DTN Gateway Architecture for Partially Disconnected Telemetry Environments

Justin P. Rohrer and Geoffrey G. Xie
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
{jprohrer,xie}@nps.edu

ABSTRACT

Telemetry networks often operate in challenged wireless environments, resulting in periods of disconnection. Our delay tolerant networking (DTN) gateway dynamically detects disruptions in connectivity and buffers telemetry data until connectivity is reestablished. When the connection is resumed, all buffered data is transmitted automatically in order to backfill any gaps in the telemetry stream. A DTN gateway may operate as a standalone device with multiple DTN client applications, or as a network of mobile DTN gateways which will perform multi-hop ad-hoc routing to deliver telemetry data across the telemetry network system (TmNS). Our DTN gateway also provides conventional IP routing and forwarding capabilities, including support for standard dynamic routing protocols, eliminating the need for a stand-alone IP router on the test article (TA). This paper presents the system architecture of our DTN gateway, along with several deployment scenarios for telemetry environments.

I. INTRODUCTION AND MOTIVATION

This paper presents the architecture of the Naval Postgraduate School (NPS) delay tolerant networking (DTN) gateway, which provides converged IP and DTN routing capabilities in a low-power, low-cost, portable platform. DTN routing is designed to mitigate the effects of disruptions in connectivity that would terminate conventional IP connections. Our gateways may operate as standalone devices, or as networks of multi-hop ad-hoc routing nodes.

The NPS DTN Gateway (NDG) leverages previous work in disruption-tolerant networking, as well as open-source Linux-based IP routing software, and commercial off-the-shelf (COTS) hardware components. Through a seamless integration of IP and DTN routing functionality, the NDG design eliminates the need for a standalone IP router, and more importantly, the additional complexity of coalescing two independently configured devices and the associated performance penalty. Another major advantage of this design is that it requires no change to user application for them to leverage DTN routing when IP routing is not feasible.

The remainder of this paper is organized as follows: Section II gives the background of DTN protocols. Section III discusses the hardware components used to build the NPS DTN gateway. Section IV describes the software components used to combine DTN and conventional IP functionality in a single platform. Section V presents preliminary performance specifications recorded in our testbed environment. Section VI describes the type of network scenario for which the DTN gateway is optimized. Section VII gives an overview of our development schedule, and Section VIII concludes and describes future directions for this work.

II. Background and Related Work

Delay- and disruption-tolerant networking (both referred to as DTN) is designed to minimize the impact of intermittent communication problems, as well as environmental limitations and anomalies.

A. Challenged Network Types

Networks which are prone to experience disruptions are commonly referred to as Challenged Networks. The standard suite of Internet protocols assume that a stable end-to-end (E2E) path exists, that the maximum round trip time is not excessive, and that the packet drop probability is small. We can categorize networks that do not have these properties as follows:

- Highly-Mobile Networks: At best these networks experience frequent route changes. They can also become partitioned unexpectedly, and in some cases an E2E path may never exist.
- Exotic Media Networks: These include satellite communications, deep space RF links, acoustic modulation (used underwater [1]), and line-of-sight (LOS) high-frequency radio or optical links. Networks using these kinds of links can experience very high RTTs, or outages due to environmental conditions.
- Military Ad-Hoc Networks: Military networks are required to operate under hostile conditions in which enemy jamming can cause interruption, and the threat of eavesdropping may trigger periods of radio silence.
- Sensor Networks: Sensor networks often have very limited resources in terms of power and transceiver range. This can result in frequent link disruptions, as well networks which are subject to partitioning.

Airborne telemetry networks tend to fall into the highly-mobile category, as well as having power and weight constraints similar to sensor networks. A wide range of approaches has been developed, from modifying traditional IP-based protocols to be more tolerant of disruption and delay, to new architectures that operate as application overlays. One of the latter approaches to building DTNs in known as the bundling protocol architecture.

B. Bundling Protocols

Delay tolerant protocols have been developed for a variety of applications [2]. In this section we discuss the history of just some of the most prevalent protocols that fall into this category.

1) Interplanetary Networks: Interplanetary Networking (IPN) presents environmental challenges that are orders of magnitude larger than those found in terrestrial networks due to the speed-of-light delay [3], [4], [5]. Interplanetary systems do have the advantage that the delays are known very exactly due to the predictable motions of the planets. Eventually it was realized that IPNs are a subset of the broader category Delay-Tolerant Networks, and that the work had terrestrial applications as well [6]. For purposes of experimentation this is also a very useful fact since it is far easier and cost efective to run networking experiments on Earth than it is to run them in space.

