Scalable DTN Distribution over Uni-Directional Links

Dirk Kutscher
TZI, University of Bremen
dku@tzi.org

Janico Greifenberg
Dampsoft GmbH
jgre@jgre.org

Kevin Loos
TZI, University of Bremen
logic@tzi.org

ABSTRACT

We present an architecture for scalable DTN communication in sparsely populated areas. Our approach is based on Uni-DTN, a unidirectional DTN convergence layer that we have developed for unicast and multicast distribution of DTN bundles. In this paper, we discuss possible application scenarios, present a suitable distribution architecture and the Uni-DTN convergence layer and describe, how specific features of the DTN bundle specification and DTN communication scenarios can be implemented using Uni-DTN.

1. INTRODUCTION

DTN concepts [2] have historically often been envisioned to be applied to sparsely populated communication scenarios, where network infrastructure and connectivity is not permanently available. Applications range from rural connectivity to nomadic/mobile access to Internet-based applications. The fundamental motivation is often the assumption that low-cost connectivity can be easier accomplished if requirements for seamless connectivity and bound round-trip delays can be softened. DTN concepts [7] [1] such as asynchronously connected nodes, store-carry-and-forwarding principles and routing for opportunistic connectivity-based networks have been proposed for different applications such as e-mail, push content distribution and limited forms of interactive communications but also very specific applications such as emergency notification.

In developing regions, connecting end-users at remote-sites via wireline access, terrestrial radio or satellite links is generally just not economically feasible as described in detail by [3]. E.g., the DakNet project [17] provides asynchronous connectivity to remote villages in India and Cambodia by installing WLAN-based information kiosks in villages and by leveraging existing communications and transportation infrastructure to distribute content to these kiosks. DakNet uses the notion of *mobile access points* — physical data transportation devices (that can reside on buses, motorcycles, bicycles etc.) with low-cost WLAN transmitters. This approach is associated with lesser capital expenditures per user as installation of new communication infrastructure at a larger scale is not required. In [17] the authors contrast DakNet with the Digital Gangetic Plains project (DGP [18]) that has installed rural connectivity testbeds based on IEEE 802.11 technology applied to directional long-range-communication. A similar model was proposed in [20] where kiosks in rural villages are used to provide network services for the local population. In these communication scenarios, the local data delivery is typically implemented on an opportunistic basis, e.g., in a kiosk model, where users and mobile devices come to a kiosk, establish a contact and exchange information units. This can be implemented by applying established concepts and technologies such as commodity WLAN devices, DTN communication infrastructure with existing and implemented DTN convergence layers [5] that can be used in an opportunistic fashion.

However, the long-haul access link to remote locations is often viewed as a critical factor for providing asynchronous connectivity to sites, as it is a significant cost factor for larger scale operations. In [3] satellite access is compared to other access alternatives, including DakNet, and it is concluded that — due to the considerable hardware and operational costs — satellite-based access is today generally not an option for individual access. However, the higher costs can be amortized among larger user groups by using satellite links for connecting DakNet hubs – thus combining the scalability and performance of satellite technology with the cost-efficiency of the DakNet approach. Adding timely and scalable distribution to a DTN-based communication infrastructure would enable to better supports many relevant applications, such as mass distribution of emergency broadcasts, public announcements, education and entertainment material.

In this paper, we present an approach for enhancing the potential coverage and the performance of a DTN-based approach to connecting sites by employing unidirectional communication, e.g., over satellites and dedicated long-range terrestrial links. We assume a general DTN-based architecture with data mules and info stations at the edges and extend this architecture to include unidirectional broadcast links that apply specifically to the distribution of mass content, i.e., in a broadcast/multicast mode. Figure 1 illustrates the general idea: we are adding a broadcast distribution link (here: a satellite link) to a DTN distribution infrastructure that is based on a data-mule distribution approach. The satellite link is used by content providers who want to distribute their offerings to a large population utilizing a nation-wide satellite service. This service is connecting the regional distribution hubs that are services by buses forwarding the received information bundles to villages where they can be delivered to kiosk systems and individual users. The main contributions of this paper are 1) a DTN convergence layer for such uni-directional broadcast links, and 2) a DTN multicast distribution architecture that leverages this convergence layer. The paper is structured as follows: In section 2 we describe some selected application scenarios for our approach, which is then pre-

NSDR'07, August 27, 2007, Kyoto, Japan.

