receive video broadcast announcements and audio/video streams. The multicast group subscription is usually a system call, which instructs the operating system's kernel to send IGMP membership report messages to the IP Multicast router. In our case, the IGMP membership reports are sent via the TAP interface to the Multicast Middleware. The Multicast Middleware interprets the IGMP membership reports and notifies the neighbor peers about the changes in the multicast routing table. This information (depending on the multicast routing protocol used in the overlay network) is propagated to other peers. The flow of the multicast subscription as described before can bee seen in a figure 2.

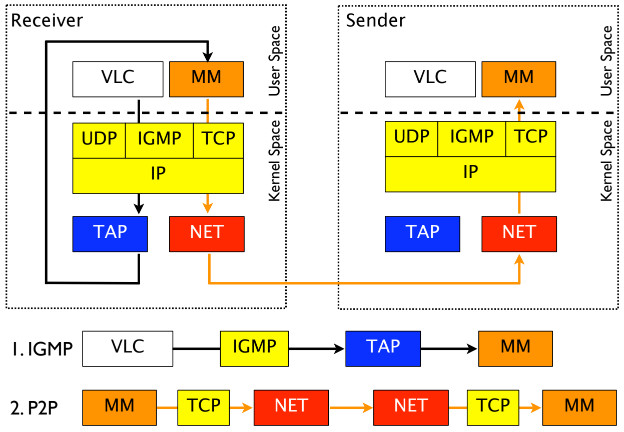

Figure 2. Subscribing to a multicast group

In the network, video broadcast announcements and audio/video streams are handled in the same way: they are delivered as multicast data packets. Sending multicast traffic does not differ from sending unicast traffic. The only difference is the destination address, which is in the case of the multicast traffic the address of an IP Multicast group and not of a host. After a data packet has been sent by the application, it is forwarded by the operating system's kernel to the appropriate multicast-enabled network device (in our case the TAP device). The Multicast Middleware process receives the outgoing multicast traffic via the TAP device. The received multicast traffic is then encapsulated into Application Layer Multicast messages. The IP Multicast destination address of the packets is translated into Application Layer Multicast addresses to which the messages are sent.

After receiving an encapsulated IP Multicast packet by Application Layer Multicast, the Multicast Middleware encapsulates the IP Multicast packet into an Ethernet frame. The Multicast Middleware then sends the Ethernet frame via the TAP interface to the operating system's kernel for processing. The operating system's kernel delivers the data to the video application. The delivery of multicast traffic from the sender through the Multicast Middleware to the receiver is presented in figure 3.

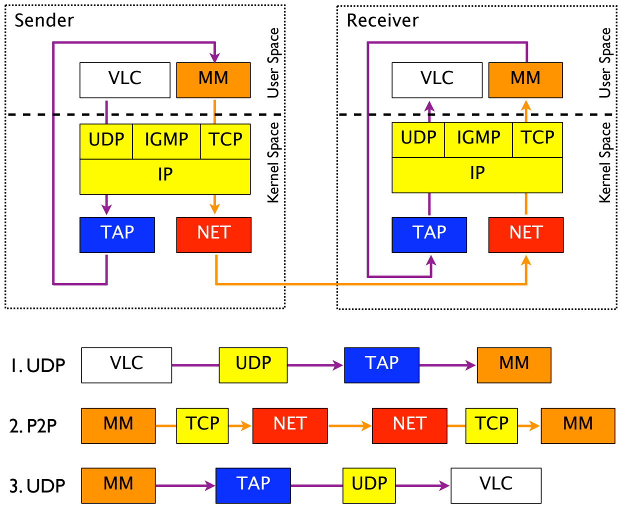

Figure 3. Sending and receiving data

3.3. Interfacing IP Multicast and Application Layer Multicast

For transporting multicast traffic we do not propose a new P2P protocol – any P2P protocol can be used and integrated. Instead we describe the requirements for the Application Layer Multicast protocol and how IP Multicast traffic can be mapped to Application Layer Multicast messages.

Every IP Multicast packet has either a source or a destination address out of the IP Multicast address range (224.0.0.0 to 239.255.255.255). Most Application Layer Multicast protocols implement their own multicast addressing scheme. Depending on the protocol's addressing scheme, the address range can be smaller, equal or larger than the IP Multicast address range. In case of a larger or equal address ranges, multicast addresses can be mapped 1-to-1 to the Application Layer Multicast addresses. For example the IP Multicast address range can be mapped to a consecutive address range of the same size in the Application Layer Multicast protocol's addressing scheme. In the case where the address range of the Application Layer Multicast is smaller than the IP Multicast address range, the IP Multicast addresses must be projected to the Application Layer Multicast address range. This can be achieved by hashing the modulo-function: $A_x = (A_{\text{IP}} - 0xe0000000) \bmod$ # of addresses.

