TalkShow Advanced Network Tips

NewTek Workflow Team

TalkShow is a powerful tool to expand a live production. While connecting in a TalkShow unit is as simple as plugging in a network cord and an SDI cable, there are best practices relating to the Skype peer-to-peer network that can be critical in establishing a trouble-free, high-quality video connection.

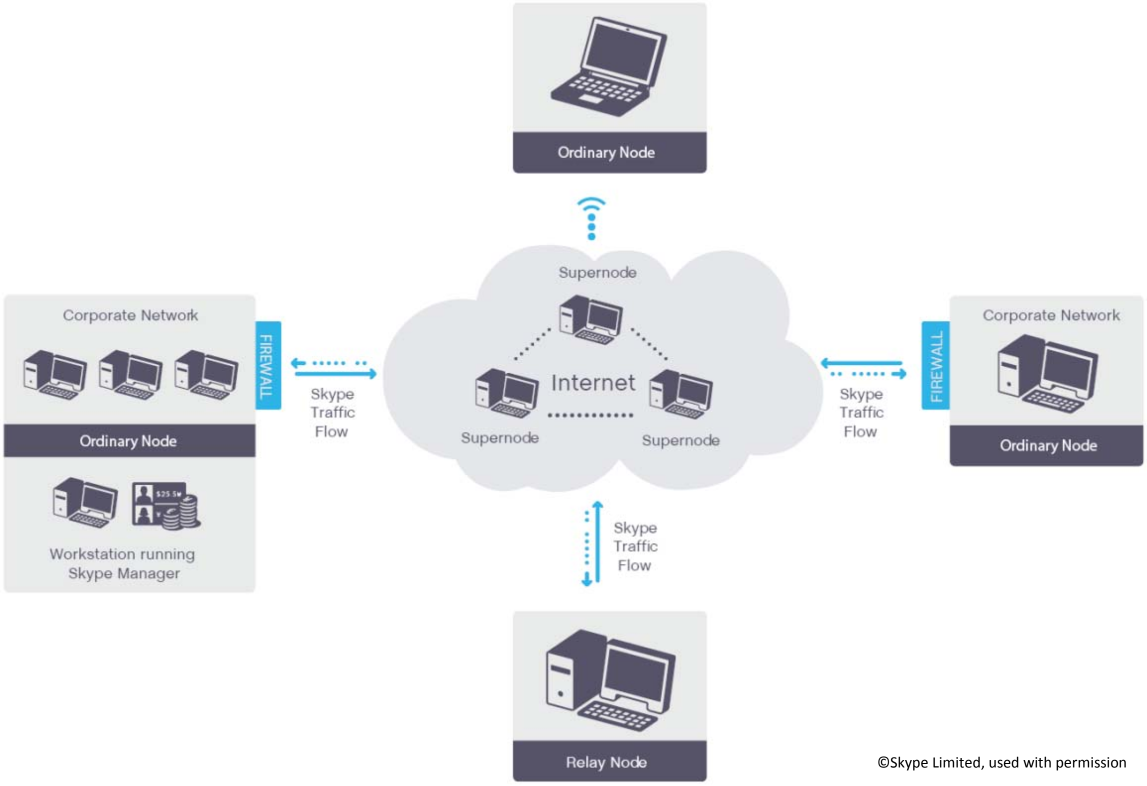
This paper will get to the heart of these best practices and is intended for professionals familiar with common networking devices and concepts. For more detailed information, see the Skype IT Administrator's Guide: http://download.skype.com/share/business/guides/skype-it-administrators-guide.pdf

Before the Call

The Technique

Skype utilizes an overlay peer-to-peer network to transmit video and voice calls, with the objective to route UDP traffic directly between peers. To arrange this – the 'overlay' – a third participant is required: a Skype "Super Node" which acts as an "introducer" for TalkShow and the remote caller.

In most situations, both TalkShow and the remote peers are behind NAT, and only the Super Node has a public IP address. When TalkShow initiates a call, it asks the Skype Super Node for an "introduction" to the remote. The Super Node responds to TalkShow with the remote caller's public IP address and UDP port number. Due to NAT, this address/port combination will differ from that of the remote client itself. The Super Node also sends a message to the remote with TalkShow's public IP address and UDP port number. Again, NAT will cause this to differ from TalkShow's private IP.