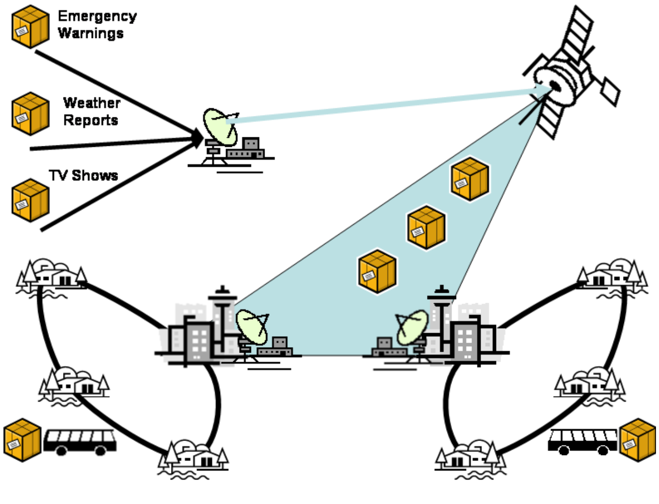

Figure 1: Integrating DTN data-mules and broadcast links

sented in section 3. Section 4 provides a description of the Uni-DTN convergence layer and our mapping of DTN bundle protocol features to it. In section 5 we discuss DTN routing issues for unidirectional convergence layers. We present implementation and evaluation results in section 6.

2. SCENARIOS

A DTN-based distribution infrastructure with broadcast transport services can be applied to different applications and communication scenarios. In this section we briefly describe two selected communication scenarios for using broadcast links in developing regions: 1) distribution of emergency broadcasts, and 2) delivering web and multimedia content to groups of users.

2.1 Distribution of Emergency Broadcasts

In natural disaster scenarios, informing the affected population is an important and often the first measure of emergency response. While some kinds of disaster may be predictable and allow for proactive counter-measures mitigating its impact, unreliable or nonexistent communication infrastructure in developing regions make it hard to communicate the imminent threat. E.g., floods or storms can be discovered before they reach land and can affect inhabited regions. However, making this information available requires communication services that can effectively reach all of the potentially affected population – in a timely and reliable fashion. In some developed countries, broadcast infrastructures exist that transmit alerts to mobile user devices, e.g., the earthquake and tsunami warning system in Japan that is currently using the TV broadcast infrastructure [12] and is being extended to a location-based service for mobile phones [6].

Naturally, deploying such services requires significant investments in infrastructure and operations, which makes them infeasible to be directly applied to developing regions. However, when we assume the existence of a DTN data-mule-based infrastructure for connecting villages in remote areas, connecting the regional *DTN hubs*, e.g., in larger villages, to the nation-wide satellite-based alert system can significantly accelerate the distribution of time-critical information with a reasonable cost efficiency: a large number of receivers can be reached with comparatively small bandwidth requirements because of the broadcast distribution.

We can extend the DTN infrastructure to include the actual sources of alerts as depicted in figure 1, i.e., a national authority such as a meteorological agency would directly transmit alerts as DTN bundles. In order to maintain the claimed scalability, this requires a **broadcast/multicast distribution service**, including a suitable **DTN addressing scheme** and **broadcast-enabled DTN transports**, i.e., network links and corresponding DTN convergence layers. Data transport must be as reliable as possible under the given circumstances. Using solicited retransmissions for transmission error recovery, however, has obvious scalability problems and does not work for receive-only-devices. Thus other **reliability mechanisms** such as forward error correction and periodic transmission should be used, as the devices cannot be expected to receive any time, due to power constraints or link failures.

2.2 Delivering Content to Groups of Users

In addition to emergency alert distribution a general DTN-based multicast infrastructure can be applied to other communication services such as *mass content distribution* to interested users. In well-connected environments, the common way to obtain news over the Internet, e.g. in the form of podcasts or web pages, is to explicitly request the data via HTTP, which is difficult in challenged networks, as sending requests may be expensive (in terms of power and money) and subject to long delays (if end-to-end paths can be established at all). A significant portion of such content (e.g., educational material or editorial content, music, videos etc.) does not really require synchronous communication and real-time-like transmission characteristics. A more scalable and cost-efficient transport for content that is of interest for groups of receivers in a network could be a "push transmission" by a sender such as an Internet/DTN gateway to all interested parties.

