ISSN: 2321-7782 (Online)
Volume 2, Issue 12, December 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

A Survey on Delay Tolerant Network

Monika. S. Shirbhate[1]
Information Technology Department
PRMIT & R, Badnera
Amravati, India

Abhishek. A. Gulhane[2]
Information Technology Department
PRMIT & R, Badnera
Amravati, India

Smeet. D. Thakur[3]
Information Technology Department
PRMIT & R, Badnera
Amravati, India

Abstract: We review the design of the Delay/Disruption Tolerant Networking (DTN) Architecture and application. A DTN is a network of smaller networks. It is an overlay on top of special-purpose networks, including the Internet. DTNs support interoperability of other networks by accommodating long disruptions and delays between and within those networks, and by translating between the communication protocols of those networks. In providing these functions, DTNs accommodate the mobility and limited power of evolving wireless communication devices. DTNs were originally developed for interplanetary use, where the speed of light can seem slow and delay-tolerance is the greatest need. However, DTNs may have far more diverse applications on Earth, where disruption-tolerance is the greatest need. The potential Earth applications span a broad range of commercial, scientific, military, and public-service applications.

Keywords: Delay/Disruption Tolerant Networking, communication protocols, Bundle Router

I. INTRODUCTION

A DTN is a network of smaller networks. It is an overlay on top of special-purpose networks, including the Internet. DTNs support interoperability of other networks by accommodating long disruptions and delays between and within those networks, and by translating between the communication protocols of those networks. In providing these functions, DTNs accommodate the mobility and limited power of evolving wireless communication devices. DTNs were originally developed for interplanetary use, where the speed of light can seem slow and delay-tolerance is the greatest need. However, DTNs may have far more diverse applications on Earth, where disruption-tolerance is the greatest need. The potential Earth applications span a broad range of commercial, scientific, military, and public-service applications. DTNs can accommodate many kinds of wireless technologies, including radio frequency (RF), ultra-wide band (UWB), free-space optical, and acoustic (sonar or ultrasonic) technologies. Communication on the Internet is based on *packet-switching*. *Packets* are pieces of a complete block of user data (e.g., pieces of an email message or a web page) that travel independently from source to destination through a network of links connected by routers. Routers switch the direction in which the packets move. The source, destination, and routers are collectively called *nodes*. Each packet that makes up a message can take a different path through the network of routers. If one link is disconnected, routers redirect the packets to use an alternate link. Packets contain both application-program user data (the payload part) and a header (the control part). The header contains a destination address and other information that determines how the packet is switched from one router to another. The packets in a given message may arrive out of order, but the destination's transport mechanism reassembles them in correct order. The usability of the Internet depends on some important assumptions:

Continuous, Bidirectional End-to-End Path: A continuously available bidirectional connection between source and destination to support end-to-end interaction.

Short Round-Trips: Small and relatively consistent network delay—milliseconds, not hours or days—in sending data packets and receiving the corresponding acknowledgement packets.

Symmetric Data Rates: Relatively consistent data rates in both directions between source and destination.

Low Error Rates: Relatively little loss or corruption of data on each link.

The Internet architecture has enabled new applications, such as e-mail, the World Wide Web and Voice over IP, which have transformed the way in which we access and create information. This architecture, however, makes strong assumptions about connectivity, such as available end-toend paths and low round-trip times, and high availability to naming, caching and search infrastructures to provide locator-based access such as DNS. This means that it is difficult to use e-mail or VoIP in a so-called *challenged* environment with unstable or high-delay connectivity, such as the dark side of the moon. This has led to the emergence of Delay Tolerant Networks (DTN) [5] as a different communication paradigm; one which is decentralized and distributed over a multitude of devices that are dynamically networked, carried by people, and embedded in everyday-life. For instance, the Haggle Project1 features a "Pocket-Switched" variant of DTN where people carry devices in their pockets, which communicate directly with other devices within their range or with infrastructure. As people move around, they can exchange messages with nearby devices, carrying a message until it is close to another device. Thus, our DTNdevice- carrying astronaut might have their e-mail messages 1http://www.haggleproject.org/ ferried by other DTN users taking their commercial flights to and from the moon.