©Skype Limited, used with permission

At this point, both TalkShow and the remote caller know each other's public IP address and UDP port number. Both TalkShow and the remote caller will attempt to send UDP messages directly to each other. If TalkShow and the remote caller happen to be behind the same NAT (with hairpinning allowed) then the IP addresses and UDP port numbers will allow peer-to-peer communication. The more common case, however, is that the peers are behind different NATs which disallow incoming connections and the originally reported IP addresses and UDP ports will be useless.

To handle this, both peers attempt to open outbound direct communication with the IP address/UDP port number reported by the Super Node. Initially, both NATs will filter out any UDP packets arriving from the other's public IP address. This gets rectified by each peer's newly-opened networking session keyed to the other's public IP address. In this way, each NAT is "tricked" into thinking that its respective internal host is the "initiator" of the new session, when actually it is fully symmetrical.[1]

The NAT Requirement

For the above technique to work, NATs must be designed so they only assign one public IP address and public UDP port to each private IP address/UDP port combination. If instead a new UDP port is assigned for each new UDP session, then direct peer-to-peer communication will fail – in essence the 'trick' won't work because the NAT will be wise that the two requests are actually different sessions, using completely different IP/port number combinations.[2]

Fortunately, Skype technology allows for successful calls even when this requirement is not met. The solution is an array of 'relay nodes' that reside at public IP addresses; however, to prevent overuse of these nodes' bandwidth, strict limits are placed on throughput and the result is lower call quality.

The Details

Now that we know what connection strategy is being employed, let's get specific about ports and security. Microsoft suggests the following in reference to ports for the standard Skype client:[3]

> To work correctly, Skype requires unrestricted outgoing TCP/UDP access to:
>
> - All destination ports above 1024
>
>   **or**
> - Ports 80 and 443
>
> If your firewall restricts access to both of these, you will need to update your firewall settings before you can use Skype.

Practically, networks tend to fall into two categories: most already allow what Skype needs while others intentionally incorporate restrictions that interfere. For the latter, opening up broad port ranges through the firewall may not be possible due to security requirements. There are more surgical methods.

When TalkShow's SkypeTX software is installed, a listening port above 1024 is chosen at random for incoming connections. By specifying a port instead of allowing a random one, only this port will need to be open for incoming connections. As far as outgoing connections are concerned, Skype tends to use ports over 30000 so the 1024 threshold can be moved up with some safety. Alternately, Skype can be set to utilize ports 80 and 443, but this may conflict with other services and require those services to themselves be moved to other port numbers.

[1] Bryan Ford, "NAT Check", http://midcom-p2p.sourceforge.net, accessed 01/15/2015

[2] Ibid.

[3] Microsoft, "Which Skype ports need to be open to use Skype for Windows desktop?", https://support.skype.com/en/faq/FA148/which-ports-need-to-be-open-to-use-skype-for-windows-desktop, accessed 01/15/2015

Skype also support SOCKS5 and https proxy services.[4] If you already utilize these servers, they present a very straightforward way to open access for Skype within your existing security framework. A more basic way to permit Skype the outgoing access it needs in a secure manner is through MAC filtering. Your firewall should allow specification of TalkShow's MAC address as a requirement to allow outgoing requests as specified, and this will provide a much more restrictive secure environment than temporarily or permanently opening outgoing port access if you otherwise have it restricted.

Regardless of the method, it is critical that the following be true of the network:

- TalkShow can listen for incoming public connections on its specified port number, which by default will also fallback to 80 and 443.
- TalkShow can open outgoing connections for ports above 30000.
- For internal calls inside the same firewall, the firewall must support packet hairpinning.

During the Call

Once a call is established there is plenty of information that can be obtained about the call and what quality to expect based off that information. In TalkShow, this technical information is displayed down the right side panel. The desktop Skype client requires a few steps to show the same information. **Skype > Tools > Options > Advanced > Connection** will allow selection of "**Display technical call info during calls.**" Thereafter Call Technical Information can be found in the Call menu during a live call.

Descriptions of the technical information are available in the TalkShow User Guide; key items are highlighted here.

