Inside Citrix chapter nineteen – The one with the ICA/HDX protocol

Citrix is an American software company founded back in 1989 by Edward Lacobucci, a former IBM developer. Citrix's first office was located in Richardson, Texas, and would later that year be relocated to Coral Springs, Florida, housing 18 FTE already. While they actually started out as Citrus Systems, shortly after an existing company claimed the trademark rights to that name they changed it to Citrix, as we know it still. In its early days the company primarily focused on developing remote access products for Microsoft Windows Operating systems, OS/2 at the time. As a result, and after two years of development, Citrix released their first product, named Citrix Multiuser, which was actually an extension to the OS/2 platform. This is where the ICA protocol, or the Independent Computing Architecture was born.

> **FMA fact**: The ICA protocol originated with Citrix Multiuser, around 1990 / 1991, meaning that the ICA protocol is actually over 25 years of age already.

They acquired / licensed the OS/2 source code from Microsoft (and basically became BFFs from then on) and built the ICA protocol from the ground up, the main ingredient for Citrix Multiuser version 1.0 at that time. The software allowed multiple users to work from separate computers remotely accessing software from a server: sounds familiar, right? Today they provide server, application and desktop virtualisation, networking, Software as a Service (SaaS), and cloud computing technologies. It has been quite a journey.

The agreement

Around the same time that Citrix released their Multiuser product, Microsoft announced to move away from the OS/2 platform over to Windows. This basically left Multiuser version 1.0 useless unless some significant changes were to be made to its base code, making it compatible with both Windows and DOS. While Citrix was on the verge of closing, multiple investors (including Microsoft) kept them on track, leading up to their second big release, Citrix Multiuser version 2.0, also referred to as Multi-Win back in 1992: fully compatible with Microsoft DOS and it allowed up to 5 users simultaneously.

In the meantime they signed a license agreement with Microsoft (which has been renewed multiple times) allowing them to use the Microsoft NT (3.5) source code for building an even more robust remote access protocol, improving the ICA, which by then was already patented by Citrix. Citrix basically modified Windows NT, turning it into a multi-user platform.

And while there were some struggles between the two, since Microsoft basically funded Citrix to develop the ICA protocol and now they wanted their piece of the pie, eventually they became friends again and Microsoft 'licensed' Citrix's ICA protocol for use with Windows NT 4.0, 5.0 and onwards. However, at this time Microsoft was still empty-handed from a remoting protocol point of view.

With this renewed agreement, for which they paid good money, by the way, Microsoft regained the opportunity to make use of Citrix's ICA technology and to come up with a potentially competing protocol themselves; this is where the RDP protocol found its origin, based and built on the technology and ideas of Citrix.

Anyway, shortly after that, in 1993, they launched a new product named WinView, which was able to run both DOS and Windows applications and the company grew to over 65 employees in total during 1994. In the meantime they also launched their first Citrix-authorised re-seller programme, signed up Tech Data as their first official national distributor and achieved a net revenue of 10 million $, impressive to say the least.

This all led up to the 1995 launch of Citrix WinFrame, a true multi-user Operating System based on Microsoft's NT technology: at that time it allowed up to 15 users simultaneously, and I guess the rest is history.

In 1997 they opened a new headquarters in Fort Lauderdale, Florida, and after serving as the Vice-President of Marketing, Mark Templeton became the new Citrix CEO, a role which he had up until 2015, and with great success, I might add. Many love Mark; unfortunately I never had the pleasure to meet him personally. When Citrix celebrated their 20th anniversary they put together a PDF document highlighting all of their milestones during those 20 years. You will find it here:

https://www.citrix.com/content/dam/citrix/en_us/documents/go/citrix_timeline.pdf

The Independent Computing Architecture

Now that you know some of its history, let's have a closer look at some of its subcomponents, how they all fit together and what happens on the inside. Let me start with a short statement on what the ICA protocol actually does from a (very) high-level perspective.

In its most simple form, the ICA protocol transports keystrokes, mouse clicks and screen updates (using standard protocols like TCP/IP, IPX, NetBEUI, SPX) from the server to the client in a highly controlled and secure manner.

