Ward Louckx

Streaming media over multicast

Bachelor's Thesis

Information Technology

May 2013

MIKKELIN AMMATTIKORKEAKOULU
Mikkeli University of Applied Sciences

DESCRIPTION

MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	**Date of the bachelor's thesis** 13.05.2013
Author(s) Ward Louckx	**Degree programme and option** Information Technology

Name of the bachelor's thesis

Streaming media over multicast

Abstract

Multicast is a routing scheme that allows efficient broadcast of media and other content over a network. By multiplication of the data in the routers on the way to their destination, it allows servers to save bandwidth by only sending one stream to multiple receivers. Though the use of multicast over the internet is rather limited. There are several reasons why multicast should be used and why in contrary it is not used. ISP's and content providers are not willing to change their systems to this efficient way of working, in fear of losing money from commercial perspective. There are ways around the blockage of ISPs. In this thesis multicast will be explained and a simulation of a multicast enabled internet will be set up. This simulation includes the setting up of a streaming server, the effects of a multicast enabled and disabled ISP and a server side workaround.

Subject headings, (keywords)

Multicast, streaming, video, network, media server

Pages	Language	URN
58	English	

Remarks, notes on appendices

Tutor	Employer of the bachelor's thesis
Matti Koivisto	Mikkeli University of Applied Sciences

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The writing of this thesis is something I could not have done alone. Only the fact that I had the opportunity to write it is due to a couple of people and institutions. Therefore it is my honor to thank them in this section for all the opportunities that they have provided me.

First of all I would like Mr. Tomi Numento for asking me if I would like to enroll in the IT double degree program of MUAS, and for the help and talks to get me enrolled as an IT student. Without him and his idea I would not have thought about it and I would have missed this great opportunity that made it to one of the biggest chapters of my life.

Of course I would have not been able to stay in Finland and at MUAS without the support of my parents who gave me their blessing and resources to live abroad for 10 months. Also the supports of all my friends, those who stayed in Belgium but also the new friends I met in Finland gave me the drive to keep going on and to bring this opportunity to a good end.

Last but not least I would like to thank my mentor Mr. Matti Koivisto for creating a challenging and interesting thesis topic that suited perfectly to my former education and also great thanks to the Mikkeli University of Applied Sciences for providing me with all the material and resources I need to complete this thesis.

1 INTRODUCTION

Today there is a lot of multimedia content available on the internet. As where internet was started to exchange information mainly by text and data, it has evolved to a stream of digital information of any form. We can reach data as in software, news and email as text, collected data in databases, graphical images, music, video, and many more. It is clear that all these forms of data are different and they might need a different approach to handle. For example a user was satisfied enough for a movie to be on a linear VHS cassette, whereas (if we would look back in the same era) for editing multiple text files he preferred a 3.5" diskette, which could be accessed non-linear. Today on the internet it is the same, every type of file has an optimal way of being transferred and read.

For multimedia content there are several ways that can be optimal, each with their specific target. Images can be downloaded non-linear as one image does not make a lot of sense when the whole file is not completed. This can be said as well about sound and video, however there are a lot more advantages when we download sound and video linear. In this way we can for example already start listening and watching when the first part has arrived. We only need the end part when we are at the end of the song or movie. So if we download linear we can listen and watch faster and earlier, we do not have to wait for the whole file to start enjoying. This is what we call streaming.

Streaming can be done several ways if you talk about the use of the network. The three main ways to stream content is through unicast, multicast or broadcast. They all have their advantages and disadvantages. However not all of them are supported by the internet backbone and the Internet Service Providers. Today most of the streaming is done by a unicast connection, where each downloader has his own connection. This is suitable for on-demand streaming, where users are streaming stored content on a server as they were downloading a simple file.

However when you have live content (a sports game, speech or concert) viewers access the same source simultaneously. They also want the stream to be live, so if they start watching it later they don't want the stream to start from the beginning. This means that the same live stream data have to be send simultaneously to different receivers. With a unicast system all those users would need a separate connection and all the load would be on that one server. This is not efficient and therefore there are the two other techniques: multicast and broadcast. They use the network structure itself to duplicate or multiply the live stream to the receivers, and take away the load from the server. The technique is rather simple but yet not supported or even blocked by Internet Service Providers and the internet backbone. In this thesis I will research how to stream media over a multicast network, how a multicast network needs to be configured and what possible other techniques exists if multicast is not supported.

To give answers to these questions it is necessary to understand how the internet works. The structure and how information is delivered over the internet will be explained in Chapter 2. Next it is necessary to know which types of digital media are out there. They will be discussed in Chapter 3. To end the theoretical part, the current capabilities of the internet regarding multicast will be summed up in Chapter 4.

The practical part will start with a simulation of a multicast enabled internet in Chapter 5, the topology of the whole setup and the configuration. The selection and configuration of the server to stream the media content will be handled in Chapter 6. After the setup of the simulation model and the server, the scenario is put to some tests in Chapter 7. Two different multicast modes and the problem case ‘what happens if my ISP doesn’t support multicast’ will be handled. Finally in Chapter 8 it is possible to find the conclusion of this thesis.

2 TRANSPORTING MEDIA

2.1 Basics of transporting data

Before talking about unicast, multicast and broadcast it is needed to understand how data travels over a network like the internet. For example a way to transfer data would be to connect the sending device directly with the receiving device by a single cable. This works perfectly when you only have two devices. To add another device it is possible to connect a cable from the third device to the first one and a cable to the second one. To add a fourth device it is again possible to connect a cable to device one, one to device two and one to device three. In this network of four devices each can directly communicate with another device by their own connection cable. However to make this network there are already 6 cables needed to connect all the computers to each other. To connect a given number of computers it is possible to use the formula:

$$T_n = \frac{n^2 - n}{2}$$

where Tn is the number of cables for a given n devices. This function however is exponential so the amount of cables grows a lot faster than the connected devices. This is not efficient, especially not with a large scale network like the internet.

To make things easier it would be better to have a dedicated central device that is directly connected to each device. This device then handles all the data and forwards it to the right destination. It works like a post sorter or telephone central. In networking it is called a switch. Now we only need to connect one extra cable for each device we want to add. However the sending computer needs to attach an address where to deliver the data, on this level, a MAC-address, a somewhat random unique address for each device (Broadband Media, 2013).

Still this is not sufficient nowadays. Imagine one central point for the internet, it means that all devices need a dedicated cable to this central point, which might be somewhere far away. Better would be to subgroup some devices by their own central switch, to make it a network and then use one cable to send data that is inter-network to another central point. To connect more networks together we need again some dedicated devices. This time those devices need to know to which network to send the data. Because of the added complexity and the need of a logic they get an IP-address which contains more information than a simple hardware address (What Is My IP Address, 2013).

It is routing that makes the internet work today. Routing decides where data goes. To manage all this routers have routing protocols and routing schemes. Protocols are used to format data, get network topology and agree on ways of communications. Routing schemes are used to define a method for the data to be delivered to the client device, to decide how the data has to get to the destination.

As written before, in the world of networking they have specific words for all the devices and other terms. To be able to explain everything it is necessary to name everything correctly.

There are a couple of different types of devices in the topology of a network. The devices that deliver the content to the network are called the servers. They store the websites, data, multimedia and all other content delivered on the internet. On the other end of the line there are the clients. This are the computers that browse the websites, smart phones that display a video from YouTube or an interactive television. In between of the clients and servers we have the network devices, mainly routers, firewalls and switches. They do everything to deliver the data, called packets, sent from the servers to the clients. Firewalls control the safety and block unwanted packets. Routers route the packets to the right network and switches deliver it to the destined client, router, switch, firewall or server. There are also devices called hubs, they send all packets to all their ports (Mitchell, 2013). Because they are inefficient and old they are hardly used these days.

2.2 Unicast

Unicast is the most general way of sending packets through a network. 'Uni' stands for 'one' or 'single'. It means in short that in unicast there is build a single cast or 'stream' of packets from the server to each client. It is a one to one relation. In Figure 1 you can see a symbolic representation of unicast.

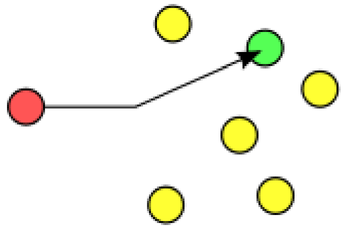


FIGURE 1. Unicast

Unicast is the standard method for transferring packets on a network. The most common services of the internet use unicast. Connecting to a web server to get a webpage over http, downloading files of an ftp server, VOIP conversations with Skype, streaming a video on YouTube, sharing files over the Bit Torrent network, sending an email with smtp or a connection through telnet, it are all example of unicast (Fairhurst, 2009). Unicast uses TCP and UDP to deliver the packets (Microsoft, 2003).

2.3 Broadcast

Broadcast is somewhat the opposite of unicast. In broadcast packets are send to everyone on the network. All clients receive the packets and have to decide for themselves if the packets are interesting or not. It is a one to many relation. In Figure two you can see a symbolic representation of broadcast.

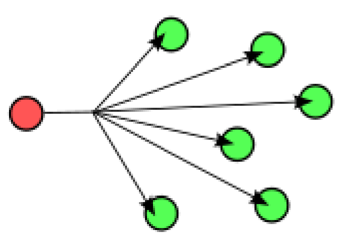


FIGURE 2. Broadcast

A typical example that uses broadcast is Address Resolution Protocol (ARP). It uses broadcast to send the ARP packet to all computers on the LAN (Fairhurst, 2009).

For broadcasting a host will send packets to the network broadcast address. The broadcast address of a network is the address where all the bits of the host part of the IP address are set to one. For example the network 192.168.1.0/24 its broadcast address is 192.168.1.255. If the host does not know the current network, like a new client in a DHCP environment, it will broadcast to the standard physical network broadcast address, where all the bits of the IP address are set to one, so for IPv4 it gives 255.255.255.255. Routers will never forward this packets out of a network (Mogul, 1984).

In it most pure form, a router will redistribute a broadcast packet to all connected networks. However this might cause packets to get into a loop (for example with redundant networks) until the Time To Live (TTL) has reached a value of zero. A subset of broadcasting is multidestination routing. With multidestination routing each packet contains a list of receivers. This way a router checks to which networks the packets have to be send, and they do not cause unnecessary flooding of the network.

There is however a more efficient way of broadcasting, called reverse path forwarding. With reverse path forwarding a router checks each source of each incoming broadcast packet. When the link where the router got this packet from is the most optimal link to get to its source, the router assumes it’s the first packet to arrive, and will broadcast it to all other connections, except for the incoming link. However if the link where the broadcast packet arrived is not the most optimal path to the source of the packet, the router will discard this packet, as it is probably a double packet. To use this form of broadcasting, the routers only need to know the most optimal paths, they are able to get this information either with distance vector routing or with link state routing. An even more efficient way is the use of a spanning tree. With a spanning tree the router knows the topology of the network, so when a broadcast packet arrives, it knows to which networks it should distribute the packets itself. This however requires link state routing (Tanenbaum & Wetherall, 2011, 398-400).

Broadcasting is by nature a connectionless routing scheme. This is because there is no real connection established between source and destination. Therefore broadcast can only use the UDP and not the TCP (Network Sorcery Inc, 2012).

2.4 Multicast

Multicast is a more efficient way of broadcast. Unlike broadcast it will only send the packages to the clients that are in the multicast group. It is like broadcast a one to many relation. In Figure 3 you can see a symbolic representation of multicast.

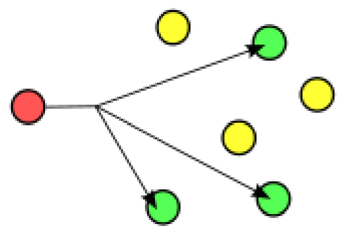


FIGURE 3. Multicast

IPTV is one of the applications that uses multicast to make efficient usage of bandwidth on the network (Klicktv, 2013a).

Multicast packets are sent to a multicast IP address. For each multicast group there is such an IP address, and computers who want to join the multicast group send and listen also on this IP addresses. These IP addresses are assigned directly by IANA, thus they are reserved (Albanna & al., 2001). In Table 1 the ranges of the multicast addresses are given.

TABLE 1. Multicast IP address ranges (Cotton & al., 2010)

Start IP	End IP	Network	Purpose
224.0.0.0	244.0.0.255	224.0.0/24	Local Network Control Block
224.0.1.0	224.0.1.255	224.0.1/24	Internetwork Control Block
224.0.2.0	224.0.255.0		AD-HOC Block I
224.1.0.0	224.1.255.255	224.1/16	ST Multicast Groups
224.2.0.0	224.2.255.255	224.2/16	SDP/SAP Block
224.3.0.0	224.4.255.255		AD-HOC Block II
224.5.0.0	224.255.255.255		RESERVED
225.0.0.0	231.255.255.255		RESERVED
232.0.0.0	232.255.255.255	232/8	Source Specific Multicast Block
233.0.0.0	233.251.255.255		GLOP Block
233.252.0.0	233.255.255.255	233.252/14	AD-HOC Block III
234.0.0.0	238.255.255.255		RESERVED
239.0.0.0	239.255.255.255	239/8	Administratively Scoped Block

For multicast it is necessary to have a type of spanning tree. The construction of this spanning tree however depends if the network is sparse or dense. With dense mode the network starts from a full broadcast, all the routers will send the packets to all other networks and destinations. When a network does not require the multicast they send back a prune message. When a router receives a prune message from all its destinations it will send on its turn a prune message towards the source. (Welcher, 2001a).

With sparse mode it works the other way around. For sparse mode there is need for a RP, a rendezvous point, a router that will keep track of where the multicast has to go. When a multicast packet enters the group of routers, it is send to the RP. If a client wants to receive the multicast, it will send an IGMP join request to its closest router. This router will send a PIM join towards the RP. Now the RP can send the multicast packets to the new client. With this method the place of the RP matters as it will be the central place to divide the multicast. To avoid this inefficient way of working the routers will try to search for the origin of the multicast once the bitrate reaches a certain threshold. By default this is 0kbps, which means it will automatically start looking after establishing the multicast stream. In Figure 4 it is possible to see how the first path was through the shared tree (green dotted line) and how it is pruned once the path along the source tree is established (top blue line) (Welcher, 2001b).

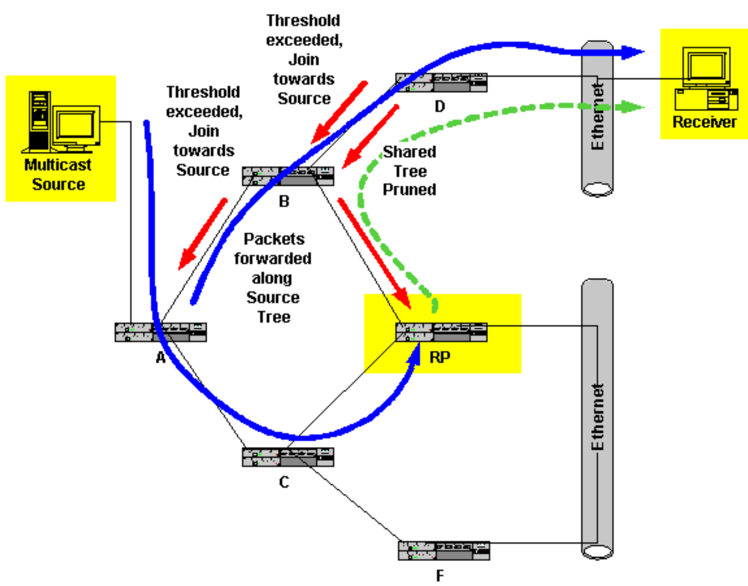


FIGURE 4. Process of PIM sparse mode (Welcher, 2001b)

For the same reason as broadcast, multicast requires UDP.

2.5 Other routing schemes

2.5.1 Anycast

Anycast is neither a real broadcast nor a real unicast routing scheme. With anycast packets will be send to one receiver that is available in a group of receivers. It is a one to one-of-many relation. In Figure 5 you can see a symbolic representation of anycast.

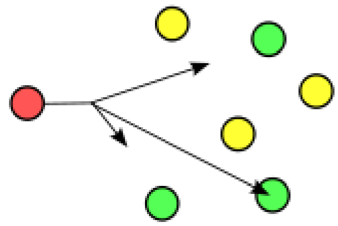


FIGURE 5. Anycast

An example of anycast is root DNS servers, to provide them with redundancy. If one of them is not available the packet will be resend to the next address in the pool of receivers. This provides a continuous availability of the DNS service that is needed for every single lookup of a domain name (Abley, 2006).

Anycast is mainly intended to use UDP but these days also more and more TCP applications are used, like content delivery services (Guan, 2011).

2.5.2 Geocast

The last routing scheme is geocast. As the name itself implies, it uses geographical conditions to deliver the packets, as sending packets to all receivers in one specific location. Geocast is a one to many relation, which makes it similar to broadcast and multicast. In Figure 6 you can see a symbolic representation of geocast.

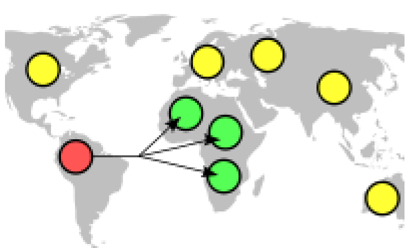


FIGURE 6. Geocast

Until today there are not any significant applications that use the geocast routing scheme. In the nineties geocast was stated as a promising routing scheme. They said it had very much opportunities, for commercial and governmental use (Imielinski & Navas, 1996).

3 STREAMING MEDIA

What is exactly streaming media? There are a lot of answers to this question as there are a lot of perspectives to look at this matter. In short the difference between streaming media and non-streaming media is that with streaming you already access the file before it is completely downloaded. It is the same difference as watching a movie on a television channel, or first ordering it on DVD or Blu-Ray to put it later in the media player. For the media it matters how it will be delivered to the media player, as there are a couple of factors that are different (Topic, 2002, 10).

Error correction is one of these factors. When a media file is streamed and you lose a couple of bytes during the stream, it can happen that the stream gets corrupted if there are no precautions made. If streaming a video would be literally showing every byte that arrives directly on the screen (like with analog television), a loss of bytes would mean a stop, a frame drop, or depending on the codec an unwanted color pattern instead of the expected video. An easy way to counter this is to buffer a part of the stream before showing it to the viewer, to build up a reserve. It gives the client time to request lost bytes to the server to be send again, this way the codec will not have to support any error correction. Though there is a part that is downloaded first, since it’s relatively small compared to the result, it still counts as streaming (Topic, 2002, 10-11).

Next to buffering also forward error correction exists. With forward error correction the encoding technology or codec sends data along with the original data, so when errors occur, bits are lost or wrongly received, the client itself can reconstruct the original data without requesting the packets again to the server. This causes that the actual data send will be bigger than the original media size, but it makes the use of a resend request unnecessary, therefore the use of the UDP transport protocol possible (Goff, 2003).

Now what is important for a file format to be streamed? It all depends on what the purpose of the stream is. For on-demand content, using UDP is possible, but also TCP offers a great possibility. With on-demand content we can use a buffer, ask for a resend of lost bits if necessary as the file is stored on the server. If we have however live content like a television channel, a live sports game, the server might not be able to resend missed bytes, and we would have less need to pause the video at any given time. Therefore the use of TCP over

UDP with real-time content is not an advantage (Pennington, 2011). This all matters for the file format. A file format for on-demand video needs to be small and compressed and on the other side a real-time file format needs to have forward error correction.

There are many formats, codecs and containers for media. The most commonly used technologies are Flash, QuickTime, Windows Media, (Grula, 2010) and the new WebM (The WebM Project, 2012a). Each of them have a specific character, behavior and possibilities.

3.1 Flash video

Flash is a commonly used format these days to display video on websites. The main advantages of Flash are the ability to stream over HTTP and the ability to be played on almost any platform (any platform that supports Adobe Flash). It is a format that also uses a high compression rate, which means small files with high quality (Conjecture Corporation, 2013).

Almost every major website like YouTube, Vimeo or Dailymotion, who provides on-demand video uses the Flash format. The key lies in the fact that the Flash platform allows each website to design their own online media player, completely customized to their own needs. Then every user with Adobe Flash installed on their device, is able to watch the videos. This was a problem for iOS device, like the iPod and iPhone, as Apple does not support Flash on their devices for several reasons. However Adobe now provides a rollover using HLS (see Chapter 3.2 QuickTime) for iOS clients (Quick, 2011).

However Flash video is still popular today, with the arrival of HTML5 the use of Flash is pushed a little bit backwards. Popular browsers now support the direct implementation of video without the use of a third party player like Flash, QuickTime Player or Windows Media Player (Refsnes Data, 2013).

Flash video comes in two container formats. The older FLV and the more recent F4V. F4V is based on the MPEG-4 ISO, using mainly H.264 video and AAC audio codec. The FLV format is available for content with codecs such as Sorensen Spark and On2 VP6 (Adobe Systems Incorporated, 2010).

The main purpose of Flash video is on-demand video streaming. When streaming with flash you have two possible protocols: HTTP and RTMP.

HTTP has the main advantage that it is possible for almost any user, even behind most corporate firewalls, to download and view the video. Also for streaming of HTTP it is not necessary to put a special server, a normal web server like Apache can serve Flash video over HTTP. The biggest disadvantages for HTTP are content protection and the need of progressive download. Every played video will be cached on the local computer so the user is able to use the file as he wants. The progressive download has the consequence that if a viewer only wants to view the last part of the video, he has to wait until all data before that point is downloaded.

With RTMP the need of progressive download is not needed, and there is an option for content protection. It is possible to immediately watch the last part of a video and the content of the video is only cached in the Flash player's memory. However to stream Flash video over RTMP it is necessary to use Adobe's Flash Media Server. It is not possible with a simple HTTP server. RTMP uses port 1935 by default. If this fails there is a first fallback on streaming through port 80 and a second fallback tunneled streaming over port 80, masking the packets as HTTP packets.

HTTP and RTMP both use TCP to establish their connection. Therefor it is unable to use it for a multicast stream, as pointed out in Section 2.4 Multicast, multicast needs UDP.

All information about the protocol use of Flash video is based on the information found on the Adobe Press website (Reinhardt, 2007).

3.2 QuickTime

QuickTime is the file format created by Apple Inc. QuickTime player was originally only available for computers running Mac OS X, but eventually QuickTime player also came available for Windows. This way Windows computers could also play the MOV file format. QuickTime was rather late to support streaming compared to the other formats. (Topic, 2002, 212-213)

QuickTime uses the RTP and RTSP protocols for streaming, with a maximum aimed bandwidth of 1.5 megabits per second. Streams can be either a QuickTime or AVI file format, in which the h264 AAC, MP3, MPEG-4 and 3GPP are the preferred encoding format. As mentioned before QuickTime uses RTP and RTSP to stream, though they also support HTTP live streaming, more commonly known as HLS. This is mainly to serve users with a lower internet connection speed and it uses a progressive download method. For firewall issues QuickTime supports HTTP tunneling to encapsulate the packets and passing them through port 80 (Apple Inc, 2005).

Multicast with QuickTime is done with the SDP announcement file, hosted on an http server. This SDP file will give the client the needed information to connect to the multicast stream (Apple Inc, 2012).

QuickTime is an excellent candidate now for streaming or multicast. It supports also commonly used encoding formats and is playable on OS X and Windows. Support for Linux is available through open source software.

3.3 Windows Media

Windows Media is a file format created by Microsoft to use with the Windows Media player, delivered with the Windows operating system. The Windows Media technology is built by Microsoft's Digital Media Division on top of the already existing DirectX core technology, also property of Microsoft. Microsoft also developed Digital Rights Management for their file format to protect media files from being distributed or copied illegally, which gave online digital media commerce a push forward. (Topic, 2002, 209-210).

In 2006 SMTPE introduced the VC-1 standard to the world. The Windows Media Video 9 codec is Microsoft's implementation of this standard. WMV9 supports the three profiles, Simple, Main and Advanced. Microsoft claims that their format can compress HD video material two until three times more effective than MPEG2 for the simple and main profile. In the advanced profile, WMV9 can deliver transport independent delivery. This way standard broadcast infrastructures can be used to deliver WMV9 content (Microsoft, 2012a).