Network - *Describes the network connection to TalkShow.*

Round Trip –The time it takes, in milliseconds, for a signal to be sent from Skype TX to the remote caller and back. *Roundtrip times 200ms or lower represent excellent conditions. If this number reaches 350ms, the need to pause during the conversation becomes significant.*
Transport – Possible values are (from best to worst): UDP, UDP with Relay, TCP, TCP with Relay. Relaying is required where a direct connection cannot be made, and may cause poor video resolution or connection reliability.
UDP Status – A value of 'good' 'ok' or 'bad' indicates the possibility of UDP connectivity in the indicated direction.
Relays – On the Skype desktop client, indicates the number of Skype Relay nodes transporting data for the connection.

System - *Technical information related to TalkShow and remote caller's computer.*

CPU Total – The total CPU load being used across all cores of the CPU. *CPU usage over 80% will almost always result in poor quality video. Attempt to reduce the number of other applications running on the same machine.*
CPU Skype – The percentage of current CPU load being consumed by Skype.
CPU Hiccups – The number of instances where the system takes more time than expected to perform a Skype-requested operation. For example, updating the Skype database on a hard drive may cause a hiccup.
Skype Version – Shows the Skype client version, if available.

Audio - *Shows the technical information of the audio to and from the remote caller.*

Audio Packet Loss % – The percentage of voice data packets being lost.
Audio Cap – The maximum audio bandwidth achievable, according to the network bandwidth monitor.
Audio Packet Length – Indicates the length of each audio data packet sent.

[4] Microsoft, "Connect Skype Through a Proxy Server", https://support.skype.com/en/faq/FA1017/can-i-connect-to-skype-through-a-proxy-server, accessed 01/19/2015

Video Capture - *Shows information about video being supplied to Skype from TalkShow.*

Width – The width of the video being supplied to Skype.
Height – The height of the video being supplied to Skype.
Camera Frame Rate – The actual frame rate of the camera being used as the capture device.
Requested Frame Rate – The user-defined frame rate of the camera being used as the capture device.

Video Stream - *Information about video being streamed to the remote caller; data about the received video is listed in the separate 'Video Receive' section.*

Width – The width of the video being streamed.
Height – The height of the video being streamed.
Upswitch Allowed – Does the receiver give the sender permission to upswitch or would it be overloaded?
Target Frame Rate – User-defined frame rate being negotiated by the client and the remote caller.
Bitrate – A measure of the bandwidth being used by the video stream. Higher is better.
Bitrate Cap – The maximum bitrate achievable, according to the network bandwidth monitor.
Video Cap – The maximum bandwidth achievable, according to the network bandwidth monitor.
MTU – Maximum Transmission Unit. The largest size video frame that can be sent.
Complexity – The measure, in levels, of the processing power needed to reconstruct the compressed data from the video steam.
Thread Count – SLIQ encoder: Indicates how many threads are doing the encoding work.
Max Threads – SLIQ encoder only: Indicates the maximum number of threads allowed to do the encoding work.

Tips and Other Things to Know

From a troubleshooting perspective, there are a few things to know about Skype when attempting to get the highest quality video.

Bandwidth

Regardless of network connection details, high quality video connections require a minimum bandwidth or better. This chart illustrates the requirements and recommendations provided by Skype:

Call type	Minimum download / upload speed	Recommended download / upload speed
Calling	30 KBps / 30 Kbps	100 Kbps/ 100 Kbps
Video calling / screen sharing	128 Kbps / 128 Kbps	300 Kbps/ 300 Kbps
Video calling (high-quality)	400 Kbps /400 Kbps	500 Kbps/ 500 Kbps
Video calling (720 HD)	1.2 Mbps / 1.2 Mbps	1.5 Mbps / 1.5 Mbps
Video calling (1080 HD)	3.5 Mbps / 3.5 Mbps	5 Mbps / 5 Mbps

Settling In

Skype "ramps up" to high quality. This may not be obvious initially and can cause wasted troubleshooting. The Skype client as well as TalkShow do a lot of calculations and adjust over time. Often the connection will start off at a lower quality setting. If you don't see an expected change right away, give it a minute or two to improve.