IP Packets can be encapsulated in Application Layer Multicast messages. If the maximal length of an Application Layer Multicast message is lower than the IP packet's length, the standard IP Packet fragmentation can be applied to the packet in order to transport the packet through the overlay network. On reception of fragmented IP Packets, the Multicast Middleware should be able to de-fragment them and to deliver them to the TAP interface. The time to live (TTL) field of the transported packets should be reduced for each P2P hop. Packets with TTL 0 should not be forwarded.

3.4. Video Broadcasting

Using the proposed solution, we can provide a video streaming service for a very large group of Internet users without the need for large investments in infrastructure. Certain requirements for sender and clients must be fulfilled. The sender of a video stream must have a Multicast Middleware installed on a computer, which is connected to the Internet. The Internet connection should support at least the bandwidth for sending the stream once. Any application supporting streaming using IP Multicast can be used (for example VLC or VLS). Each client, which wants to receive a video stream must install the Multicast Middleware on his computer and must have an Internet access. Any video application with IP Multicast support (like VLC) can be used for receiving the video stream.

The session announcement protocol (SAP)[16] can be used to announce running or scheduled video broadcasts over IP Multicast. The SAP announcements include session description protocol SDP [15] stream descriptions encapsulated in UDP packets and are sent to a predefined IP Multicast group (for example IPv4 global scope session announcements are sent to 224.2.127.254) and port (9875). Since the Multicast Middleware enables IP Multicast on the end-host, SAP can be used for announcing video broadcast transmissions.

4. Prototype of the Multicast Middleware

As a proof of concept for our proposal, we have implemented a prototype of the Multicast Middleware. In order to support as many operating systems as possible, we have implemented the core of the Multicast Middleware in the JAVA programming language. The only non-portable component of the Multicast Middleware prototype is the communication module for the TAP interface. This module differs for each operating system and is usually implemented in C language.

4.1. Implementation of the Prototype

Each network packet received through the TAP device is passed unchanged to the JAVA core of the Multicast Middleware. The Multicast Middleware converts the received raw data Ethernet frame into a set of objects representing the Ethernet Frame. The result of the conversion is for example one object representing the Ethernet frame. This object contains another object representing the IP packet, which contains an object representing the UDP Packet. This kind of packet representation in JAVA is not optimal regarding processing speed. However it enables a clean object-oriented approach to packet handling.

We have implemented a simple P2P protocol for the initial test. The protocol is based on TCP connections between the peers through, which the messages are exchanged. The message format resembles the format of messages used in the Common Open Policy Service (COPS) protocol [10]. The reason for using the COPS protocol message format is its extendibility and efficiency. Each message consists of a header, which defines the length and type of the message and a list of objects, which contain additional information. The types and order of objects in a message depend on the type of the message. Each object consists of a header and data. The header defines the length, class and type of the object. The object data describes the object. Depending on the class and type, the semantic of the object data differs. For example, the object data can be an IP address or a encapsulated IP packet. This kind of design ensures easy protocol extension by introducing new types of messages or by adding new classes and types of objects into the existing messages. Currently, five types of messages have been defined:

The *Open* message is exchanged between the peers as soon as a P2P connection has been established. With the Open message each peer informs its communication partner about his preferences for receiving multicast traffic. The peer defines the UDP port on which the peer wishes to receive the multicast traffic encapsulated in UDP unicast packets. If the port is undefined, the peer can only receive multicast traffic through the already established P2P TCP connection.

The *Keep Alive* (KA) message is sent by each peer periodically to verify the TCP connection. If no Keep Alive message is received within the predefined timeout period (10 seconds) the TCP connection is considered invalid and is closed.

The *Add Route* and *Remove Route* messages are used to signal to peers changes in the multicast routing table.

The *IP Data* message contains an encapsulated IP packet (for example a IP Multicast packet). This message is used for transporting multicast traffic through the P2P network.