Windows Media uses either the RTSP, MMS or HTTP protocol to deliver a stream. MMS is the proprietary protocol of Microsoft to deliver streams over UDP (MMSU) or TCP

(MMST). For multicast it is necessary to use UDP on either RTSP or MMS. When using multicast the announcements are made with ‘nsc’ files. These files contain all the needed information for the media player to connect to the multicast group. For compatibility and rollover functionality, the server will always try to negotiate the best usable protocol, supported by server and client. Therefore there is a rollover option to stream over HTTP. However as said before, multicast cannot be streamed over HTTP (Nelson, 2007).

Windows Media is supported on any computer running Windows Media player or Windows Media extensions for QuickTime. On Linux Windows Media is supported for example by VLC Media player, however it cannot receive a multicast stream from a Windows Media server (VideoLAN, 2013).

3.4 WebM

WebM is a fairly new open source video format supported by Google. The development of WebM is in hands of the WebM Project and is based on the open source Matroska file format, a popular container format to serve high definition video. However to maintain compatibility and need of only a few codecs, WebM only supports a few stream types compared to Matroska. WebM is created with the aim on streaming video over HTTP through web browsers, embedded in an HTML5 webpage. However it still is in development the format is already supported in some browsers and provided as a beta test on YouTube.

The codecs used in WebM are only VP8 for video and Vorbis for audio. The VP8 codec was previously owned by On2 Technologies but bought by Google in February of 2010. On the website of WebM there is an example how to use WebM over UDP (The WebM Project, 2012b). Since the VP8 codec is used the streaming protocol next to HTTP is RTP. A draft is available on IETF how to encapsulate VP8 into RTP (Westin, 2013).

WebM might be able to support multicast on later time as it is capable of using UDP and RTP. Considering it open source character and support in some popular browsers it might be worth watching this format for future streaming purposes.

4 THE INTERNET AND MULTICAST

4.1 What is the Internet

The internet today is a huge network of tier 1, tier 2 and tier 3 ISP’s and thousands of servers and clients. Connections go through fiber optic, satellite, Ethernet, coax, telephone lines, Wi-Fi and many more. All the connections, clients, servers and ISP’s are connected to each other by routers and switches.

On the internet data is send as packets from device to device, forwarded through a complex path of connections to reach the destination. The path is chosen based on routing algorithms to get the most efficient way out of the thousands of possibilities. The speed is incredibly fast and is not always related to the distance between the server and the client.

However on this magic network not everything is possible by default. A lot of the functionalities depends on the support for the protocols used by all devices connected. For example a simple web page request with a web URL in the browser needs the DNS protocol and HTTP. Most of these protocols use unicast and sometimes broadcast.

4.2 Multicast support

For multicast to work all over the internet it is required for every device, or at least the nodes. Until today this is not the case. It is not a question if the devices are capable or not, because they are, but more a question if it is favorable to activate the support for multicast. A lot of ISP’s for example charge users by the amount of data they download, or give users a monthly bandwidth. With multicast they would not be able to see all the traffic passing through per user. Also content providers like to know how much users are watching their content. As it is much easier with unicast to count the users, which are needed for commercial income, they do not tend to use multicast over unicast. The cost of providing unicast does not weigh against the revenues of statistics which are valuable in the commercial world.

That being said it does not mean multicast over the internet is completely impossible. There are other ways to achieve multicast routing. One example are overlay networks. These networks are application based networks on top of the current internet infrastructure, visualized in Figure 7. By using tunneling protocols and other methods they build a virtual network on top of the Internet. In this virtual network it is possible to enable multicast, as the routers on the internet see the packets traveling on the physical internet as unicast packets (Kumar, 2006).

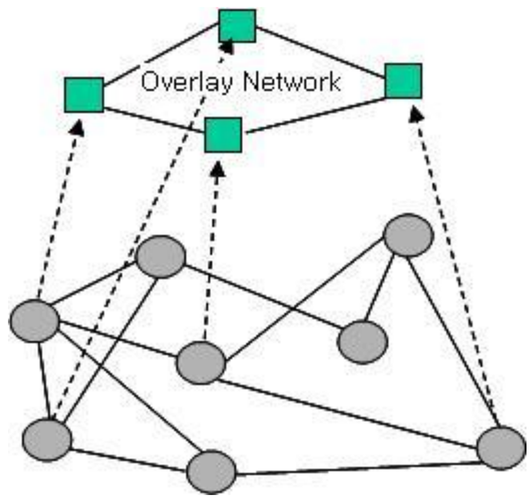


FIGURE 7. Visualisation of an overlay network (Bok, 2002)

Although multicast might not be available on the internet, it is commonly used in corporate networks and maybe a better example is IPTV providers. To distribute IPTV on the own network of an ISP they can enable multicast to work within their network and with their devices. This offers the ISP a bandwidth efficient way to distribute television over their existing network (Klicktv, 2013b). An example of an ISP distributing multicast IPTV is the Belgian company Belgacom (Belgacom SA, 2012).

4.3 Past projects

In the past there have been some bigger projects to make multicast supported on the internet. Two of them are worth mentioning: the Mbone and Mcast.

4.3.1 Mbone

The Mbone was a try to create a backbone on the internet to route multicast traffic. Like the current backbone they wanted to create a backbone where it was possible to connect to for providing multicast support. The Mbone got its rise from 1995 and connected subnetworks to each other over the internet through tunneling. Each subnetwork was multicast enabled and the communicated with each other using Distance Vector Multicast Routing Protocol, or DVMRP. It was a simple mechanism but it required a manual administration to set up all the connections between the subnetworks. As the network grew toward the end of the nineties DVMRP itself required too much bandwidth to operate. Newer protocols like PIM and BGP replaced the Mbone completely after 2001 (Internet2, 2004).

4.3.2 Mcast

Mcast was a project from the Vrije Universiteit Brussel trying to enable multicast access for everyone on the internet. With tunneling technology they tried to capsulate the packets over the existing network to other multicast nodes. The configuration of the tunnel clients running on the client computers would be automatic (Goossens, et al., 2006).

The researchers hope was to show ISP's and content providers the usefulness of multicast. If enough clients would be using multicast, they could be convinced of investigating in multicast on the ISP's network. However after the project professor Marnix Goossens said they under estimated the current business model. As for now there is nobody at the Vrije Universiteit Brussel researching multicast anymore (Goossens, 2013).

5 SIMULATING INTERNET

5.1 Scenario

The basic idea of this thesis is to stream a video file over internet which is multicast enabled. So it is needed to simulate the following components: a streaming server, the internet, ISP's and clients. When the simulation model is set up there will be two main case studies: multicast with PIM in dense mode and multicast with PIM in sparse mode. For more information about PIM dense and sparse mode, refer to section 2.4 Multicast.

The case for dense mode has the target to quickly get the network going. It is easy to set up, requires not much of configuration, and therefore it is a good way to start the basic testing. In this case all the routers will be configured in dense mode and packet traces will be done to see how the traffic flows. Also the consequences of a multicast disabled ISP will be covered.

The second case is the spare mode configuration. It requires a bit more configuration and will be more similar to a real life situation on the internet. This time the internet routers will be configured in sparse mode, independently from the ISP routers. Again there will be packet traces to look at the flow of the packets and the effects of a multicast disabled ISP.

5.2 Topology

The construction of this simulation can be done in several ways. For this project it is needed to simulate internet. On the internet routers route packets to the right destination, therefore it is necessary to have at least one network which resembles the internet or the cloud. For more testing purposes, especially to see how multicast traffic flows, the internet backbone will be resembled by three routers and three networks in this thesis.

The ISPs will each be resembled by one router and switch. There will be two ISPs, one multicast enabled (ISPE) and one multicast disabled (ISPD). They will both have three clients, resembled by computers.

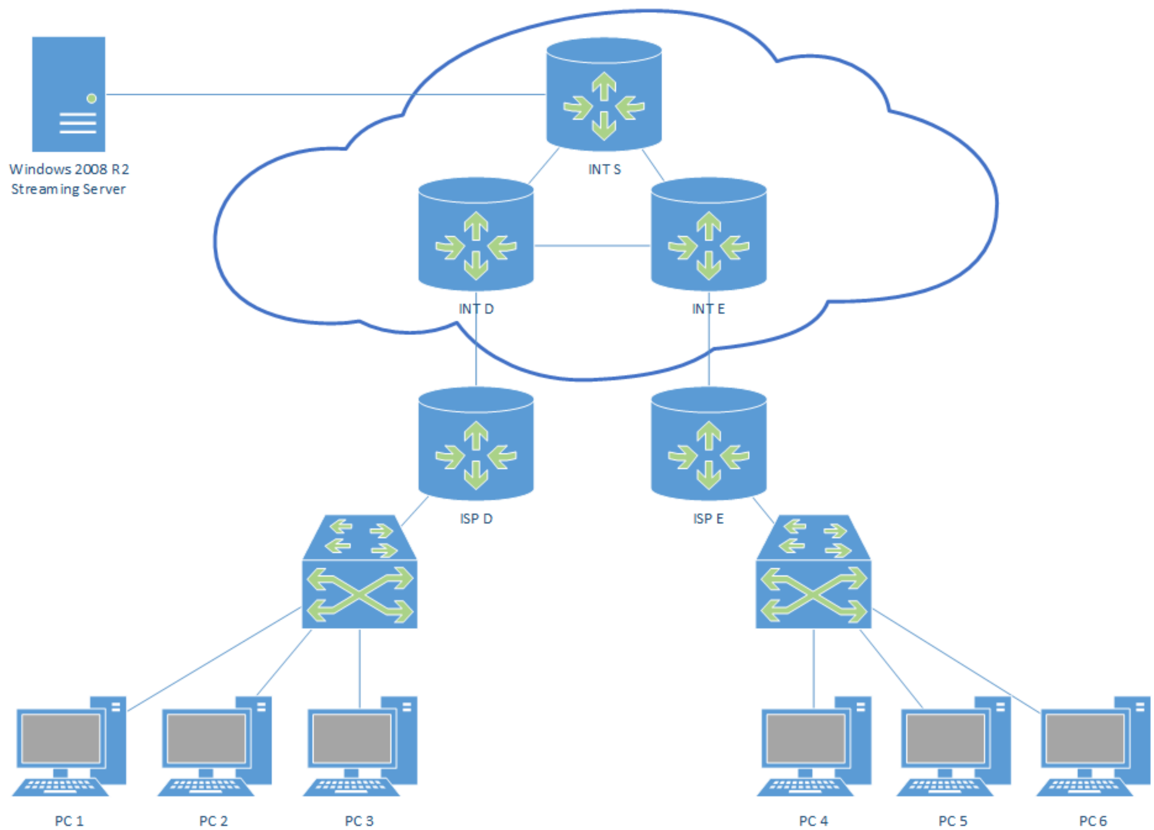


FIGURE 8. Network Topology

The server will be directly connected to the simulated internet backbone without an ISP or other router as this would not provide any relevant testing opportunities in this scenario. A full visual representation of the network is shown in Figure 8.

The network hardware used in the topology are five Cisco 2911 routers and two Cisco Catalyst 2960 switches. The router type supports multicast routing and has three Gigabit Ethernet connections. The switches have twenty four Fast Ethernet and two Gigabit Ethernet uplink ports. The connection of all devices is made by Cat5E Ethernet cable. The choice for connecting the routers with crossover Ethernet cable over serial connections is a pure bandwidth decision. Also there is no need for special WAN protocols supported by the serial interfaces.

5.3 Configuration of the routers

To configure the routers it must be thought how to subnet the whole network topology. The subnets were configured according to the information in Table 2. To increase the ease of use the Gigabit Ethernet port 0/0 is pointed outward of the internet backbone for the three internet routers (INTS, INTE, INTD). For the ISP routers the Gigabit Ethernet port 0/0 is the connection with the internet backbone and Gigabit Ethernet port 0/1 serves the internet customers. Each of the internet routers also gets a loopback interface which will be needed for some multicast configurations later.

TABLE 2. Router address configuration

Router	Interface	IP-address	Subnet mask	Network
INTS	Gi0/0	10.0.1.1	255.255.255.0	10.0.1.0
	Gi0/1	10.0.0.5	255.255.255.252	10.0.0.4
	Gi0/2	10.0.0.1	255.255.255.252	10.0.0.0
	Lo0	10.1.1.1	255.255.255.0	10.1.1.0
INTE	Gi0/0	10.0.2.1	255.255.255.0	10.0.2.0
	Gi0/1	10.0.0.6	255.255.255.252	10.0.0.4
	Gi0/2	10.0.0.9	255.255.255.252	10.0.0.8
	Lo0	10.1.2.2	255.255.255.0	10.1.2.0
INTD	Gi0/0	10.0.3.1	255.255.255.0	10.0.3.0
	Gi0/1	10.0.0.2	255.255.255.252	10.0.0.0
	Gi0/2	10.0.0.10	255.255.255.252	10.0.0.8
	Lo0	10.1.3.3	255.255.255.0	10.1.3.0
ISPE	Gi0/0	10.0.2.2	255.255.255.0	10.0.2.0
	Gi0/1	172.16.0.1	255.255.192.0	172.16.0.0
ISPD	Gi0/0	10.0.3.2	255.255.255.0	10.0.3.0
	Gi0/1	172.16.64.1	255.255.192.0	172.16.64.0

The configuration of the routers is done through terminal software running on a simple computer which is connected with a serial cable to the router. In Appendix 1 the initial router configuration can be found. In Section 7.1 Dense mode the routers will be configured the first time to work for multicast.

To simulate the internet there are the three INTx routers and the two ISPy routers (where x stands for S, D or E and y for D or E). Configuring all this routers with static routes would take a lot of time, so it is better to configure them with OSPF so they can learn the routes dynamically. Enabling OSPF is done by creating an OSPF process in the routers global configuration mode by entering the following command:

```
INTS(config)# router ospf 1
```

In this command the number ‘1’ represents the process number. After the command the router allows you to add the networks with its wildcard mask (the inverse of a subnet mask) and area number. The networks entered will be broadcasted to the other routers. Every router needs to broadcast its own connected networks and also the static routes, which will be configured later. Here follows an example of the INTS router commands. The other routers are configured analog:

```
INTS(config-router)# network 10.0.0.0 0.0.0.3 area 0
INTS(config-router)# network 10.0.0.4 0.0.0.3 area 0
INTS(config-router)# network 10.0.1.0 0.0.0.255 area 0
INTS(config-router)# network 10.1.1.0 0.0.0.255 area 0
```

To redistribute the static networks, they first need to be added to the router. The INTE and INTD both have a static route to the network of the ISP. This static route can be described in the simulation as the IP range that IANA gives to a certain ISP. The commands to do this are as followed:

```
INTE(config)# ip route 172.16.0.0 255.255.192.0 10.0.2.2
INTE(config)# router ospf 1
INTE(config-router)# network 172.16.0.0 0.0.63.255 area 0
INTE(config-router)# redistribute static subnets
```

The ISP routers will not be configured with OSPF. The ISP routers will use a default route to forward all traffic that is not destined to the ISP its network to the internet. The configuration of the default route is done by the following command:

```
ISPE(config)# ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0
```

There is no need to make static routes for directly connected networks. They are learned by the router automatically.

The network configuration part of the topology of the simulation is now done. What is left is the configuration of the clients and the configuration of the server.

5.4 Configuration of the clients

The clients need a basic configuration. Clients will be running either Windows XP or Windows 8 with Windows media player. The IP configuration of the clients can be found in Table 3. The hardware specifications of the clients are: Intel Core 2 Duo E7500 Processor (x64) running at 2.93GHz with 4 to 8GB RAM and an 80GB hard drive. If necessary extra software like VLC media player can be installed. This however depends on the selection of the server system and media format.

TABLE 3. Client address configuration

Computer	IP-address	Subnet mask	Network	Gateway
PC1	172.16.64.11	255.255.192.0	172.16.64.0	172.16.64.1
PC2	172.16.64.12	255.255.192.0	172.16.64.0	172.16.64.1
PC3	172.16.64.13	255.255.192.0	172.16.64.0	172.16.64.1
PC4	172.16.0.14	255.255.192.0	172.16.0.0	172.16.0.1
PC5	172.16.0.15	255.255.192.0	172.16.0.0	172.16.0.1
PC6	172.16.0.16	255.255.192.0	172.16.0.0	172.16.0.1

The way to configure the IP addresses differs per operating system and version. For Windows XP this can be found in the properties of the network adapter in the control panel section ‘Network Connections’. For Windows 7 and higher it is found in ‘Change adapter settings’ in the ‘Network and Sharing center’.

6 STREAMING SERVER

6.1 Selection of server system

The selection of the right sever system to stream video content depends on a lot of factors. First it is necessary to look at the requirements of the system and which formats that will be streamed. For the scenario our main requirement is that the server needs to be able to stream over multicast.

In the world of servers, able to run on standard computers, there are three groups of servers: UNIX (mainly Linux), Windows and Mac OSX servers. In Table 4 there is a short comparison of the relevant requirements and features supported by each server system.

To stream media over multicast it is possible to any of the three families except for the latest Windows server version. The choice will be mainly between ease-of-use, price, and supported formats.

Only considering the supported formats, the Linux based servers have a great advantage. However configuring a Linux server with VLC as streaming component is difficult work and references found on the internet push the administrator to the terminal, away from an easy graphical user interface. Pricewise the options give a disadvantage for OS X. Also the software of OS X does not run on any Intel based computer.

Therefor the choice for Windows Server 2008 R2 is suitable for this scenario. The ease-of-use makes up the fact it lacks a wide variety of supported formats. Since all clients run Windows XP or 8 in the setup environment and Microsoft delivers a free Windows Media format encoder we can rule out any compatibility issues by staying with the same software vendor.

TABLE 4. Server comparison

Server Family	Unix (Linux)	Windows (2008 R2 and 2012)	Mac OS X
Hardware	Any[1]	x86-64 (only x64 from version 2012)[2]	Only Apple based x64[3]
GUI	optional	Yes (core system optional)	Yes
Streaming	Yes (through VLC)	Yes	Yes
Multicast	Supported through VLC	Supported on 2008 R2 through WMS, not available on 2012[4]	Yes[5]
Supported formats[6]	• MPEG-2 (TS) • MPEG-4 (TS) • H/I 263 • H.264/MPEG-4 • MPEG layer 1/2/3 audio • AC3 • MPEG-4 Audio (AAC) • Speex • PCM • µ-law/A-law	• WMA • WMV • ASF • MP3	• Quicktime • MPEG-4 • 3GGP
Price	Free (open-source)	Free through Dreamspark (MSDN Academic Alliance)	$19.99 on Mac running Mountain Lion
Ease-of-use	Difficult	Easy	Easy

[1] (Merrill, 2004)
[2] (Microsoft, 2007) (Microsoft, 2012b)
[3] (Apple Inc, 2013)
[4] (Bristol, 2012)
[5] (Apple Inc, 2005)
[6] (VideoLAN, 2013) (Microsoft, 2013) (Apple Inc, 2005)

6.2 Setting up server

First it is necessary to have the installation medium (a DVD for example) and a computer which will run the server software. For this project the server software was downloaded from Dreamspark and burned to a bootable DVD. The server version is Windows 2008 R2 Datacenter with Service Pack 1. The computer is an Intel Core 2 Duo E7500 Processor (x64) running at 2.93GHz with 3.43GB RAM and an 80GB hard drive.

The installation of the server itself is guided by an installation wizard. The wizard guides you easily through the installation process. First of all it is important to set the right local language and keyboard settings (shown in Figure 9). This makes sure the keyboard acts the way it supposed to and configures the right time zone settings for the local area. The settings in this installation are put to Finnish.

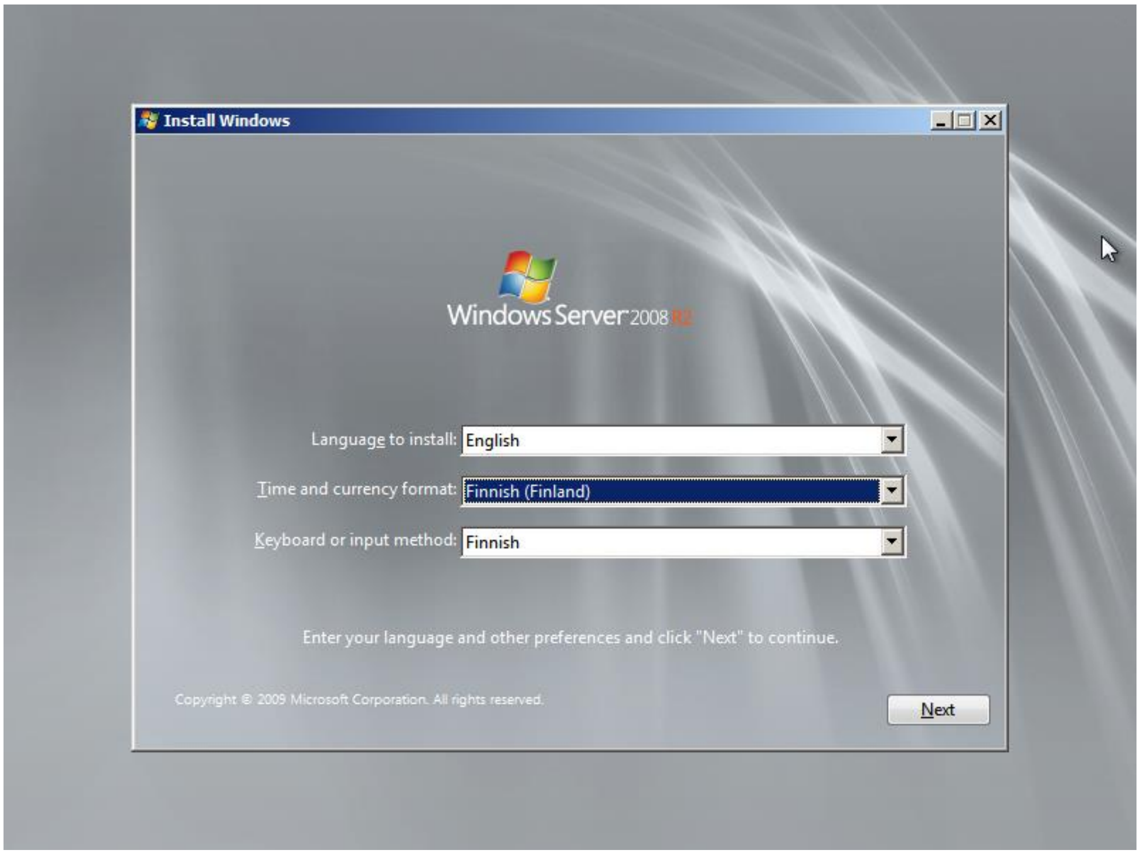


FIGURE 9. Selecting system local settings

Next is probably the most important section in the whole installation process: the choice of which server type to install (Figure 10). Since Windows Media services require at least Enterprise or Datacenter to provide multicast (Microsoft, 2010) it is important to choose at least one of those. The choice between Full Installation and Server Core Installation is up to server administrator himself. The Full Installation comes with a complete GUI and all tools to configure windows server. With this type of installation almost all configuration is done through the GUI, wizards and dialog boxes. With the Server Core Installation Windows installer will only install the necessary core of Windows Server, together with a very limited GUI, a command prompt and PowerShell. Windows Media services works with both versions, although it is easier to install and configure with the Full Installation. However when the server will need all the recourses for streaming it might be better to opt for the Core version which runs less system recourses, is more stable and more secure thanks to the missing layer of an advanced GUI. In the test scenario recourses are not that important due to the scale of which the server will serve so it is better to opt for the more user friendly Full Installation.