2) Delay-Tolerant Networking Research Group: The IETF delay-tolerant networking research group (DTNRG) protocols are largely a continuation of the work started in the IPN project, but extend the concepts to include networks with unpredictable round-trip times caused by a variety of challenges in addition to speed-of-light delays [7], [8]. The DTNRG developed two main protocols, the Bundle Protocol [9] and the Licklider Transmission Protocol (LTP) [10]. The Bundle Protocol is a overlay store-and-forward network that sends packages of application data over a wide range of underlying network types using a sequence of gateways that serve as nodes in the overlay network. This represents the mainstream approach within the DTNRG group. A prominent example implementation of the bundling protocol is the SPINDLE 3 system developed by BBN [11], which we are evaluating for use in the NPS DTN gateway. Several other DTN implementations were recently compared by Pöttner et. al. in [12]. LTP is a point-to-point protocol that deals with individual long delay links by freezing timers that would otherwise expire before an acknowledgement was received. It relies on a lower layer scheduler to tell it exactly when and how much to transmit. Because it is only designed for dedicated point-to-point links LTP does not handle congestion or routing issues [13].

C. Non-IP Protocols

An alternative to using native IP or application-layer overlays in the telemetry network environment, is to translate telemetry data into a custom protocol stack designed for highly dynamic environments. A recent approach using this method is the ANTP suite [14], [15], which is composed of the AeroTP transport layer [16], [17], the AeroNP network layer [18], and the AeroRP routing layer [19]. Translation between these and traditional IP protocols is performed by the AeroGW gateway [20].

Developing non-IP protocols is a long-term approach to the problem that has benefits in reducing overhead associated with IP, as well as improving cross-layer information sharing. The downside is that retrofitting a network designed around IP-based protocols to use another network layer is difficult and costly, and for these reasons not suitable for use in the NPG.

Fig. 1. NPS DTN gateway

III. Hardware Components

The NPS DTN gateway hardware is based on the AMD Brazos platform, chosen for it high I/O-bandwidth capability, low cooling requirements, and high performance vs. cost efficiency (more details on this in Section V). The Zacate CPU is a dual-core package fabricated using a 40 nm process, and running at 1.6 GHz. It communicates with other onboard components using a high-bandwidth UMI interface. System storage and DTN buffering are provided using high-speed synchronous flash, accessed via a 6.0 Gbit/s serial ATA interconnect. In addition to the onboard gigabit ethernet interface, we provide four additional routable gigabit interfaces that are interconnected to the CPU using a four PCI-express 2.0 channels (20 Gbit/s aggregate). Eight gigabytes of dynamic random access memory operate at 1.33 GHz. The entire system is enclosed in a steel chassis 8.7" wide, 12.9" deep, and 3.8" tall, as shown in Figure 1. Power consumption is approximately 35 W under typical load, which is low enough that active cooling fans are not required under most circumstances.

IV. Software Components

The software components of the NPS DTN gateway consist of a Linux-based software IP router, a DTN bundling protocol agent, and an integration suite that intelligently determines when messages should be sent using one mechanism or the other.

The version of Linux used is derived from the Debian distribution [21]. We are currently running version 2.6.37 of the kernel, but are evaluating updated versions for use in the gateway as they mature.

A. IP Routing

The Linux-based software IP router utilizes the Quagga [22] routing implementation to support major dynamic routing standards including OSPFv2 [23], OSPFv3 [24], RIPv2 [25], RIPng [26], and BGPv4 [27]. Of these OSPFv3, RIPng, and BGPv4 include support for IPv6.

This platform provides high-speed, stable IP packet forwarding through the network as long as coherent end-to-end paths exist.

B. DTN Routing

The DTN bundling component operates on the principle of custody transfer to provide reliability, instead of the end-to-end reliability typically provided by TCP. A DTN application registers itself as an endpoint with a DTN router, either running on the same host, or on a separate gateway communicating over standard IP protocols. Messages are passed from the application to the gateway, at which point the gateway assumes responsibility for delivery to the final destination and the application flushes the messages from its buffer. The DTN gateway must then determine a next-hop, which could consist of a destination application registered to the same gateway, or another DTN gateway to which the destination application is registered, or another DTN gateway that will serve as an intermediate hop. The difference between this and conventional MANET routing is that the DTN nodes are only intermittently connected, so a message may remain buffered for a significant interval of time before a connection to the next hop becomes available.

The bundling agent provides a plug-in interface for routing modules, allowing the routing protocol to be selected based on the characteristics of the network topology. A number of DTN routing protocols have been proposed recently, including MaxProp [28], PRoPHETv2 [29], and

RAPID [30]. Each of these uses an algorithm to learn the connectivity patterns established over time, and attempts to correctly predict the best next hop for messages based on these patterns.