It is optimised for Wide Area Networks with high latency and offers various levels of Quality of Service. These types of protocols, since there are more besides ICA, are often referred to as remoting protocols.

> **FMA fact**: By default, the ICA protocol uses TCP port 1494. If Session Reliability is enabled a.k.a. the Common Gateway Protocol, or CGP then ICA traffic will be encapsulated through TCP port 2598. Note that any network traces that you might run will also show 2598 instead of 1494.

When Citrix released WinFrame back in 1995 it also introduced ICA version 3.0, which included ThinWire 1.0. Back then, ICA functionality was still limited to ThinWire (screen updates), printing, client drive mapping and audio. Before that there was the ICA version 1.0 in 1992 and

the ICA version 2.0, which was also released in 1992 as part of the Citrix Multiuser launch discussed earlier.

Virtual Channels

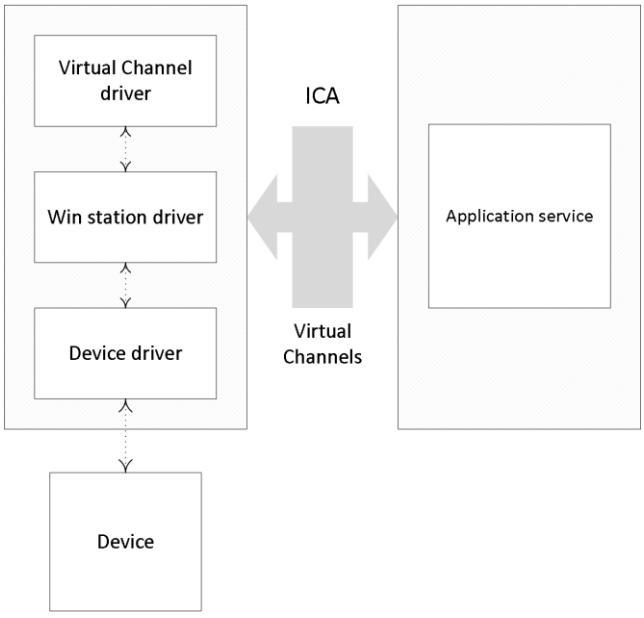
I already touched on the ICA protocol and its 32 virtual channels in the chapter 'The Citrix Receiver' as part of the nine main FMA components. This is where the client-server set-up highlighted earlier becomes even more apparent. Let's do a quick résumé since the virtual channels are an essential part of the ICA protocol stack.

A big part of the communication between the client and server takes place over and through what Citrix refers to as virtual channels. This is where most ICA / HDX features live. Each virtual channel consists of a client-side virtual driver (Receiver) that communicates with a server-side application (the VDA). I say 'most', because Receiver also offers and supports a whole bunch of additional features and functionalities that do not involve or need a virtual channel.

Virtual channels (there can be 32 channels in total) are mainly used for some of the bigger well-known features where a bigger than average and direct communication path between the client and server is needed, like client drive mapping, smart cards, clipboard, printing, audio, video and so on.

> **FMA fact**: As a (security) best practice Citrix recommends disabling any virtual channels that are not in use.

And of course from time to time, new virtual channels are released with new versions of XenDesktop and Receiver to provide additional functionality. Take Framehawk and ThinWire Plus, for example. Those were released as part of Feature Pack 2 and 3, respectively, for XenDesktop 7.6, including a new Receiver on the client side. Each virtual channel represents a specific feature or functionality on its own.



Virtual Channels - Client / Server overview

What happens in a nutshell

- When a new session is established, at client load time, first the client (Receiver) connects to the XenDesktop / XenApp Server (VDA).
- The client passes information about the virtual channels it supports to the server. This is where the version of the Receiver combined with VDA installed matters.
- The server-side application starts, obtains a handle to the virtual channel, and optionally queries the client for any additional information about the channel.
- As soon as additional data has been sent and received, the server virtual channel application is completed and it closes the virtual channel to free up any resources that may have been allocated.