In fact, some 3G and digital broadcast networks in developed regions provide commercial services for mass push content distribution, e.g., the EZ Channels Plus service by the Japanese operator KDDI [4], which demonstrates the case for scalable, asynchronous mass distribution of self-contained information units. In [13] we have described DTN-based web access in different configurations and have performed measurements with different link characteristics.

Applying these concepts to a DTN-based scenario such as depicted in figure 1 requires an approach for **multicast distribution** that is able **to deliver specific content**, such as a new information item on a specific "push channel" to the group of interested receivers. A corresponding **channel concept** (ideally mapped to channels of underlying networks such as IP multicast groups or radio channels) should be applied in order to use transmission (and receiver) resources efficiently. Senders should **repeat the content periodically**, as they cannot assume that all receivers are reachable at the same time. In order to use possibly expensive broadcast resources such as satellite links efficiently, it should be possible to apply **prioritizing schemes**, i.e., in order to allow for differentiated services, e.g., repeated distribution of entertainment material should be transmitted with a lower priority than emergency alerts.

Unlike the broadcast distribution of emergency alerts, the mass distribution of multimedia content requires some way of **signaling**, i.e., in order to allow interested receivers to *subscribe* (and *un-subscribe*) to content. E.g., content may be of regional interest only and it should be avoided to unnecessarily flood local distribution infrastructures.

3. DISTRIBUTION ARCHITECTURE

The analysis of the requirements for the scenarios described in section 2 has led to the development of a DTN-based distribution

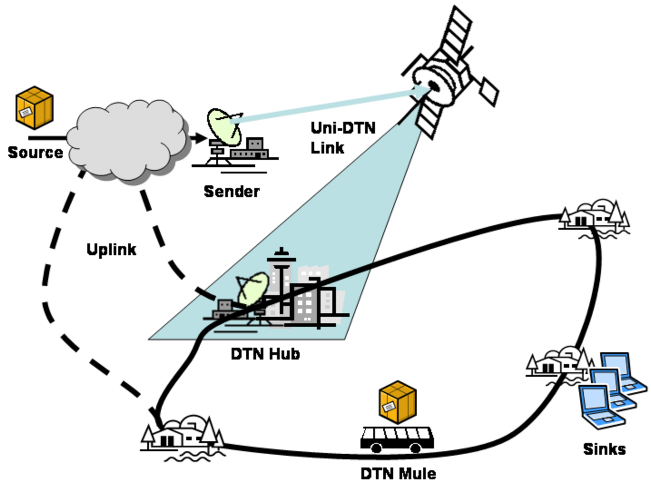

Figure 2: Distribution Architecture

architecture that is intended to provide scalable DTN bundle distribution in certain, essentially source-specific, multicast scenarios. In the scenarios described above the senders in the un-challenged part of the network (here: *sources*) are well connected while the receivers in the challenged part (here: *sinks*) are not. We assume that the source has access (directly or indirectly) to a wide-area broadcast medium that covers the area where the receivers are located. Examples of such media are satellite links, terrestrial broadcast systems like DVB-T or ISDB-T, and long-distance WLAN links. For most of these link layer technologies, it is unrealistic to assume that each sink, i.e., each user, has the capability to receive data over the broadcast medium. For reasons of cost-efficiency and power conservation, receivers on broadcast links would rather be established as routers or kiosks acting as *DTN hubs* and forwarding data via a local network such as regular WLAN links to data-mules, which would then eventually deliver bundles to the intended sinks in a certain region. Users at local sites may also have a way to send data using either a data mule service or a wide-area wireless link, which may be used for sending content subscriptions — however this is optional in our architecture.

Our system model is illustrated in figure 2. It consists of the following elements: 1) **A source** distributing data via the DTN overlay network. It is not necessarily the originator of the data itself, but may also be an aggregator or an application gateway. 2) **A sender** (a DTN bundle router) capable of transmitting over a broadcast link. It may be the source itself, but can also be a different endpoint multiple hops away from the source. 3) **One or more DTN hubs** capable of receiving from a broadcast link. They serve as DTN routers for the DTN mules and the sinks. 4) **DTN mules** may optionally act as intermediaries and carry data from the hubs to 5) **any number of sinks** that are the final recipient of the DTN bundles. The sinks may be co-located with DTN hubs but this is not required. Optionally, sinks can have access to egress routers such as the regular DTN mule that allow them to transmit data to the source.