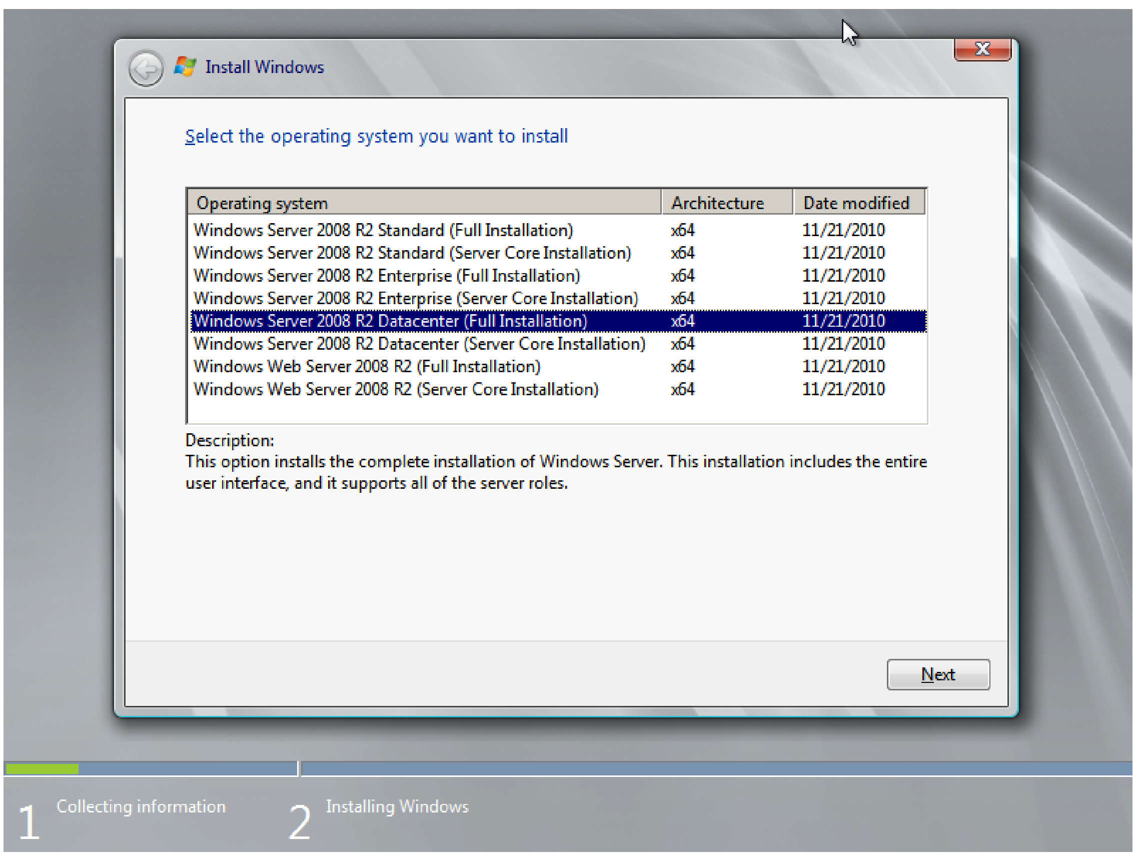


FIGURE 10. Selection of server installation

After this selection, reading and accepting the license terms, and choosing for a full install instead of upgrade, it is up to the choice where to install the machine. Since it is an empty hard drive and it no extra partitions are required it is best to use all the allocated space as the primary hard drive. To do this was a simple click on the unallocated space and click next as shown in Figure 11. After this step the installation will start copying the files. Several reboots may be possible.

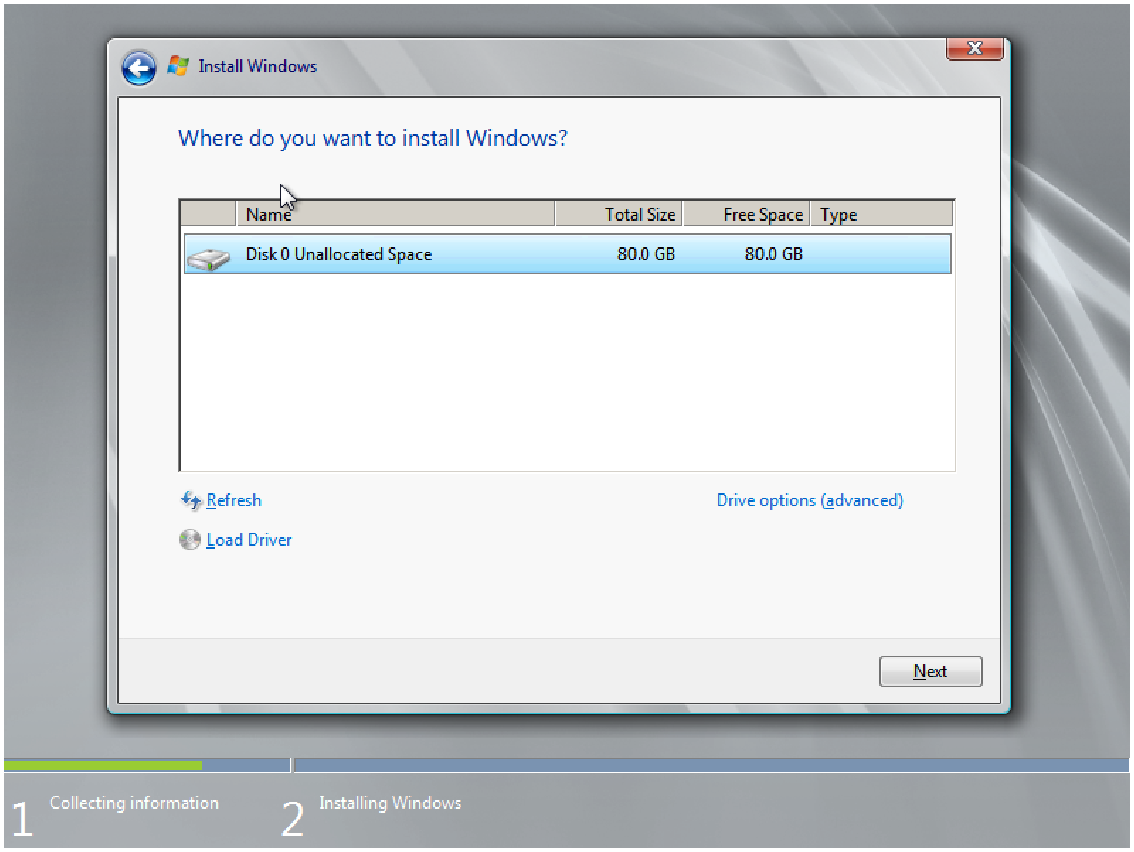


FIGURE 11. Install location

Just before the first login Windows asks to configure an administrative password. After creating a password it is possible to log in and Windows welcomes you with the Initial Configuration Tasks window as shown in Figure 12.Windows Server 2008 R2 is now ready to be configured.

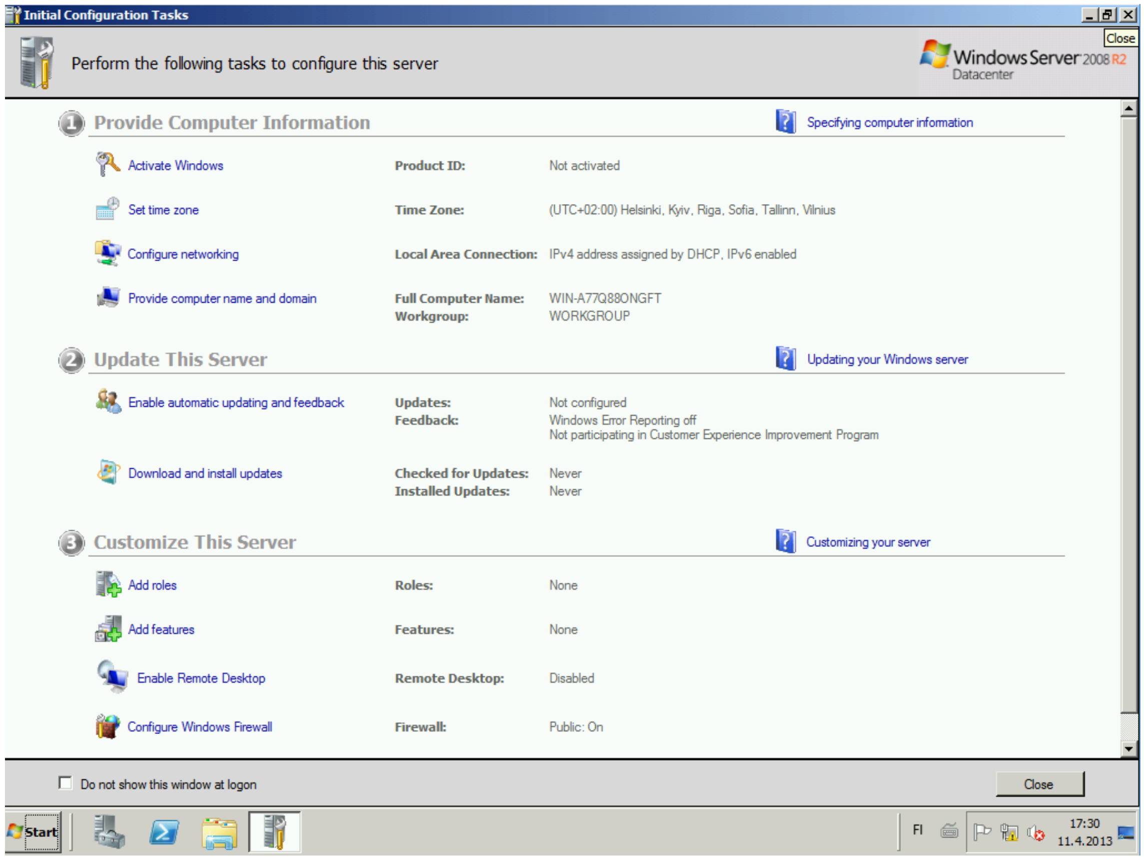


FIGURE 12. Initial Configuration Tasks window

6.3 Configuring server

Now that the server is installed it is time to configure the server. Before installing or even thinking about adding any services it is best to configure the network adapters first. By configuring the network adapters the server will be able to communicate with the network and will fit in the scenario topology. The settings for the server are configured as in Figure 13.

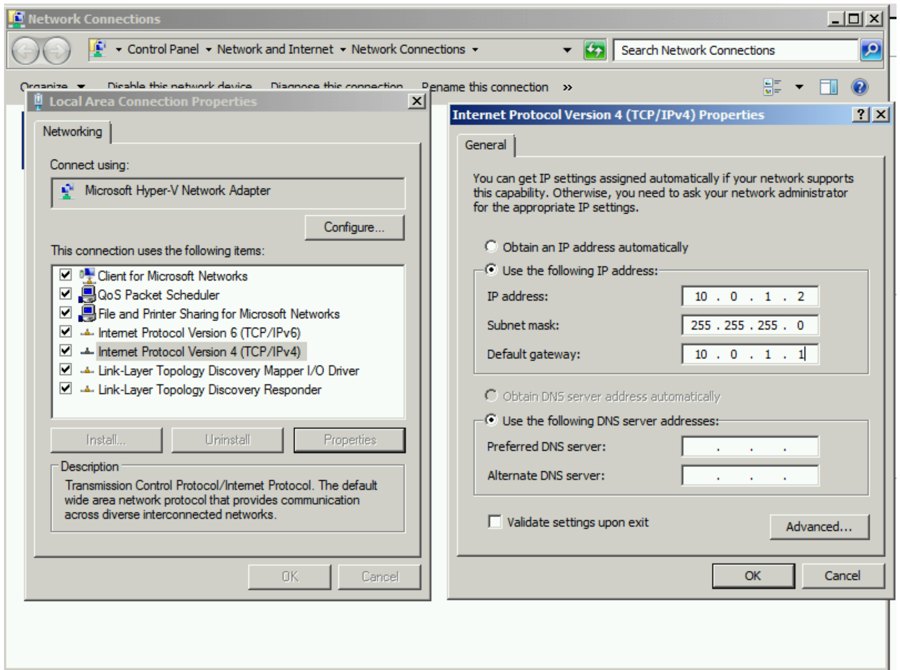


FIGURE 13. Network adapter configuration

In Windows Server the server provides its services through so called roles and features. When installing a role or a feature you give the server the capability to serve the clients. These roles and features can be a DNS server, web server, Active directory service, routing, SMTP server and many more.

To configure the server it is important to know what the server will need to do. The main purpose of this server is to stream over multicast. This can be done through the Windows Media Services role. However it is also necessary to serve the announcement files to the clients. To announce these files the server will also need to act as a basic HTTP web server. If there would be already a web server running it is possible to place these announcement files on that webserver and use the server only for streaming. If the server needs to stream over HTTP it is impossible to run both a web server and streaming server on the same port. Either two servers are required or one of the server services has to run on a different port than 80.

Before installing the web server and the streaming role, the streaming role has to be downloaded as it is not included by standard in Windows Server 2008 R2. The streaming role can be found easily in the Microsoft Download center. After choosing the right language it is possible to download the file. Since the server in the scenario was not connected to the real internet, the file was burned to a DVD. As seen in Figure 14 the package runs as a standalone Windows update. After accepting the license terms, the package installs the support for Windows Media Services on the server. However it does not install the role. The role has to be added to the server later.

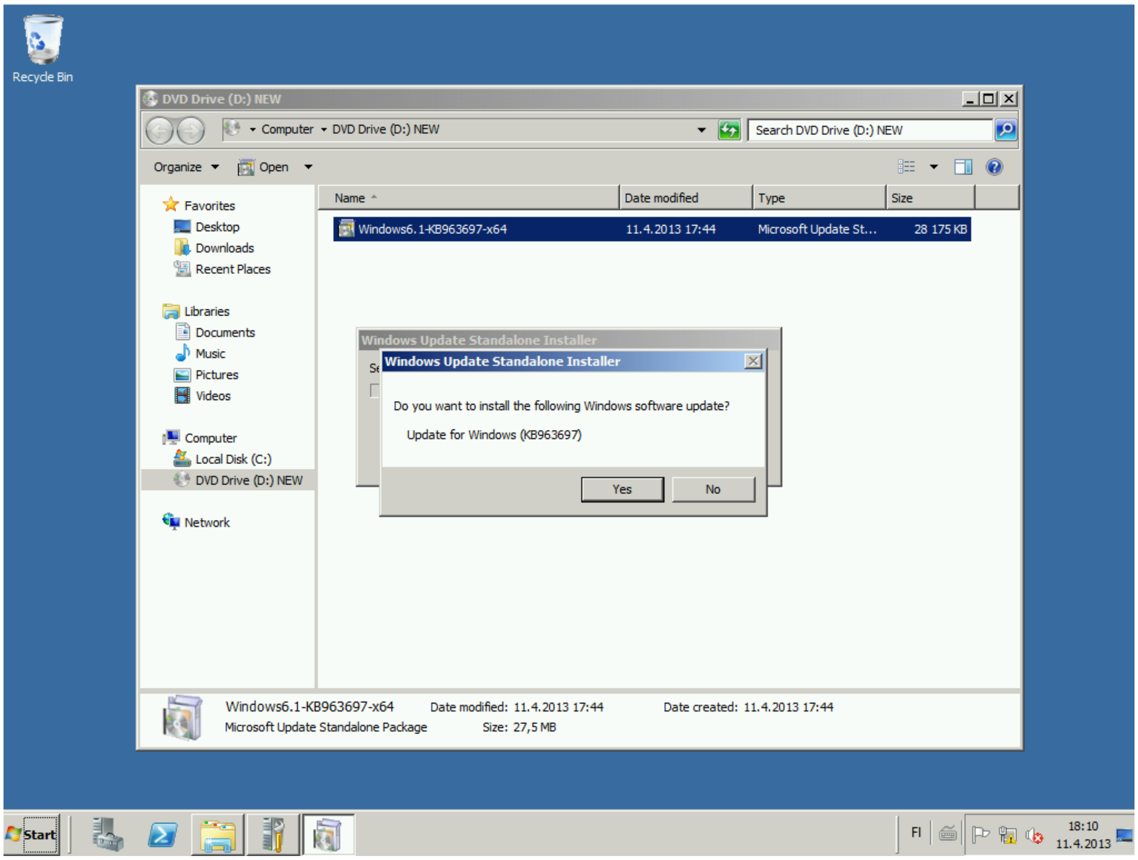


FIGURE 14. Installation of Windows Media Services package

Now that all the required files, software and data is present on the server it is time to install and configure the roles. To start installing the roles it is as easy as clicking ‘Add Roles’ in the Initial Configuration Tasks window. *Note that a lot of steps as updating, configuring a name, activating the server and configuring the security are skipped. This is because in the scenario these settings are not important. However when running a streaming server in a real environment it is hardly recommended to configure and secure the server decently!*

When adding roles the ‘Add Roles Wizard’ shows up. The introduction page reminds the administrator to make sure he did not forget to do any steps, the ones that are skipped in this scenario. On the second dialog, as shown in Figure 15 the wizard asks which roles it has to install. The scenario needs both Streaming Media Services and Web Server (IIS).

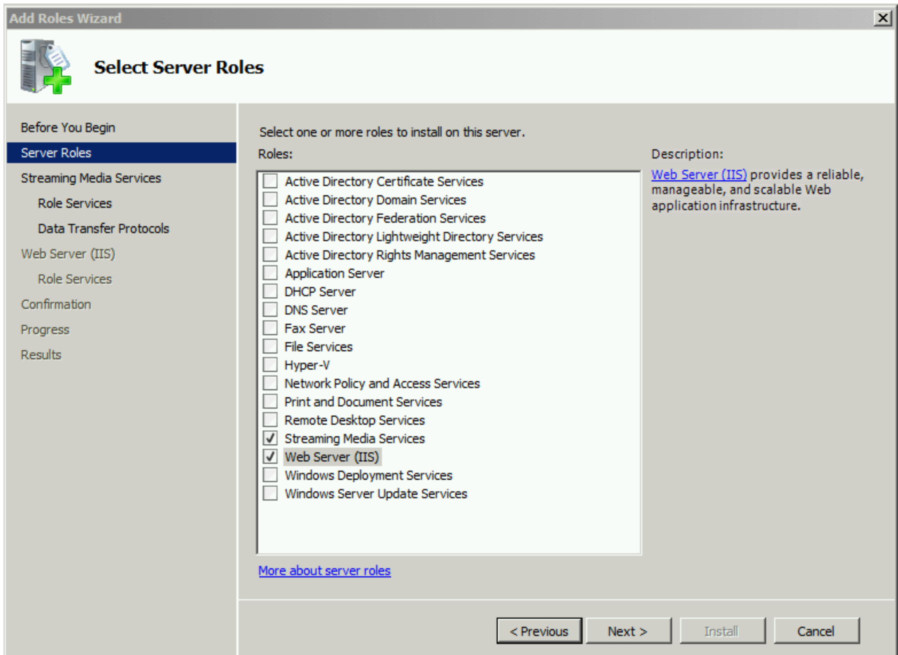


FIGURE 15. Select server roles

The wizard will now ask some initial configuration settings. For this scenario the basic settings proposed by the wizard are sufficient. The setup will only give a warning as shown in Figure 16 regarding HTTP streaming. If the server would need to be able to stream over HTTP it is not possible to run a web server at the same time on the same port. To avoid this either do not install web server or install the web server first and configure the web server not to run on port 80 before installing the streaming services. At the end of all the questions the wizard will install and configure the roles automatically.

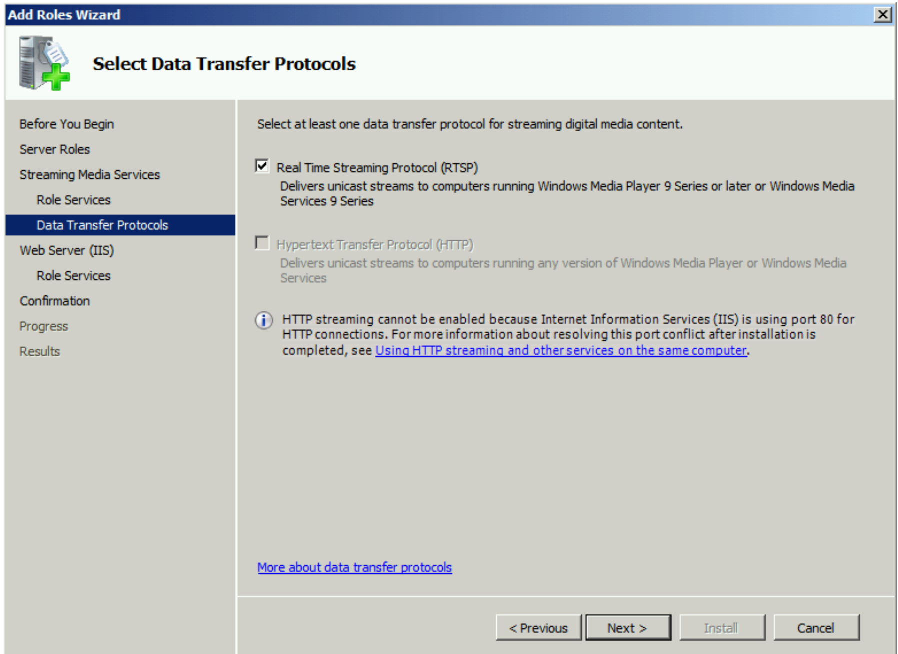


FIGURE 16. Selection of protocols

Before continueing configuring the server it is better to first explore some folders on the hard drive. The web server and streaming role both made a folder on the system drive, in this case drive C. The folders are called 'inetpub' for the web server role and 'WMPub' for the streaming role. In those folders there is a root folder, wwwroot for the web server and wmroot for streaming.Those folders are the root for their service, which means it is the base of the public environment for either HTTP or streaming. These folders will be important for later configuration.

Now it is finally time to set up the actual streaming. Streaming in Windows Server is done by publishing points. Each publishing point serves a specific stream or playlist. There are two different types of publishing points: On-demand and Broadcast points. On-demand broadcast points are for non live video streaming. The client can ask at any given time any available media and control it (pause, play, forward, reverse). Broadcast points are for live events. With these type of streaming the client can watch the stream but cannot control the stream. A publishing point is created by clicking the 'Add Publishing Point' button on the Publishing Points screen in the Server Manager, shown in Figure17.

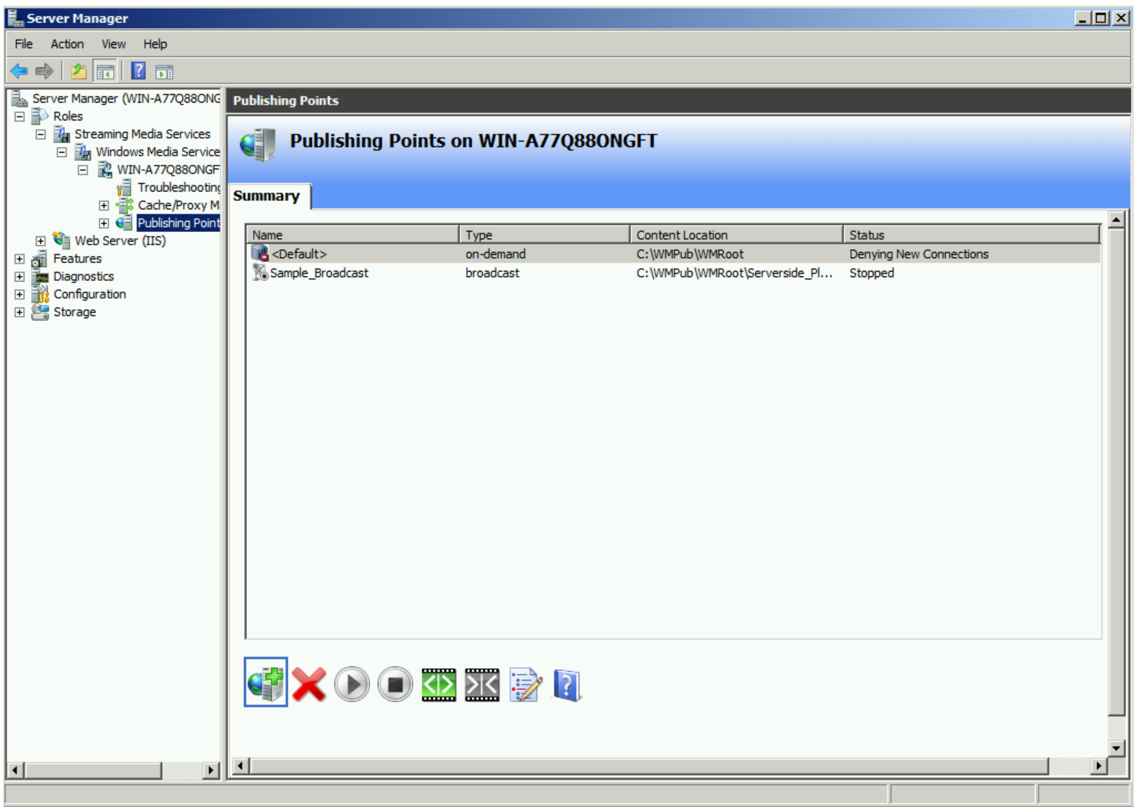


FIGURE 17. Publishing Points Manager

A wizard guides the administrator through the creation of a publishing point. The wizard is fairly easy and straight forward. Therefore only the most important steps will be discussed in this chapter. A full list of the steps of the wizard can be found in Appendix 3.

