

Chapter 4

Delay/Disruption Tolerant Networking



Ronald in 't Velt, Ingrid Mulders, Arwid Komulainen, and Michael Goetz

Delay and/or Disruption Tolerant Networking (DTN) protocols are important when dealing with partitioned wireless communication clusters, i.e., when dynamic connections are often broken for longer times than the maximum allowed network latency. A combination of ad-hoc and DTN functionalities is key for underwater sensor networking where AUVs can temporarily leave the network area or where a submarine can go silent for a while [1–3].

DTN is an active research field, especially in terrestrial and space applications such as wildlife tracking, vehicular networks (mobile ad-hoc networks, MANETs), interplanetary networks, or in situations after a disaster, where the protocol enables transport of the data between groups of disconnected mobile nodes, if the communication path between senders and receivers is completely broken or disconnected for longer time periods (e.g., emission control) [4–7]. Several delay-tolerant networking protocols have been developed in the past years [1–27]. Amongst others, the (now disbanded) DTN Research Group of the Internet Research Task Force (IRTF) made some significant contributions by developing an architecture [28] and specifying protocols for DTN. The most important product of this group is the Bundle Protocol specification [29]. The DTN Working Group of the Internet Engineering Task Force (IETF) is revising and standardizing the Bundle Protocol.

R. in 't Velt · I. Mulders
Netherlands Organisation for Applied Scientific Research (TNO), The Hague, Netherlands

A. Komulainen
Swedish Defence Research Agency (FOI), Stockholm, Sweden

M. Goetz
Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE),
Bonn, Germany

4.1 Store-Carry-Forward Paradigm

Before explaining the Bundle Protocol specification, the Store-Carry-Forward paradigm [5] is introduced, a key element of DTN. In a conventional packet-switched (non-flooding) network layer such as IP, a network node that is to forward a packet consults its Forwarding Information Base (FIB) to determine the Next Hop address associated with the destination address of the packet. If a Next Hop address is found, then the packet may sit in a queue of the associated outgoing interface for a while, waiting its turn, but it will eventually be transmitted: this is referred to as Store-and-Forward. However, if no Next Hop address for the given destination is present in the FIB, then the node has no other option than to discard the packet. DTN allows a node to hold on a Protocol Data Unit (PDU) for which a Next Hop is not readily available. If the node is mobile, then its movements may bring it within communication range of the final destination of that PDU or within range of a suitable Next Hop: hence *Store-Carry-Forward*.

4.2 Bundle Protocol Specification

One of the main implementations of the Store-Carry-Forward paradigm is the Bundle Protocol specification. The idea behind the Bundle Protocol is that, in an environment with long communication delays and frequent disruptions, protocol interactions between communication endpoints should be kept to a minimum. Data to be exchanged should be grouped together with all necessary meta data and protocol information into a self-contained Protocol Data Unit, the *Bundle*. In terms of the OSI (or TCP/IP) model, the Bundle Protocol can be thought of as residing at the Application Layer, forming an overlay over the traditional protocol stack. The entity instantiating the Bundle Protocol layer at a node is called the *Bundle Protocol Agent* (BPA). The transfer of a Bundle from one BPA to another BPA is end-to-end communication from the OSI point of view, but several of such transfers in sequence may be required to convey a Bundle from its source DTN endpoint to its destination DTN endpoint, with intermediate BPAs acting as Bundle relays, see Fig. 4.1. Bundles are made up of blocks, always including a single Primary Block and a single Payload Block, and optionally one or more Extension Blocks. The Primary Block contains Bundle Protocol header information, including a Source and a Destination Endpoint Identifier (EID). These EIDs take the form of URLs, consisting of a scheme name followed by a scheme-specific part (e.g., `dtn://ucomms/auv-1`). Bundles can be much larger than (IP) packets, with the size of the latter typically being constrained by the Maximum Transmission Unit (MTU) of the underlying link layer technology (packet fragmentation should generally be avoided). Breaking down Bundles into segments that the Transport Layer can handle is a task for the Convergence Layer that resides below the Bundle Protocol Layer. Convergence Layers for different Transport Layer