We have identified two main technical challenges in this system. 1) **Defining a DTN convergence layer operating over a unidirectional broadcast link**, providing reliability and allowing for prioritization of certain bundles, which is addressed in section 4. 2) A scheme for **routing over a broadcast link** that is general enough to be applicable to a wide range of scenarios and still simple enough to be actually useful and deployable in existing DTN configuration, which is discussed in section 5.

4. UNI-DTN

For uni-directional transport of DTN bundles, we have defined a DTN convergence layer (CL) based on the FLUTE protocol named Uni-DTN [8][1]. This section provides a brief introduction to FLUTE and its operation within the Uni-DTN CL and describes how services such as periodic, concurrent transmissions of bundles and prioritization can be realized. It should be noted that, in principle, the notions of convergence layer multicast as presented in this section and DTN multicast are orthogonal to each other, i.e., it is possible to forward a DTN bundle to a singleton endpoint over a multicast Uni-DTN channel, but it also possible to treat a unicast Uni-DTN channel as a point-to-point DTN link that is used to convey both unicast and multicast DTN bundles.

4.1 FLUTE

FLUTE [15] is a file delivery protocol for unidirectional transport. It provides reliable transport by combining forward error correction (FEC) and repeat transmission in a data carousel. Transmitted files are divided into *source blocks*, to which FEC a sender adds parity symbols (using a configured FEC algorithm and a configured redundancy rate). This block-based FEC mechanisms allows senders to configure the redundancy rate according to the expected (or observed) packet loss rates on a given link, so that the average number of required retransmission through the data carousel can be kept to a minimum. From the underlying Asynchronous Layered Coding (ALC, [10]) and Layered Coding Transport (LCT [11]) building blocks, FLUTE inherits the concept of a session, which is named "file delivery session". A session consists of a set of logically grouped ALC/LCT channels associated with a single sender sending packets with ALC/LCT headers for one or more objects. A channel is defined by the combination of a sender and an address associated with the channel by the sender.

For FLUTE, the objects transmitted in ALC/LCT sessions are either files or File Delivery Tables (FDTs), a concept introduced by FLUTE for specifying properties of the transmitted files (or the delivery itself) such as file size, FEC parameter specifications, aggregate sending rate, file identification, MIME type etc. Multiple files may be transmitted in a single session, i.e., the FDT of a FLUTE session is a set of description entries for the files to be delivered in a corresponding FLUTE session.

4.2 Convergence Layer

The Uni-DTN convergence layer is based on FLUTE's notion of file delivery sessions. A Uni-DTN CL link is mapped to a FLUTE file delivery session, and DTN bundles are delivered as files in these sessions. Especially in multicast scenarios, FLUTE is typically used in a periodic transmission mode, i.e., files are transmitted more than once in order to give late-joiners in a session the chance to receive them completely, and also to compensate for transmission problems. We use Uni-DTN in a similar fashion, i.e., for long-lived sessions where bundles are periodically transmitted.[2]

[1]See https://prj.tzi.org for details and implementations

[2]It should be noted that, although Uni-DTN provides a unidirectional DTN bundle transport only, DTN transport services such as custody transfer that rely on bi-directional communications, are not necessarily excluded in DTN scenarios that use Uni-DTN links. It depends on the specific connectivity graph in a given DTN scenario, e.g., on the existence of DTN paths between custodians, whether custody transfer can be provided.

The Uni-DTN CL extends the FLUTE FDT to include DTN specific information such as the EIDs of the destination and the router sending the bundle (i.e., the FLUTE sender). The unique bundle identification is conveyed in the content location field of the original FLUTE FDT.

4.3 Concurrent Transmission of Multiple Objects within One Flute Session

FLUTE splits files into blocks which may be encoded with FEC schemes such as Reed Solomon. These blocks are split into pieces, to which an identification header is assigned before they are sent as a packets over the FLUTE session. Transmission of multiple files can be realized by sequentially or concurrently multiplexing pieces of different files onto a single FLUTE session. A FLUTE session may consist of multiple channels (i.e., IP multicast groups), which can be used for receiver-controlled congestion control.