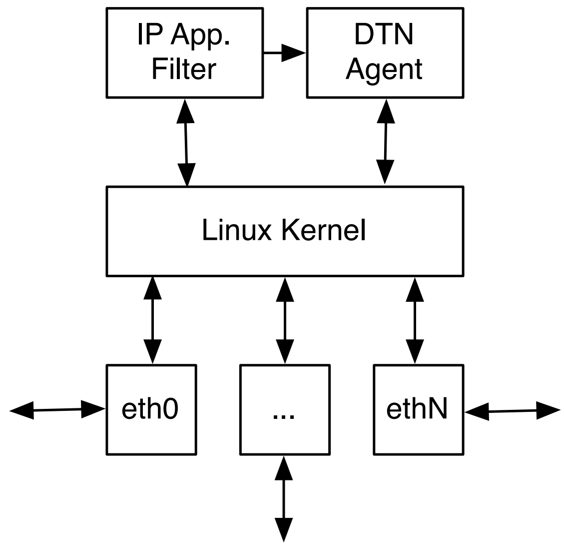

Fig. 2. NPS Gateway architectural components

C. Integration

At a basic level, the NPS DTN gateway consolidates two devices (a traditional IP router, and a DTN server) into a single device that compares favorable in terms of cost, size, weight, power, and performance with either of the two devices it replaces. However, in order to provide a truly integrated service more is required. To this end we are integrating the IP and DTN routing functionalities such that IP traffic may be buffered and forwarded by the DTN service in cases where a coherent end-to-end IP path does not exists and the connections would otherwise have failed without the intervention of the DTN service. The mechanism for this is an IP application filter (shown in Figure 2) that intervenes before packets are dropped by the Linux kernel due to lack of an IP route to the destination. The filter first checks to see if the packet belongs to an application that will tolerate the expected delays of the DTN network. If not the packet is returned to the kernel to be dropped with standard ICMP response. If the packet does belong to an application that is expected to tolerate some delay, it is inserted into a DTN bundle and passed to the DTN bundle agent to be buffered and forwarded according to the DTN routing semantics.

This approach will not be applicable to all traffic categories, for example real-time voice and video application data cannot be buffered and forwarded for later delivery, however for others such as periodic telemetry data readings a small delay in delivery is greatly preferable to loosing the data permanently.

Compared with using only DTN-aware applications, our approach has the advantage of not requiring every IP-based application to be rewritten to support communication via a bundling protocol agent as described in Section IV-B.

V. Performance

As mentioned in Section III, the device supporting these functions is relatively small, consumes little power, and requires little cooling. Naturally these characteristics lead to questions concerning the capabilities of the system.

In our preliminary testing we found the system to be capable of simultaneously routing two flows of 800 Mb/s each, effectively saturating the unidirectional capacity of 4 of the 1 Gb/s interfaces. This represents a routed traffic load of over 120,000 pkts/s. We used OSPF for route discovery and enabled the DTN bundling agent during these tests. While under this traffic load we observed that the average system load remained below 2%, and system memory usage remained below 150 MBytes out of the available 8192 MBytes.

We will be continuing to increase the number of clients and DTN nodes in our testing environment, resulting in both additional routes and traffic flows, but these preliminary performance results indicate an ability to scale networks of our DTN gateways well beyond the current intended deployment size.

