



APPS & DESKTOPS

CITRIX TECH BYTES

NOV 30, 2017

HDX Adaptive Transport and EDT: ICA's New Default Transport Protocol (Part I)

**Fernando Klurfan**

9



*This post was co-authored by **Fernando Klurfan** and **Georgy Momchilov**.*

As members of the Citrix HDX team, we've been wanting, for a long time, to write a blog post about **HDX Adaptive Transport** and Enlightened Data Transport (EDT) ; ICA is our favorite protocol and these are some of its biggest improvements ever.

In this two-part series, we will clear up misconceptions around EDT and explain what are we doing, why we are doing it, and where are we headed.

In the XenApp & XenDesktop 7.16, HDX Adaptive Transport is set to "preferred," by default, so receivers will use EDT whenever possible. Therefore, we want to give you a thorough background on the enhancements. We always like to start with a little bit of history, because it is essential to understanding the present.

NOTE: *This is a companion to the **blog** written by **Derek Thorslund**, Director of Product Management for HDX.*

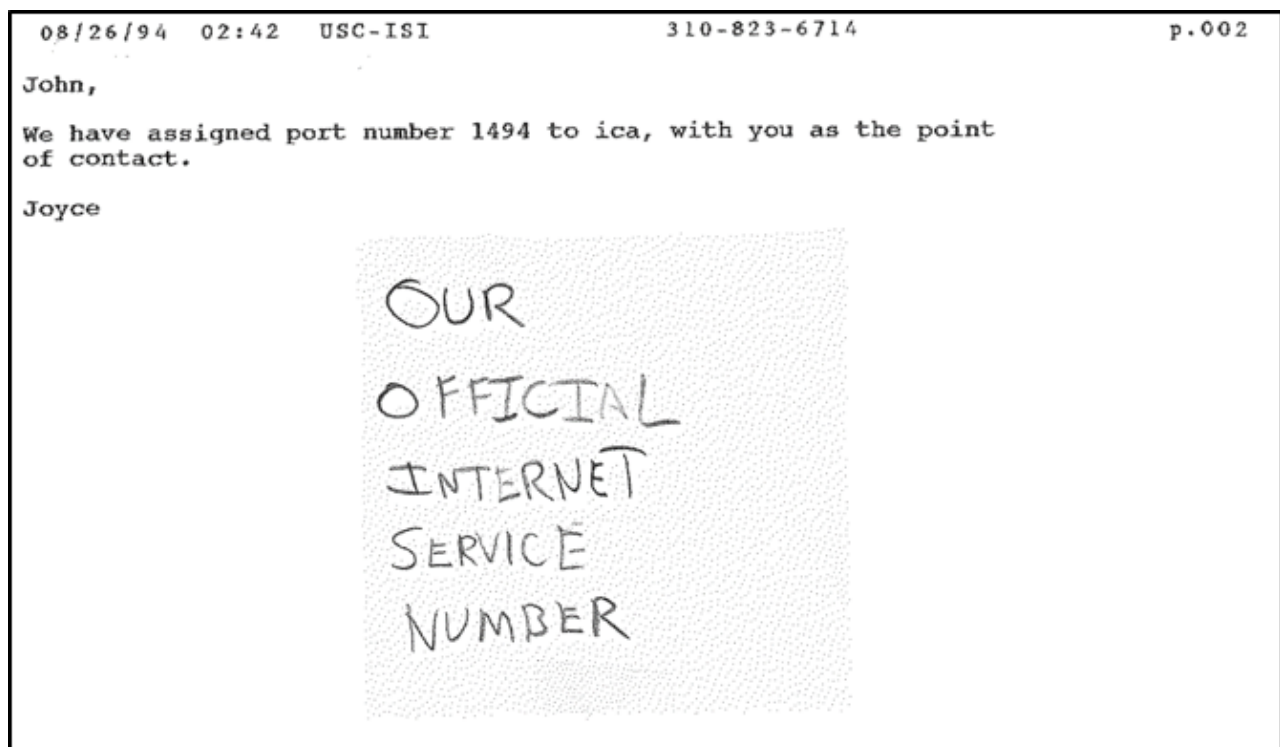
TCP: *de facto* transport protocol of the internet

If you are ~40 or older, then you know that TCP wasn't always king. I (Fernando) remember my first year in college (1997), where an old calculus lab was still based on a Novell farm using IPX/SPX as the transport layer protocol.