In order to provide receiver-controlled FEC-based reliability and congestion-control for challenged environments, we have designed the Uni–DTN FLUTE implementation so that it supports distributing packets over multiple channels: The channel with the lowest bandwidth distributes the file once, while the other channels are filled with FEC. Thus, a receiver receiving on the slowest channel only can eventually receive a full copy, while other receivers can optionally use the additional FEC data to reconstruct the file as soon as enough packets are received. The simplest way to send multiple files concurrently is to rotationally send one piece of every file on each channel as depicted in figure 3) . The complete transmission rate is divided equally between the files.

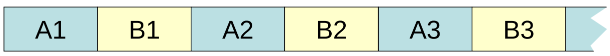

Figure 3: Concurrent transmission of two files

In order to allow for different priority classes when transmitting multiple bundles in a single session, we have added a flexible prioritization scheme that is based on configuring different sending rates for file packets in a FLUTE session. E.g., figure 4 depicts a multiplex scheme for transmitting two files with two thirds of the available bandwidth for one file and one third of the bandwidth for the other file on all channels.When adding a file to a FLUTE session we specify the packet ratio as proportion rate of $1/x$ where x depicts the number of packets to be used when rotating through all available files while filling the output buffer. I.e., in the example above the first file (A) would be assigned a proportion of $x = 2$ and the second file a proportion of $x = 1$. The DTN bundle specification [19] defines a two bit priority field with three classes of service ("bulk", "normal", and "expedited") that indicate the bundles priority. These priorities are mapped to flute priorities when used with Uni-DTN.

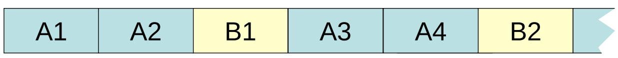

Figure 4: Concurrent transmission at different rates

5. ROUTING

DTN (Multicast) routing is currently still an open research issue. For applying Uni-DTN to scenarios such as those as described in section 2, we are proposing a specific, pragmatic approach that is presented in the following.

Identifying interested receivers for some content that can be distributed as multicast DTN bundles over DTN-Uni is based on a naming concept for multicast content/channels, a subscription-based group membership management approach, and a receiver-based selection process for contacts between routers (and sinks).

The **naming concept** is straightforward: we assume a concept of channels (multicast groups) that are identified by DTN endpoint identifiers (EIDs), e.g., `dtn://alerts.met.agency/tsunami`. Potential sinks use these identifiers to **subscribe** to corresponding channels when they establish contact with DTN routers, e.g., the DTN data mule. After establishing a contact between two nodes `A` and `B` (using the DTN Neighbor Discovery protocol[3]), the following exchange takes place:

1. Node `A` sends (over a local DTN link) `SUBSCRIBE` messages for each channel it is interested to node `B`. These messages may carry parameters such as the subscription lifetime and qualifiers for selecting specific content from a channel, such as a content ID.

2. Node `B` process the subscription and adds the subscribed channel to its list of channels it is interested in.

3. If node `B` already has content for that channel that can satisfy node `A`'s request, it tries to deliver the corresponding bundles over the existing DTN link.

It should be noted that this exchange can happen in both directions, i.e., there is no clear receiver-router relationship. It should also be noted that routers do not maintain a mapping of channel EID to subscriber EIDs. At each contact, the potential receiver issues their subscription requests and matching bundles will be delivered.

The DTN hub, i.e., the FLUTE receiver is one of these routers. We assume that the subscription requests of a region will eventually reach it, e.g., over the data mule-based forwarding service. The DTN hub has access to different Uni-DTN sessions on different FLUTE-channels. Relying on the Uni-DTN CL identification mechanism it can match FDT entries to its subscription list and *tune into* FLUTE sessions with relevant content for its subscribers. The accumulated DTN bundles will be stores and then later be forwarded to DTN data mules applying exchange described above. Subscriptions are timed-out based on the specified life-time, which eventually results in a pruning of the distribution paths. As an optimization, the subscriptions may also be delivered to the actual sources (over a potentially slower uplink as depicted in figure 2), thus allowing them to adapt the content distribution to the actual demand. In anyway, the source sends multicast bundles to the senders using bundle-in-bundle-encapsulation [21] so that the intermediary routers do not have to be aware of the multicast traffic. We assume that sources either know the senders' EIDs (e.g., by static configuration) or that they can learn them dynamically, i.e., by receiving subscription message from the DTN hubs.