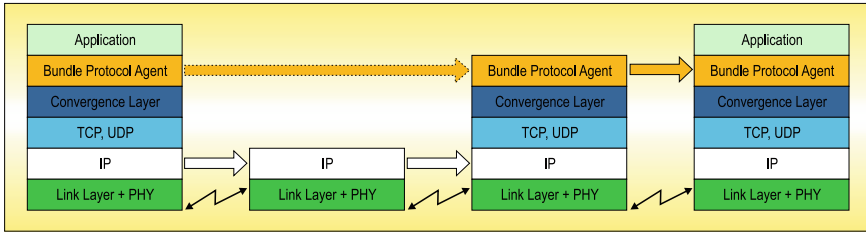


Fig. 4.1 Protocol stacks in a combined DTN Bundle-relaying and packet-forwarding configuration

protocols have been specified, e.g., for TCP [30] and for datagram-based transport [31].

In an environment where node movement patterns are largely unpredictable,¹ BPAs need to become aware of each other's presence before Bundles can be exchanged. This necessitates a mechanism for DTN *Neighbour Discovery*, which can either involve the use of a special kind of Bundles (in-band) or be based on a dedicated protocol external to the BPA (out-of-band). Moreover, when a DTN node encounters another DTN node that does not hold the Destination Endpoint for some of the Bundles that the first node is carrying, it has to decide whether or not to transfer or copy those Bundles to the other node. Essentially, it has to assess whether the encountered node is a viable candidate for getting the Bundles closer to their Destination Endpoint. This process is referred to as *DTN routing*. It should be realized, however, that this is very different from routing at the packet level. Many different DTN routing strategies can be found in literature. A distinction is made between *single-copy* and *multi-copy* schemes. The former allows only a single instance of a given Bundle to be present in the network at any time, whereas the latter uses duplication of Bundles to increase the probability of their delivery and/or decrease latency. Examples of multi-copy routing strategies are Epidemic [8, 32] and Spray-and-Wait [33], and an example of a single-copy scheme is the custody transfer mechanism [3].

4.3 DTN for Underwater Acoustic Networks

Bundle-Protocol-based DTN is developed for terrestrial and interplanetary networks, and a valid question would be to what extent it is applicable to underwater acoustic communications. The Bundle Protocol and its supporting Convergence Layer introduce additional protocol overhead, for which the time and bandwidth in the underwater acoustic network may be limited. An interesting new development to reduce communication overhead is the context-based adaptation in DTN [27], in

¹In one significant area of application of DTN, space communications, disruptions and re-establishment of communication links often are predictable: it can be calculated ahead of time when a planet orbiter will emerge from the shadow of the planet, when a satellite will come into range of a ground station, etc.

which DTN protocols can be adapted to variations in network conditions. The adaptation is guided by predefined context parameters that can be collected by the node itself. DTN does not only require a (distributed) storage capability within the network, but the network should also be able to detect that a destination node cannot be reached before sending messages to that node (Store-Carry-Forward).

Furthermore, Bundle-Protocol-based DTN is targeting adverse communication conditions, including long propagation delays and frequent disruptions, but not necessarily very low transmission rates. In fact, it can be argued that the Bundle Protocol works best in environments where node encounters are of an opportunistic nature. I.e., when during contact periods a relatively high transmission rate can be used, presumably over short distances, for example using optical communication. However, the Store-Carry-Forward paradigm of DTN is deemed to be valuable in the context of underwater communications. It may be worth exploring whether this paradigm can somehow be implemented at the Network Layer instead of the Application Layer, i.e., packet-based DTN instead of Bundle-based DTN. Unfortunately, there are no known specifications to fall back on for such an alternative approach to DTN. The required technology would thus have to be developed from scratch.