Georgy was in charge, in 1993, of setting up the computer farm in his college dorms, also running Novell.

In fact, TCP only started to take off when ARPANET (the precursor to the modern internet) switched to TCP/IP in 1983 (replacing the earlier NCP protocol).

When the ICA protocol was commercially launched (1991), it did not even support TCP until an engineer named Jeff Muir decided to give it a try in 1994 (make sure you read his [blog post](#) about it; it's a fascinating journey back in time).



The original 1494 TCP port assignment letter from USC-ISI/IANA to Citrix engineer John Richardson!

The internet boom changed the technology landscape for everyone, and the conservative mechanics of TCP made a lot of sense for an emerging "network of networks."

There are two main pillars to TCP: **flow control** and **congestion control** (with many honorary titles like selective acknowledgements, fast re-transmit, fast recovery, the whole RFC1323 — *e.g. window scaling* — and more).

Flow control, a window-based mechanism (*'rwnd'*) between sender and receiver, is designed to prevent the receiver from being overwhelmed by incoming packets. (A separate *rwnd* window is maintained independently by each receiving side and advertised in the TCP three-way handshake).

We would say **congestion control (and avoidance)** is the most difficult problem by far, and the main differentiator in all the TCP flavors (Tahoe, Reno, Vegas, New Reno, Compound TCP — default on Windows, BIC, CUBIC — default on Linux, experimental on Windows 10 1703, etc.). Its goal is to effectively utilize the network bandwidth and back off when congestion is detected. It's a window-based mechanism (*"cwnd"*) between the sender and the network, maintained by the sender, but never advertised in the three-way handshake.

The algorithm used in TCP for congestion control/avoidance is called AIMD (**A**dditive **I**ncrements of packet sending rate up to the estimated bandwidth, and **M**ultiplicative **D**ecrements to cope with loss), which unfortunately is inefficient over high speed WANs. TCP takes longer to build up to the available bandwidth, and each time a loss occurs it retracts drastically, resulting in TCP not saturating the available bandwidth.

These two inefficiencies are detrimental to any remoting technology with use cases ranging from bulk data transfer (file transfer or USB traffic) to interactivity with user input and display remoting (Thinwire).

UDP: the Null Protocol

But rewriting TCP is impractical both for technical and commercial reasons: Operative System kernels will take a long, long time to update their TCP stacks. So, we started looking at its cousin, the User Datagram Protocol (UDP).

UDP was added to the core internet protocol suite in 1980, right around the time the TCP/IP specifications were being split to become two separate RFCs. UDP, at its core, is the encapsulation of an IP packet into a datagram, with four additional fields: source port, destination port, length of packet and a checksum. While IP can establish a communication between hosts, UDP does it between processes or applications on those hosts. And that's pretty much it (hence the "null" reference).

UDP is not something new to ICA. Framehawk, RTP Audio, and the HDX Real-time Optimization Pack for Skype for Business all rely on it. For these scenarios, an unreliable "best effort" protocol works very well.

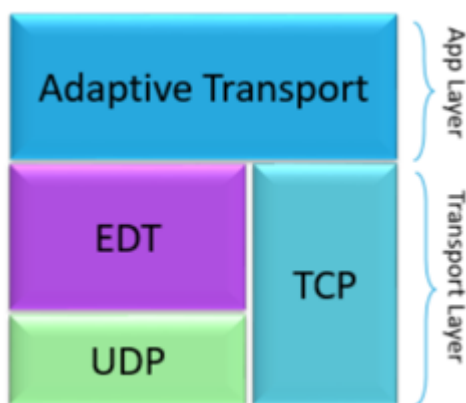
The best aspect of UDP is not what extra features it introduces, but rather **what features it omits** from TCP: no guaranteed delivery, no orderly delivery, and no congestion control.