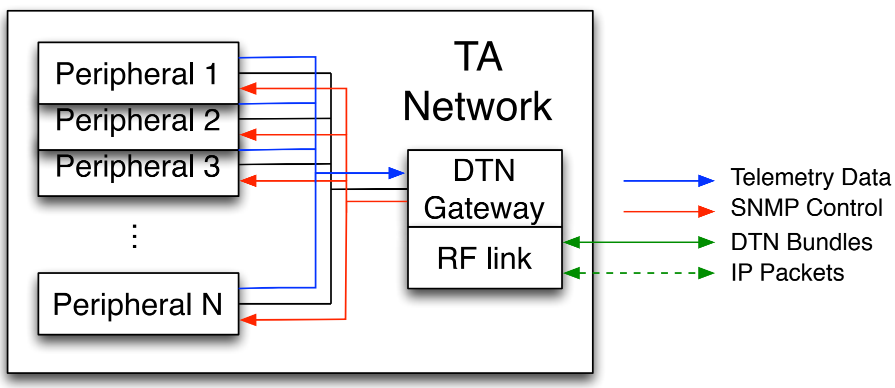

Fig. 3. Test article network overview

VI. Use Case Scenarios

The NPS DTN gateway is designed to replace conventional IP routers in vehicle and ground networks (vNETs and gNETs). When test articles antennas are within line-of-sight (LOS) of the ground station antenna conventional IP routing is used to forward packets, however if the connection to the ground station is temporarily lost the DTN behavior will automatically buffer packets, as well as search for multihop alternatives to the direct ground station connection, as illustrated in Figure 4. There are many environmental conditions that can result in a temporary outage of the TA to GS link, including terrain and areal maneuvers in which part of the aircraft structure interrupts LOS between the antennas. When either a multihop option is found, or the direct connection is restored, the DTN agent on the vNET forwards its stored bundles to the DTN agent at the next hop. When these bundles reach the gNET, the DTN agent there forwards them to the IP application filter, which in turn unpacks them and passes the IP packets to the Linux kernel to be forwarded to the destination.

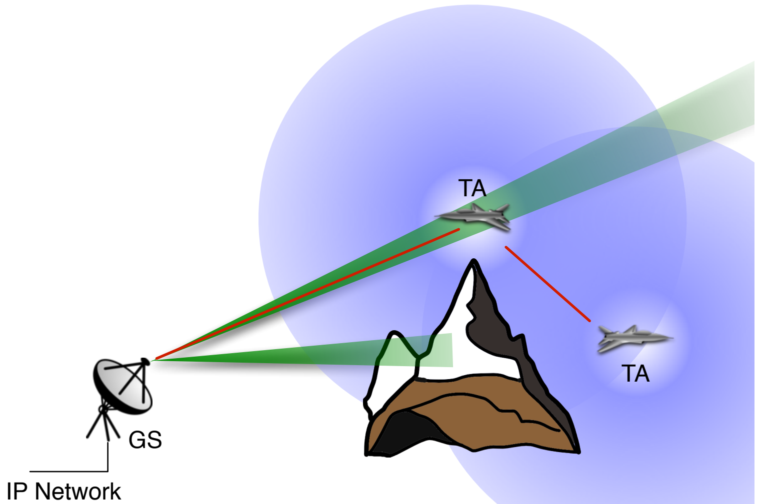

Fig. 4. Obstructed signal scenatrio

VII. Development Timeline

The underlying DTN protocols implemented by our gateway have been under development since the 1990s with considerable investment from both DARPA and NASA making the relatively mature at this time. Our concept of creating a unified IP/DTN gateway came together during the winter and early spring of this year. As of this writing we are operating several NPS DTN gateways in a testbed environment, and continuing to develop benchmarking tools to examine the converged IP and DTN capabilities. The next phase, anticipated to take place in later summer, will be to field test the gateways by mounting them on vehicles that are interconnected with a variety of radio networks as described in Section VI.

VIII. Conclusions and Future Work

In this paper we have presented an overview of our design architecture for implementing the NPS DTN Gateway. At a basic level, the NPS DTN gateway consolidates two devices (a traditional IP router, and a DTN server) into a single device that compares favorable in terms of cost, size, weight, power, and performance with either of the two devices it replaces.

As we continue this project, we will continue to test more complex and advanced scenarios, and assess IP applications' ability to tolerate delay in these scenarios, when there data is being transparently forwarded by the DTN agents.

Acknowledgments

The authors would like to thank the US Marine Corps for their support. This work was funded in part by the Marine Corps Systems Command, Program Manager (PM) MAGTF Command, Control, and Communications (MC3). We would like to thank USMC Capt. Josh Dixon for his involvement in discussions leading to this work.

REFERENCES

[1] J. Partan, J. Kurose, and B. N. Levine, “A survey of practical issues in underwater networks,” in *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, (New York, NY, USA), pp. 17–24, ACM, 2006.

[2] D. Waitzman, “Standard for the transmission of IP datagrams on avian carriers.” RFC 1149 (Experimental), Apr. 1990. Updated by RFC 2549.

[3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, and H. Weiss, “Interplanetary internet (IPN): Architectural definition,” internet draft, work in progress, IETF, May 2001.

[4] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, “Delay-tolerant networking: an approach to interplanetary internet,” *IEEE Communications Magazine*, vol. 41, pp. 128–136, June 2003.

[5] R. C. Durst, G. J. Miller, and E. J. Travis, “TCP extensions for space communications,” in *MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 15–26, ACM Press, November 1996.

[6] I. F. Akyildiz, Ö. B. Akan, C. Chen, J. Fang, and W. Su, “Interplanetary internet: state-of-the-art and research challenges,” *Comput. Networks*, vol. 43, no. 2, pp. 75–112, 2003.