The above is also referred to as the client server handshake.

The earlier mentioned client-side virtual drivers can be found in the following Registry Key:

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ICA Client\Engine\Configuration\Advanced\Modules\ICA 3.0

If you would like to disable certain client functionality, you can do so by editing the Registry Key mentioned above. Simply remove the functionality you would like to disable, like clipboard (clipboard mapping) or ClientDrive (client drive mapping).

> **FMA fact**: As mentioned, there are 32 virtual channels in total; however, Citrix reserves 17 of those. Third-party companies and customers who want to design and implement their own virtual channels are free to use the other ones. These are also referred to as dynamic virtual channels or DVCs.

Most features and functionalities configured at server level (mainly through policies) will need to be supported at the client side as well; there is a strong dependency between the two (think about the ICA handshake mentioned earlier).

Creating your own

Although XenDesktop / XenApp products ship with various virtual channels included, supported by both the VDA and Receiver, they are also designed to allow customers and third-party vendors to create their own virtual channels by using one of the provided SDKs, or Software Development Kits.

When you want or need to create a virtual channel of your own you basically have two choices. One: you can use the Virtual Channel SDK; or two: you can use the ICA Client Object (ICO) SDK. Citrix offers several resources for you to leverage when it comes to creating your own VC. Here is a shot excerpt from the Virtual Channel SDK page:

> The Citrix Virtual Channel Software Development Kit (VCSDK) allows software engineers to write both host-side applications and receiver-side drivers to support additional virtual channels using the Citrix ICA protocol. The host-side virtual channel applications run on XenApp or XenDesktop, and the client-side portion of the virtual channel runs on the local device where Citrix Receiver resides. This SDK provides support for writing new virtual channels for the Citrix Receiver.

Citrix offers the following online resources:

- The Citrix Virtual Channel Software Development Kit (login with My Citrix needed).
- CTX113279 – How to Allow Custom Virtual Channels Created with ICO in Version 10.0 of the CTX Windows Client. Although somewhat outdated, it does provide some offer some interesting additional information.
- Client Object API Specification Programmer's Guide.
- The Citrix Developer Network. Home of all technical resources and discussions involving the use of Citrix SDKs.

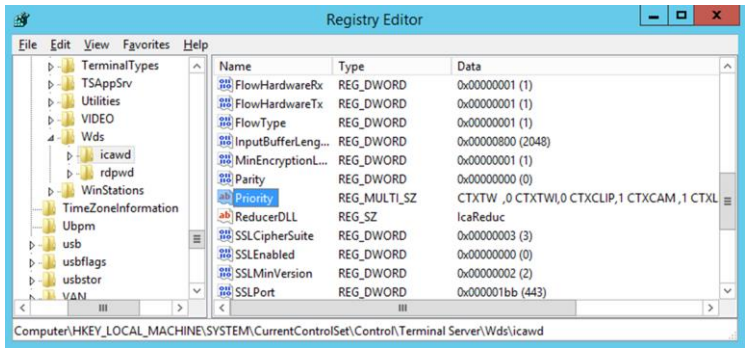
Life is all about priorities

As we have seen, the ICA protocol consists of various virtual channels each offering its own functionality. By default, each of these virtual channels is given a priority, ranging from 0 to 3: the lower the number, the higher the priority.

For example, printing has a default priority of 3, which means that it has the lowest priority and will therefore be allowed less bandwidth then virtual channels with a higher priority (lower number) like Audio and ThinWire (Windows screen updates).

While it is possible to manually change these priorities, it isn't that common to do so. You need to be aware that when giving a higher priority, and thus more bandwidth to one virtual channel it also means that you are taking away potential bandwidth from another virtual channel. However, in the rare occasion when you may want to assign a higher priority to one of the virtual channels, this is how it is done. In Registry locate the key:

HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\icawd\Priority.



Virtual Channels (priority) Registry Key

There you will find various abbreviations like: CTXCAM, CTXTWI, CTXFLASH and so on, accompanied by a number ranging from 0 to 3 (their current priorities). These two combined represent a single virtual channel.