The Multicast Middleware implements currently IGMP version 1 [9] for handling IP Multicast subscriptions. The Multicast Middleware sends periodically IGMP host membership query packets on the TAP interface. The operating system's kernel replies to each host membership query with one IGMP host membership report message for each IP Multicast group to which at least one application has subscribed. The removal of a host membership is not signaled explicitly in IGMPv1. Hosts simply do not send membership reports for the group to which they are not interested any more so that the membership for the group expires. As consequence, a certain lag between an application signaling the operating system to leave the group is introduced. This which delays the propagation of this information to multicast routers. This issue is resolved in IGMPv2 [12]. The implementation if IGMPv2 and IGMPv3 [7] are planned extensions of our Multicast Middleware implementation.

The Multicast Middleware forwards the multicast traffic between the local TAP device and other peers according to a multicast routing table. The multicast routing table does not distinguish between TAP interfaces and connections to other peers. Each entry in the multicast table consists of a multicast group and a set of "IP Multicast destinations" to which the packet is sent. The multicast table changes through external events such as receiving add/remove route messages from connected peers. It also changes after receiving IP Multicast subscriptions from local applications through the TAP interface using the IGMP protocol.

4.2. Evaluation of the Multicast Middleware

For the prototype evaluation the P2P network is setup statically with configuration files. The P2P structure has to be built as a cycle-free tree. Multicast group subscriptions are flooded through the P2P network for the prototype evaluation. We have tested our prototype implementation of the Multicast Middleware by streaming a Video CD (MPEG 1 [18] stream with 1.4 Mbit/s). The test-bed consisted of four recent as well as low-end laptop computers (Sony Vaio P4-2.6 GHz, Apple PowerBook G4 1 GHz, Dell P3 1.2 GHz and 0.7 GHz) connected with Fast Ethernet. The PowerBook was running on Mac OS X 10.4 Tiger, the other laptops were running Fedora Core 3 with a Linux Kernel 2.6.9. All laptops used Java Version 5 and had the TUN/TAP kernel modules installed.

The overlay network and routing was configured using our introduced P2P messaging system. The IP Multicast data was transported encapsulated in P2P messages through the overlay network with TCP connections between the peers. The structure of the P2P network between the involved hosts is depicted in Figure 4. The Vaio P4-2.6 GHz laptop was used as the sender and all other laptops were the receivers of the stream,. The Dell laptop with 1.2 GHz had to replicate the received stream two times to send it to the remaining laptops (PowerBook 1 GHz and Dell P3-0.7 GHz).

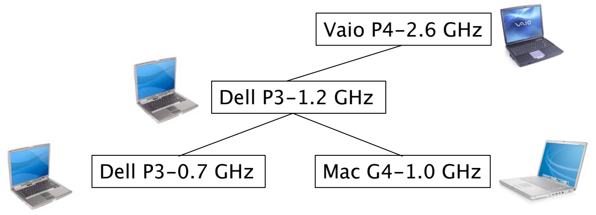

Figure 4. P2P setup for the testbed

We were able to stream a full-length movie (141 minutes) without serious quality penalties. The only (occasionally) issue perceived at the receiver of the video stream was caused by packet delivery jitter. The jitter was caused by the JAVA garbage collection mechanism. To solve this issue, we plan to implement the forwarding of IP Multicast packets through the P2P links natively in the C language, which will allow better scalability and support more streams concurrently. Note that the Dell P3-1.2 GHz in the center of the distribution tree had the triple network and processing load compared the other three laptops. It had to receive the stream and then replicate it twice to serve the remaining peers in the network.

Future evaluations will include test and performance measurements in heterogeneous network environments connect all over Europe. The EuQoS project will provide such a test-bed environment in the near future. This kind of evaluation of the Multicast Middleware has been planned within the EuQoS project for scalability, reliability, QoS and performance analysis.

5. Security and Privacy Considerations

In contrast to IP Multicast, where routers in the transporting network replicate data packets, Application Layer Multicast relies on end-systems to replicate data packets. As a consequence, not only ISPs have the possibility to monitor the traffic, but also end-users can "see" the traffic, which their neighbors are receiving. As a consequence, the security and privacy of the end-users are even more threatened than with IP Multicast.

End systems, which are used as relays for multicast data can accumulate knowledge about the preferences of their neighbors regarding the reception of video streams. This information can be used for targeted marketing of products or as a component of user surveillance. This effect can be amplified through the collusion of relaying peers, which can exchange the surveillance data about neighbors. This allows creating a more complete picture of the monitored peers.