Some efforts have already been made to adapt known DTN mechanisms for the underwater network environment with untrustworthy links. A first step in this direction was the ACommsNet10 trial in September 2010, performed by CMRE (La Spezia, Italy), by using new local, low-overhead, adaptive routing schemes [34]. Additionally, the capability to exchange data between separated network clusters with AUVs was tested within a German national WTD 71/FKIE cooperation. For this purpose, the network protocol was extended, among other enhancements, by a so-called postman functionality. The protocol extension was tested with two SeaCat AUVs and three bottom nodes during a sea trial near Bornholm in November 2014 and in Summer 2017 in the North Sea. G UWAL was used as application language, which was already used in the RACUN demonstration and was only extended by DTN network control packets, e.g., for the postman handshake. An advantage of G UWAL are the timestamps within the packets, which can be used to decide if a delayed packet is still valid and should be exchanged with a postman or should be dropped.

4.4 Data Muling

A commonly used application for DTN is Data Muling. Data muling is the activity of transporting data between (static) nodes using one or more mobile nodes. In Underwater Sensor Networks (UWSNs), data muling usually refers to an AUV transporting data between nodes, or clusters of nodes, between which there are no (long-range) communication links. Typical scenarios that employ data muling are postman scenarios, where communication between disjoint parts of a network is assisted by a mobile postman, data offloading from static sensor nodes to mobile nodes, and data offloading from mobile (inspection) nodes to static gateway nodes (wireless docking). Data

offloading is typically performed using some form of high-speed communication technology. Optical communication is commonly used [35–37], and another option is to use a high-frequency acoustic link.

Some prerequisites for efficient data muling are: AUV localization of and distance estimation to sensor nodes using, for instance, the methods presented in the previous chapter; high-speed communication links, high-frequency acoustic or electromagnetic; and protocol stacks designed to handle intermittent links (delay/disruption-tolerant-networking). Depending on the purpose of the data mule and the scenario in which it is used, efficient route planning in order to visit sensor nodes in an energy-efficient manner can also be an important requisite. In [38], Hollinger et al. treated route planning as a traveling salesman problem, with the problem formulation modified to account for the unreliable communication links.

In [36], the AUV is equipped with a down-facing camera and is assumed to have low-accuracy maps of sensor node locations. The AUV also surfaces occasionally to correct drift in its position estimate. The AUV is tasked with visiting all stationary nodes to offload data. The sensor nodes are equipped with low-rate acoustical modems which can be used for signaling events and beaconing, but this is not demonstrated in the paper. When the AUV is in the vicinity of a node, it processes the images from the camera to locate and hover over the sensor node. When the AUV has started hovering it uses optical communication to offload data before moving to the next node. The system is demonstrated in a pool containing three sensor nodes and one AUV.

The system demonstrated in [37] uses acoustic beaconing to let the AUV locate the sensor nodes using either a stochastic gradient descent approach or a particle filter. Both methods are shown to successfully localize the sensor node, with the particle filter performing best. The AUV is said to need only a rough estimate of the sensor node position, with an error margin similar to the range of the acoustic beacon. The sensor node sends acoustic beacons every six seconds and constantly streams optical data. When a packet is successfully received by the AUV over the optical link it switches from using the acoustic beacons to the optical signal to stay hovering over the sensor node.

In [39], the medium access problem that occurs when a data mule visits a cluster of nodes is examined. More specifically, the authors compare random access schemes with a proposed polling-based scheme called *UW-polling*. The protocol consists of three parts: a neighborhood discovery phase where the AUV determines which nodes to poll, a data prioritization phase where the nodes communicate what data to send and finally the AUV polls the nodes in an order determined in the previous phase. *UW-polling* is compared to random-access schemes, CS-Aloha and Distance-aware collision avoidance protocol (DACAP), in terms of throughput, packet delivery ratio, delay and energy efficiency. The CS-Aloha protocol is modified in the sense that the AUV transmits trigger packets which triggers the Aloha procedure in the sensor nodes for a given time. The modification prevents sensor nodes to transmit blindly when the AUV is out of range. At low source power levels *UW-polling* offers benefit in terms of robustness compared to the other schemes, however, for higher source levels, the performance is similar to CS-aloha. In [40], *UW-polling* is compared to

another protocol proposed by the same authors called *U-Fetch*, in which sensor nodes forward data to cluster heads, which are appointed for communicating with the AUV. It is shown that U-Fetch provides lower latency than UW-polling, at the cost of lower packet delivery ratio.