By simply changing the number, you will change the priority of the associated virtual channel. These priorities, 0 to 3, are also commonly referred to as priority groups. And remember, always be careful when editing the Registry, make sure to back up the key, or keys beforehand.

> **FMA fact**: Other ways to accelerate ICA traffic would include Citrix policies, which can then be applied either per user or per server, or to the whole Site. Implementing a physical accelerator like the Citrix CloudBridge, formerly known as Branch Repeater, is always optional as well.

Multi-Stream ICA

While this (see previous section) does offer us some level of control with regard to ICA traffic acceleration, it is still fairly limited. By implementing, or activating a feature named Multi-Stream, or Multi-Port ICA we can configure true Quality of Service (QoS) on all or parts of the ICA / HDX traffic sent throughout our network. Note that I am referring to network-based QoS, which is different from prioritising a single virtual channel. Here we would like to be able to accelerate ICA traffic on an network (TCP/IP port) level rather than from within the ICA protocol itself. Without Multi-Stream ICA we can only accelerate ICA traffic as a whole on TCP/IP port 1494 or 2598 (Session Reliability) as discussed previously.

With Multi-Stream ICA enabled we can assign separate TCP/IP ports to each of the earlier mentioned priority groups a.k.a. streams within Multi-Stream ICA. This means that we can configure and assign a TCP/IP port for all priority 0 virtual channels, and another separate TCP/IP port for all priority 1 virtual channels, and so on. Meaning that there can be four Multi-Stream ports in total.

Each virtual channel will by default already have a priority assigned to it, as we've seen, ranging from 0 to 3, which correlates to very high (real-time activities, such as webcam content), high (interactive elements, such as screen, keyboard, and mouse), medium (for bulk processes, such as client drive mapping) and low (background activities, such as printing). This is true for single-stream ICA traffic (without Multi-Stream ICA enabled and configured) as well as Multi-Stream ICA traffic (with Multi-Stream ICA enabled and configured). However, as highlighted earlier, these priorities can be manually changed if and when needed.

The accompanying Multi-Stream ICA Registry Key, when enabled, can be found at:

HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\icawd\MultiStreamIca

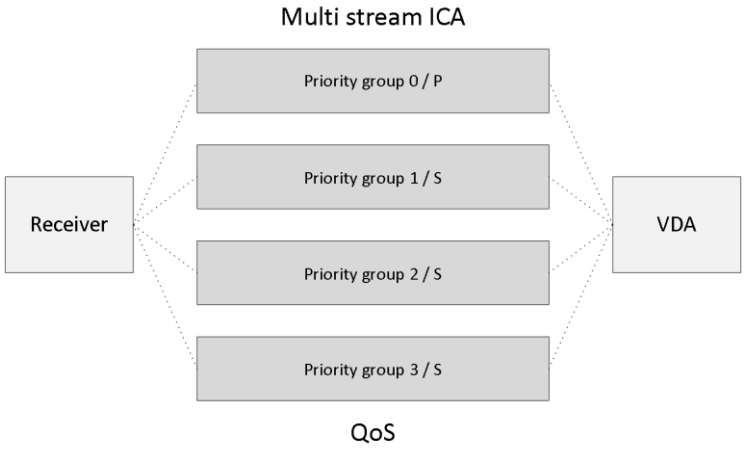
It will consist of two subkeys: Stream and VirtualChannels.

Within the Stream Registry Key we can manually configure the various stream priorities and assign them to be either primary or secondary. Or we can use the default configuration instead.

The format used is *Stream#, Stream type*. For example: 0,S will mean; all virtual channels with the priority 0 will be secondary, and 1,P means that all virtual channels with a priority of 1 will be primary and so on. Note that there can be only one primary stream, the rest will be secondary. The default configuration is: 0,S;1,P;2,S;3,S.

Within the VirtualChannel Registry Key we can manually configure the virtual channel stream pairs (binding a VC to a stream), which basically means that we assign a priority to a virtual channel, just like before. Or we can leave the default configuration in place. The format used is: *VirtualChannelName, Stream#*. CTXCAM,1 means that the virtual channel CTXCAM has been assigned to the stream 1.