The privacy of end-users can be improved by using Application Layer Multicast routing protocols, which change the delivery path of the multicast data over time or use parallel paths for receiving data. In this way, one peer is not always relaying data for the same peer and is not able to accumulate the information about involved end-users.

Since the peers, which are subscribed to one multicast group, do not only receive multicast data for that group, the relaying peers can also receive the multicast data they are forwarding. To protect commercial content, some kind of content encryption has to be introduced. A possible content encryption and authentication solution is described in [2]. Another alternative is to construct an overlay network consisting only from receiving peers for each multicast tree. The disadvantage of this approach is the number of overlay networks in which one receiver is participating if he wishes to receive more than one video stream.

Malicious peers can also alter the video stream they are relaying. For example, a malicious peer could inject commercials or logos into the video stream. Such behavior can be detected [14] and appropriate actions (e.g. excluding the peer from the overlay network) can be taken.

6. Quality of Service

Quality of Service (QoS) support in existing networks is still not widely deployed. Different approaches exist to cope with the problem of introducing QoS services and reliability to overlay networks [20, 17] and multicast routing [33]. Others cope with introducing QoS to the Internet by the means of using overlay networks [29]. Supporting heterogeneity and congestion control [23] is also a major requirement for a QoS enabled overlay network. Quality of Service and Denial of Service go hand in hand since one requirement to QoS should also be providing protection against Denial of Service attacks [27]. Different mechanisms have to be introduced to make an overlay network robust against fundamental attacks and to ensure reliability in order to support basic QoS requirements.

The presented Multicast Middleware is being developed for the EuQoS project [30] that aims to provide a facility for providing inter-domain end-to-end QoS services. As a consequence the Multicast Middleware will be using the EuQoS QoS capabilities to setup its unicast based overlay network links with QoS reservations. This will enable QoS not only for the reception but also the redistribution of the received encapsulated multicast data. The Multicast Middleware will also support measurement based Best-Effort QoS. This will enable basic QoS services in non EuQoS-aware networks or on the paths in the networks that do not support QoS. The Best-Effort QoS denotes providing QoS based on measurement and prediction of QoS parameters (round trip time, bandwidth, jitter etc.) in a network, which does not support QoS.

Multicast streams can be prioritized inside the Multicast Middleware using different queuing schemes to provide user-driven QoS capabilities. Also priorities have to be arranged internally and assigned to flows that have to redistributed in order to provide a fully functional multicast overlay distribution facility. Algorithms and methods on how to build the overlay network and how to group peers locally with respect to QoS properties have been discussed in [25] and [26].

7. Conclusion

IP Multicast is a necessary service for efficient video streaming over the Internet. Although commercial ISP providers are not willing to enable IP Multicast services for end-users, it is possible to provide multicast-like services using only unicast communication.

In this paper we have described a way to provide IP Multicast services on end-systems without changing existing IP Multicast applications or deploying infrastructure to the Internet. Our approach uses a virtual network device for capturing the multicast traffic and forwarding it to a user application (Multicast Middleware). The Multicast Middleware then transports the data using an Application Layer Multicast to other receivers in the Internet. We have also described how existing Application Layer Multicast protocols can be integrated into our approach. We have outlined the possibility of QoS provision for Multicast Middleware over the Internet. We have presented a proof-of-concept implementation of the Multicast Middleware, which was successfully tested by broadcasting a video stream. We have also described the security and privacy issues problems, which arise from using Application Layer Multicast.

We plan to extend our prototype implementation of the Multicast Middleware with native forwarding services to reduce jitter of the transmitted data. Also QoS support (EuQoS and "Best Effort" based) has to be integrated. A better Application Layer Multicast routing protocol has to be integrated and the P2P protocol has to be extended to improve Multicast transport. Further evaluations in large heterogeneous networks and considerations about IPv6 have to be done in order to evaluate scalability and future applications.

8. Acknowledgments

This work was funded by the European Union 6th Framework Programme under contract IST FP6 IP 004503 EuQoS Integrated Project.

References

[1] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.

[2] R. Balmer and T. Braun:. Resource control and authentication for a video streaming service in a diffserv/ip multicast network. In *3rd Conference on Security and Network Architectures (SAR04)*, La Londe, Cote d'Azur France, June 21-25 2004.

[3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proceedings of the ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '02)*, pages 205–217, New York, NY, USA, 2002.

[4] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. Bimodal multicast. *ACM Trans. Comput. Syst.*, 17(2):41–88, 1999.