II. A Brief History of DTN Research

DTN research started with Vint Cerf and the Interplanetary Internet initiative [6], which proposed new architecture that could work over both terrestrial and interplanetary links. This architecture could enable applications such as the remote operation of scientific experiments on other planets, controlled using TCP/IP from Earth. The Interplanetary Internet ideas were generalized by the IRTF DTN Research Group [7], which focuses on any kind of challenged environment where end-to-end connectivity may not always be available, and the DARPA Disruption Tolerant Networking program [8], which concentrates on developing protocols for bundling application-layer data units into DTN-layer protocol bundles for transport by DTN nodes. Opportunistic networks [9] and message ferrying [10] concentrate on mobile ad hoc DTNs as in the previously-discussed Haggle example, where routes are built dynamically between source and destination, and any possible intermediate node can be used *opportunistically* to ferry data as required. At the same time, applied EE/CS researchers, dissatisfied with the inefficacy of MANETs that they were trying to build in the real world, have started to do more and more experimental work, mapping out the world in terms of real radio propagation experiences, and actual mobility traces of genuine users (pedestrian, vehicular, zebras, badgers, albatrosses and so on). An important aspect of DTN route building is therefore an understanding of mobility patterns, and as such recent DTN research has studied social networks, both in humans [11, 12, 8] and other species [13]. These two strands of research can be seen to have converged in attempts to build realistic models for high-variance density MANETs. As well as being interesting from a systems and experimental point of view, research problems in DTNs, rather like MANETs before them, are also academically fascinating. A large class of models from the world of *network science* and *complex systems* can be extended covering the graph theoretic properties, and using physical analogs for the propagation of information through their dynamic topologies (percolation, diffusion, statistical thermodynamic/entropic models, and many more). In some senses, there is a body of work which stands on its own. In other senses, the reality gap between these theoretical models and the real results can be seen in the differences between predicted and measured performance of test applications. We are not trying to write a "defense of the dark arts of DTNs" syllabus here. Instead, we concentrate on questioning the realism of DTN research.

III. Architectures

At roughly the same time, there was growing interest in wireless sensor networks (WSNs), a topic which itself has spawned numerous conferences, journals, theses, and some commercial activities [5]. Much of the activity in WSNs has been devoted to power management, routing, and other tasks such as software updates and programming environments. Common to most WSN systems is a form of *gateway*—a communication node that often implements an application layer gateway able to effectively translate Internet (TCP/IP) protocols to the specialized protocols used within WSNs. It is typical for such gateways to participate in routing domains on both the Internet and within the WSN. Some of these gateways also possess storage, used to hold data collected from the sensor network until consumed by an application. Such gateways were being constructed in an ad-hoc manner, specific to each WSN, limiting interoperability between them. The DTN architecture was designed to accommodate not only network connection disruption, but also to provide a framework for dealing with the sort of heterogeneity found at sensor network gateways (and other gateways, more generally). Whereas the Internet (IP protocol) model supports heterogeneity as well, it does so by requiring every node to use a common network layer host identifier (IP address), packet format with universally-obeyed semantics, and routing methodology that assumes a connected routing graph in order to achieve interoperability. Supporting other addressing formats or semantics in conjunction with IP has resulted in widespread use of *overlay* networks, where the IP protocol is essentially used as a link protocol. As we shall see in more detail later, DTN uses naming, layering, encapsulation, and persistent storage to interconnect heterogeneous portions of a larger network, irrespective of formal layer.