Enlightened Data Transport: a reliable UDP solution

A challenge with our previous UDP solutions was that they were restricted to graphics and audio. So, Georgy, our HDX Architect, decided to devise a UDP-based reliable protocol solution that will unlock the value of UDP for all of HDX and serve as a solid foundation for future enhancements.

The main goals were to:

- Build a transparent transport layer for all HDX that works in all network scenarios
- Seamlessly seek UDP first with fallback to TCP, just like a hybrid car, which is electric first but falls back to gasoline
- Leverage custom congestion and flow control algorithms for performance far better than TCP on WAN and on par with TCP on LAN
- Improve performance for both interactive and bulk data transfer scenarios
- Achieve faster convergence to the network's bandwidth capacity
- Take into account modern networks where packet loss is mostly due to end-point interference (stochastic loss) and not congestion. Some packet loss can be tolerated without 'throughput panicking'
- Fair-share available bandwidth with other EDT or TCP streams
- Avoid affecting Single Server Scalability
- Avoid any impact to connection time



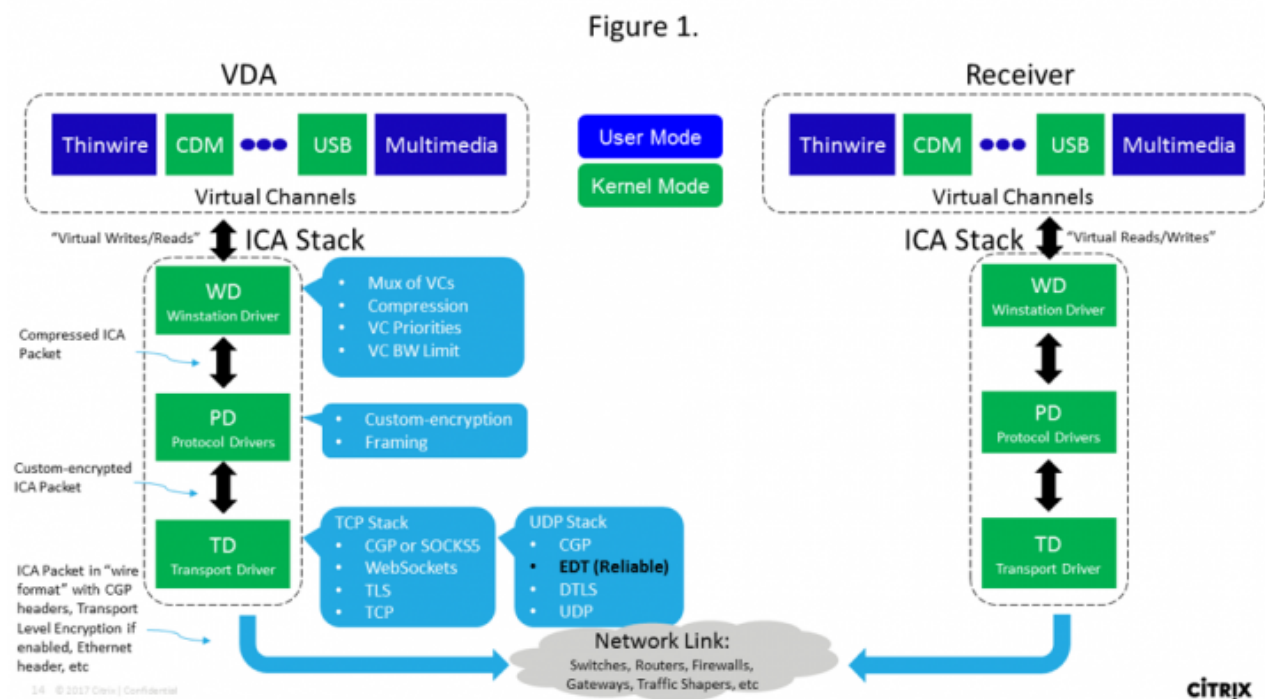
We introduced EDT in XenApp/XenDesktop 7.13 and have been improving it steadily ever since. In reality, we introduced HDX Adaptive Transport, which is a superset of EDT, orchestrating the **fallback** from EDT to TCP if the network does not allow UDP or if EDT fails for any reason.