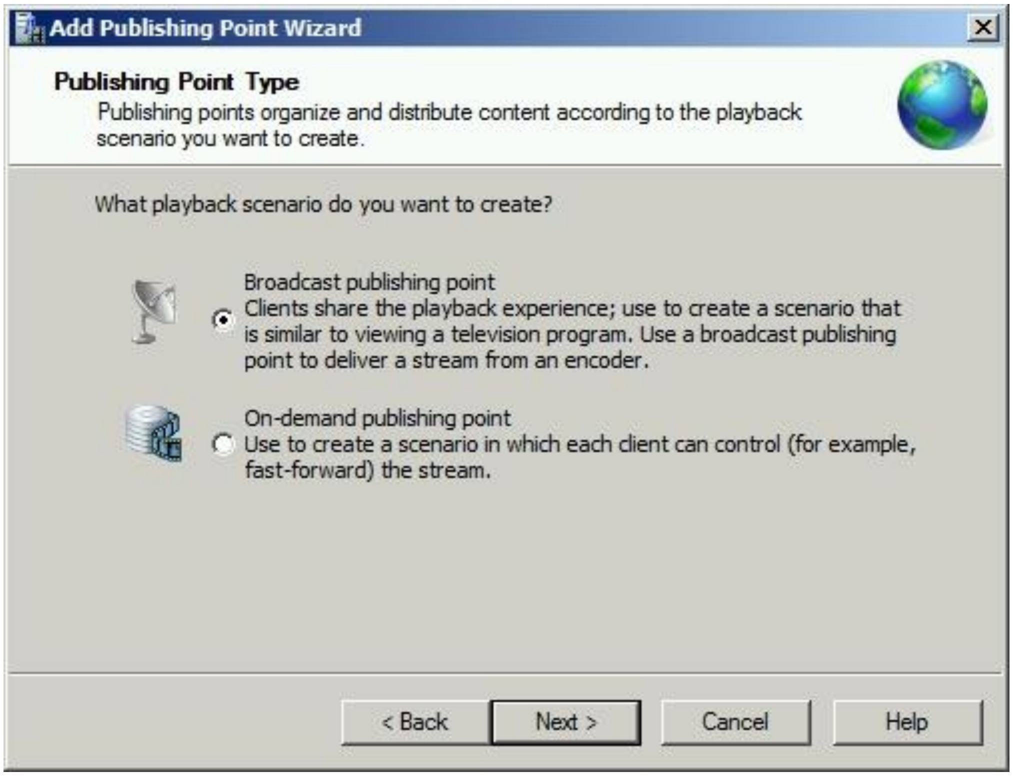


FIGURE 18. Publish Point Type selection

After selecting a name and type of video source the wizard asks which type of publishing point is needed. For a multicast a broadcast point is needed, as shown in Figure 18. Next up the wizard will ask which type of broadcast it should create. There is no need for a unicast, but multicast. Also for now a unicast rollover must be unchecked as it might step in action when the multicast does not work. For troubleshooting and results of the research this can be a problem. The dialog must look like Figure 19 before continuing.

Now the wizard will ask where the source material is located, and then give you a summary of the publishing point it will create. The last screen asks the administrator to create a multicast information file. This is necessary for the clients to connect to the multicast. Select the options as in Figure 20.

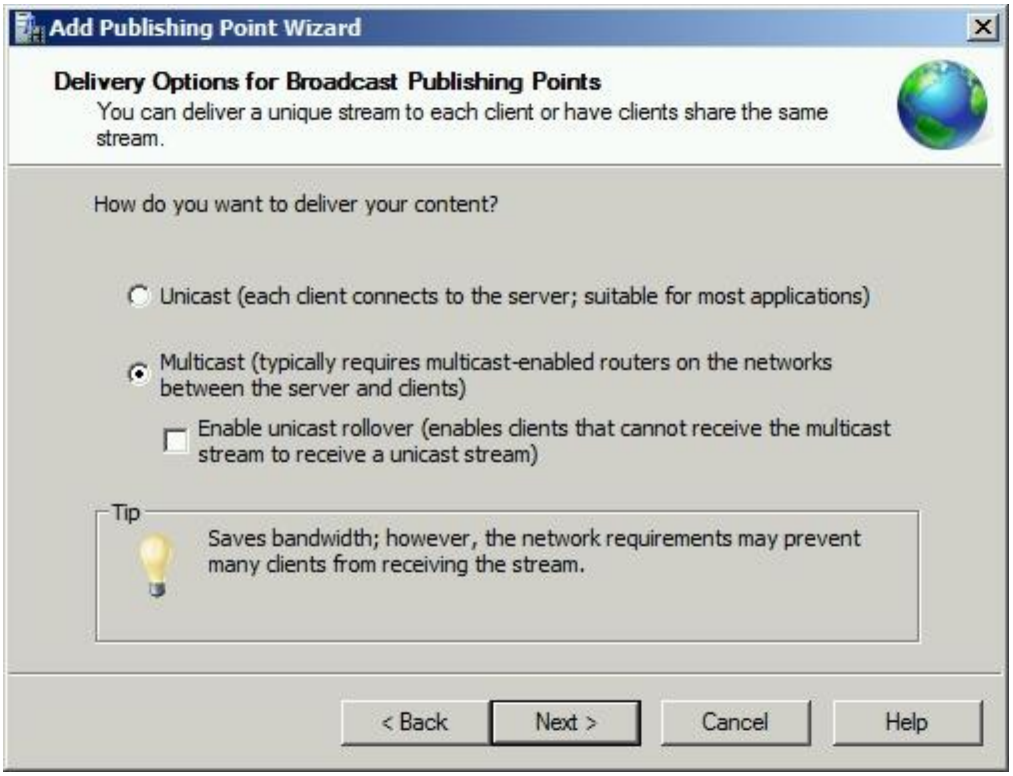


FIGURE 20. Delivery method selection

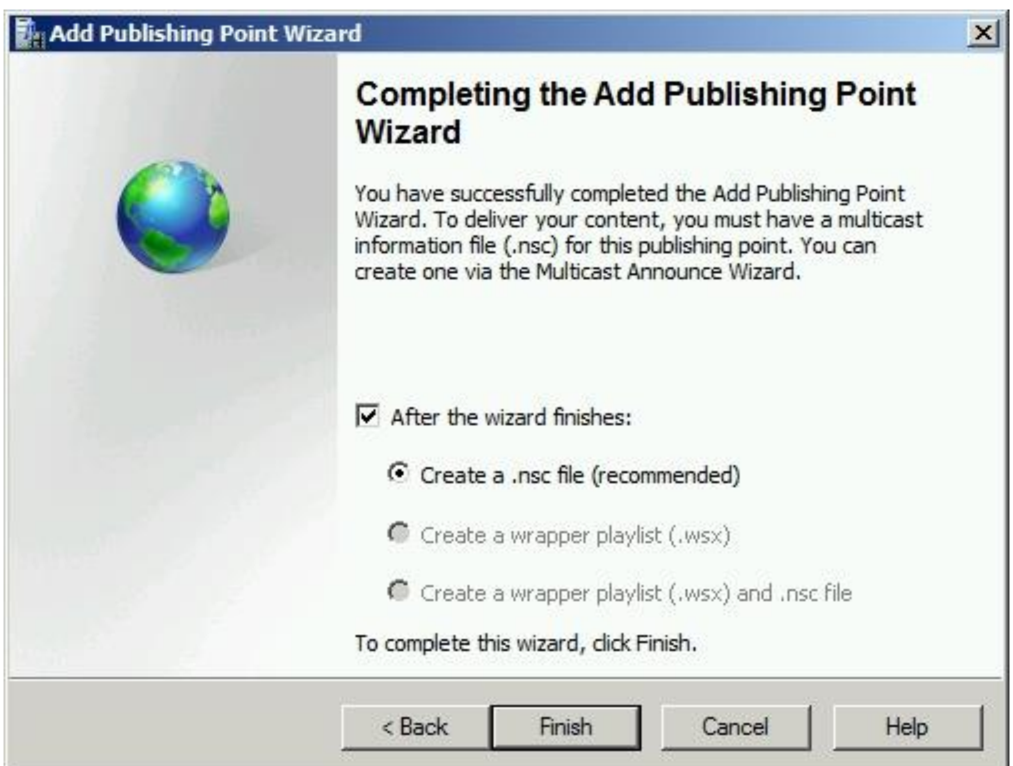


FIGURE 19. Publishing Point Wizard completion

The next wizard for the creation of a multicast announcement shows up. Also this wizard is very straight forward. There are only a couple of things to pay attention to. To make things easier it is recommended to create both a multicast information file (.nsc) and an announcement file (.asx). A web page is not necessary. The wizard will also ask you which file types you are streaming. This is needed to determine the format which you are streaming to add it to the multicast information file. It is possible to add multiple formats to the list. If all the files or sources have the same format, only one source needs to be listed. After a while the wizard will ask the location to save the multicast information file and the announcement file. It is compulsory for the viewers to access these files by an http connection, so they should be stored on a web server. By default the wizard gives you a location in the WMPub folder. This is not where the web server is running, therefore the location should be changed as shown in Figure 21.

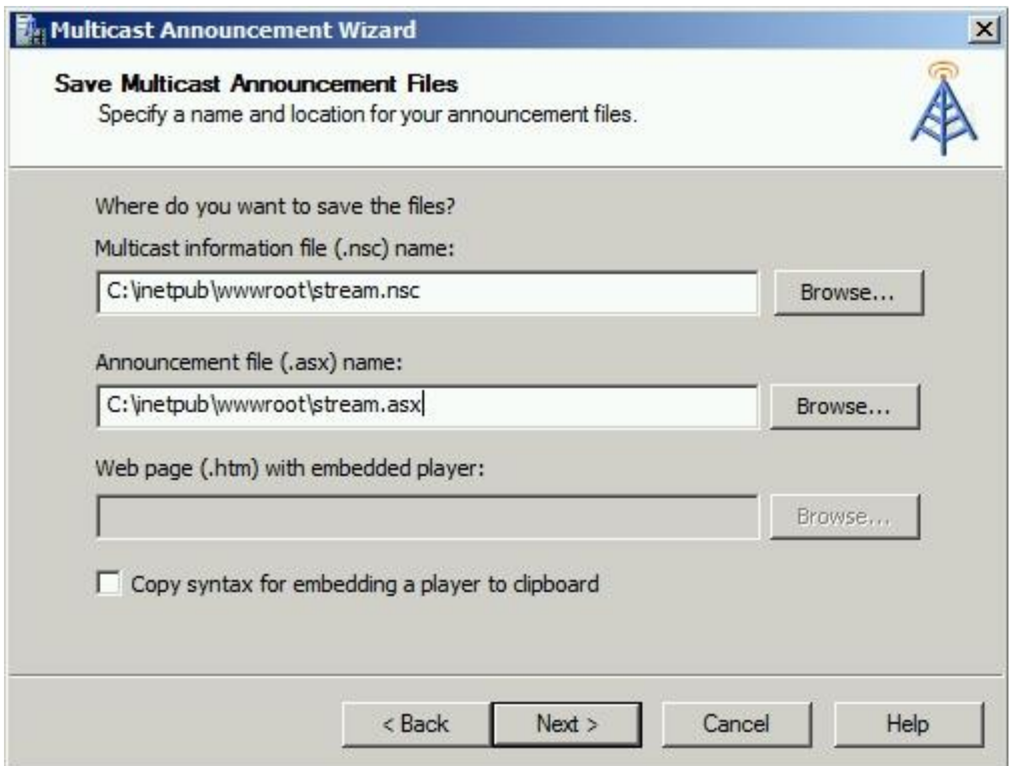


FIGURE 21. Announcement Files save location

The last important configuration is to point out where the users will be able to find the multicast information file. This information is stored in the announcement file. Since the multicast information file was stored in the root of webserver the location needs to point there. The configuration for this server is shown in Figure 22.

After finishing the wizard the configuration of the server is done. The simulation is now ready for analyses and testing. This will be done in the next chapter, Simulation.

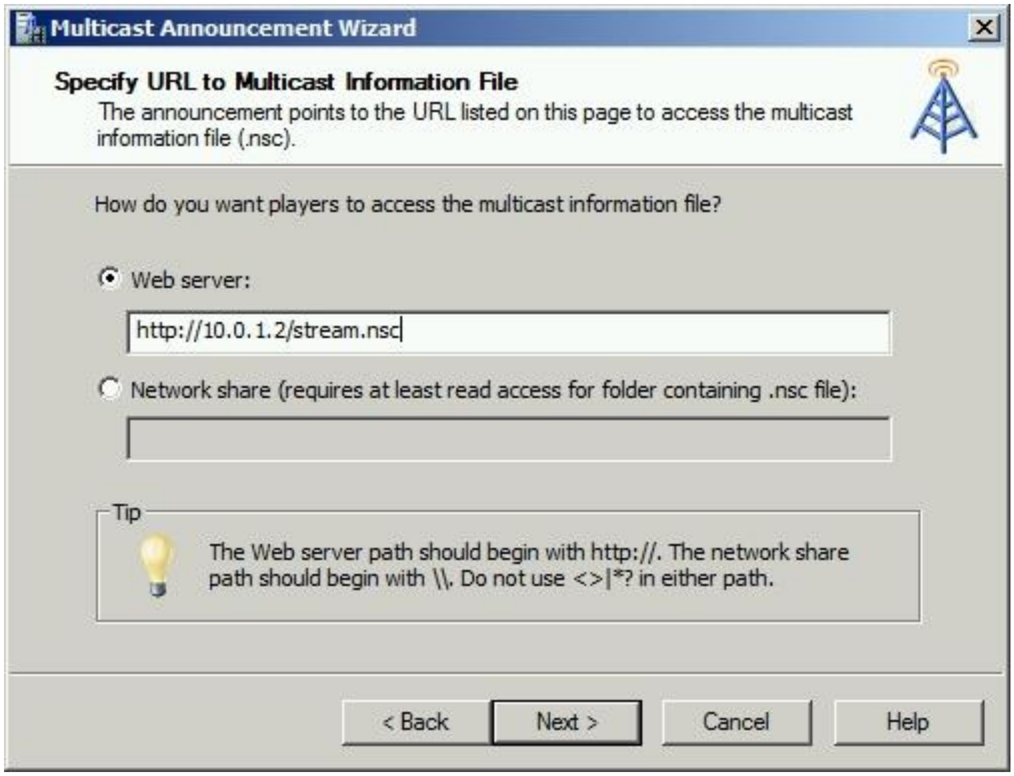


FIGURE 22. Location of the multicast information file

7 SIMULATION

Now that all devices are configured to work it should be possible to reach the server basic web page from the clients. Any errors for the setup to be malfunctioning can be type errors, wrong or faulty cables, or a misconfiguration.

The simulation will exist of two main test cases, the dense mode and the sparse mode, in which the effects of the multicast enabled and multicast disabled ISP will be reviewed. After the tests there will be a work around for the multicast disabled ISP on the server side. For the difference between dense and sparse mode, refer to Section 2.4 Multicast.

But before starting with these cases the routers must be activated to handle multicast traffic. This is done by running the following command on all of the routers, except of course ISPD which will resemble the multicast disabled ISP:

```
INTS(config)# ip multicast-routing
```

Now the routers are ready to handle multicast traffic, however some configuration needs to be done, which are dependent on the cases.

7.1 Dense mode

To configure the dense mode on the routers a command needs to be added to each of all the routers interfaces, with exception of the ISPD router. The command is short and easy:

```
INTS(config-if)# ip pim dense-mode
```

After configuring all the interfaces, also on the INTD router, the internet section and the ISPE router should forward multicast traffic. When the publishing point on the server is started and a computer in the ISPE network requests the announcement file, Windows Media player should start playing the multicast stream.

7.1.1 Multicast enabled ISP

In the multicast enabled ISP the multicast video stream shows nicely on the client computer. As multicast promises, a single stream should serve multiple clients. With packet sniffing between ISPE and the switch it is possible to see this is true. In Figure 23 the first client connects (shown by the TCP traffic) and in Figure 24 another client connects. By looking at the UDP traffic it is possible to see there is no difference between those two cases, which points to a single working multicast stream.

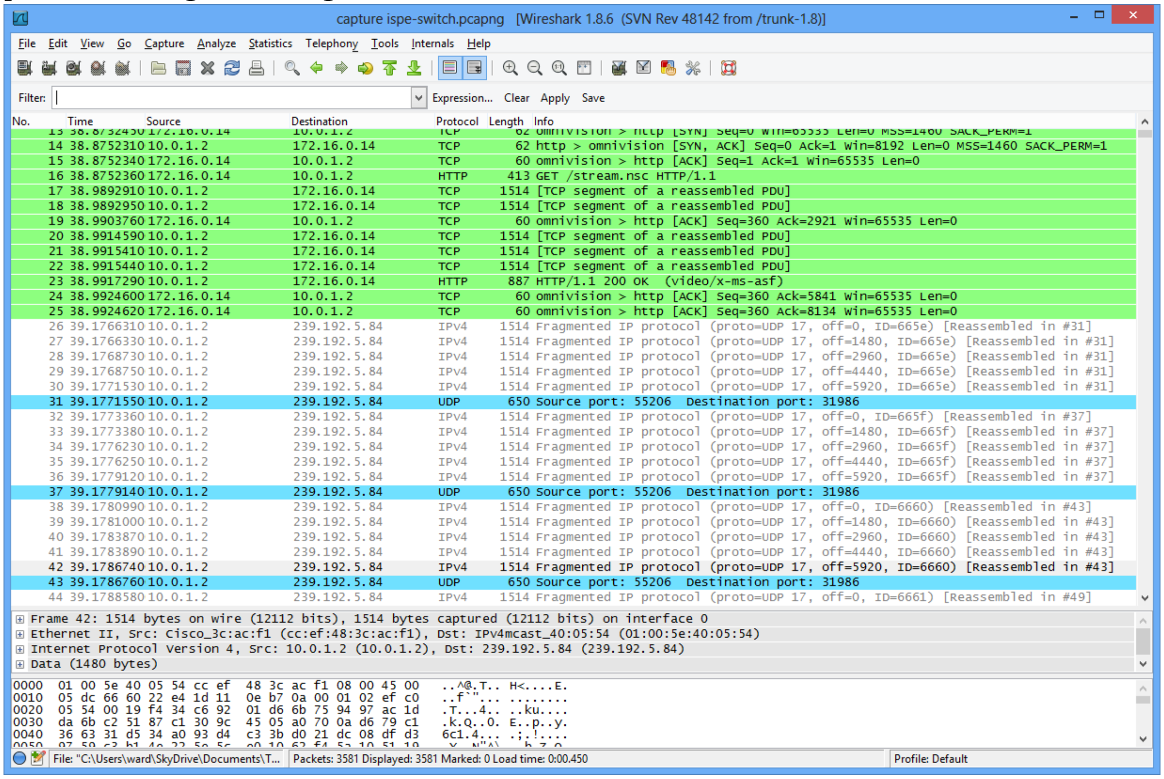


FIGURE 23. Dense mode, first client connects

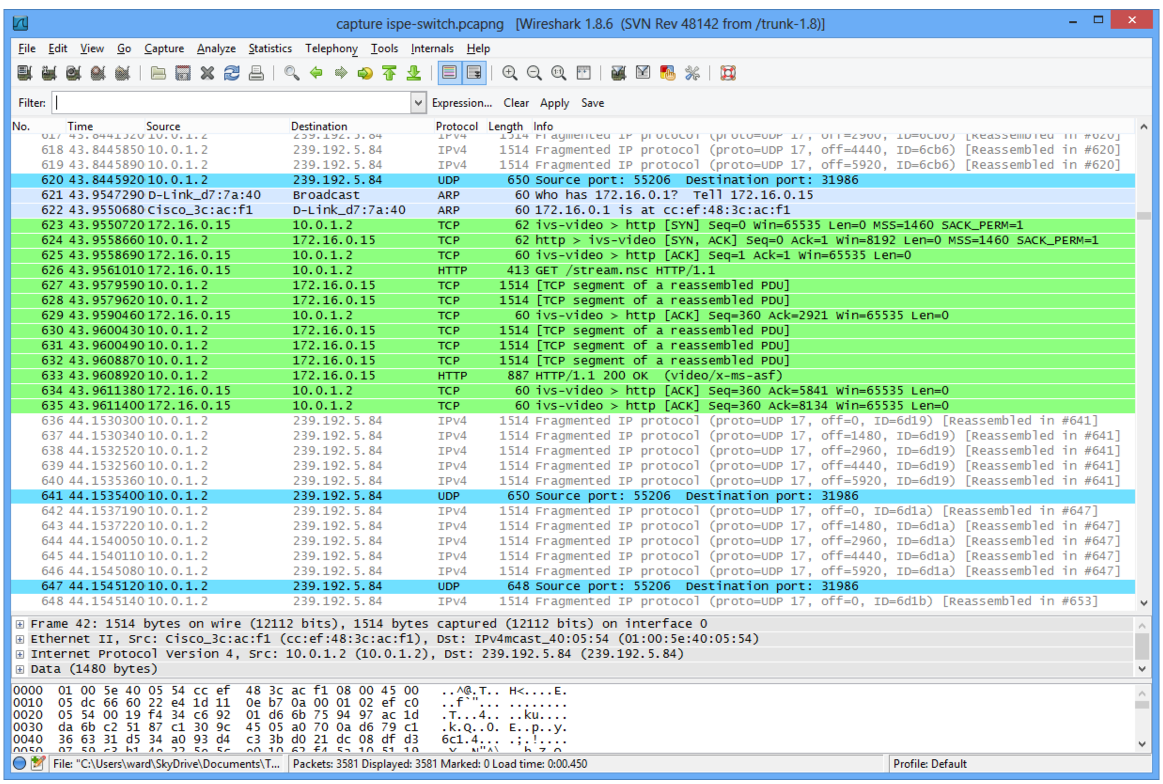


FIGURE 24. Dense mode, second client connects

By following up the path towards the server, packet tracing between the server and INTS, INTS and INTE, INTE and ISPE all reveal the same packet stream. More remarkable however is when tracing between INTE and ISPE when no computers request the multicast stream, the packets are still flowing. After tracing for over 3 minutes as shown in Figure 25 the packets still flow and no prune message has been send. Apparently something causes the routers not to send any prune message towards the source.