6. EVALUATION AND CONCLUSION

In order to assess the feasibility and the performance of our approach, we have implemented the Uni-DTN convergence layer and have performed a series of measurements. The implementation is based on two main components: 1) *RDTN*, our Ruby-based DTN bundle router and 2) *Papageno*, our RFC 3926 compliant FLUTE

[3]http://www.dtnrg.org/wiki/NeighborDiscovery

implementation in C++[4]. RDTN implements three DTN convergence layers: TCP, UDP and Uni-DTN.

We have conducted a series of bundle transmission tests using our implementations with the Kasuari emulation framework [14][5]. Kasuari is an emulation framework that provides emulating the behavior of network links between virtual Linux nodes on a physical host computer. The link behavior is emulated in real-time, e.g., according to mobility simulations, using an adopted version of the ns2 network simulator [6]. Kasuari is based on *Xen*[7] virtual machines. These systems are connected via virtual network interfaces that can be controlled from the host system, so that the properties of a challenged link can be modeled. This allows us to use our actual implementation for the experiments instead of one that is only written for a simulator.

The emulation setup for Uni-DTN uses a dedicated sender that transmits bundles at regular intervals to a group of receivers. The links are interrupted frequently, so that the receivers need to rely on forward error correction and redundancy to be able to receive bundles. The loss rate has been varied over different measurement series. The main objective of these tests was to show that a FLUTE-based convergence layer can reliably transmit arbitrary DTN bundles in a uni-directional communication scenario in the presence of packet loss while a naive UDP convergence layer is simply not sufficient. Therefore, we have implemented a simple UDP convergence layer that is intended to behave as the corresponding convergence layer in the DTN reference implementation[8]: bundles are sent directly as packets without any additional headers. Consequently, the UDP convergence layer cannot handle arbitrary packet sizes as FLUTE can, so we need to apply fragmentation. In order to obtain comparable results, the number of completely reassembled bundles and not received fragments is taken into account.

As can be expected, when using the unmodified UDP convergence layer the bundle delivery rate is heavily affected by packet loss and is well beyond 50% for loss rates of 10% (bundle size of 1MB). For the FLUTE case, the delivery rate essentially depends on the redundancy rate and the number of retransmissions one is willing to accept. In typical configurations, the redundancy rate is therefore tuned to enable a complete reception in one transmission for average conditions. For uniformly distributed packet losses, it is possible to mathematically compute the performance under different configurations. See [16] for a detailed performance analysis. E.g., the authors of [16] have analyzed the performance of FLUTE's FEC mechanism under uniformly distributed errors considering different loss rates and FEC rates. For a loss rate of 10%, a FEC rate of 5% yields an average (and maximum) number of required transmission rounds of 2. For a 25% FEC rate both the average and maximum number of requires transmission go down to 1. This has been confirmed by earlier real-world measurements that we had done with our FLUTE implementation [9].

In summary, leveraging FLUTE as a transport for a uni-directional DTN convergence layer is clearly useful and certainly preferable over a UDP-only approach. FLUTE can cope with transmission failures with reasonable efficiency (without requiring feedback loops) and can be adjusted to different scenarios and different requirements with respect to link error rates and bandwidth constraints. The Uni-DTN convergence layer is directly using FLUTE's file transmission abstraction, which seems to be a good fit to DTN bundle transmission. Our extensions allow receivers to efficiently identify bundles and their destinations, which is an important feature for enabling efficient receiver operations and bandwidth utilization. The prioritization scheme is a straightforward implementation of the priority classes concept for the DTN bundle transport, which fits well to satellite environments and carousel-based transmission.

Our distribution architecture is a pragmatic solution to mass content distribution in specific environments. Although it is not intended as a general purpose multicast architecture, we believe that it can be applied to other scenarios, too, where only source-specific multicast is required – and where sinks remain in a geographic context with their upstream routers. The fundamental concept is to extend the scope of the DTN network onto the large-scale distribution network and not to limit it to local opportunistically connected nodes. We believe that the value of such networks for developing regions will be much higher when overall transport delays can be shortened and mass content can be made available. The DTN hub concept and the integration with local data mule-based delivery is intended to make this approach economically feasible.

So far, results from practical experiences with that architecture are not yet available. We are currently working on a real-world implementation that includes the Neighbor Discovery and multicast signaling procedures as proposed in section 5. Our future work will also include integration of other DTN-layer multicast approaches such as DTN multicast custody transport, and we are working on applying our architecture to DTN multicast distribution in other scenarios.

[4]More information on the implementation and its components is available at http://prj.tzi.org.