The scenarios where HDX Adaptive Transport performs an evaluation of the use of UDP vs. TCP transport are:

- Initial connection
- Roaming from one Receiver device to another (same as initial connection)
- Session Reliability (SR) Reconnect and Auto Client Reconnect (ACR)
- High-availability failover from one Gateway instance to another
- Proactively seek UDP even when TCP transport is not broken (Q4 2017 Receivers only)

EDT and ICA: show me the Stack

A common misconception is that EDT is a Graphics protocol. The confusion might stem from the word “adaptive” in Adaptive Transport, which sounds similar to Adaptive Display, the latter being a suite of encoding technologies, selectively using H264 for transient regions of the screen and lossless encoding for text. (HDX is pixel-perfect where it matters). This is Thinwire+, our graphics virtual channel, the one in charge of display-remoting through encoding/decoding.

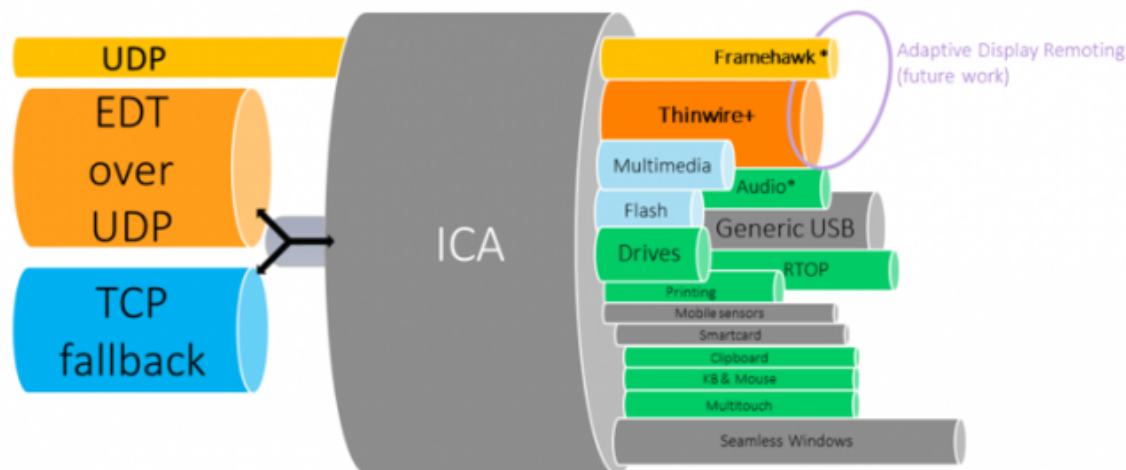


From Figure 1 above, you can clearly see that EDT is part of our Transport Driver, and it can be leveraged by every virtual channel: Thinwire, printing, CDM, multimedia, USB, and so on.

Exceptions: Framehawk (our “big guns” graphics solution for mobile workers on broadband wireless connections) has its own **UDP data transport layer** based on network gearing (ability to monitor network conditions and immediately react to changes), so it does not use EDT or TCP. In the future we plan on cross-pollinating Thinwire and Framehawk and converging on a single Display Remoting mode over a unified ICA stack with EDT.

Also, VoIP still performs better over **pure UDP/RTP Audio** (outside ICA). In the future we plan to merge real-time audio into the unified ICA stack with EDT.

