



Delay-Tolerant Networking: a New Type of Space Cloud Network Architecture

Scott Burleigh
Jet Propulsion Laboratory
California Institute of Technology

16 May 2018

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. (c) 2018 California Institute of Technology. Government sponsorship acknowledged.



What's New About It?

- Unlike the Internet, DTN doesn't rely on brief communication round-trip times.
 - No conversational communication structures. Client/Server is mostly eliminated.
 - The End-to-End Principle is reconsidered.
 - It's still true that everything should be done only between the endpoints unless performing some network operations inside the network – between adjacent nodes of the end-to-end path – confers an important performance advantage.
 - In the environments that DTN is designed to tolerate, intra-network operations always confer important performance advantages.
 - So core network operations are performed at every node, as needed, not end-to-end.



Why?

- Communications based on brief communication round-trip times will not perform well...
 - ...if transmission between nodes is very time-consuming (large distances, slow signal propagation rates), introducing point-to-point delay, or...
 - ...if transmission between nodes is frequently but temporarily interrupted (occlusion, occultation, disruption), introducing round-trip delay.



Ambient Connectivity

- With Internet architecture, you have got network connectivity whenever you have continuous end-to-end data flow.
 - If any link anywhere along the path is disrupted, you are no longer connected.
- With DTN architecture, you have got network connectivity whenever you are connected to any other node.
 - If all links along the end-to-end path are functioning, great!
 - If not, then communication will take a little longer, but your message will still get through eventually as the link(s) are restored.



Satellite Cloud

- A satellite-based Cloud will be wireless, so all links (whether high-rate R/F or optical) will be subject to noise, fading, weather interference, jamming, etc.
- When links are disrupted, contemporaneous round-trip communication is lost.
- But DTN simply waits for the next link to come back to life, then uses it, one after the next. It's patient and relentless.



What About When All is Fine?

- The core network protocol of the DTN architecture is Bundle Protocol (BP).
- BP is an overlay protocol, just like the Internet Protocol (IP).
 - Just as IP runs over subnetwork protocols (Ethernet, 802.11, SONET) to interconnect subnetworks, BP runs over network protocols (IP, 5G) to interconnect delay-intolerant networks.
- When the underlying networks are running without any problems, BP is just a container for application data, adding a small increment of overhead.
- Whenever there's a failure in the underlying network, BP automatically helps out until everything is great again.



So, Ditch the Internet?

- Of course not. The Internet is terrific.
- This is not either/or. Because BP is an overlay protocol, it never replaces IP; it runs on top of IP, carried by TCP/IP traffic. (And on top of other networks or link protocols as well, where needed.)
- When there's a TCP/IP failure for some reason, BP recovers from the failure automatically.



We Already Do That

- Certainly, many organizations have had to develop mechanisms to deal with the same kinds of failures that DTN is designed for. This is not rocket science.
- But DTN was designed from the outset to support an open-ended set of operating scenarios in a single standard way.
 - It's already a published standard of the Consultative Committee for Space Data Systems and is being broadly deployed by NASA.
 - It's currently being standardized by the Internet Engineering Task Force as well.
 - It's being tested by multiple national space agencies, will be a vehicle for secure international interoperability.



And the Catch is...

- There is some complexity in some of the DTN mechanisms, especially routing. But the implementations of DTN insulate applications from this complexity.
- Storage needs to be provided at nodes, so that bundles can be securely retained while waiting for an outbound link to be established.
- Your applications need to be delay-tolerant too.



Delay-Tolerant Applications

- Actually, the most heavily used Internet applications are already innately delay-tolerant:
 - Email.
 - Social media: you post something, and later on someone reads it.
 - Messaging: if it's instant, great, but if nobody is paying attention right now the message will be read a little later.
 - Announcements, breaking news items.
 - Alarms, distress calls – you really want someone to respond immediately, but if no one gets the message right now you still want them to respond as soon as possible.
 - Even streaming audio and video are delay-tolerant, to the extent that the user is delay-tolerant.
- But nothing conversational. DTN is no good for VOIP.



A Special Focus: Multicast

- In DTN, retransmission of lost data is performed between adjacent nodes in the path – not end-to-end.
- So every arc of the multicast tree is independently reliable.
- So the aggregate multicast is reliable at every destination node. No extraordinary data accounting needed at the source.



A Multicast Application: DTKA

- Delay-Tolerant Key Administration (DTKA) is a delay-tolerant public key infrastructure.
 - User nodes generate key pairs locally, never disclose their private keys anywhere, and multicast their public keys to the lobes of a distributed key authority (KA).
 - The KA lobes periodically reach consensus on a key update bulletin and cooperatively multicast the bulletin to user nodes.
 - The design of the KA ensures that there is no single point of failure, no single point of authority, no single point of veto.
 - Compromise of one or more KA lobes is no problem, and none of the traffic need be encrypted.
- The same structure works for any source of information that is critical but not secret.



DTN Security

- Security was built into the DTN architecture from day 1.
 - Individual blocks of a bundle can be individually signed and/or encrypted in any number of different keys.
 - So bundles are automatically secured at rest as well as in transit.
 - A symmetric key – signed and/or encrypted – can be included in the bundle and asymmetrically verified and/or decrypted.
 - The source and destination in the primary block can be asymmetrically signed, ensuring authenticity.
 - So data can safely be forwarded by untrusted nodes.
 - For defense against traffic analysis, the entire bundle can form the encrypted payload of an encapsulating bundle whose source and destination need not be secret.



Information-Centric Networking

- Information-centric networking (ICN) – then called “self-forming Akamai” – was a goal of the original DARPA DTN project.
- DTN-based ICN is yet another ICN architecture, which couldn’t be realized until BP multicast was available – and still will require substantial work to implement. But it seems to offer some advantages:
 - No namespace issues.
 - Computation is widely distributed.
 - User privacy is protected.
 - All transmission is reliable.



It's Ready Now

- DTN is not on the drawing board. It's running on the International Space Station today.
- It's compatible with all the protocols you already use.
- It's going to be an Internet standard.
- And it's free.



Thanks!

Questions?