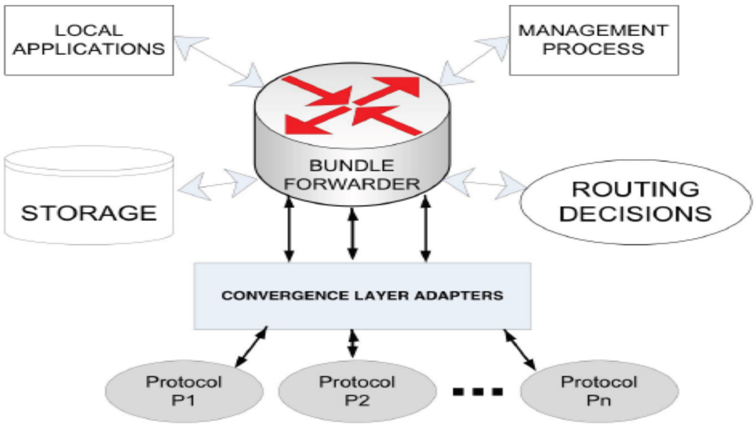


Fig. 1 Architecture shows how a bundle forwarder interacts with storage

DTN can use a multitude of different delivery protocols including TCP/IP, raw Ethernet, serial lines, or hand-carried storage drives for delivery. As each of these protocols provide somewhat different semantics, a collection of protocol-specific *convergence layer adapters* (CLAs) provide the functions necessary to carry DTN protocol data units (called *bundles*) on each of the corresponding protocols. Figure 1 gives the relative position of the convergence layer adapters in a conceptual implementation architecture: In this conceptual implementation architecture, a central forwarder is responsible for moving bundles between applications, CLAs, and storage according to decisions made by routing algorithms. Arrows indicate interfaces, which may carry either bundles (in the case of storage, CLAs, and applications) or directives (routing decisions, management, applications). In some cases, implementing these interfaces using inter-process communication facilities rather than conventional procedure call has been useful to promote the ability to develop system components independently.

IV. THE BUNDLE PROTOCOL

The DTN architecture implements store-and-forward message switching by overlaying a new transmission protocol—called the *bundle protocol*—on top of lower-lower protocols, such as the Internet protocols. The bundle protocol ties together the lower-lower protocols so that application programs can communicate across the same or different sets of lower-lower protocols under conditions that involve long network delays or disruptions. The bundle-protocol agent stores and forwards entire bundles (or bundle fragments) between nodes. A single bundle protocol is used throughout a DTN. By contrast, the lower-lower protocols below the bundle protocol are chosen to suit the characteristics of each communication environment. The figure below (top) illustrates the bundle-protocol overlay and (bottom) compares the Internet protocol stack (left) with a DTN protocol stack (right).

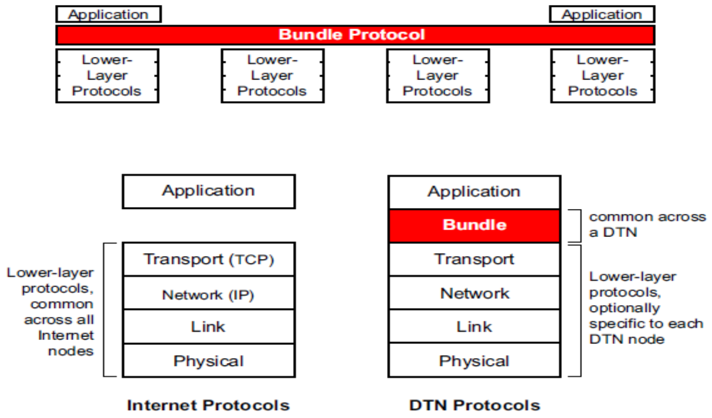


Fig. 2 Bundle Protocol

Bundles consist of three things: a bundle header consisting of one or more DTN blocks inserted by the bundle-protocol agent, a source-application's user data, including control information provided by the source application for the destination application that describes how to process, store, dispose of, and otherwise handle the user data, and an optional bundle trailer, consisting of zero or more DTN blocks, inserted by the bundle-protocol agent (not shown in the figure below). Like application-program user data, bundles can be arbitrarily long. Bundles extend the hierarchy of data-object encapsulation performed by the Internet protocols. The example below shows how bundle encapsulation works in the context of the lower-layer TCP/IP protocols. The bundle protocol does not alter the Internet-protocol data; it merely encapsulates the data. A bundle-protocol agent may break whole bundles into fragments (not shown in the figure below), just as the IP protocol may break whole data grams into fragments. If bundles are fragmented, the bundle-protocol agent at the destination reassembles them.