Encoding Quality

Skype does not give control over the video encoder. Although it is adaptive, one thing a user can do to produce the best quality image is to ensure there is contrast between the subject and the background. For more on best practices, see the Remote Caller Guide for TalkShow. You can download it here:

http://www.newtek.com/images/talkshow/pdf/Live-Skype-Video-Remote-Caller-Guide.pdf

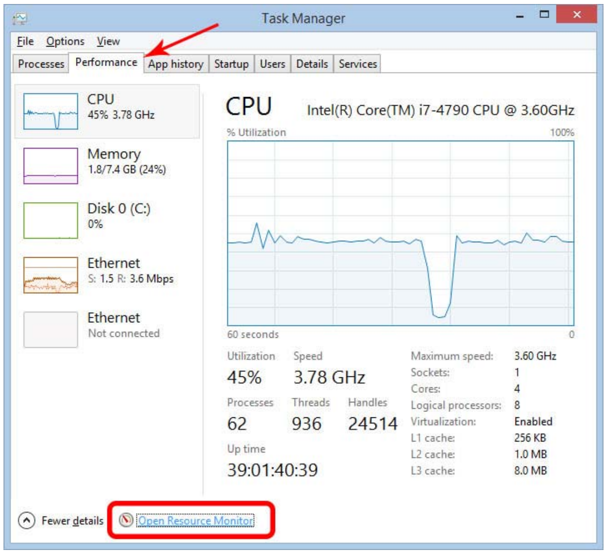
Local Loop Control

As can be seen by the connection process outlined above, the set up of the 'local loop' network, those components linking the TalkShow or remote caller's computer to the internet backbone, are a critical part of establishing a high-quality link. While many ISPs provide relatively clear paths to the backbone, some more-complex providers, such as cell providers, need to manage their data traffic more carefully and, in doing so, can create paths which Skype cannot traverse without using relay nodes and their inherent limits.

Basic Network Monitoring

Lastly, the Skype client and TalkShow software will show bandwidth being used in packets, but that doesn't always help when we want to know how much bandwidth is being used. To get this information, we can use the Resource Monitor in Windows 8.[6]

To open Resource Monitor, use the keyboard shortcut Ctrl + Shift + ESC to open Task Manager.

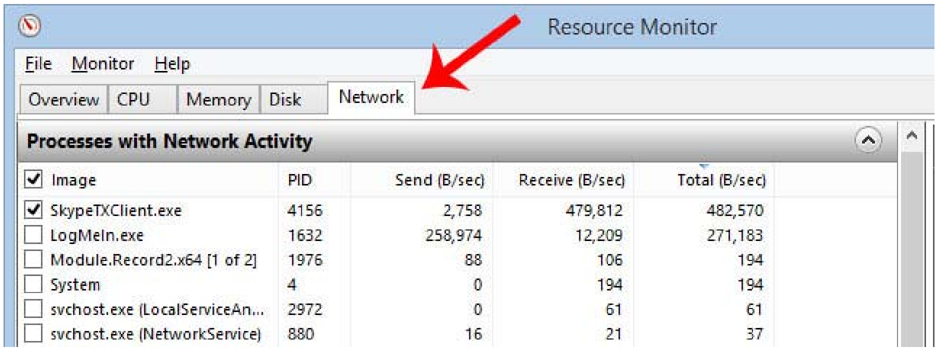


Utilization 45%
Speed 3.78 GHz
Processes 62
Threads 936
Handles 24514
Up time 39:01:40:39

Maximum speed:	3.60 GHz
Sockets:	1
Cores:	4
Logical processors:	8
Virtualization:	Enabled
L1 cache:	256 KB
L2 cache:	1.0 MB
L3 cache:	8.0 MB

Then, go to the *Performance* tab. At the bottom of that tab, Click a button or link saying *"Resource Monitor*".