A multi-modal data mule scenario using a combination of acoustic and optical communications is analyzed in [41]. The system uses the previously mentioned CS-Aloha-Trig protocol with a fixed physical-layer modality during each trigger period. Between trigger periods, the AUV can decide to switch between acoustic and optical communication, based on the received power for each modem. Network simulations using the DESERT framework [42], extended to model multi-modal communications, are used to determine the throughput in different water conditions. To handle difficult, turbid conditions, the data mule needs to stand still longer at each node in order to correctly determine when to switch physical-layer modality.

References

1. Anitha S, Micheal G (2013) An evaluation of different routing schemes in dtn. *Int J Comput Sci Mob Comput* 2(5):41–49
2. Desai CB, Pandya VN, Hadia SK (2013) A survey on knowledge based classification of different routing protocols in delay tolerant networks. *Int J Comput Sci Mob Comput* 2(3):83–88
3. Yu Y, Chen X (2013) Research on custody transfer service in delay tolerant network. *J Netw* 8(8):1713–1719
4. Omidvar A, Mohammadi K (2014) Particle swarm optimization in intelligent routing of delay-tolerant network routing. *EURASIP J Wirel Commun Netw* 147
5. ZhuK (2014) Social-based Data Routing Strategies in Delay Tolerant Networks. PhD thesis, University of Göttingen
6. Ono M, Sawai K, Suzuki T (2014) Development of rest facility information exchange system by utilizing delay tolerant network. *Int J Adv Comput Sci Appl* 5(2):83–89
7. Wang S-Y, Torgerson JL, Schoolcraft J, Brenman Y (2014) The deep impact network experiment operations center monitor and control system. <http://hdl.handle.net/2014/44657>
8. Marandi A, Faghi Imani M, Salamatian K (2014) Practical bloom filter based epidemic forwarding and congestion control in dtns: a comparative analysis. *Comput Commun* 48:98–110. Elsevier
9. Ramesh S, Ganesh Kumar P (2014) Bcr routing for intermittently connected mobile ad hoc networks. *Int J Eng Technol* 6(1):66–74
10. Zhang Z, Jin Z, Ma M (2014) Ccs-dtn: clustering and network coding-based efficient routing in social dtns. *Sensors* 15(1):285–303
11. Papaj J, Dobos L (2014) Trust based algorithm for candidate node selection in hybrid manet-dtn. *Adv Electr Electron Eng* 12(4):271–278
12. Jain S, Kishore N, Chawla M (2014) Soares VNGJ (2014) Composite mechanisms for improving bubble rap in delay tolerant networks. *J Eng* 1:1–7
13. Garcia G, Robles S, Sanchez A, Borrego C (2014) Information system for supporting location-based routing protocols. *RECSI*, pp 203–208
14. Moreira WA Jr (2014) Social-aware opportunistic routing. PhD thesis, Universidade de Aveiro
15. Saha S, Verma R, Saika S, Paul PS, Nandi S (2014) e-one: enhanced one for simulating challenged network scenarios. *J Netw* 9(12):3290–3304
16. Zou S, Wang W, Wang W (2013) A routing algorithm on delay-tolerant of wireless sensor network based on the node selfishness. *EURASIP J Wirel Commun Netw* p 212