V. Blocks

The first or *primary* block of each bundle, illustrated in Figure 3, contains the DTN equivalents of the data typically found in an IP header on the Internet: version, source and destination EIDs, length, processing flags, and (optional) fragmentation information. It also contains some additional fields, more specific to the bundle protocol: report-to EID, current custodian EID, creation timestamp and sequence number, lifetime and a *dictionary*. Strings are placed in the dictionary, and offsets are used as pointers to the beginnings of strings in an effort to reduce space that would otherwise be devoted to duplicate strings. Most fields are variable in length, and use a relatively compact notation called *self-delimiting numerical values* (SDNVs) [2]. Early designs for the primary bundle block used more fixed-length fields, but the relative merit of choosing a fixed-length field for simplicity was ultimately found to be less compelling than the flexibility offered by SDNVs. SDNVs are discussed in more detail in Section IV-C. The bundle processing control flags indicate a number of special circumstances associated with the containing bundle: fragmentation condition (fragmented, allowed), type (regular or administrative), special requests (custody, acknowledgment generation, delivery status), class of service indication, and if the destination endpoint is known to be a *singleton* (that is, a single entity as opposed to a multicast endpoint). This last indicator is used when forwarding using custody transfer to alert a custodian that multiple nodes may be responding with custody transfer acknowledgments. By setting various bits in the bundle processing control flags, the sender can request a report for any of the following events: receipt at destination node, custody acceptance at a node, bundle forwarded/deleted/delivered en route, and receipt by destination application. Clearly, these capabilities need to be used with caution because of the amount of report traffic they may generate, may be limited in live operations by policy, and are really intended for diagnosing network problems, as with ICMP in the Internet. While there is some hesitation as to the value of having such a rich set of report types, requiring support for them in protocol implementations has already proven itself extremely useful during interoperability test and debugging sessions held in conjunction with IETF.2 To the best of our knowledge, the report-to EID is unique to the bundle protocol. It allows a sender of a bundle requesting one or more status reports to have the reports directed to node(s) other than it. This is in contrast to Internet ICMP messages which are specified to be sent back to the sender of the packets that caused the ICMP messages to be generated. This capability is useful in the DTN context because some senders may not be expected to exist beyond the time required to transmit a bundle they have sent. Examples include expend- 2http://www3.ietf.org/proceedings/06nov/slides/DTNRG-1/sld1.htm able sensor nodes that are lost or destroyed after reporting their sensor readings to a nearby DTN relay node. The origination time in each bundle indicates the real time at which the bundle was sent from its origin. The lifetime is a positive offset of real time from the origination time. If a bundle is found to be queued at the end of its lifetime, it can be discarded. This is one of the ways excess bundles can be cleared from the network. It also provides a basis for implementing policy: a network operator could arrange for bundles beyond some age to be expired early (or late). The use of real time in bundles imposes a requirement on each participating DTN node: that real time is synchronized, at least roughly. This requirement was considered repeatedly, as it represents a significant departure from common practice in the Internet today. To date, we have identified four reasons for imposing it. First, most applications for which DTN were designed are time sensitive; resources are consumed at particular points in space and time. A DTN node not knowing the time renders the DTN far less useful for most applications which themselves require time. Second, in most of the cases where DTN has been tested, and in most cases for which it is planned, access to real-time is already provided by some mechanism (including in deep space and underwater environments). Third, routing using scheduled connectivity is inherently tied to link availability at a certain time. Fourth, network management tasks, including tracing and debugging are considerably easier when a common time reference is used throughout the network. Other than the required primary block appearing at the beginning of a bundle, additional blocks are optional but use a common basic format. The common format includes an 8-bit block type (like the extension header type in IPv6), processing flags and block length. The processing flags indicate whether the block is to be copied in any fragment created, whether a status report should be issued if the block type is unknown to the node forwarding the bundle and whether the bundle should be dropped in this case. The indication to copy the block to each fragment is really designed for blocks carrying meta-data associated with delivery of the bundle contents such as handling restrictions, retention guidelines, digital rights management, or sensitivity labels. In the environments that require them, such meta-data are typically mandatorily bound close to the data they describe.

