A Brief Introduction to Delay Tolerant Networking

Frank Brickle, AB2KT
ab2kt@arrl.net

ARRL/TAPR DCC 20008

Abstract

Delay Tolerant Networking is the art and science of moving bits around in bad situations, such as between distant endpoints in deep space, or among unknown mobile users at uncertain positions. We give a brief introduction to the technology that has been developed to provide network connectivity in such situations.

1 History and Overview

Since the beginning of digital networking, the issue of how to accommodate migrant users has been the subject of much discussion and addressed by many ad hoc solutions. In the early 1990's, Vint Cerf, one of the most prominent Internet pioneers, turned his attention to the problem of providing connectivity to extreme cases of migrant usage, namely, vehicles and travelers in deep space. The idea was to define a simple, comprehensive solution for passing bits to and fro through deep space that would still be cheap, flexible, and robust.

The results of this work were formalized in a pair of Internet Engineering Task Force documents, RFC 4838 and RFC 5050, published in 2007. Rigorous definitions of the problem are thus quite recent, as are also some comprehensive software packages that realize the proposed solutions.

1.1 Not Just Space

It is immediately clear that the networking methodology for Deep Space applies equally well to the problem of achieving connectivity in far more mundane scenarios. To rephrase the problem, Delay Tolerant Networking is the art and science of moving bits around in bad situations of any sort.

Difficult conditions obtain, for example, between distant endpoints in deep space; or into and out of disaster areas; or among mobile user passing through uncertain access points; or among sites that need to be kept covert. DTN provides ways of sustaining digital links in such scenarios, where connection paths are usually intermittent, unpredictable, or unreliable.

1.2 Amateur Radio Applications

DTN technology can have an immediate home in several time-honored amateur radio activities.

1.2.1 Satellites

In particular, DTN is a prime candidate for implementing a sophisticated mobile point-to-point messaging system via amateur satellite, where only one layer, the convergence layer, needs to be realized through gateways to the satellite uplinks and downlinks, with the remainder being handled as ordinary TCP or UDP/IP traffic.

1.2.2 EMCOMM

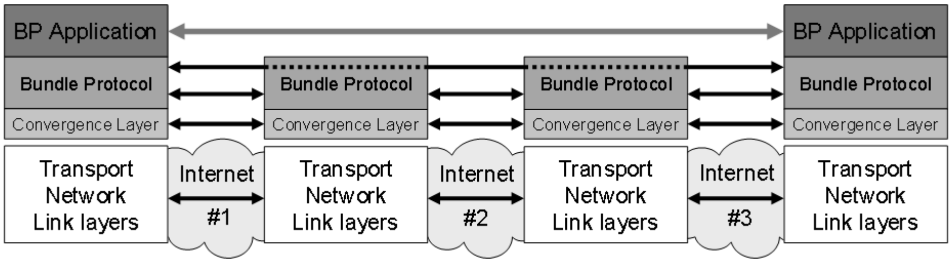
DTN is also capable of providing the digital backbone for a messaging system supporting disaster relief. It is especially well-suited for situations where resources are scarce or unreliable for running communications nodes, and where they might need to be moved frequently and unpredictably.

1.3 Ready Application Development

There are several well-developed and comprehensive systems in already in existence to provide DTN capabilities at the software level. In the following section we give a high-level overview of the technology. Later we offer pointers to commonly-available resources for information and software capabilities that are already can be had by amateur developers. Finally, in a forthcoming paper, we also describe some ready-to-hand potential applications, and outline some experiments at exploitation that have already been carried out or are currently in preparation.

2 The Major DTN Components

The overall structure of DTN is extremely simple, which in turn requires only modest complexity in the protocol that keeps the structure in motion.



2.1 Bundles

The central idea of the IETF proposals is an enhancement called the *bundle*, which operates at the higher levels of network protocol. The general scenario is this. You have a chunk of data which you want to pass along to a specific recipient. Senders and recipients are known by their *Endpoint Identifiers* or *EID*s.

What you don't know is how to get the chunk of data from you to the recipient. So what you do is package up your data along with various other pieces of information about you, the recipient, and other things, according to an agreed-upon set of rules. That's the bundle: your data + metadata + broad routing info.

2.1.1 Bundle Protocols

One of the important features of the "other things, according to agreed-upon rules" is that they're general enough so that the bundle can be passed over (effectively) any sort of transport, by unknown agents, hand-to-hand and step-by-step until it gets to its recipient. The aggregate rules for packaging up data and pasting on an address constitute the *bundle protocol*. Of necessity the protocol is simple but capable of supporting necessary features such as authentication, privacy, priority, and expiration dating.