As a result, it will be part of the stream pair 1,P, as highlighted earlier. The default configuration is: CTXCAM,0; CTXTW,1; CTXTWI,1; CTXLIC,1; CTXVFM,1; CTXPN,1; CTXSBR,1; CTXSCRD,1; CTXCTL,1; CTXEUEM,1; CTXMM,2; CTXFLSH,2; CTXGUSB,2; CTXCLIP,2; CTXCDM,2; CTXCCM,3; CTXCM,3; CTXLPT1,3; CTXLPT2,3; CTXCOM1,3; CTXCOM2,3; CTXCPM,3; OEMOEM,3; OEMOEM,2.



Multi-Stream ICA

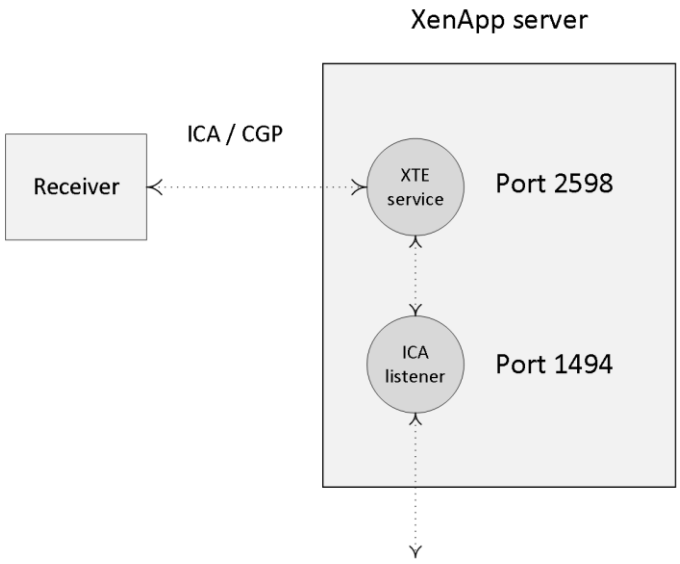
> **FMA fact**: When not using a CloudBridge appliance, formerly known as Branch Repeater, Session Reliability must be enabled for Multi-Stream ICA to function.

Once you have set up and configured the stream priorities and pairs you will have to configure a so-called multi-port Citrix policy where you configure separate TCP/IP ports to the primary (one) and secondary (three) streams as explained in the previous section.

Once that is out of the way you can actually go on and configure and apply QoS policies at network level on a per TCP/IP port level. This way you can apply QoS on grouped ICA virtual channels instead of a single or a couple of virtual channels within the ICA protocol.

Session Reliability

I already mentioned this feature once or twice throughout this chapter; here I will address it in a bit more detail. When Session Reliability is enabled, the ICA Client tunnels its ICA traffic inside the Common Gateway Protocol (CGP) and sends the traffic to port 2598 instead 1494. The XTE service acts as a relay; removing the Common Gateway Protocol layer and then forwarding traffic to the ICA listener on port 1494 internally as shown below:



Session Reliability

Internally, all ICA traffic coming from the XenApp server destined for the end-user's client device will be sent through or via the XTE service as well; it basically works the same way, only vice versa. Session Reliability has the ability to buffer ICA traffic when the CGP connection between the client and the XTE service is somehow broken; it will then temporarily store all ICA data until the connection is restored.

During that time, as long as the XTE service is buffering the ICA data, the user session will not go into a disconnected state; instead the session will remain active on the server.

From a client perspective, the session seems frozen while the client is attempting to reconnect with the XTE service over the Common Gateway Protocol. Once the session is restored, all buffered ICA data will be flushed and sent over to the client device and the session will continue as usual.

Configuration specifics

By default, Session Reliability (SR) is configured via policy and set to 180 seconds, or three minutes before the user session will be dropped and put into a disconnected state. However, this time-frame can be changed when needed. The default port used by SR is TCP/IP port Nr 2598 but can also be changed when desired.