[5]http://www.kasuari.org

[6]http://www.isi.edu/nsnam/ns/

[7]http://www.cl.cam.ac.uk/research/srg/netos/xen/

[8]http://www.dtnrg.org

7. REFERENCES

[1] Eric Brewer, Michael Demmer, Bowei Du, Melissa Ho, Matthew Kam, Sergiu Nedevschi, Joyojeet Pal, Rabin Patra, Sonesh Surana, and Kevin Fall. The Case for Technology in Developing Regions. *IEEE Computer*, 38(6):25 – 38, 2005.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H.Weiss. Delay-Tolerant Network Architecture. RFC4838, April 2007.

[3] C. Chyau and J.F. Raymond. First Mile Solutions' Daknet Takes Rural Communities Online. Technical report, World Resources Institute, October 2005.

[4] KDDI Corporation. KDDI to Introduce "EZ Channel Plus" Programming Service for au. Press Release, August 2006. http://www.kddi.com/english/index.html.

[5] Michael Demmer and Jörg Ott. Delay Tolerant Networking TCP Convergence Layer Protocol. Internet Draft draft-demmer-dtnrg-tcp-clayer-00.txt, Work in Progress, October 2006.

[6] NTT DoCoMo and KDDI Corporation. NTT DoCoMo and KDDI Develop Platform for Immediate Notification of Emergency Earthquake Alerts. Press Release, May 2007. http://www.kddi.com/english/index.html.

[7] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. Proceedings of ACM SIGCOMM 2003, Computer Communications Review, Vol 33, No 4, August 2003.

[8] Dirk Kutscher, Kevin Loos, and Janico Greifenberg. Uni-DTN: A DTN Convergence Layer Protocol for Unidirectional Transport. Internet Draft draft-kutscher-dtnrg-uni-clayer-00.txt, IETF, 2007. Work in progress.

[9] Dirk Kutscher, Jörg Ott, and Steffen Bartsch. Supporting network access and service location in dynamic

environments. TERENA Networking Conference, May 2007.

[10] Michael Luby, Jim Gemmell, Lorenzo Vicisano, Luigi Rizzo, and Jon Crowcroft. ALC protocol instantiation. RFC 3450, December 2002.

[11] Michael Luby, Jim Gemmell, Lorenzo Vicisano, Luigi Rizzo, Mark Handley, and Jon Crowcroft. Layered Coding Transport (LCT) Building Block. RFC 3451, December 2002.

[12] Early Warning Sub-Committe of the Inter-Ministerial Committee on International Cooperation for Disaster Reduction. Japan's Natural Disaster Early Warning Systems and International Cooperative Efforts. Technical report, Government of Japan, March 2006.

[13] Jörg Ott and Dirk Kutscher. Bundling the Web: HTTP over DTN. In *Proceedings of WNEPT 2006*, August 2006.

[14] Jörg Ott, Dirk Kutscher, and Christoph Dwertmann. Integrating DTN and MANET Routing. In *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)*, September 2006.

[15] Toni Paila, Michael Luby, Rami Lehtonen, Vincent Roca, and Rod Walsh. FLUTE. RFC 3926, October 2004.

[16] J. Peltotalo, S. Peltotalo, J. Harju, and R. Walsh. Performance Analysis of a File Delivery System Based on the FLUTE Protocol. *International Journal of Communication Systems*, October 2006.

[17] Alex Pentland, Richard Fletcher, and Amir Hasson. DakNet: Rethinking Connectivity in Developing Nations. 37(1):78–83, January 2004.

[18] Bhaskaran Raman and Kameswari Chebrolu. Experiences in Using WiFi for Rural Internet in India. *IEEE Communications Magazine*, January 2007. Special Issue on New Directions In Networking Technologies In Emerging Economies.

[19] Keith L. Scott and Scott C. Burleigh. Bundle Protocol Specification. Internet Draft draft-irtf-dtnrg-bundle-spec-08.txt, Work in progress, December 2006.

[20] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost Communication for Rural Internet Kiosks Using Mechanical Backhaul. In *MOBICOM 2006*, September 2006.

[21] S. Symington, R. Durst, and K. Scott. Delay-Tolerant Networking Bundle-in-Bundle Encapsulation. Internet Draft draft-irtf-dtnrg-bundle-encapsulation-00, IETF, July 2006. Work in progress.