[5] A. Bozdog, R. van Renesse, and D. Dumitriu. Selectcast: a scalable and self-repairing multicast overlay routing facility. In *SSRS '03: Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems*, pages 33–42, New York, NY, USA, 2003. ACM Press.

[6] M. Brogle and D. Milic. EuQoS Multicast Midddleware: Basic Architecture Overview and Concepts. Technical Report IAM-05-002, Institute of Computer Science and Applied Mathematics, Bern, June 2005.

[7] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet group management protocol, version 3. RFC3376, October 2002.

[8] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: high-bandwidth multicast in cooperative environments. In *Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03)*, pages 298–313, New York, NY, USA, 2003.

[9] S. Deering. Host Extensions for IP Multicasting. RFC1112, August 1989.

[10] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) protocol. RFC2748, January 2000.

[11] H. Eriksson. Mbone: the multicast backbone. *Commun. ACM*, 37(8):54–60, 1994.

[12] W. Fenner. Internet group management protocol, version 2. RFC2236, November 1997.

[13] A. Ganjam and H. Zhang. Connectivity restrictions in overlay multicast. In *Proceedings of the 14th ACM international workshop on Network and operating systems support for digital audio and video (NOSSDAV '04)*, pages 54–59, New York, NY, USA, 2004.

[14] A. Habib, D. Xu, M. Atallah, B. Bhargava, and J. Chuang. A tree-based forward digest protocol to verify data integrity in distributed media streaming. In *IEEE Transactions on Knowledge and Data Engineering*, volume 17, July 2005.

[15] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC2327, April 1998.

[16] M. Handley, C. Perkins, and E. Whelan. Session announcement protocol. RFC2974, October 2000.

[17] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr. Overcast: Reliable Multicasting with an Overlay Network. pages 197–212.

[18] JTC1/SC29/WG11. ISO/IEC 11172:1993: Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – parts 1 to 5, 1993.

[19] M. Kwon and S. Fahmy. Path-aware overlay multicast. *Comput. Networks*, 47(1):23–45, January 2005.

[20] Z. Li and P. Mohapatra. QRON: QoS-aware routing in overlay networks, 2003. IEEE JSAC, 2003, to appear.

[21] H. Ma and K. G. Shin. Multicast video-on-demand services. *SIGCOMM Comput. Commun. Rev.*, 32(1):31–43, 2002.

[22] C. K. Miller. *Multicast Networking and Applications*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.

[23] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. Supporting Heterogeneity and Congestion Control in Peer-to-Peer Multicast Streaming. In *IPTPS*, pages 54–63, 2004.

[24] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of internet content delivery systems. *SIGOPS Oper. Syst. Rev.*, 36(SI):315–327, 2002.

[25] M. Scheidegger. New Results in the XBAC Project. Technical Report IAM-05-002, Institute of Computer Science and Applied Mathematics, Bern, June 2005.

[26] M. Scheidegger, T. Braun, and F. Baumgartner. Endpoint Clustering for Improving Distributed Services. submitted for publication.

[27] S. Shalunov and B. Teitelbaum. Quality of service and denial of service. In *RIPQoS '03: Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS*, pages 137–140, New York, NY, USA, 2003. ACM Press.

[28] A. Sobeih, W. Yurcik, and J. C. Hou. Vring: A case for building application-layer multicast rings (rather than trees). In *Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04)*, pages 437–446, Washington, DC, 2004.

[29] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. OverQoS: offering Internet QoS using overlays. *SIGCOMM Comput. Commun. Rev.*, 33(1):11–16, 2003.

[30] EuQoS project. http://www.euqos.org/.

[31] Universal TUN/ TAP. http://vtun.sourceforge.net/tun/.

[32] Videolan client. http://www.videolan.org/.

[33] S. Yan, M. Faloutsos, and A. Banerjea. Qos-aware Multicast routing for the Internet: the design and evaluation of QoSMIC. *IEEE/ACM Trans. Netw.*, 10(1):54–66, 2002.

[34] J. Zhang, L. Liu, C. Pu, and M. Ammar. Reliable peer-to-peer end system multicasting through replication. In *Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, pages 235–242, Washington, DC, 2004.

[35] R. Zhang and Y. C. Hu. Borg: a hybrid protocol for scalable application-level multicast in peer-to-peer networks. In *NOSSDAV '03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, pages 172–179, New York, NY, USA, 2003.