HDX with Enlightened Data Transport



* Framehawk actually uses its own UDP data transport layer based on gearing
 * Audio still performs better when it runs on it's own RTP/UDP socket

2/4 © 2017 Citrix | Confidential

CITRIX

Key Takeaways

We would like to close this Part I by listing a few key takeaways.

- Is EDT the same as Adaptive Transport? Almost! Adaptive Transport = EDT (UDP) + TCP (Fallback).
- Is Adaptive Transport same as Adaptive Display? No, Adaptive Display is a Graphics stack suite.
- EDT is not designed to necessarily save bandwidth:
 - It is **not** a compression protocol nor a more efficient encoder.
 - It just gets the data where it needs to go faster (**it's a transport protocol**).
 - Might actually use more bandwidth if it is available (but this is good for interactivity and bulk data transfer speed).
- That said, EDT is fair to other streams: EDT, TCP, HTTP, etc.
- EDT works on a LAN too. It really shines on WAN: Up to 2.5x Interactivity depending on nature of the load and network conditions; Up to 10x faster file transfer.
- Framehawk is **not** EDT:
 - Framehawk is a Graphics stack, with a Receiver-side Intent Engine and a tightly coupled UDP-based transport featuring network gearing.

- EDT is just a transport protocol, available to every Virtual Channel (except Framehawk and RTP Audio today).

In [Part 2](#), we will roll up our sleeves and get into the details of the protocol and discuss a few configuration topics!

Follow [@CitrixFerK](#) Follow [@CitrixGeorgyM](#)

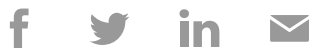
***Citrix TechBytes** – Created by Citrix Experts, made for Citrix Technologists! Learn from passionate Citrix Experts and gain technical insights into the latest Citrix Technologies.*

[Click here](#) for more TechBytes and [subscribe](#).

Want specific TechBytes? Let us know! tech-content-feedback@citrix.com

Citrix TechBytes

Created by Citrix Experts,
made for Citrix technologists!



Tagged under:

Adaptive Transport

Apps & Desktops

Citrix Tech Bytes

Citrix Virtual Apps & Desktops (XenApp & XenDesktop)

HDX



Fernando Klurfan

Principal Product Manager in HDX. Multimedia, Skype for Business, Teams.

9 Comments **Citrix Blogs****1 Login** ▾ **Recommend** 5 **Tweet** **Share****Sort by Best** ▾

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS **Mayunk Jain** • 2 years ago

Excellent recap of how technology comes full circle, Fernando. My own career started with this thing called Frame Relay, and for a brief period, we lived under the illusion that the bloated TCP-IP with a bloated 40 byte header will be DoA against the (arguably) better, lean-and-mean 2 byte header of Frame Relay. Now, I can't even remember if I flaunted those values correctly ;-)

YOU MIGHT BE INTERESTED IN

CITRIX TECH BYTES

NETWORKING

YESTERDAY

Citrix SD-WAN programming with Windows PowerShell – Part 1

Citrix SD-WAN has available a powerful NITRO API that enables configuration and monitoring of the product...



0

Stay Informed.
Subscribe Today.

email address

SUBSCRIBE

Yes, I'd like
to hear
about offers

and services from Citrix by email.

☐ Yes ☐ No

Citrix will process your data according to our [Privacy Policy](#)

APPS & DESKTOPS

CHANNEL & PARTNERS

2 DAYS AGO

Cloud market opportunities take center stage at Microsoft Inspire

Spotlighting new ways to grow your business, boost customer value, and drive revenue with Citrix-Microsoft...



Tricia Atchison

0

[APPS & DESKTOPS](#)[CITRIX WORKSPACE SERVICES](#)

JUL 17

Native clipboard experience is here for Citrix Workspace app for HTML5

Are you a user of the Citrix Workspace app for HTML5? Do you feel the existing clipboard experience — four...



Shruthi Udayakumar

0

[CITRIX.COM](#)[ABOUT](#)[CONTACT](#)[TERMS OF USE](#)[COOKIE PREFERENCES](#)

© 2019 Citrix. All rights reserved.