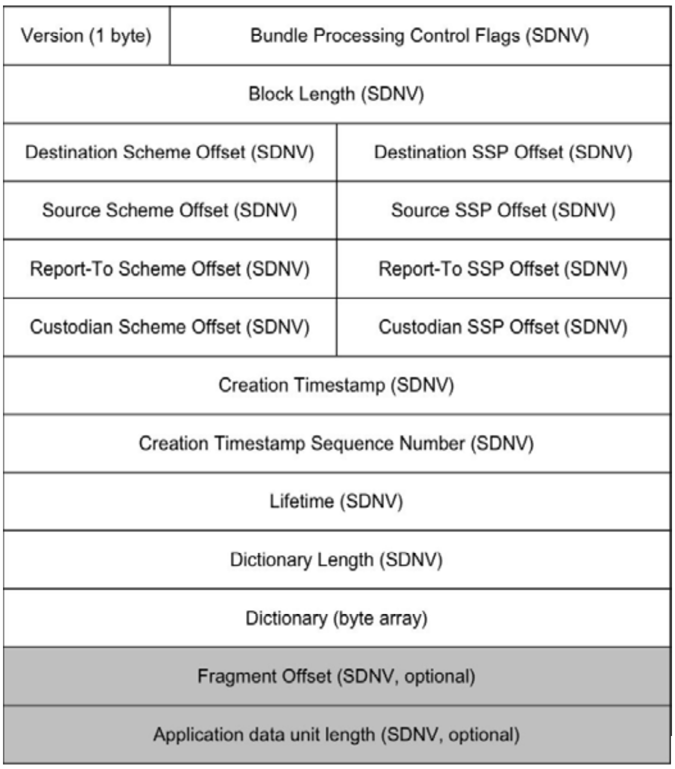


Fig. 3 The structure of the primary block of a bundle contains a fixed-length version field followed by a collection of variable-length fields encoded as self-delimiting numeric values (SDNVs).

VI. Applications

1. Bulk data distribution in urban areas (e.g. free customized newspapers/adverts to cell phones on the metro or tube). This is ideal for a DTN since capacity in a DTN will always be much higher than in an infrastructure network [11]. At the same time you will still require some infrastructure, due to the phase transition[14] seen for reach ability, and also for the injection of original data, and perhaps for identity and payment mechanisms.

2. Sharing of individual contents (rather than bulk data) in urban areas. For example city commuters may stay together in the same bus or train for a good amount of time, using their short-range radio equipped mobile devices, individuals can share content with fellow travelers. Historical collocation and social information can be used to determine the best content sources. [15]

3. Disconnected kiosks in rural areas. Rural Internet kiosks in developing regions can provide a variety of useful services such as birth, marriage, and death certificates, land records, and medical and agricultural consulting to the poorest sections of society. Kiosk Net uses low-cost computing devices as kiosks and use vehicle as mechanical backhaul to move data to Internet gateways [16]. They have run pilot deployment in Anandapuram, a village in South India, as early as in year 2006.

4. Mobile location-aware sensing applications. B-MAD is a Bluetooth and WAP push based location-aware mobile advertising system [1]. A Bluetooth sensor deployed in the environment discovers the nearby cell mobiles and sends the Bluetooth addresses over a WAP connection to the Ad Server. Ad Server will push related advertisements to the end users. This is not really a DTN application, but similar concept can be used in DTN to push bulk data to end users.

5. Social mobile applications. Haggle project has a social mobile application called MobiClique, which uses social network information from Facebook6 to bootstrap the data forwarding structure of the DTN, and at the same time social networks are also created from the daily contact patterns. Combining with location services and the assistance of limited amount of infrastructures, DTN can provide a big variety of city-wide social services.[1]

VII. Conclusion

The DTN store-and-forward message switching architecture is a generalization of work originally conceived to support the InterPlaNetary Internet (IPN). The primary goals are interoperability across network environments, and reliability capable of surviving hardware (network) and software (protocol) failures. Although DTNs were originally conceived for interplanetary use, they may have a far greater number of applications on Earth.

In this paper the survey DTN architecture aims to provide interoperable communications between a wide range of networks which may have exceptionally poor and disparate performance characteristics. The design embraces the notion of message switching with in-network storage and retransmission, late binding of names, and routing tolerant of network partitioning to construct a system better suited to operations in challenged environments than most other existing network architectures, particularly today's TCP/IP based Internet.