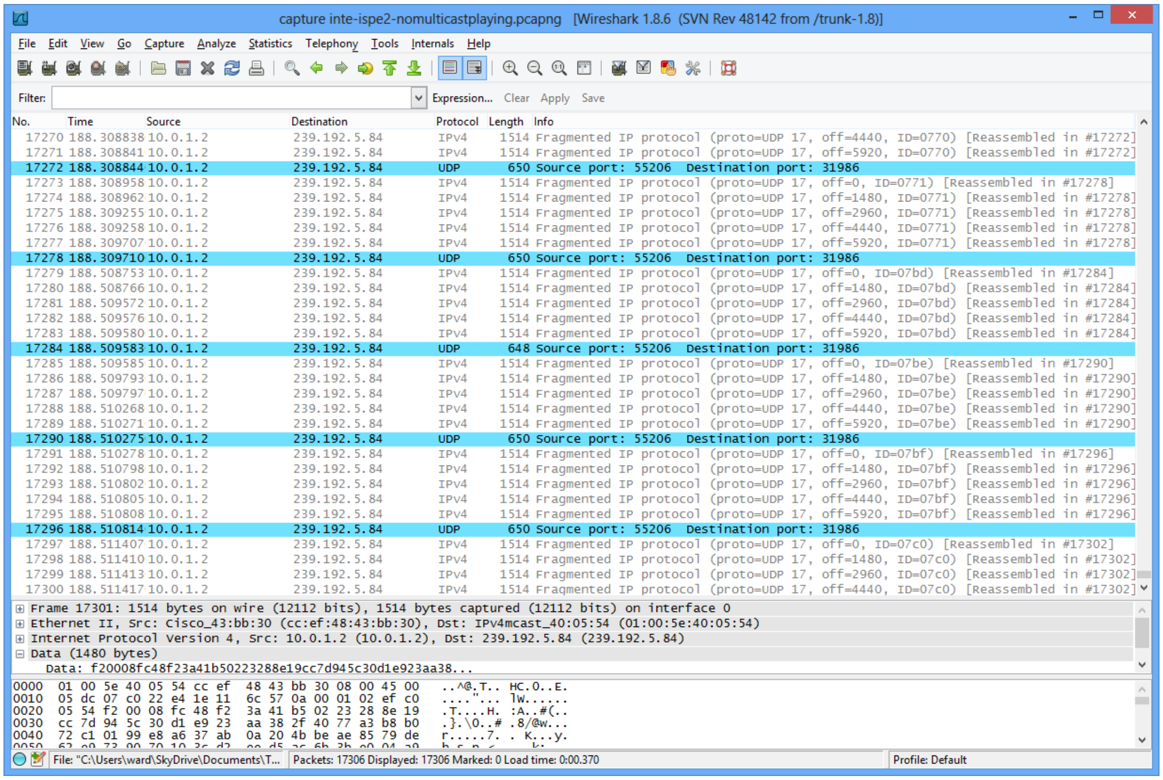


FIGURE 25. Dense mode, traffic keeps flowing

7.1.2 Multicast disabled ISP

When trying to open the multicast stream on a client on the multicast disabled ISP router ISPD, windows media player shows an error message as shown in Figure 26.

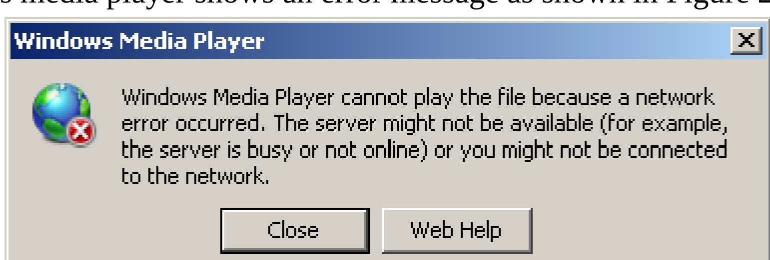


FIGURE 26. Windows Media Player error

It is clear that when the ISP router does not support multicast, it is not possible to get the data from a multicast stream.

When analyzing the traffic flow through the network, the packet sniffer reveals however a somewhat unnecessary packet flow between the internet routers INTS and INTD as shown in Figure 27. Logically there should not be any traffic as there is no multicast requests on the INTD side of the internet. The flow of packets is due to the characteristics of the dense mode and probably maintained by the redundancy link between INTD and INTE. In this scenario the dense mode behaves more like broadcast than as a multicast in the network. More investigation is needed to explain the behavior of this packet flow in the network.

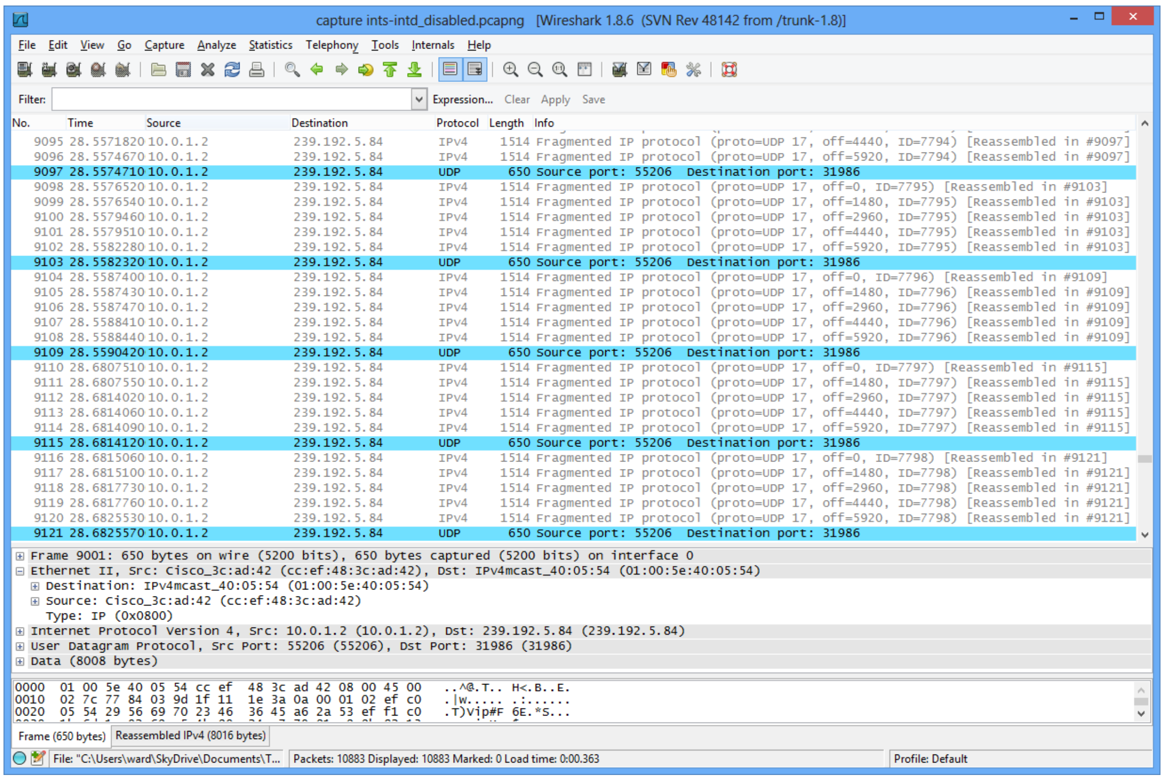


FIGURE 27. Dense mode, packet flow between INTS and INTD

7.2 Sparse mode

As it shows that dense mode is not really efficient and resembles a lot like broadcast traffic, it might be better to configure the network in sparse mode. As told in section 2.4 Multicast, in sparse mode there is no initial flow of traffic over the whole network as it is with dense mode.

To configure the routers in sparse mode all of the internet routers interfaces should be configured in sparse mode. This is done by first removing the dense mode line and then adding the sparse mode line:

```
INTS(config-if)# no ip pim dense-mode
INTS(config-if)# ip pim sparse-mode
```

The configuration of the ISP routers will be done later in section 7.2.2 Stub Multicast.

7.2.1 Rendezvous Point

When selecting the RP in an efficient way, the flow of traffic can be reduced to a minimum. The best location for a RP is as close to the source as possible. This is not necessary but it is more efficient as the initial multicast flows to the RP. In this scenario the internet router closest to the source, the server, is INTS.

The other routers need to know the address of the rendezvous point. The address of the loopback interface is therefore an excellent choice. In all three internet routers the following line must be added:

```
INTS(config)# ip pim rp-address 10.1.1.1
```

The INTS router will also be configured as bootstrap router. A bootstrap router is a router that will collect any RP candidate information if the manually configured RP selection fails. The creation of the bootstrap router is done by the following command:

```
INTS(config)# ip pim bsr-candidate Loopback0 0
```

The rendezvous point is now configured. Multicast traffic can flow over the internet section of the topology.

7.2.2 Stub Multicast

The ISP routers have not been configured with pim sparse mode. The reason is that for sparse mode all routers need to be in the same OSPF area. In this scenario the ISPs are configured to communicate with the internet through a default route and a static route, which is a closer resemblance to the real life situation than adding them to the same OSPF area. Therefore a stub multicast is needed.

With stub multicast routing a router will act as an IGMP proxy. In this scenario it will be the INTD and INTE which will act as a proxy. The ISPE router will forward all IGMP packets (like PIM messages) to the proxy.

First of all the routers ISPE and INTE cannot become PIM neighbors because ISPE will not fully take part in the multicast process, but only act as some kind of gateway. To border the PIM domain a standard access list and a filter on the ISP side of the internet router need to be configured. This can be done in both INTE and INTD, if later ISPD decides to support multicast traffic.

```
INTE(config)# access-list 1 deny 10.0.2.2
INTE(config)# interface gi0/0
INTE(config-if)# ip pim neighbor-filter 1
```

Next the ISP router needs to forward all multicast requests to the internet routers. This is done by configuring the client side of the router with an IGMP helper, the closest internet router. So in the GigabitEthernet0/1 interface of ISPE the next line needs to be added:

```
ISPE(config-if)# ip igmp helper-address 10.0.2.1
```

Note that the interfaces on ISPE are still configured with dense mode. On client side the ISP can choose the type itself, on the internet side it needs to be configured in dense mode so it will flood any multicast. Because the client side network in this topology is rather basic, dense mode works best in this case. Configuring sparse mode will require a new RP and since there are no other routers this will be useless.

The configuration of the network is now done and multicast should flow through all configured routers when requested. The end configuration of the routers can be found in Appendix 2.

7.2.3 Multicast enabled ISP

For the multicast enabled ISP side there should not be much of a difference with dense mode. However this time it is clear to see that there is no traffic until it is requested. In Figure 28 it is possible to see that the traffic starts flowing between the INTE and ISPE router after an IGMPv2 membership packet for group 239.192.5.84 (the multicast IP address configured on the server).

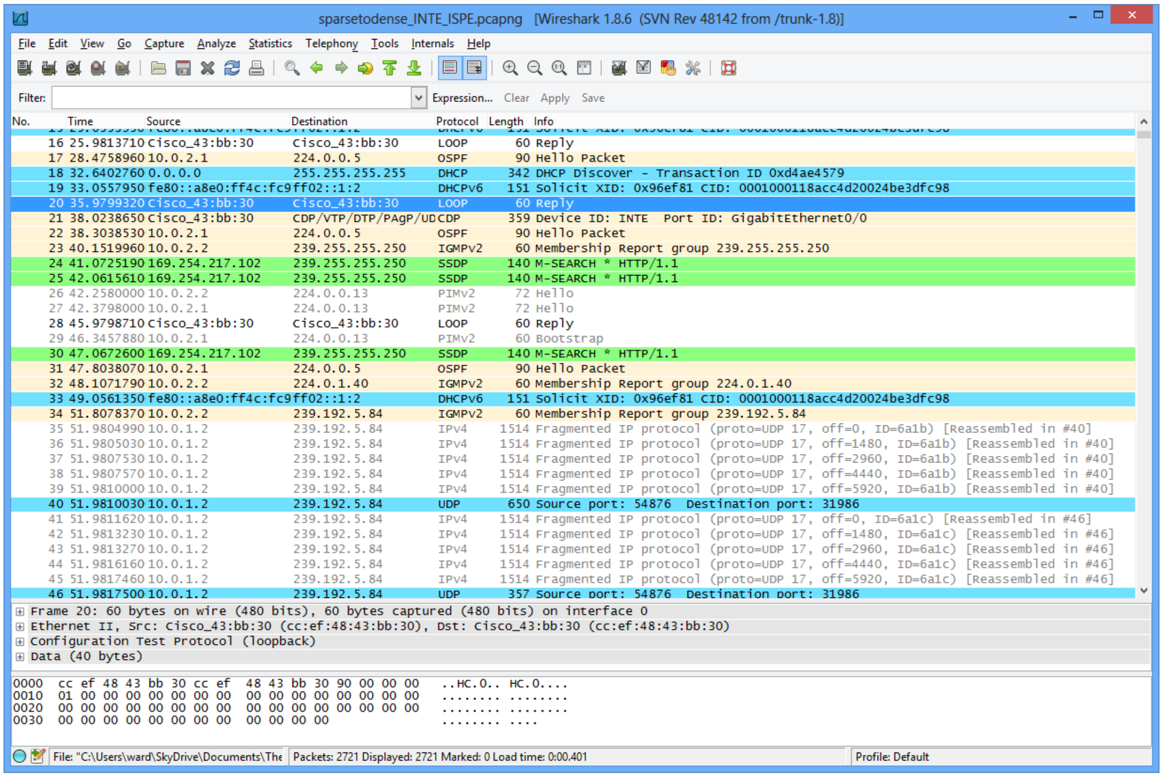


FIGURE 28. Sparse mode, IGMPv2 membership packet

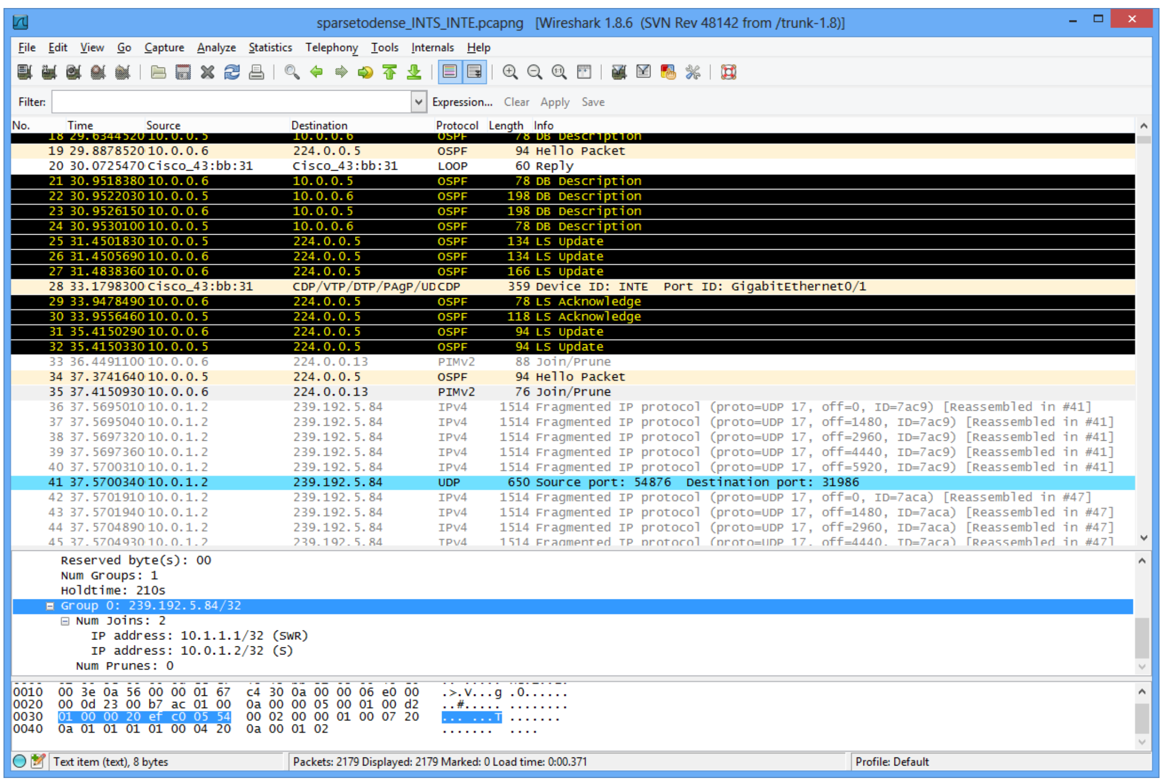


FIGURE 29. Sparse mode, PIMv2 Join/Prune packet

On the side of the internet, between routers INTE and INTS it is shown that this IGMP packet is translated into a PIMv2 join/prune packet at shown in Figure 29. It is also possible to see the multicast IP address in the packet.

7.2.4 Multicast disabled ISP

Now that the networking is working in sparse mode, there should not be any traffic flowing towards the ISP disabled router ISPD. This can be checked by investigating the traffic between INTS and INTD. In Figure 30 it is possible to see that there is no multicast traffic flowing originating from the live stream. This shows that sparse mode will only send multicast traffic upon request, as where dense mode starts by flooding the network.

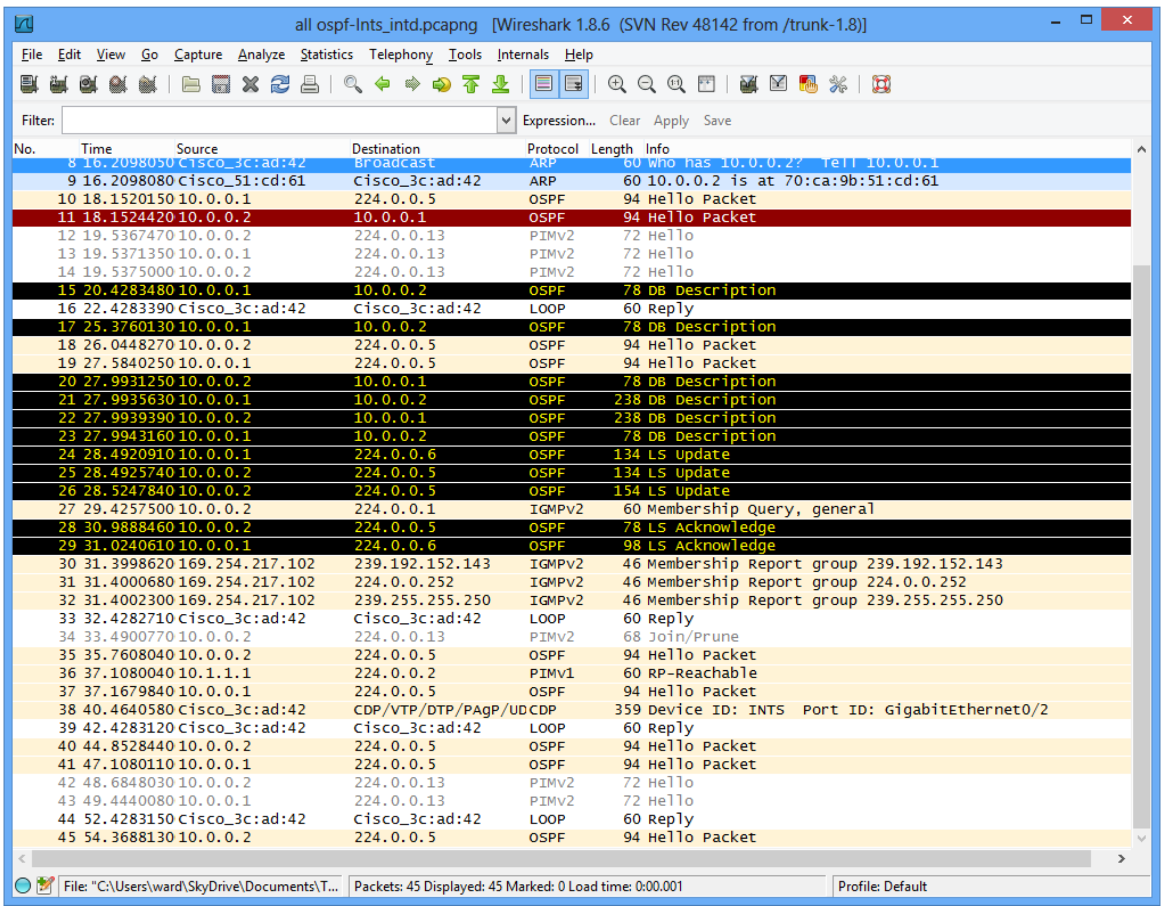


FIGURE 30. Sparse mode, no multicast traffic to INTD

7.3 Server side workaround

The multicast traffic flows as supposed to all computers behind the multicast enabled ISP. The network resources are lowered to a minimum and the server only has to send one stream. The content provider, in this case the manager of the server, will be happy that he only has to get a bandwidth connection which is just a little bit higher than the bandwidth of the stream he wants to send. Despite these cost efficiency he loses a lot of public who are not able to watch the stream because their ISP does not provide multicast traffic. This means he can lose a lot of income generated by commercials.

Assuming the lack of multicast support is the only limiting factor, a way around this problem is to also provide a way to connect to the server with a unicast connection when the initial multicast connection fails. This is called a unicast rollover, and is supported by the streaming component of Windows Server 2008 R2.

7.3.1 Providing unicast rollover

To provide unicast rollover on Windows Server 2008 R2, it is possible in two ways. The first way is checking the box of unicast rollover during the setup of the publishing point, the second way is by clicking the 'Allow new unicast connections' button in the Publishing points screen, as shown in Figure 31.

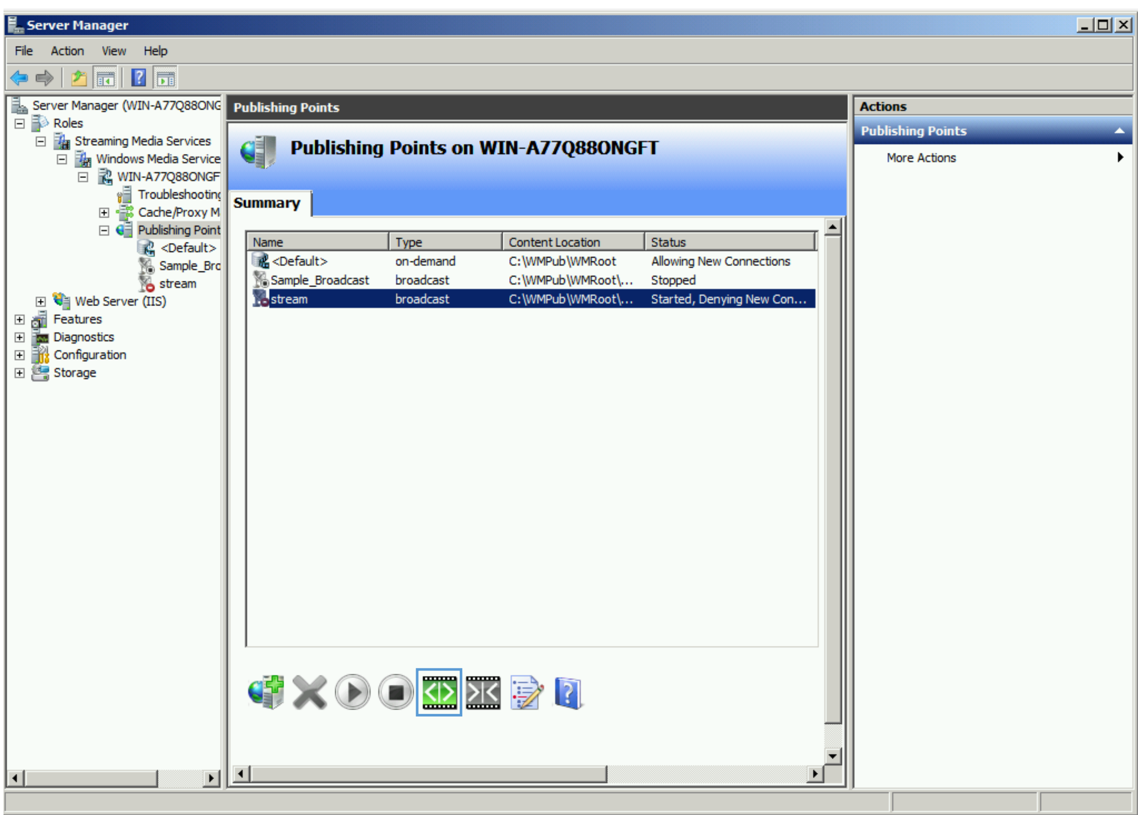


FIGURE 31. Allow new unicast connections

Because the announcement file contains the information to reach the server, there is no configuration needed on the client side. When the client connects to the server they will both automatically decide the most optimal way of connecting.