17. Xie LF, Chong PHJ, Guan YL (2013) Routing strategy in disconnected mobile ad hoc networks with group mobility. *EURASIP J Wirel Commun Netw* p 105
18. Fiems D, Altman E (2009) Markov-modulated stochastic recursive equations with applications to delay-tolerant networks. Technical Report RR-6872, INRIA. An optional note
19. Liang H (2013) Resource management in delay tolerant networks and smart grid. PhD thesis, University of Waterloo
20. Poonguzharselvi B, Vetriselvi V (2012) Trust framework for data forwarding in opportunistic networks using mobile traces. *Int J Wirel Mob Netw* 4(6):115–126
21. Ramesh S, Ganesh Kumar P (2013) Spray and wait routing with agents in intermittently connected manets. *Int J Eng Technol* 6(1):66–74
22. Wang S, Ma R (2013) Name: A naming mechanism for delay/disruption-tolerant network. *Int J Comput Netw Commun* 5(6):231–241
23. Fowjiya S, Udhayachandrika A, Kathirvel A (2013) Architectural overview of delay tolerant network. *Int J Eng Sci Res Technol* 2(10)
24. Gao L, Li M, Zhou W, Shi W (2013) Anonymous data forwarding in human associated delay tolerant networks. In: *IEEE 33rd international conference on distributed computing systems workshops*, July 2013. Philadelphia (PA), USA
25. Gao L, Li M, Bonti A, Zhou W, Yu S (2013) Multi-dimensional routing protocol in human associated delay-tolerant networks. *IEEE Trans Mob Comput* 12(11):2132–2144
26. Bindra HS, Sangal AL (2012) Need of removing delivered message replica from delay tolerant network - a problem definition. *Int J Comput Netw Inf Sec* 4(12):59–64
27. Petz A (2012) Context-based adaptation in delay-tolerant networks. PhD thesis, University of Texas, Austin, USA
28. Cerf V, Burleigh S, Hooke A, Torgerson L, Durst R, Scott K, Fall K, Weiss H (2007) Delay-tolerant networking architecture. RFC-4838, IRTF network working group
29. Scott K, Burleigh S (2007) Bundle protocol specification. RFC-5050, IRTF network working group
30. Demmer M, Ott J, Perreault S (2014) Delay-tolerant networking tcp convergence-layer protocol. RFC-7242, IRTF network working group
31. Kruse H, Jero S, Ostermann S (2014) Datagram convergence layers for the delay- and disruption-tolerant networking (dtm) bundle protocol and licklider transmission protocol (ltm). Technical Report RFC-7122, IRTF network working group
32. Lu X, Hui P (2010) An energy-efficient n-epidemic routing protocol for delay tolerant networks. In: *Fifth IEEE international conference on networking, architecture and storage*, Macau SAR, China
33. Spyropoulos T, Psounis K, Raghavendra CS (2005) Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: *Proceedings of 2005 ACM SIGCOMM workshop on delay-tolerant networking*, Philadelphia (PA), USA
34. Merani D, Berni A, Potter J, Martins R (2011) An underwater convergence layer for disruption tolerant networking. In: *Baltic congress on future internet and communications*, Riga, Latvia
35. Vasilescu I, Kotay K, Rus D, Dunbabin M, Corke P (2005) Data collection, storage, and retrieval with an underwater sensor network. In: *Proceedings of the 3rd international conference on embedded networked sensor systems*, SenSys '05. ACM, New York, NY, USA, pp 154–165
36. Dunbabin M, Corke P, Vasilescu I, Rus D (2006) Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. In: *Proceedings 2006 IEEE international conference on robotics and automation*, 2006. ICRA 2006, pp 2091–2098
37. Doniec M, Topor I, Chitre M, Rus D (2013) Autonomous, localization-free underwater data muling using acoustic and optical communication. Springer International Publishing, Heidelberg, pp 841–857
38. Hollinger GA, Choudhary S, Qarabaqi P, Murphy C, Mitra U, Sukhatme GS, Stojanovic M, Singh H, Hover F (2012) Underwater data collection using robotic sensor networks. *IEEE J Sel Areas Commun* 30(5):899–911
39. Favaro F, Casari P, Guerra F, Zorzi M (2012) Data upload from a static underwater network to an auv: polling or random access? In: *2012 Oceans - Yeosu*, pp 1–6

40. Favaro F, Brolo L, Toso G, Casari V, Zorzi M (2013) A study on remote data retrieval strategies in underwater acoustic networks. In: 2013 OCEANS - San Diego, pp 1–8
41. Campagnaro F, Favaro F, Guerra F, Calzado VS, Zorzi M, Casari P (2015) Simulation of multimodal optical and acoustic communications in underwater networks. In: OCEANS 2015 - Genova, pp 1–7
42. Casari P, Tapparello C, Guerra F, Favaro F, Calabrese I, Toso G, Azad S, Masiero R, Zorzi M (2014) Open source suites for underwater networking: Woss and desert underwater. *IEEE Netw* 28(5):38–46