2.1.2 Bundle Agents

The critical players along the way are called bundle *transfer agents*. The transfer agents all know the rules, and they also know a few things about how to keep bundles moving. What the agents know are, basically, two things:

1. A list of the other bundle agents they've had contact with recently, and
2. A practical schedule for blasting out copies of your bundle to those agents, with the schedule capable of being tuned for maximum efficiency according to defined criteria.

Critically, the bundle agents also are capable of using different kinds of transport depending on the beighbor agents they need to contact, although for the contact transactions themselves they depend on another protocol layer.

3 Bundling Protocols

3.1 Routing and Heuristics

Traditional protocols will normally try to establish a route before moving data. This is a provably hard problem all by itself. A DTN is further complicated by the property that it might have no discernible instantaneous end-to-end paths. However, it can take advantage of intermittent or opportunistic connections. Therefore a DTN agent will be on the lookout constantly for one or more good opportunities to advance your data, ready to take them when they appear. Most of the routing protocol is devoted to computing which is the best bet among a number of possible opportunistic events.

3.1.1 End-to-end Paths

What can wind up happening to your bundle is that it gets passed from agent to agent over a wide variety of point-to-point links, some of them wireless, some of them terrestrial and wired, and so on, as transient opportunities dictate. The extra information bundled with your payload is sufficient to tell any agents along the way how to keep it moving forward towards the recipient.

The path between two endpoints might never happen twice in exactly the same way, since bundle agents can move in and out of the neighborhoods of other agents. All you know is that the information you've bundled with your data, and the conforming behavior of the agents, will promise a best effort at moving your payload to your intended recipient.

3.1.2 Heuristics

Since the arrival of transfer opportunities in a DTN will be nondeterministic, routing will be suboptimal. Indeed, it can be shown to be arbitrarily far from optimal. Even under the best of circumstances, optimal routing is NP-hard. Thus at its core, DTN routing will necessarily rely on heuristics – that is, it consists of a probabilistic strategy that rests on sending multiple copies of data via different potential opportunistic paths. Thus, when routing a bundle, a DTN agent will "look ahead" in a breadth-first fashion and develop a preferential list of nearby agents to whom it can pass off the bundle for further transport to the goal. Factored into this computation will be the competing goals of

- getting throughput – that is, guaranteeing delivery of as much data as possible,
- avoiding logjams – giving fair treatment to new data when old data is undelivered and backed up,
- don't thrash the system – avoiding network congestion.

There are several approaches to mediating among these requirements, but they all rest on the fundamental principle of spraying a limited number of copies of your bundle into the local neighborhood of bundle agents. Where they disagree is primarily in their philosophies of limiting the number of copies.

3.2 Transport

DTN makes educated guesses about the right steps to move data forward. What it doesn't do is stipulate how, physically, the bits get moved.

3.2.1 The Convergence Layer

From the standpoint of traditional network layering, the transport mechanism is far below the bundle agents, down towards the cold metal of the network stack. Nevertheless the bundling protocol needs to be able to adopt a unified view of all the transport possibilities operating below it in the stack. From the standpoint of the bundling agents, then, the various transport capabilities are themselves presented as a single *convergence layer*.

4 A Successful Experiment

In 2006-2007, Darren Long G0HWW and John Ronan EI7IG reported success in transferring email between ordinary Linux clients exposed to the internet, using DTN bundling with AX.25 at the convergence layer. We have been unable to find any followup to this work, apart from a synpathetic critique of AX.25 as convergence layer on a set of webpages by G0HWW which, unfortunately, survive only in Google caches.

While this experiment appears to have been limited in scope, its significance should not be missed. To carry it out, the developers made minimal changes to existing TCP/IP code in a publicly available DTN implementation. There is clearly room for a lot more experimentation based on amateur radio infrastructure already in place. More information on AX.25 and DTN can be found at `http://theronans.com/j0n/?p=260`

5 Conclusion

It should be pretty clear at this point that DTN amounts to a high-octane generalization of the familiar *store-and-forward* scenario. Since data moving across a DTN can traverse many different kinds of underlying local networks, it itself should be thought of as an *overlay* network resting on opportunistic contacts with single, diverse network nodes.

There are a number of reasons why DTN is attractive. Among them:

1. Somebody's already "thought big" about and has then devised a usable solution for a wide swath of our mobile-, satellite- and EMCOMM-related problems.
2. The protocol is (or at least can be) used effectively in a way that's completely transparent to a wide range of existing applications.
3. The capabilities make it possible to extend existing applications, as-is, into novel areas.
4. There's enough decent Open software around even at this time to start looking at applying it to projects of interest to us.

If you're looking for further information, source code, and more, the place to start is the Delay Tolerant Networking Research Group homepage, `http://www.dtnrg.org/wiki.`