References

1. Kevin Fall, and Stephen Farrell. “DTN: An Architectural Retrospective” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 26, NO. 5, JUNE 2008
2. Evan P.C. Jones and Paul A.S. Ward. “Routing Strategies for DelayTolerant Networks”
3. Kevin Fall Intel Research, Berkeley. “A Delay-Tolerant Network Architecture for Challenged Internets”
4. Sushant Jain, Kevin Fall and Rabin Patra. “Routing in a Delay Tolerant Network”
5. Fall, K. “A delay-tolerant network architecture for challenged internets.” In Proceedings of ACM SIGCOMM (2003), pp. 27–34.
6. Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., and Weiss, H. “Delay-tolerant networking: an approach to interplanetary internet.” IEEE Communications Magazine 41, 6 (June 2003), 128–136.
7. Aguilera, M. K., Strom, R. E., Sturman, D. C., Astley, M., and Chandra, T. D. “Matching events in a content-based subscription system.” In Proceedings of PODC ’99 (1999), pp. 53–61.

8. Bigwood G, Rehunathan D, Bateman M, Henderson T., and Bhatti, S. "Exploiting self-reported social networks for routing in ubiquitous computing environments." In Proceedings of the 1st International Workshop on Social Aspects of Ubiquitous Computing Environments (SAUCE) (Avignon, France, Oct. 2008).

9. Pelusi, L., Passarella, A., and Conti, M. "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks." IEEE Communications Magazine 44, 11 (2006), 134–141.

10. Zhao, W., Ammar, M., and Zegura, E. "A message ferrying approach for data delivery in sparse mobile adhoc networks." In Proceedings of ACM MobiHoc (New York, NY, USA, May 2004), pp. 187–198.

11. Grossglauser, M., and Tse, D. N. C. "Mobility increases the capacity of ad hoc wireless networks."IEEE/ACM Transactions on Networking 10, 4 (Aug. 2002), 477–486.

12. Mtibaa, A., Chaintreau, A., Lebrun, J., Oliver, E., Pietilainen, A. K., and Diot, C. "Are you moved by your social network application?" In Proceedings of the First ACM SIGCOMM Workshop on Online Social Networks (WOSN) (Seattle, WA, USA, Aug. 2008).

13. Lindgren, A., Mascolo, C., Lonegan, M., and Mcconnell, B. "A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks." In Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS08) (Oct. 2008).

14. Hui, P., and Lindgren, A. "Phase transition of opportunistic communications." In Proceedings of the ACM Workshop on Challenged networks (CHANTS) (Sept. 2008).

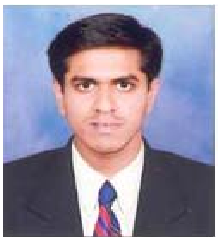
15. Mcnamara, L., Mascolo, C., and Capra, L. "Media sharing based on colocation prediction in urban transport." In Proceedings of ACM MobiCom 2008 (Sept. 2008).

16. Guo, S., Falaki, M. H., Oliver, E. A., Rahman, U. S., Seth, A., Zaharia, M. A., and Keshav, S. "Very low-cost internet access using kiosknet." SIGCOMM Computer Communication Review 37, 5(Oct. 2007), 95–100.

AUTHOR(S) PROFILE

Monika. S. Shirbhate, received the M.E degree in Information Technology from Prof.Ram Meghe institute of Technology & Research, Badnera-Amravati in 2012 and 2013, respectively. Currently working as an Assistant professor in Information Technology Department of Prof.Ram Meghe Institute of Technology & Research Badnera Amravati (Maharashtra).

Abhishek. A. Gulhane, received the M.E degree in Information Technology from Prof.Ram Meghe institute of Technology & Research, Badnera-Amravati in 2011 and 2012, respectively. Currently working as an Assistant professor in Information Technology Department of Prof.Ram Meghe Institute of Technology & Research Badnera Amravati (Maharashtra).



Smeet. D. Thakur, received the M.E degree in Information Technology from Prof.Ram Meghe institute of Technology & Research, Badnera-Amravati in 2013 and 2014, respectively. Currently working as an Assistant professor in Information Technology Department of Prof.Ram Meghe Institute of Technology & Research Badnera Amravati (Maharashtra).