The *Overview* window shows *CPU* activity by default. Click on the tab for *Network.*



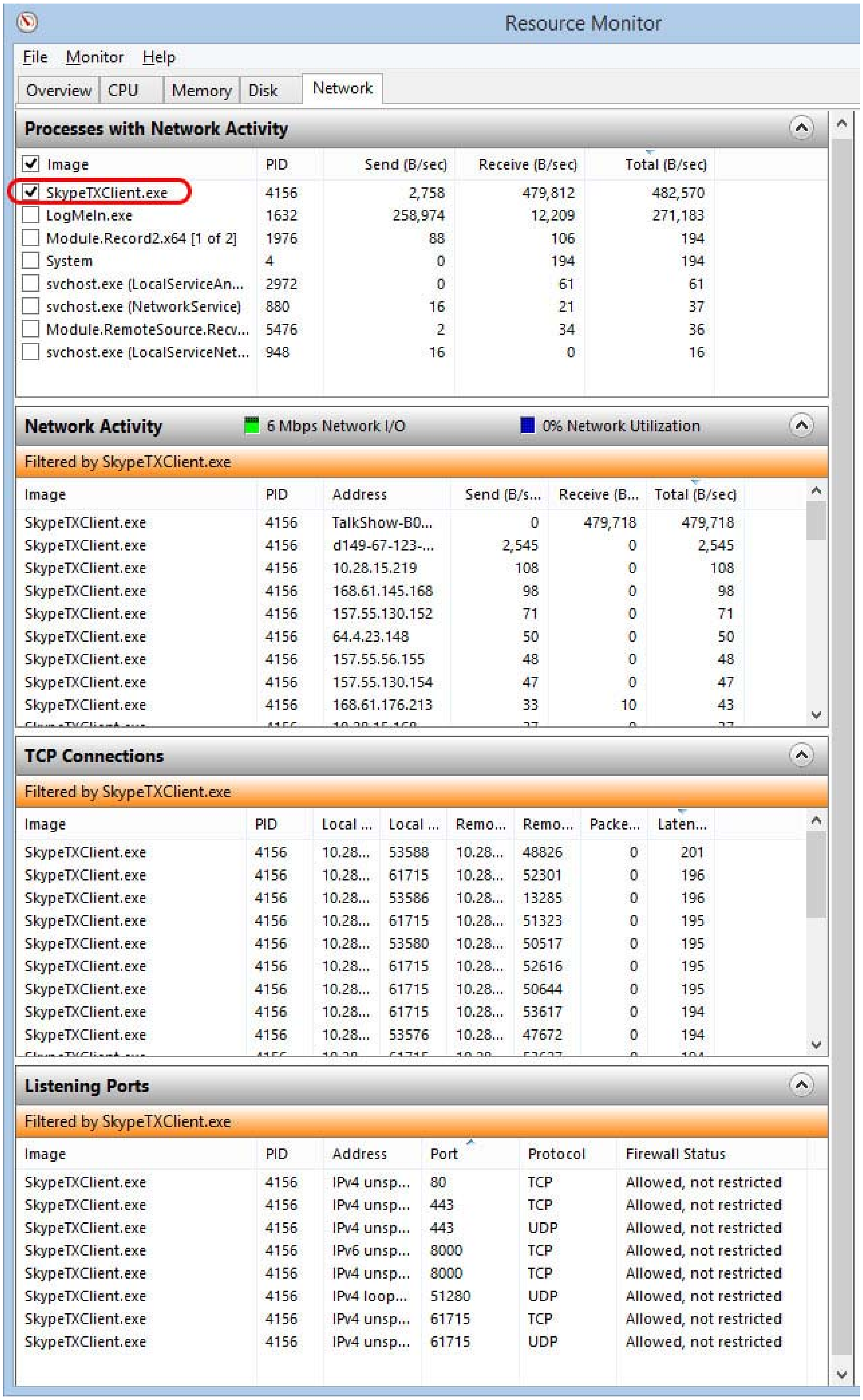
Processes with Network Activity

☑ Image	PID	Send (B/sec)	Receive (B/sec)	Total (B/sec)
☑ SkypeTXClient.exe	4156	2,758	479,812	482,570
□ LogMeIn.exe	1632	258,974	12,209	271,183
□ Module.Record2.x64 [1 of 2]	1976	88	106	194
□ System	4	0	194	194
□ svchost.exe (LocalServiceAn...	2972	0	61	61
□ svchost.exe (NetworkService)	880	16	21	37

You'll see sections for *Processes with Network Activity, Network Activity, TCP Connections,* and *Listening Ports*. The first section is the only one you can do anything with; the others are for your information but you cannot manipulate or change anything in them. *TCP Connections* and *Listening Ports* contain information that is useful to more advanced users.