7.3.2 Unicast performance consequences

Now that the unicast rollover is provided, all clients that are not compatible with multicast will make a unicast connection. This has consequences for the performance of the server. The server will need to send a stream for every new unicast client. In Figure 32 it is possible to see how the connecting clients influence the use of bandwidth on the Ethernet adapter. At the starting point of measuring there were three computers connected with multicast, PC4, PC5 and PC6 behind the multicast enabled ISPE router. After an interval of about thirty seconds a computer, PC1, behind the multicast disabled ISPD router requested a unicast stream from the server. Also PC2 and PC3 were connected after another interval of each thirty seconds. In the end the unicast connected computers were disconnected one by one.

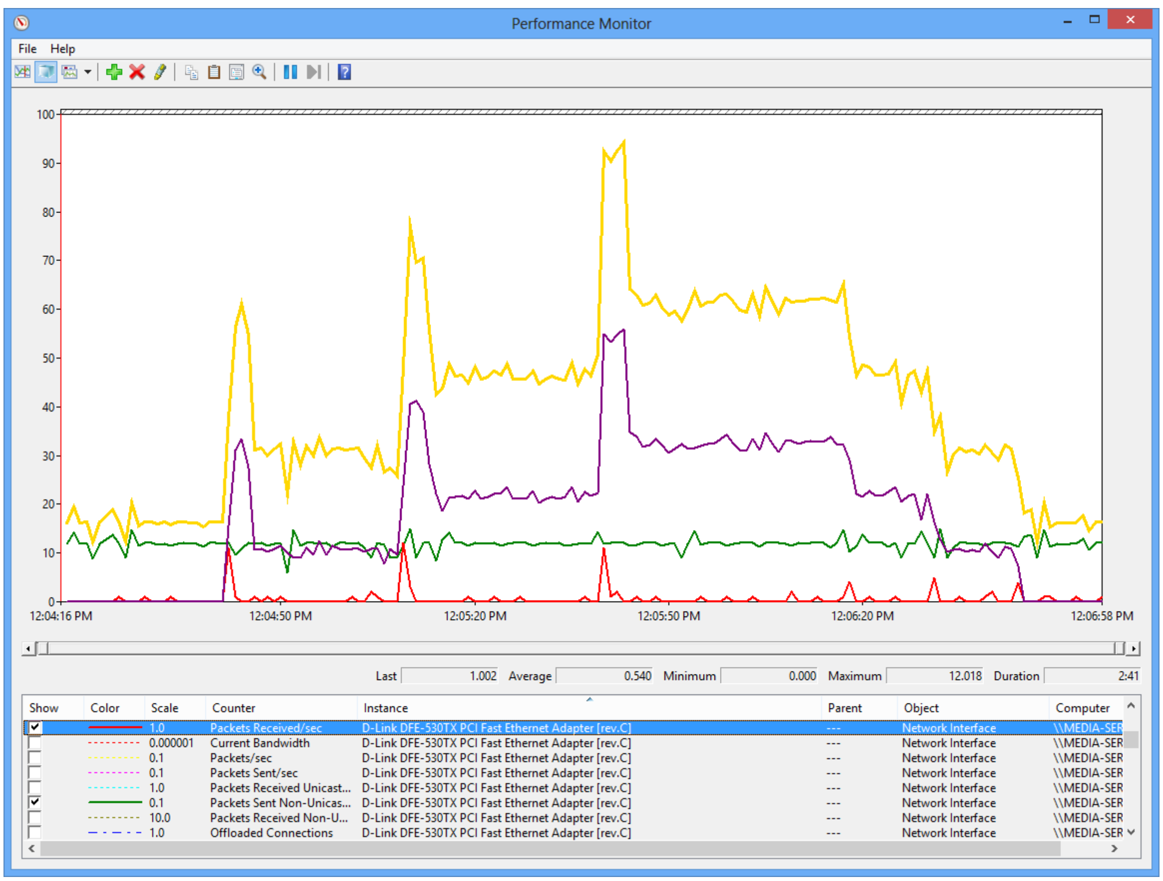


FIGURE 32. Server bandwidth statistics

On the graph it is clear to see how the overall bandwidth used (yellow line) rises for each new connected unicast client. The green line shows how the multicast traffic stays the same. Note that the bandwidth already almost doubles when connecting the first unicast client. This proofs that the multicast only needs the amount of one unicast client to serve three multicast clients. The purple line shows the amount of unicast traffic send by the server. The peak in the beginning of each connection is due to the buffering process that take place with the unicast connection.

In multicast there is also buffering but this is only on client side where the client wait for a predefined time (e.g. five seconds) of multicast to be stored in the local buffer. The last line that is visible on the graph is the red line which represents incoming unicast packets. These peaks are the unicast requests by the clients.

If this graph shows the effective throughput of data it should be visible on the wire between the server and the INTS router as well. It is possible to see this is indeed true. In Figure 33 it is possible to see how the second client, PC2 connects when PC3 is already connected. In Figure 34 it is possible to see the multicast traffic as UDP and fragmented IPv4 packets with the multicast group address as destination address and the unicast traffic as RTP traffic with the clients IP address as destination address.

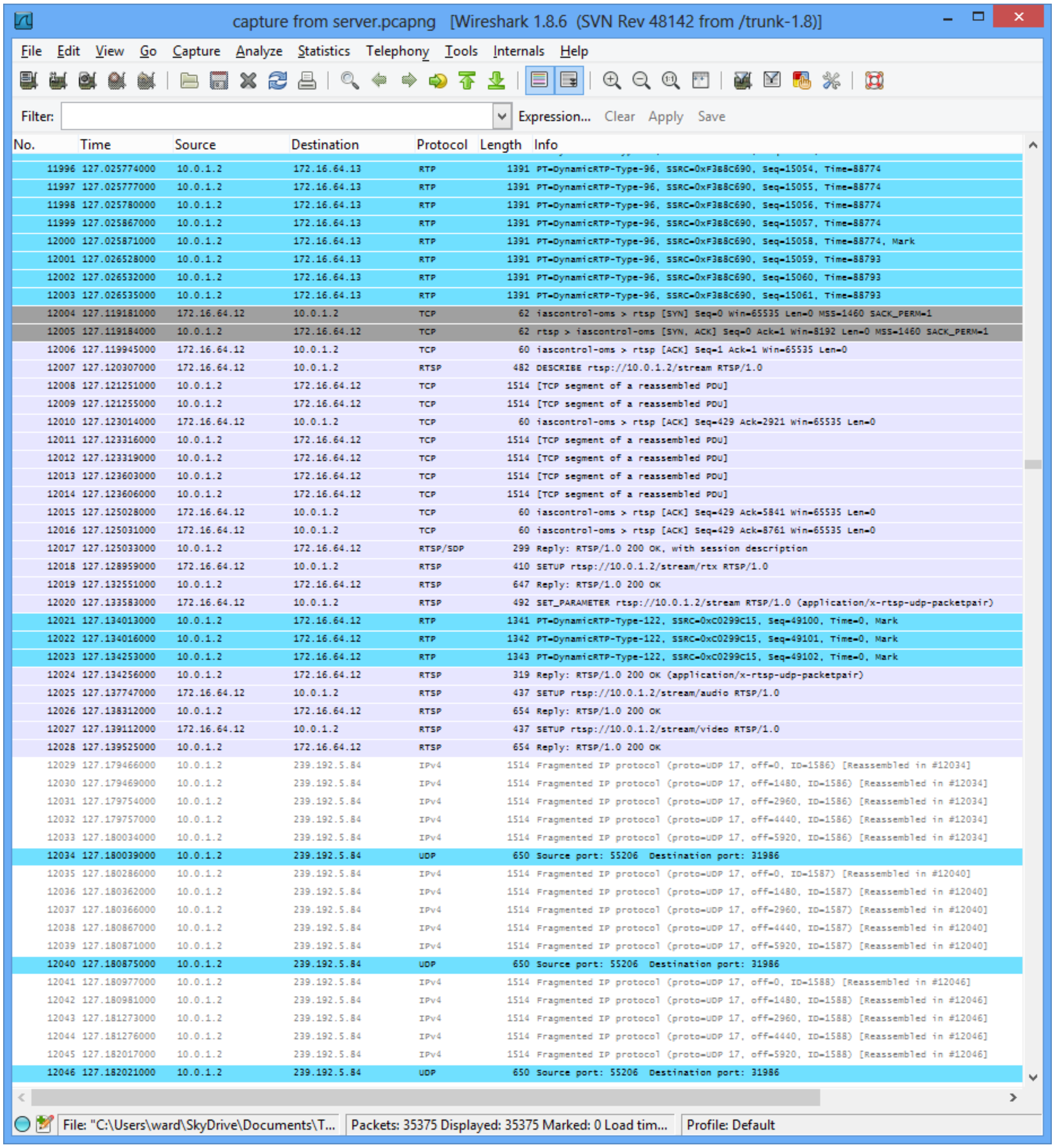


FIGURE 33. PC2 connects for a unicast stream

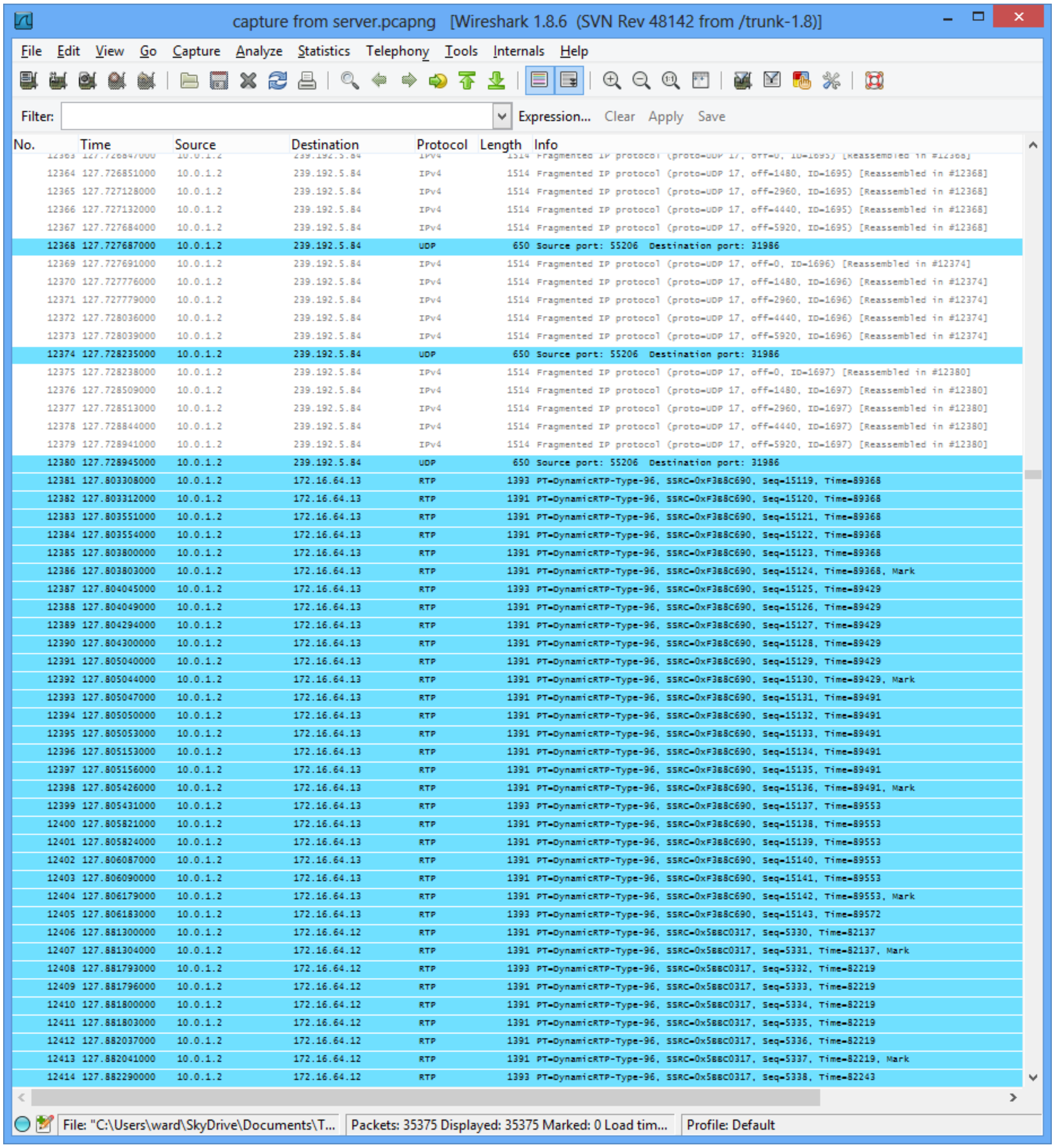


FIGURE 34. Multicast and unicast traffic on the wire

8 CONCLUSION

To sum up what was done in this thesis is the comparison and small history of unicast and multicast, the selection of the most suitable server and media format for the simulation model and the effects of the difference between dense and sparse mode multicast.

It was clear that multicast provides a very efficient way of broadcasting live streams over a network. The optimal use of bandwidth compared to multiple unicast connections shown in Chapter 7.3.2 gave a clear view why it is a good choice. Also the efficiency of using only the needed paths with sparse mode compared to broadcasting or dense mode gave multicast the opportunity to not flood the whole network. The question ‘why use dense mode?’ is answered by the fact that some networks, for example where all clients need the multicast, gain profit from it. In the case of this scenario it was for example for all clients behind the ISPE router. The internet section however was more suited by the sparse mode configuration.

So why is there for the moment hardly any support for multicast on today’s internet? Why do internet service providers and also content providers not provide multicast? The answer lies as stated in section 4.2 Multicast support in money and a business model. As stated by professor Goossens they underestimated the powerful need of a decent business model.

Then why does multicast still exist? Multicast is still useful for local network, or bigger corporate networks. Multicast can be used in a network where many but not all people need to have access to the same streaming source. By using multicast for example at a corporate presentation by the CEO, only those who want or need receive the stream and the network load is reduced to a minimum. Also IPTV uses multicast because their business model lies closer to general television broadcast as it lies to streaming internet video.

But next to corporate networks there are some workarounds. In this thesis the use of a multicast rollover to unicast was one option to provide a connection to the stream for people without multicast connection. This might be not the most efficient way. In Chapter 4.3 there were two projects that used a type of tunneling or masking to make multicast possible over non supported networks. Further research for a work around on client side or internet side can give more opportunities than the given server side rollover solution in this thesis.

In the future, with the rise of IPv6 things might change as IPv6 does not have broadcast anymore but only built in support for multicast. This thesis was based purely on IPv4 because today it is still the standard and the equipment used was not really suitable for the use of IPv6. If this thesis was based on IPv6 the outcome might have been totally different. Though all hope must not rely only with the come of IPv6. In the UK the BBC has contracts with certain providers to enable multicast for the BBC broadcasts. Customers of those providers can access the BBC’s multicast streams and watch the broadcasts on their computer. This evolution might be an idea for other national television providers in other countries, certainly with the current rise of IPTV solutions.

As end conclusion it must be said that multicast is an efficient way of working, with a lot of advantages, but for now with the current factors there is only place for streaming media over unicast on the internet.

BIBLIOGRAPHY

Abley, J., 2006. Operation of Anycast Services. WWW-document. http://tools.ietf.org/pdf/rfc4786.pdf. Referred 15 March 2013.

Adobe Systems Incorporated, 2010. Adobe Flash Video File Format Specification. WWW-document. http://download.macromedia.com/f4v/video_file_format_spec_v10_1.pdf. Referred 1 April 2013.

Albanna, Z. & al., 2001. IANA Guidelines for IPv4 Multicast Address Assignments. WWW-document. http://tools.ietf.org/html/rfc3171. Referred 26 April 2013.

Apple Inc, 2005. Mac OS X Server. WWW-document. http://www.apple.com/quicktime/pdf/QT_Streaming_Server_v10.4.pdf. Referred 12 April 2013.

Apple Inc, 2012. QuickTime Streaming Server: General Information. WWW-document. http://support.apple.com/kb/ta25288?viewlocale=en_us. Referred 12 April 2013a.

Apple Inc, 2013. OS X Server technical specifications. WWW-document. http://www.apple.com/osx/server/specs/. Referred 12 April 2013.

Belgacom SA, 2012. Multicast Services. WWW-document. http://www.belgacomwholesale.be/wholesale/gallery/content/documents/multicast/Multicast_Welcome_Pack_v1.0.pdf. Referred 12 April 2013.

Bok, C. J., 2002. Overlay Network, Overlay Model. WWW-document. http://ktword.co.kr/abbr_view.php?m_temp1=3481. Referred 18 April 2013.

Bristol, D., 2012. Windows Media Services not supported on Windows Server 2012. WWW-document. http://blogs.msdn.com/b/randomnumber/archive/2012/11/14/windows-media-services-not-supported-on-windows-server-2012.aspx. Referred 12 April 2013.

Broadband Media, 2013. What is a MAC Address?. WWW-document. http://www.iplocation.net/tools/mac-address.php. Referred 14 March 2013.

Conjecture Corporation, 2013. What Are the Advantages of FLV Format?. WWW-document. http://www.wisegeek.com/what-are-the-advantages-of-flv-format.htm. Referred 1 April 2013.

Cotton, M. & al., 2010. IANA Guidelines for IPv4 Multicast Address Assignments. WWW-document. http://tools.ietf.org/html/rfc5771#section-224.5.0.0. Referred 27 April 2013.

Fairhurst, G., 2009. Unicast, Broadcast, and Multicast. WWW-document. http://www.erg.abdn.ac.uk/~gorry/eg3567/intro-pages/uni-b-mcast.html. Referred 14 March 2013.

Goff, D., 2003. Fiber Optic Video Transmission. 1st ed. Woburn: Focal Press.

Goossens, M., 2013. Email discussion on 15.2.2013 s.l.:Vrije Universiteit Brussel.

Goossens, M., Liefooghe, P. & Swinnen, A., 2006. The CastGate project. WWW-document. http://www.nordu.net/conference2006/presentations/We11_NORDUnet2006.pdf. Referred 12 April 2013.

Grula, L., 2010. An Overview of Internet Video File Formats - Video Containers. WWW-document. http://www.reelseo.com/basics-web-video-file-formats-video-containers/. Referred 30 March 2013.

Guan, 2011. CDN with TCP anycaste lite. WWW-document. http://guan.dk/tcp-anycast-lite-cdn. Referred 2013 March 15.

Imielinski, T. & Navas, J., 1996. GPS-Based Addressing and Routing. WWW-document. http://tools.ietf.org/html/rfc2009. Referred 15 March 2013.

Internet2, 2004. Internet2 Multicast Workshop. WWW-document. http://andrew.triumf.ca/AG/multicast/internet2-multicast-workshop-may-2004-1-overview.pdf. Referred 14 April 2013.

Klicktv, 2013a. IPTV Multicasting Explained. WWW-document. http://www.klicktv.co.uk/tv-distribution-solutions/iptv/multicasting.html. Referred 14 March 2013.

Klicktv, 2013b. All about IPTV. WWW-document. http://www.klicktv.co.uk/iptv/all-about-iptv.html. Referred 12 April 2013.

Kumar, S., 2006. Why multicast is irrelevant to the Internet. WWW-document. http://www.arl.wustl.edu/~jst/reInventTheNet/?p=161. Referred 12 April 2013.

Merrill, D. C., 2004. The Linux Kernel. WWW-document. http://www.tldp.org/FAQ/Linux-FAQ/kernel.html. Referred 12 April 2013.

Microsoft, 2003. Differences Between Multicast and Unicast. WWW-document. http://support.microsoft.com/kb/291786. Referred 14 March 2013.

Microsoft, 2007. Windows Server 2008 System Requirements. WWW-document. http://msdn.microsoft.com/en-us/windowsserver/cc196364.aspx. Referred 12 April 2013.

Microsoft, 2010. Release Notes for Windows Media Services 2008. WWW-document. http://technet.microsoft.com/library/cc771560.aspx#WMS_010. Referred 13 April 2013.

Microsoft, 2012a. About the Windows Media Codecs. WWW-document. http://msdn.microsoft.com/en-us/library/gg153556(v=vs.85).aspx. Referred 12 April 2013.

Microsoft, 2012b. Installing Windows Server 2012. WWW-document. http://technet.microsoft.com/en-us/library/jj134246.aspx. Referred 12 April 2013.

Microsoft, 2013. Supported file types. WWW-document. http://technet.microsoft.com/en-us/library/cc731194.aspx. Referred 12 April 2013.

Mitchell, B., 2013. Switch - Definition of Network Switch. WWW-document. http://compnetworking.about.com/od/hardwarenetworkgear/g/bldef_switch.htm. Referred 14 March 2013.

Mogul, J., 1984. BROADCASTING INTERNET DATAGRAMS IN THE PRESENCE OF SUBNETS. WWW-document. http://tools.ietf.org/html/rfc922. Referred 26 April 2013.

Nelson, D., 2007. Firewall Information for Windows Media Services 9 Series. WWW-document. http://www.microsoft.com/windows/windowsmedia/forpros/serve/firewall.aspx. Referred 12 April 2013.

Network Sorcery Inc, 2012. UDP, User Datagram Protocol. WWW-document. http://www.networksorcery.com/enp/protocol/udp.htm. Referred 30 March 2013.

Pennington, M., 2011. networking - TCP vs UDP on video stream - Stack Overflow. WWW-document. http://stackoverflow.com/questions/6187456/tcp-vs-udp-on-video-stream. Referred 30 March 2013.

Quick, D., 2011. Adobe finally delivers Flash video to iOS devices. WWW-document. http://www.gizmag.com/adobe-flash-ios/19790/. Referred 17 March 2013.

Refsnes Data, 2013. HTML5 Video. WWW-document. http://www.w3schools.com/html/html5_video.asp. Referred 30 March 2013.

Reinhardt, R., 2007. Protocols: HTTP vs. RTMP > Beginner's Guide to Distributing Flash Video. WWW-document. http://www.adobepress.com/articles/article.asp?p=1014968&seqNum=2. Referred 1 April 2013.

Tanenbaum, A. S. & Wetherall, D. J., 2011. Computer Networks. 5th ed. Boston: Pearson.

The WebM Project, 2012a. The WebM Project | About WebM. WWW-document. http://www.webmproject.org/about/. Referred 30 March 2013.

The WebM Project, 2012b. The WebM Project | FAQ. WWW-document. http://www.webmproject.org/about/faq/. Referred 12 April 2013.

Topic, M., 2002. Streaming Media Demystified. New York: McGraw-Hill.

VideoLAN, 2013. Streaming features list. WWW-document. http://www.videolan.org/streaming-features.html. Referred 12 April 2013.

Welcher, P. J., 2001a. PIM Dense Mode. WWW-document. http://www.netcraftsmen.net/resources/archived-articles/376-pim-dense-mode.html. Referred 30 March 2013.

Welcher, P. J., 2001b. PIM Sparse Mode. WWW-document. http://www.netcraftsmen.net/resources/archived-articles/424-pim-sparse-mode.html. Referred 3 March 2013.

Westin, P., 2013. RTP Payload Format for VP8 Video. WWW-document. http://tools.ietf.org/html/draft-ietf-payload-vp8-08. Referred 12 April 2013.

What Is My IP Address, 2013. What is an IP Address?. WWW-document. http://whatismyipaddress.com/ip-address. Referred 14 March 2013.

APPENDIX 1(1).

```
! ROUTER INTS INITIAL CONFIGURATION
! Last configuration change at 09:45:26 UTC Tue Feb 26 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname INTS
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720T5
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 10.0.0.5 255.255.255.252
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 ip address 10.0.0.1 255.255.255.252
 duplex auto
 speed auto
```

APPENDIX 1(2).

```
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
End


! ROUTER INTE INITIAL CONFIGURATION
! Last configuration change at 05:27:14 UTC Sun Mar 8 2009
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

APPENDIX 1(3).

```
hostname INTE
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720TH
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 10.0.0.6 255.255.255.252
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 ip address 10.0.0.9 255.255.255.252
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
```

APPENDIX 1(4).