> **FMA fact**: When Session Reliability is enabled users will be automatically reconnected as soon as the network connection is reinstated, and they will do so without needing to reauthenticate. Configuring the 'Auto client reconnect authentication' policy to prompt users to reauthenticate can change this behaviour.

Auto client reconnect is a feature used to detect unintended disconnected ICA sessions and will reconnect the user session automatically. As mentioned, users then do or do not have to reauthenticate, depending on how you configure the accompanying policy. If both Session Reliability and Auto client reconnect are used they will work in sequence, meaning that first the Session Reliability policy will be applied, and as soon as the user session disconnects because the configured SR time-frame has elapsed, the Auto client reconnect policy will kick in.

As an alternative to Session Reliability you can also configure ICA Keep-Alive. This feature prevents a session from going into a disconnected state when a session seems broken. If configured, it will send a constant stream of ICA packages every few seconds (configurable) to detect if the user session is active. Only after the session has been marked as inactive will the session be put in a disconnected state. However, in practice Session Reliability is almost always preferred over ICA Keep-Alive.

Citrix HDX

I just wanted to briefly touch on HDX (High Definition Experience) since it is more than 'just' the ICA protocol and it is often misunderstood. In fact, HDX technologies are built on top of the ICA protocol: and they are not meant as a replacement at all. HDX technologies extend the ICA protocol. According to Citrix: HDX technologies offer a set of capabilities that deliver a 'high-definition' experience to users of centralised applications and desktops, on any device and over any network.

It does this by trying to optimise the user experience, decrease the overall bandwidth consumption, and increase the user density per server. The HDX portfolio offers several innovative and industry-first technologies further enhancing and extending ICA, still the Nr. 1 remoting protocol in the industry today. A couple of examples of HDX technologies are: Flash and Windows media redirection, 4K monitor support, HDX 3D Pro GPU acceleration and sharing support (separate VDA), acceleration of printing and scanning, optimisation of USB traffic and more.

> **FMA fact**: Remember, Citrix HDX isn't a replacement for the ICA protocol. HDX technologies are meant as an extension and as such operate on top of the ICA protocol.

There are two HDX-related technologies, which I would like to highlight in particular: these are ThinWire compatibility mode and Framehawk.

ThinWire compatibility mode

Although this may sound like something completely new, ThinWire on its own has been part of the ICA protocol almost from the beginning: it was part of the WinFrame release back in 1995, or ICA 3.0 as mentioned earlier in this chapter. Then again, the technology introduced with ThinWire compatibility mode is indeed (very) new, innovative even. ThinWire compatibility

mode, which has had various names along the way, like project snowball, enhanced ThinWire, ThinWire Plus, and so on, was released as part of Feature Pack 3 for XenDesktop / XenApp version 7.6. It delivers a great user experience while keeping the CPU and bandwidth footprint as small as possible. In fact with the release of version 7.8 of XenDesktop / XenApp it has again been improved with enhanced lossless visual quality, sharpening fuzzy images at a faster rate when compared to version 7.7.

ThinWire compatibility mode is actually a fall-back mode for the current 'video codec for compression' which is used by default. It uses H.264 compression and delivers a high-quality and superior graphics and video experience for most users by default. The accompanying policies (based on templates) are configured out of the box and aim to deliver the best user experience possible. However, there is a trade-off to all this. This method consumes a lot more resources to encode and decode, which requires a relatively high-power processor on the client side and impacts user density on the server side.

If the default method fails to kick in, because the user's end point device doesn't have enough compute / CPU power on-board, for example, or an older unsupported version of Citrix Receiver, then the ThinWire compatibility mode will be leveraged automatically, hence the fall-back mentioned earlier. And since it uses a combination of low-cost algorithms, which are compatible with almost every Operating System out there, it can be used in almost all circumstances. As highlighted, it also offers a very effective, and above all efficient CPU and bandwidth footprint. So even when the default mode does work you might want to consider switching to compatibility mode anyway, enhancing user density, saving on valuable resources and network bandwidth in general.

> **FMA fact**: Make sure you check out the HDX policy templates in Studio. There are 6 in total.