[7] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 27–34, ACM, 2003.

[8] F. Warthman, “Delay-tolerant networks (dtns): A tutorial.” March 2003.

[9] K. Scott and S. Burleigh, “Bundle Protocol Specification.” RFC 5050 (Experimental), Nov. 2007.

[10] M. Ramadas, S. Burleigh, and S. Farrell, “Licklider transmission protocol–specification,” Internet draft, Experimental draft-irtf-dtnrg-ltp, IETF, January 2008.

[11] R. Krishnan, P. Basu, J. M. Mikkelson, C. Small, R. Ramanathan, D. W. Brown, J. R. Burgess, A. L. Caro, M. Condell, N. C. Goffee, R. R. Hain, R. E. Hansen, C. E. Jones, V. Kawadia, D. P. Mankins, B. I. Schwartz, W. T. Strayer, J. W. Ward, D. P. Wiggins, and S. H. Polit, “The SPINDLE disruption-tolerant networking system,” in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, (Orlando, FL, USA), pp. 1–7, October 29–31 2007.

[12] W.-B. Pöttner, J. Morgenroth, S. Schildt, and L. Wolf, “Performance comparison of DTN bundle protocol implementations,” in *Proceedings of the 6th Proceedings of the 6th ACM workshop on Challenged networks (CHANTS)*, pp. 61–64, September 23 2011.

[13] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, “When TCP breaks: Delay- and disruption- tolerant networking,” *IEEE Internet Computing*, vol. 10, no. 4, pp. 72–78, July-Aug. 2006.

[14] J. P. Rohrer, A. Jabbar, E. Perrins, and J. P. G. Sterbenz, "Cross-layer architectural framework for highly-mobile multihop airborne telemetry networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, (San Diego, CA, USA), pp. 1–9, November 2008.

[15] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, E. Perrins, and J. P. Sterbenz, "Highly-dynamic cross-layered aeronautical network architecture," *IEEE Transactions on Aerospace and Electronic Systems (TAES)*, vol. 47, pp. 2742–2765, October 2011.

[16] J. P. Rohrer, E. Perrins, and J. P. G. Sterbenz, "End-to-end disruption-tolerant transport protocol issues and design for airborne telemetry networks," in *Proceedings of the International Telemetering Conference*, (San Diego, CA), October 27–30 2008.

[17] J. P. Rohrer and J. P. G. Sterbenz, "Performance and disruption tolerance of transport protocols for airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (Las Vegas, NV), October 2009.

[18] A. Jabbar, E. Perrins, and J. P. G. Sterbenz, "A cross-layered protocol architecture for highly-dynamic multihop airborne telemetry networks," in *Proceedings of the International Telemetering Conference (ITC)*, (San Diego, CA), October 27–30 2008.

[19] A. Jabbar and J. P. G. Sterbenz, "AeroRP: A geolocation assisted aeronautical routing protocol for highly dynamic telemetry environments," in *Proceedings of the International Telemetering Conference (ITC)*, (Las Vegas, NV), October 2009.

[20] E. K. Çetinkaya and J. P. G. Sterbenz, "Aeronautical gateways: Supporting TCP/IP-based devices and applications over modern telemetry networks," in *Proceedings of the International Telemetering Conference*, (Las Vegas, NV), October 26–29 2009.

[21] "Debian linux distribution." http://www.debian.org/, June 2012.

[22] "Quagga routing suite." http://www.quagga.net/, December 2009.

[23] J. Moy, "OSPF Version 2." RFC 2328 (Standard), Apr. 1998. Updated by RFC 5709.

[24] R. Coltun, D. Ferguson, J. Moy, and A. Lindem, "OSPF for IPv6." RFC 5340 (Proposed Standard), July 2008.

[25] G. Malkin, "RIP Version 2." RFC 2453 (Standard), Nov. 1998. Updated by RFC 4822.

[26] G. Malkin and R. Minnear, "RIPng for IPv6." RFC 2080 (Proposed Standard), Jan. 1997.

[27] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271 (Draft Standard), Jan. 2006.

[28] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, April 2006.

[29] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a DTN routing protocol – PRoPHETv2," in *Proceedings of the 6th ACM Workshop on Challenged Networks (CHANTS)*, ACM, Sept. 2011.

[30] A. Balasubramanian, B. N. Levine, and A. Venkataramani, “Replication routing in DTNs: A resource allocation approach,” *IEEE/ACM Transactions on Networking (TON)*, vol. 18, pp. 596–609, Apr. 2010.