```
no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
End


! ROUTER INTD INITIAL CONFIGURATION
! Last configuration change at 10:01:26 UTC Tue Feb 26 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname INTD
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
```

APPENDIX 1(5).

```
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ154420V0
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.3.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 10.0.0.2 255.255.255.252
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 ip address 10.0.0.10 255.255.255.252
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
```

APPENDIX 1(6).

```
no ip http secure-server
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
End


! ROUTER ISPE INITIAL CONFIGURATION
! Last configuration change at 08:52:31 UTC Tue Feb 26 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ISPE
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip cef
!
```

APPENDIX 1(7).

```
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720T8
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.2.2 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.0.1 255.255.192.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/1/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 2000000
```

APPENDIX 1(8).

```
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
End


! ROUTER ISPD INITIAL CONFIGURATION
! Last configuration change at 10:16:57 UTC Tue Feb 26 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ISPD
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
```

APPENDIX 1(9).

```
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720TD
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.3.2 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.64.1 255.255.192.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
```

APPENDIX 1(10).

```
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
End
```

APPENDIX 2(1).

```
! ROUTER INTS END CONFIGURATION
! Last configuration change at 08:50:22 UTC Fri Mar 1 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname INTS
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
ip multicast-routing
!
no ip domain lookup
ip host INTS 10.1.1.1
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720T5
!
redundancy
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
 ip pim sparse-mode
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.1.1 255.255.255.0
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
```

APPENDIX 2(2).

```
 ip address 10.0.0.5 255.255.255.252
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 ip address 10.0.0.1 255.255.255.252
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
router ospf 1
 network 10.0.0.0 0.0.0.3 area 0
 network 10.0.0.4 0.0.0.3 area 0
 network 10.0.1.0 0.0.0.255 area 0
 network 10.1.1.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
ip pim rp-address 10.1.1.1
ip pim bsr-candidate Loopback0 0
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
```

APPENDIX 2(3).

```
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end


! ROUTER INTE END CONFIGURATION
! Last configuration change at 05:30:02 UTC Wed Mar 11 2009
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname INTE
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
ip multicast-routing
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720TH
!
redundancy
!
interface Loopback0
 ip address 10.1.2.2 255.255.255.0
!
interface Embedded-Service-Engine0/0
```

APPENDIX 2(4).

```
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.2.1 255.255.255.0
 ip pim neighbor-filter 1
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 10.0.0.6 255.255.255.252
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 ip address 10.0.0.9 255.255.255.252
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
router ospf 1
 redistribute static subnets
 network 10.0.0.4 0.0.0.3 area 0
 network 10.0.0.8 0.0.0.3 area 0
 network 10.0.2.0 0.0.0.255 area 0
 network 10.1.2.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.63.255 area 0
!
ip forward-protocol nd
!
ip pim rp-address 10.1.1.1
no ip http server
no ip http secure-server
```

APPENDIX 2(5).

```
!
ip route 172.16.0.0 255.255.192.0 10.0.2.2
!
access-list 1 deny   10.0.2.2
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end


! ROUTER INTD END CONFIGURATION
! Last configuration change at 11:41:59 UTC Fri Mar 1 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname INTD
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
```

APPENDIX 2(6).

```
ip multicast-routing
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ154420V0
!
redundancy
!
interface Loopback0
 ip address 10.1.3.3 255.255.255.0
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.3.1 255.255.255.0
 ip pim neighbor-filter 1
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 10.0.0.2 255.255.255.252
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 ip address 10.0.0.10 255.255.255.252
 ip pim sparse-mode
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
```

APPENDIX 2(7).

```
 clock rate 2000000
!
router ospf 1
 redistribute static subnets
 network 10.0.0.0 0.0.0.3 area 0
 network 10.0.0.8 0.0.0.3 area 0
 network 10.0.3.0 0.0.0.255 area 0
 network 10.1.3.0 0.0.0.255 area 0
 network 172.16.64.0 0.0.63.255 area 0
!
ip forward-protocol nd
!
ip pim rp-address 10.1.1.1
no ip http server
no ip http secure-server
!
ip route 172.16.64.0 255.255.192.0 10.0.3.2
!
access-list 1 deny   10.0.3.2
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end


! ROUTER ISPE END CONFIGURATION
! Last configuration change at 09:44:20 UTC Fri Mar 1 2013
version 15.2
```

APPENDIX 2(8).

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ISPE
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
ip multicast-routing
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720T8
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.2.2 255.255.255.0
 ip pim dense-mode
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.0.1 255.255.192.0
 ip pim dense-mode
 ip igmp helper-address 10.0.2.1
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 no ip address
```

APPENDIX 2(9).

```
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/1/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
login
```

APPENDIX 2(10).

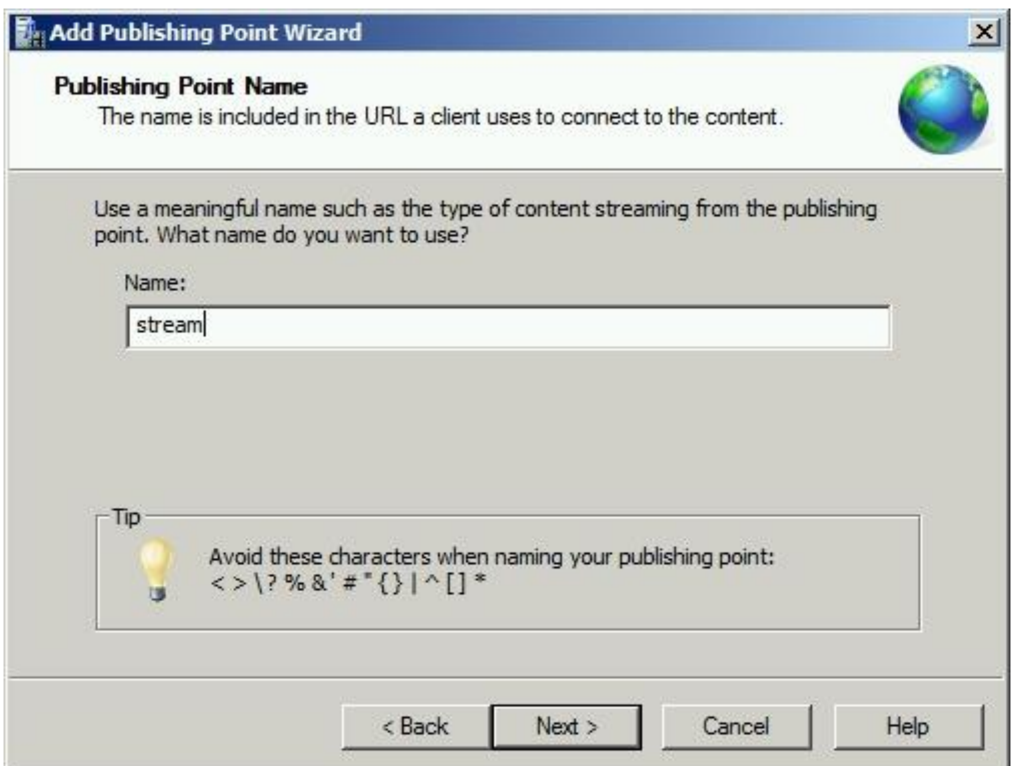
```
transport input all
!
scheduler allocate 20000 1000
!
end


! ROUTER ISPD END CONFIGURATION
! Last configuration change at 11:42:57 UTC Fri Mar 1 2013
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ISPD
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ipv6 cef
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2911/K9 sn FCZ153720TD
!
redundancy
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.0.3.2 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
ip address 172.16.64.1 255.255.192.0
```

APPENDIX 2(11).