Framehawk

Framehawk was first introduced with Feature Pack 2 for XenDesktop / XenApp version 7.6 combined with a new Receiver, version 4.3. And while it has been around a bit longer than ThinWire compatibility mode, it is still considered fairly new and is constantly being improved. Framehawk is mainly aimed at mobile workers who depend on Wi-Fi / 4G networks where things like packet loss and high latency are usually a problem. Framehawk can fix this for you. I have read about latency's reaching up to 500 ms and situations where 35% of packet loss was no exception and Framehawk still delivered a more than acceptable user experience. That is pretty impressive.

> **FMA fact**: If you go to YouTube and search for Citrix Framehawk you will find multiple comparison clips of Framehawk vs. other technologies. Guess who comes out on top?

On average Framehawk does tend to consume a lot more bandwidth and CPU resources when compared to ThinWire compatibility mode. However, with the latest 7.8 releases Framehawk has again been improved significantly with reductions in memory footprint of 40% and up to 20% in CPU efficiency.

They also gained up to 50% in bandwidth efficiency when scrolling via touch input, once again improving the overall user experience. Framehawk is now also compatible with the latest, or near latest, release of NetScaler Gateway (including the Unified Gateway as of release 11.0-F) and the Citrix Receiver for Windows and iOS. Just be aware that there will still be a difference between the two, and it will all depend on the use case at hand.

Key takeaways

- Edward Lacobucci founded Citrix in 1989.
- Initially they started developing a multi-user platform for Microsoft's OS/2.
- Citrix actually started out as Citrus.
- They licensed the OS/2 source code from Microsoft and started developing Multiuser, which would later become their first major release.
- ICA was introduced when Citrix Multiuser was launched, which was around 1990 / 1991.
- Shortly after Citrix launched Multiuser, Microsoft announced that they would drop OS/2 and move to Windows.
- With some help of other companies, Microsoft included, Citrix managed to stay in business.
- In the meantime Citrix patented ICA and they started working on a new and improved version of ICA.
- Eventually a new agreement was signed giving Microsoft access to the ICA source code. This is how the Microsoft RDP protocol came to exist.
- ICA supports most, if not all, standard protocols today.
- It uses TCI/IP port 1494 by default, and is tunnelled through port Nr. 2598 when Session Reliability is enabled.
- The ICA protocol consists of 32 virtual channels in total, 17 of which are reserved by Citrix.
- The client capabilities are negotiated at session launch time, also referred to as the handshake.
- Virtual channels consist of, and communicate through, virtual drivers at the client side and server-side applications on the server side.
- Customers and other third parties have the ability to develop their own virtual channels.
- Each virtual channel has a default priority assigned to it, ranging from 0 to 3, with 0 being the highest, or most important. A higher priority means more bandwidth.
- By editing the registry you can manually change priorities. Be careful with this, giving more priority to one VC means you also take away priority (bandwidth) somewhere else.
- Multi-Stream ICA works by assigning separate TCP/IP ports to groups of priorities, or streams, establishing true QoS.
- Session Reliability ensures that the user session is not disconnected and that the user's session freezes, while in the background the ICA traffic is buffered.
- All buffered ICA traffic will be flushed out to the user's device once the user session reconnects.
- Session Reliability can leverage the Auto client reconnect feature to enforce users to reauthenticate when a session is reconnected.
- HDX is an extension to the ICA protocol and is in no way intended to replace ICA. It works on top of the ICA protocol.
- The Citrix ThinWire technology had multiple names: it is known as ThinWire Plus, ThinWire Advanced, Legacy ThinWire and ThinWire compatibility mode. They all have

one thing in common: ThinWire is all about compressing data and enhancing the overall user experience.

- ThinWire has a small CPU and memory footprint and doesn't need much bandwidth.
- Framehawk is all about packet loss and high latency connections, delivering a more than acceptable user experience under challenging circumstances.
- In general, Framehawk needs more CPU and bandwidth than ThinWire, although this has been greatly enhanced with the latest 7.8 release.