[6] "How to use the Resource Monitor", http://www.7tutorials.com/how-use-resource-monitor-windows-7, accessed on 01/15/2015

Take a look at the *Processes with Network Activity* section. Here you'll find a list with programs you're running that are connecting to your network and to the Internet. One very useful thing you can do in this tab is to select one process or a group you are interested in and the data in the lower sections becomes filtered, so you can see the *Network Activity, TCP Connections* or *Listening Ports* for only the selection you've made. Selecting SkypeTXClient.exe will show you details relevant to your Skype call.



Processes with Network Activity

☑ Image	PID	Send (B/sec)	Receive (B/sec)	Total (B/sec)
☑ SkypeTXClient.exe	4156	2,758	479,812	482,570
□ LogMeIn.exe	1632	258,974	12,209	271,183
□ Module.Record2.x64 [1 of 2]	1976	88	106	194
□ System	4	0	194	194
□ svchost.exe (LocalServiceAn...	2972	0	61	61
□ svchost.exe (NetworkService)	880	16	21	37
□ Module.RemoteSource.Recv...	5476	2	34	36
□ svchost.exe (LocalServiceNet...	948	16	0	16

Network Activity 🟩 6 Mbps Network I/O 🟦 0% Network Utilization

Filtered by SkypeTXClient.exe

Image	PID	Address	Send (B/s...	Receive (B...	Total (B/sec)
SkypeTXClient.exe	4156	TalkShow-B0...	0	479,718	479,718
SkypeTXClient.exe	4156	d149-67-123-...	2,545	0	2,545
SkypeTXClient.exe	4156	10.28.15.219	108	0	108
SkypeTXClient.exe	4156	168.61.145.168	98	0	98
SkypeTXClient.exe	4156	157.55.130.152	71	0	71
SkypeTXClient.exe	4156	64.4.23.148	50	0	50
SkypeTXClient.exe	4156	157.55.56.155	48	0	48
SkypeTXClient.exe	4156	157.55.130.154	47	0	47
SkypeTXClient.exe	4156	168.61.176.213	33	10	43

TCP Connections

Filtered by SkypeTXClient.exe

Image	PID	Local ...	Local ...	Remo...	Remo...	Packe...	Laten...
SkypeTXClient.exe	4156	10.28...	53588	10.28...	48826	0	201
SkypeTXClient.exe	4156	10.28...	61715	10.28...	52301	0	196
SkypeTXClient.exe	4156	10.28...	53586	10.28...	13285	0	196
SkypeTXClient.exe	4156	10.28...	61715	10.28...	51323	0	195
SkypeTXClient.exe	4156	10.28...	53580	10.28...	50517	0	195
SkypeTXClient.exe	4156	10.28...	61715	10.28...	52616	0	195
SkypeTXClient.exe	4156	10.28...	61715	10.28...	50644	0	195
SkypeTXClient.exe	4156	10.28...	61715	10.28...	53617	0	194
SkypeTXClient.exe	4156	10.28...	53576	10.28...	47672	0	194

Listening Ports

Filtered by SkypeTXClient.exe

Image	PID	Address	Port	Protocol	Firewall Status
SkypeTXClient.exe	4156	IPv4 unsp...	80	TCP	Allowed, not restricted
SkypeTXClient.exe	4156	IPv4 unsp...	443	TCP	Allowed, not restricted
SkypeTXClient.exe	4156	IPv4 unsp...	443	UDP	Allowed, not restricted
SkypeTXClient.exe	4156	IPv6 unsp...	8000	TCP	Allowed, not restricted
SkypeTXClient.exe	4156	IPv4 unsp...	8000	TCP	Allowed, not restricted
SkypeTXClient.exe	4156	IPv4 loop...	51280	UDP	Allowed, not restricted
SkypeTXClient.exe	4156	IPv4 unsp...	61715	TCP	Allowed, not restricted
SkypeTXClient.exe	4156	IPv4 unsp...	61715	UDP	Allowed, not restricted