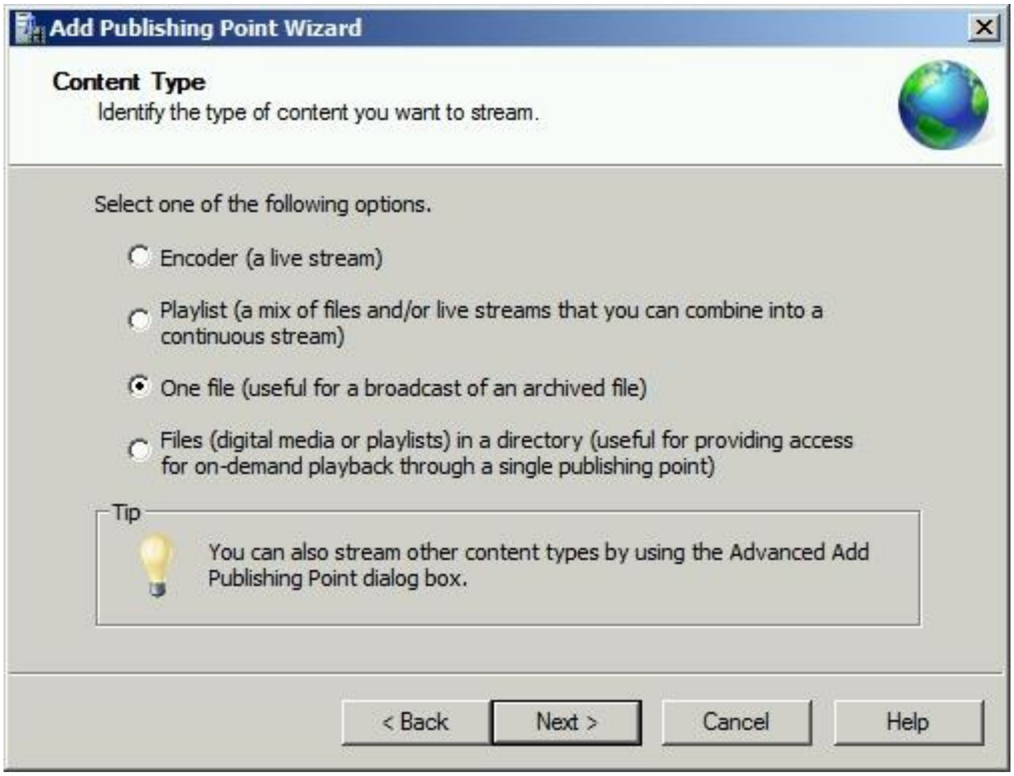
```
duplex auto
 speed auto
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

APPENDIX 3(1).


Add Publishing Point Wizard
Welcome to the Add Publishing Point Wizard
This wizard helps you add a new publishing point, including:
- Selecting a publishing point type and name
- Identifying the content to stream
- Enabling logging for the publishing point
Do not show this Welcome page again
To continue, click Next.
< Back
Next >
Cancel
Help

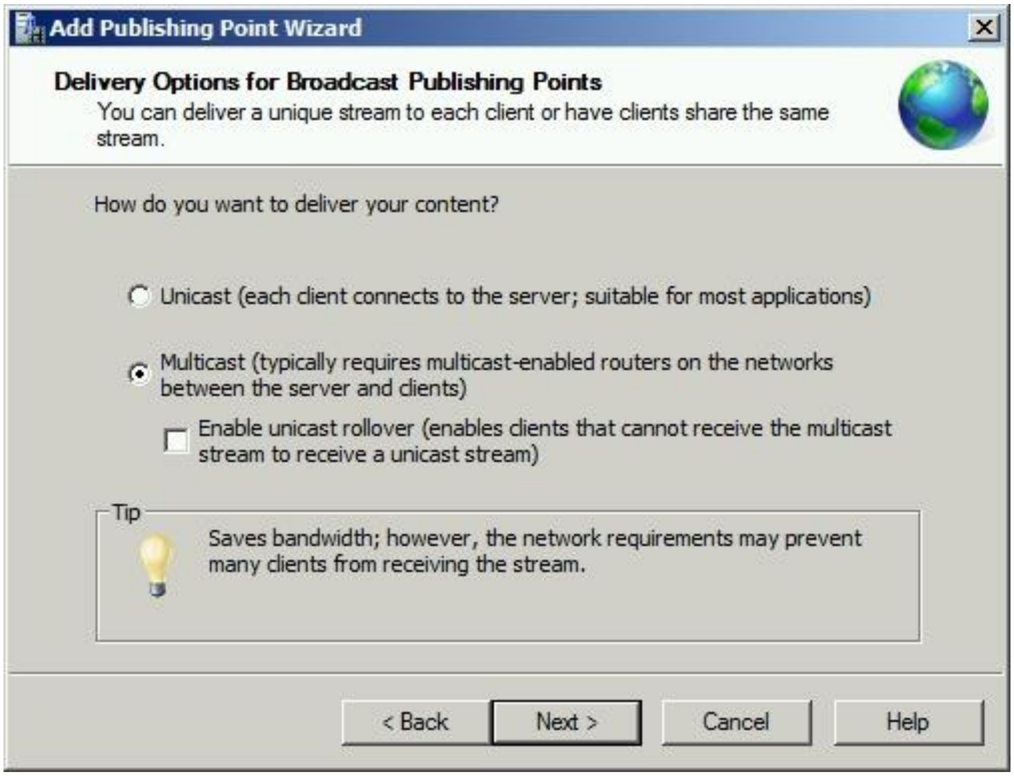

Add Publishing Point Wizard
Publishing Point Name
The name is included in the URL a client uses to connect to the content.
Use a meaningful name such as the type of content streaming from the publishing point. What name do you want to use?
Name:
stream
Tip
Avoid these characters when naming your publishing point:
< > \ ? % & ' # " { } | ^ [ ] *
< Back
Next >
Cancel
Help

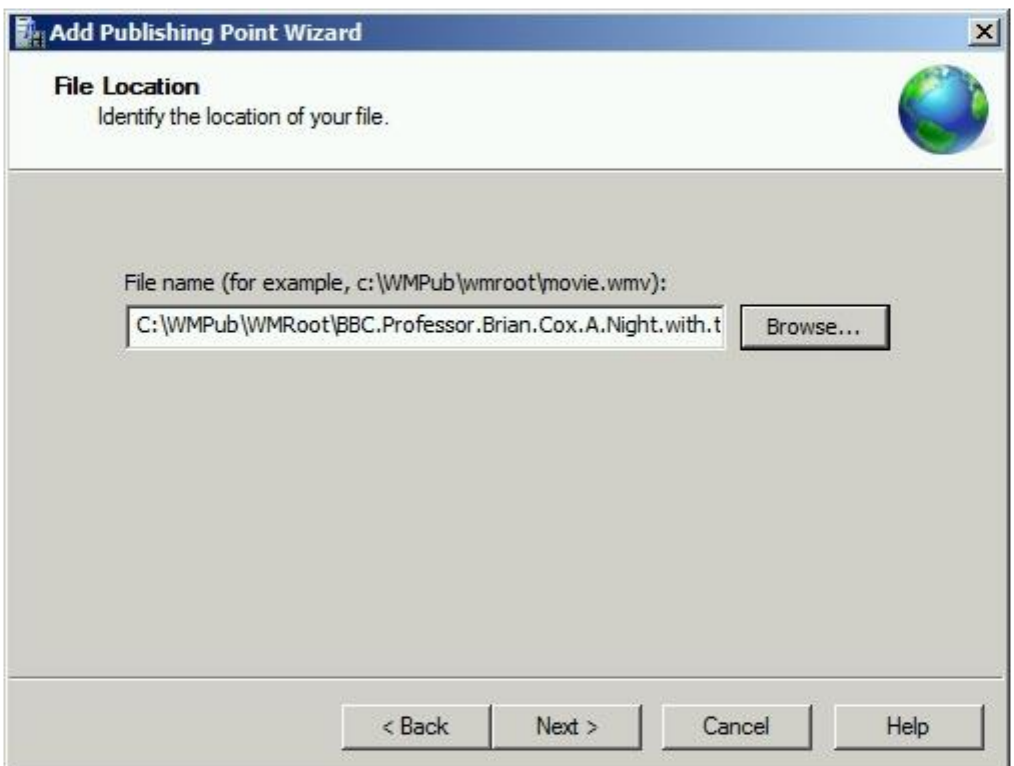
APPENDIX 3(2).


Add Publishing Point Wizard
Content Type
Identify the type of content you want to stream.
Select one of the following options.
Encoder (a live stream)
Playlist (a mix of files and/or live streams that you can combine into a continuous stream)
One file (useful for a broadcast of an archived file)
Files (digital media or playlists) in a directory (useful for providing access for on-demand playback through a single publishing point)
Tip
You can also stream other content types by using the Advanced Add Publishing Point dialog box.
< Back
Next >
Cancel
Help

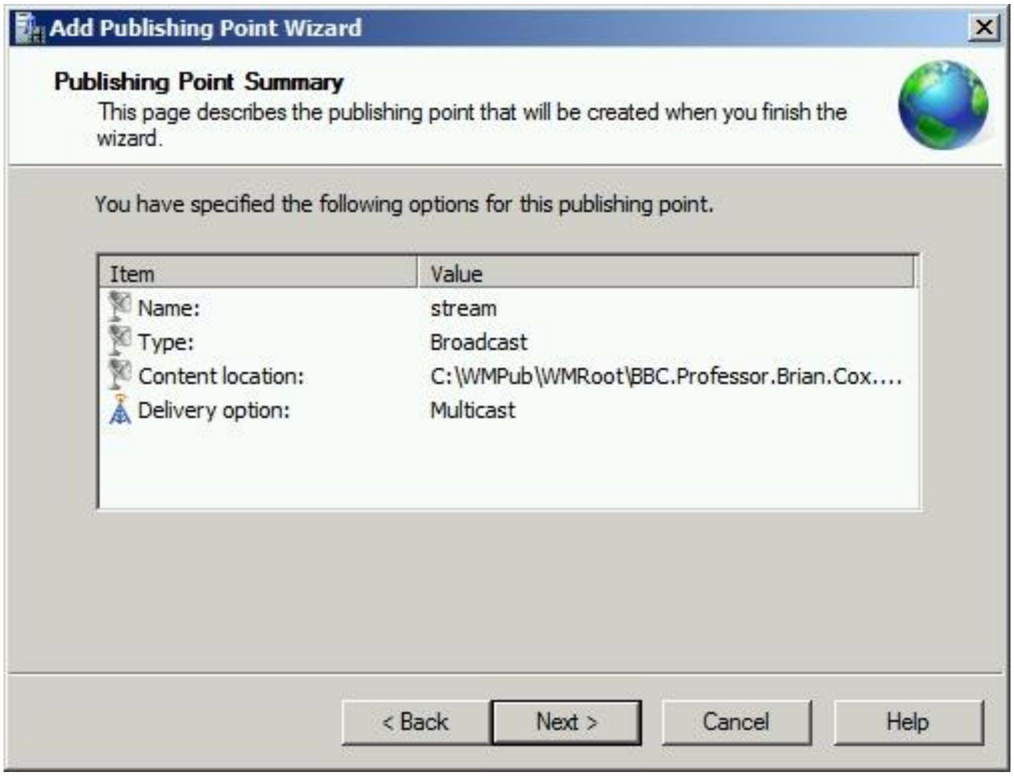

Add Publishing Point Wizard
Publishing Point Type
Publishing points organize and distribute content according to the playback scenario you want to create.
What playback scenario do you want to create?
Broadcast publishing point
Clients share the playback experience; use to create a scenario that is similar to viewing a television program. Use a broadcast publishing point to deliver a stream from an encoder.
On-demand publishing point
Use to create a scenario in which each client can control (for example, fast-forward) the stream.
< Back
Next >
Cancel
Help

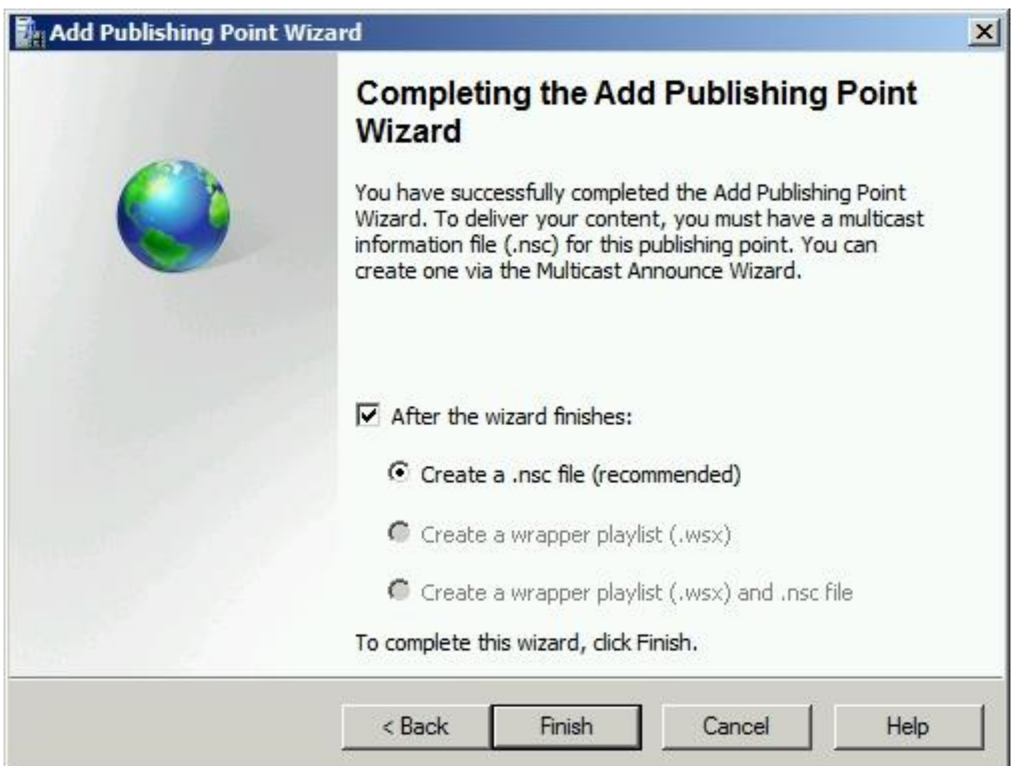
APPENDIX 3(3).


Add Publishing Point Wizard
Delivery Options for Broadcast Publishing Points
You can deliver a unique stream to each client or have clients share the same stream.
How do you want to deliver your content?
Unicast (each client connects to the server; suitable for most applications)
Multicast (typically requires multicast-enabled routers on the networks between the server and clients)
Enable unicast rollover (enables clients that cannot receive the multicast stream to receive a unicast stream)
Tip
Saves bandwidth; however, the network requirements may prevent many clients from receiving the stream.
< Back
Next >
Cancel
Help


Add Publishing Point Wizard
File Location
Identify the location of your file.
File name (for example, c:\WMPub\wmroot\movie.wmv):
C:\WMPub\WMRoot\BBC.Professor.Brian.Cox.A.Night.with.t
Browse...
< Back
Next >
Cancel
Help

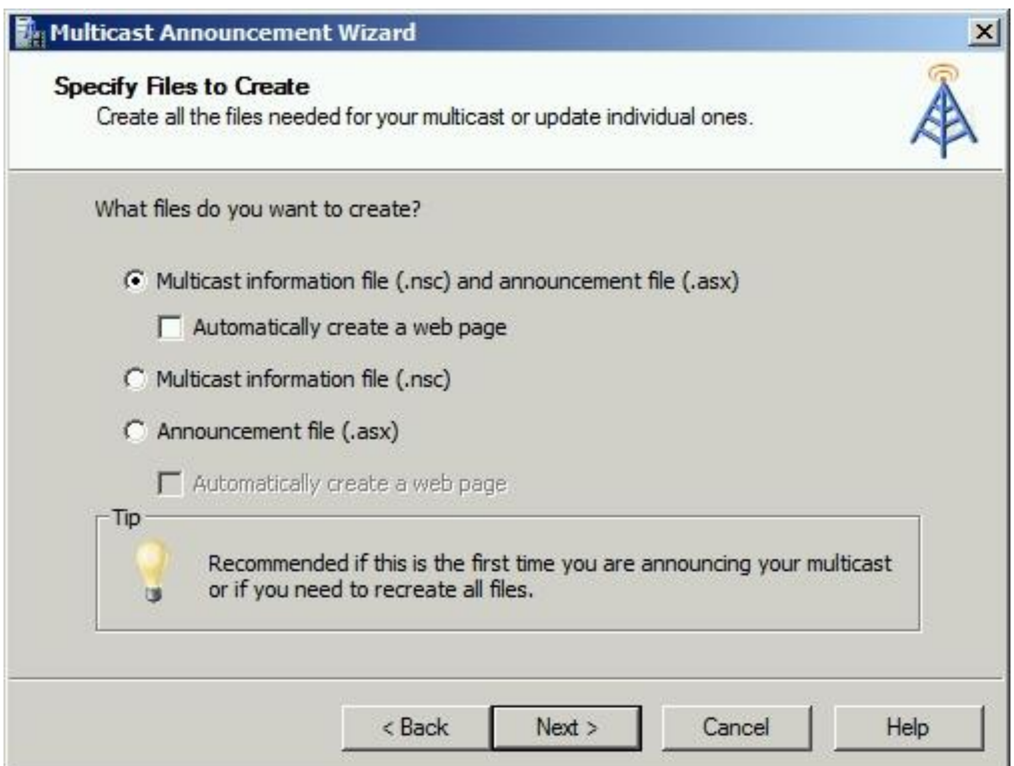
APPENDIX 3(4).


Add Publishing Point Wizard
Publishing Point Summary
This page describes the publishing point that will be created when you finish the wizard.
You have specified the following options for this publishing point.
Item
Value
Name:
stream
Type:
Broadcast
Content location:
C:\WMPub\WMRoot\BBC.Professor.Brian.Cox....
Delivery option:
Multicast
< Back
Next >
Cancel
Help

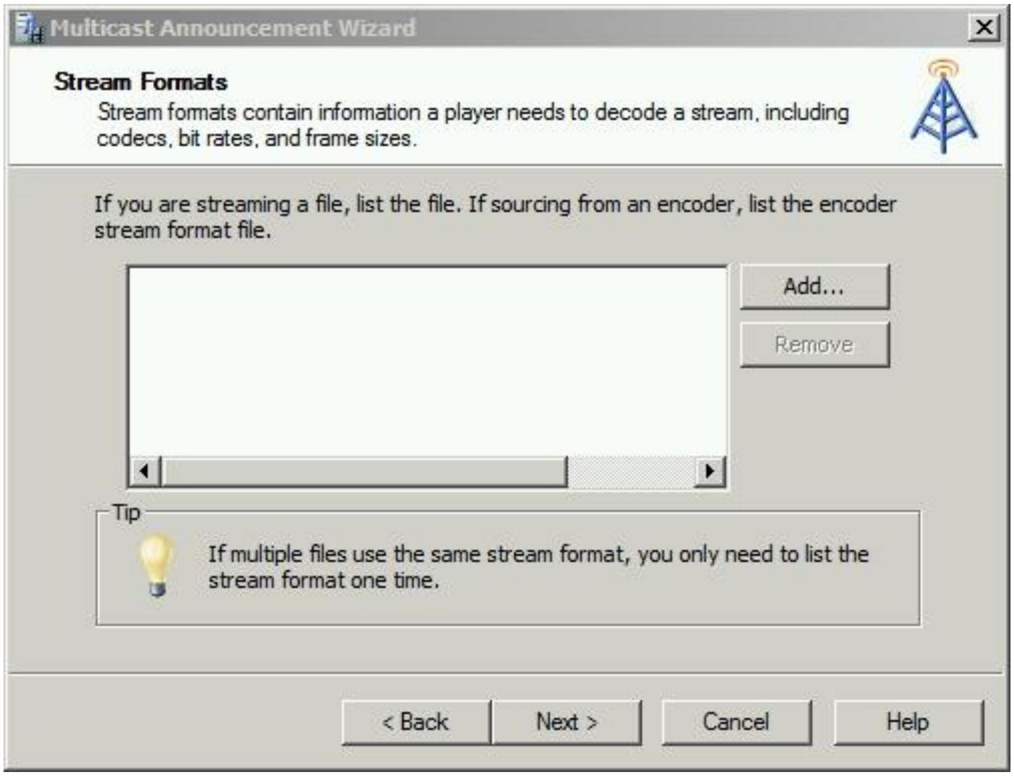

Add Publishing Point Wizard
Completing the Add Publishing Point Wizard
You have successfully completed the Add Publishing Point Wizard. To deliver your content, you must have a multicast information file (.nsc) for this publishing point. You can create one via the Multicast Announce Wizard.
After the wizard finishes:
Create a .nsc file (recommended)
Create a wrapper playlist (.wsx)
Create a wrapper playlist (.wsx) and .nsc file
To complete this wizard, click Finish.
< Back
Finish
Cancel
Help

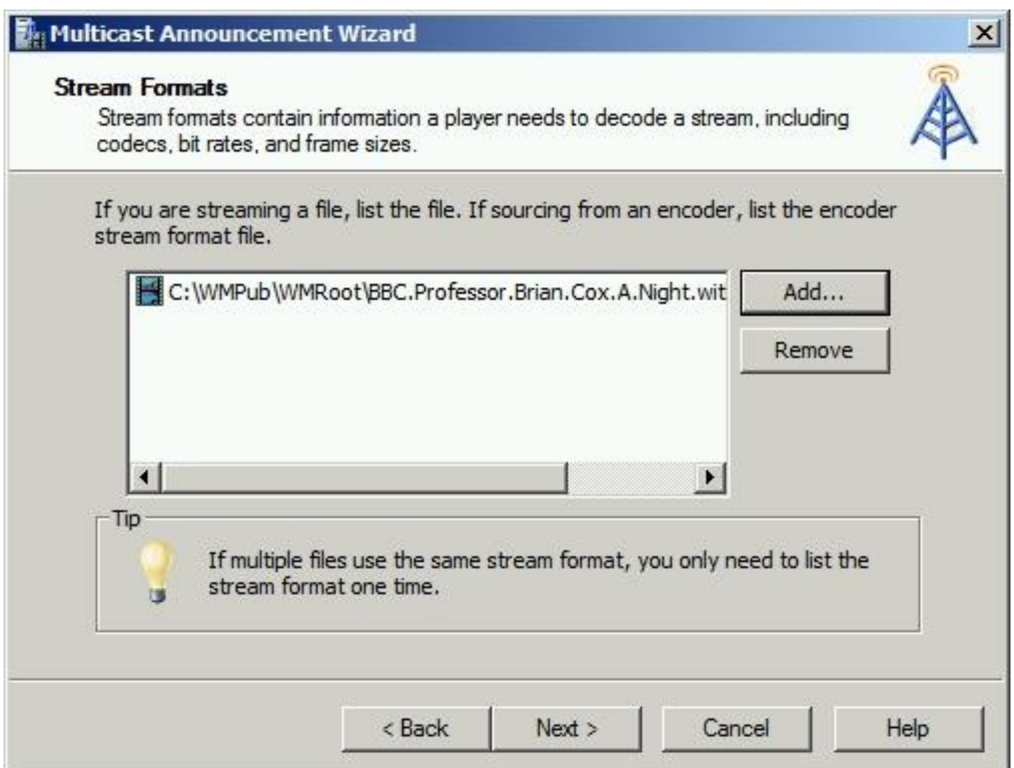
APPENDIX 3(5).


Multicast Announcement Wizard
Welcome to the Multicast Announcement Wizard
A multicast requires that several files be present in order for the player to successfully view the stream. This wizard helps you create or modify those files, which include:
- A mulitícast information file (.nsc). This file enables Windows Media Player to decode the stream.
- An announcement file (.asx). This file directs Windows Media Player to the location where your content is stored.
- A Web page. This page is optional and contains an embedded player and a link to your content.
Do not show this Welcome page again
To continue, click Next.
< Back
Next >
Cancel
Help

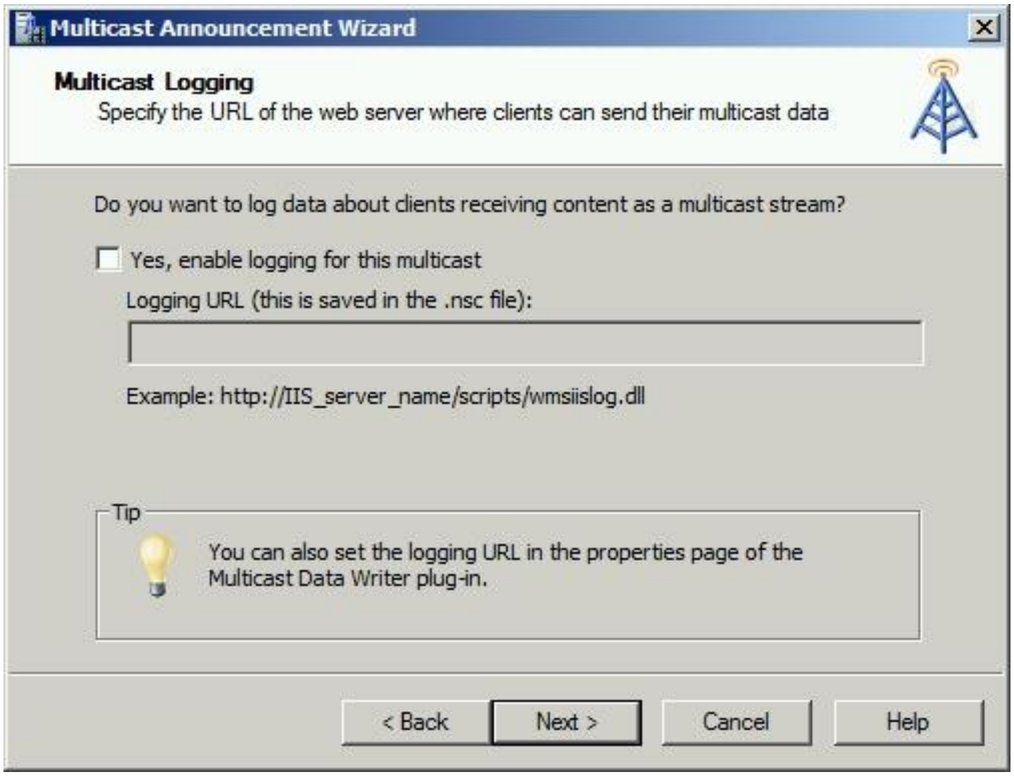

Multicast Announcement Wizard
Specify Files to Create
Create all the files needed for your multicast or update individual ones.
What files do you want to create?
Multicast information file (.nsc) and announcement file (.asx)
Automatically create a web page
Multicast information file (.nsc)
Announcement file (.asx)
Automatically create a web page
Tip
Recommended if this is the first time you are announcing your multicast or if you need to recreate all files.
< Back
Next >
Cancel
Help

APPENDIX 3(6).

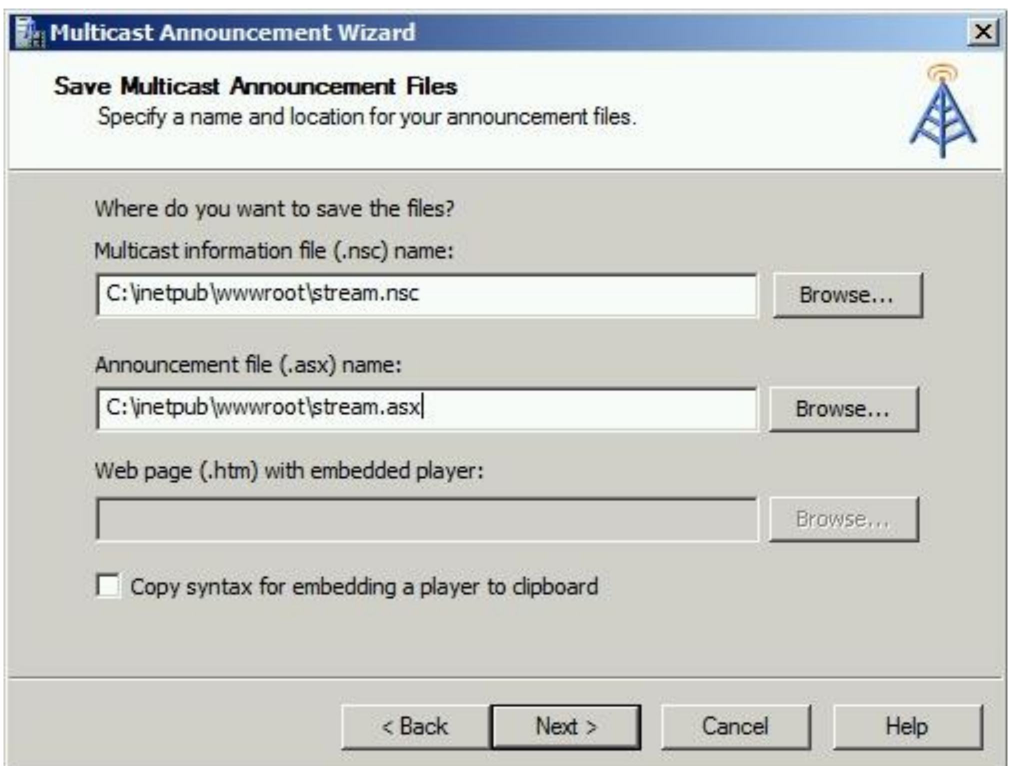

Multicast Announcement Wizard
Stream Formats
Stream formats contain information a player needs to decode a stream, including codecs, bit rates, and frame sizes.
If you are streaming a file, list the file. If sourcing from an encoder, list the encoder stream format file.
Add...
Remove
Tip
If multiple files use the same stream format, you only need to list the stream format one time.
< Back
Next >
Cancel
Help


Multicast Announcement Wizard
Stream Formats
Stream formats contain information a player needs to decode a stream, including codecs, bit rates, and frame sizes.
If you are streaming a file, list the file. If sourcing from an encoder, list the encoder stream format file.
C:\WMPub\WMRoot\BBC.Professor.Brian.Cox.A.Night.wit
Add...
Remove
Tip
If multiple files use the same stream format, you only need to list the stream format one time.
< Back
Next >
Cancel
Help

APPENDIX 3(7).

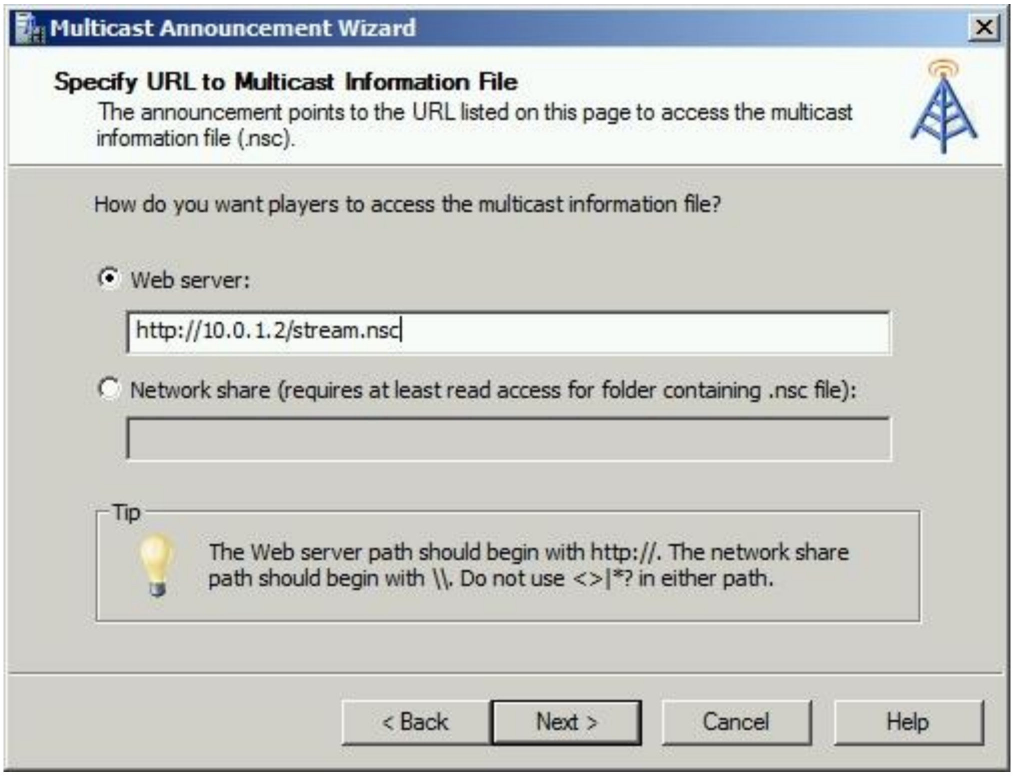


Multicast Announcement Wizard
Multicast Logging
Specify the URL of the web server where clients can send their multicast data
Do you want to log data about clients receiving content as a multicast stream?
Yes, enable logging for this multicast
Logging URL (this is saved in the .nsc file):
Example: http://IIS_server_name/scripts/wmsiislog.dll
Tip
You can also set the logging URL in the properties page of the Multicast Data Writer plug-in.
< Back
Next >
Cancel
Help

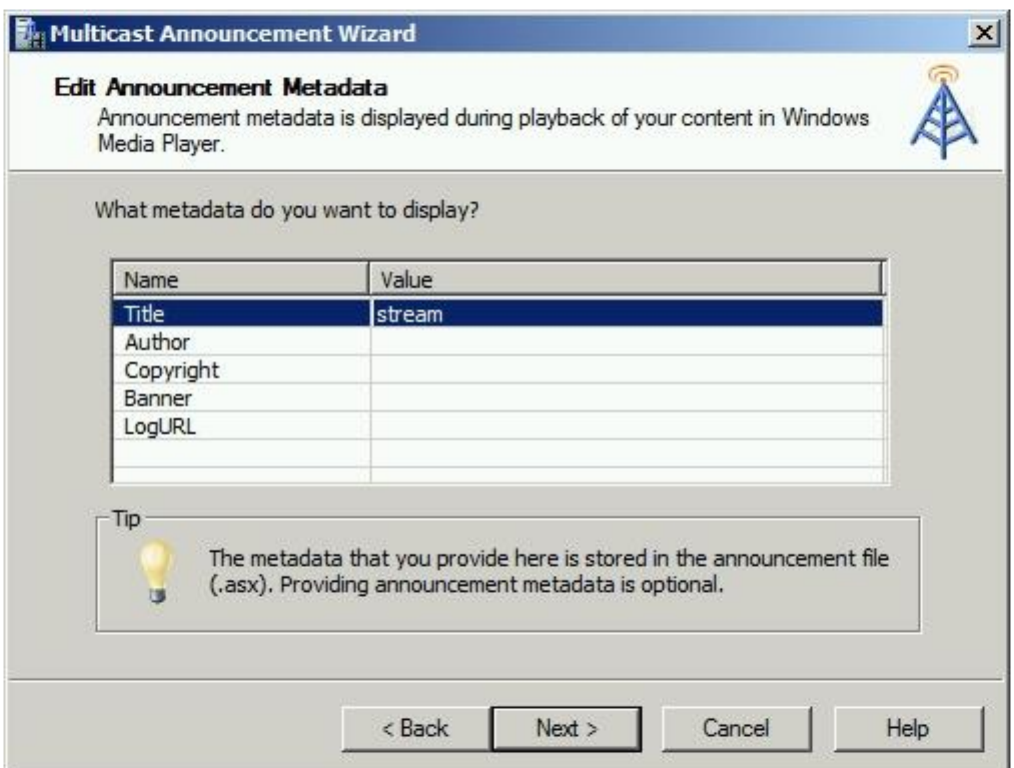


Multicast Announcement Wizard
Save Multicast Announcement Files
Specify a name and location for your announcement files.
Where do you want to save the files?
Multicast information file (.nsc) name:
C:\inetpub\wwwroot\stream.nsc
Browse...
Announcement file (.asx) name:
C:\inetpub\wwwroot\stream.asx
Browse...
Web page (.htm) with embedded player:
Browse...
Copy syntax for embedding a player to clipboard
< Back
Next >
Cancel
Help

APPENDIX 3(8).

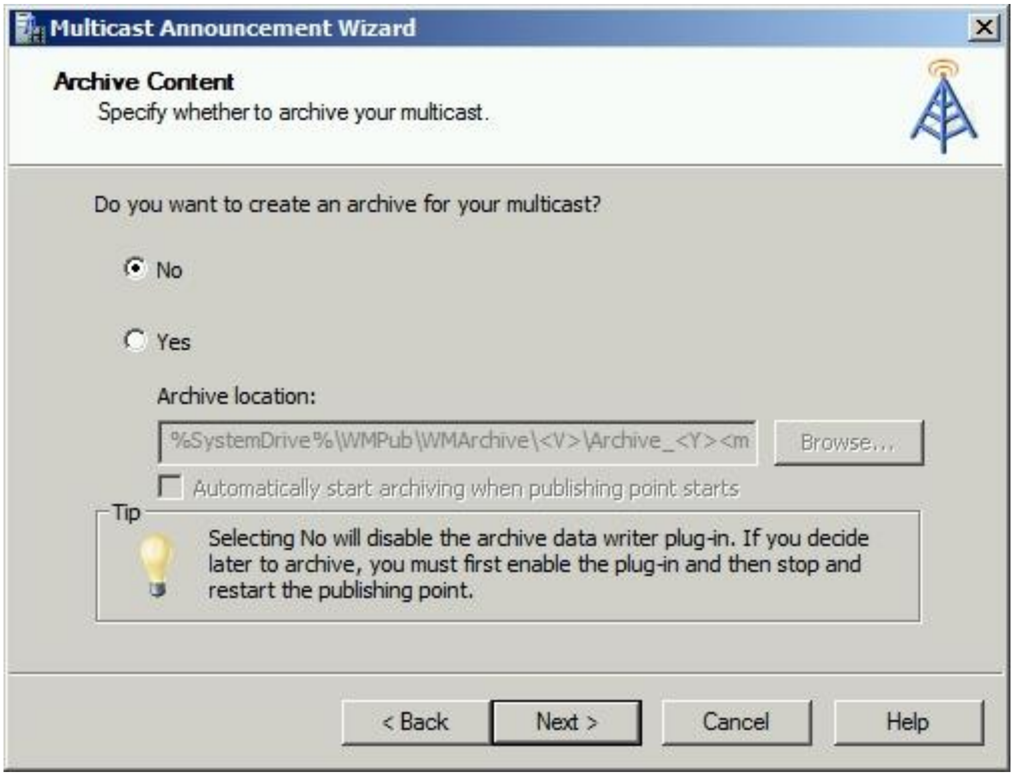


Multicast Announcement Wizard
Specify URL to Multicast Information File
The announcement points to the URL listed on this page to access the multicast information file (.nsc).
How do you want players to access the multicast information file?
Web server:
http://10.0.1.2/stream.nsc
Network share (requires at least read access for folder containing .nsc file):
Tip
The Web server path should begin with http://. The network share path should begin with \\. Do not use <>|*? in either path.
< Back
Next >
Cancel
Help



Multicast Announcement Wizard
Edit Announcement Metadata
Announcement metadata is displayed during playback of your content in Windows Media Player.
What metadata do you want to display?
Name
Value
Title
stream
Author
Copyright
Banner
LogURL
Tip
The metadata that you provide here is stored in the announcement file (.asx). Providing announcement metadata is optional.
< Back
Next >
Cancel
Help

APPENDIX 3(9).


Multicast Announcement Wizard
Archive Content
Specify whether to archive your multicast.
Do you want to create an archive for your multicast?
No
Yes
Archive location:
%SystemDrive%\WMPub\WMArchive\<V>\Archive_<Y><m
Browse...
Automatically start archiving when publishing point starts
Tip
Selecting No will disable the archive data writer plug-in. If you decide later to archive, you must first enable the plug-in and then stop and restart the publishing point.
< Back
Next >
Cancel
Help


Multicast Announcement Wizard
Completing the Multicast Announcement Wizard
You have successfully completed the Multicast Announcement Wizard for the publishing point stream.
The following files are created when you click Finish.
C:\inetpub\wwwroot\stream.nsc
C:\inetpub\wwwroot\stream.asx
Test files when this wizard finishes
Start publishing point when wizard finishes
To close this wizard, click Finish.
< Back
Finish
Cancel
Help