IP over DTN: Large-Delay Asynchronous Packet Delivery in the Internet

Hideya Ochiai
The University of Tokyo/NICT
jo2lxq@hongo.wide.ad.jp

Kenichi Shimotada
The University of Tokyo
ken@hongo.wide.ad.jp

Hiroshi Esaki
The University of Tokyo/NICT
hiroshi@wide.ad.jp

***Abstract*—The well-known DTN divides an end-to-end communication path into hop-by-hop sessions and allows asynchronous message delivery over physically delay or disruptive network environments. They have introduced Bundling, an overlay network for delay tolerant networking in its architecture, which should be globally deployed and operated with tremendous effort. In this paper, we propose a new DTN architecture that are more practically and easily deployable in the Internet space. Our approach is IP over DTN; DTN message delivery framework works as a data link to deliver IP packets over network boundaries. The transport protocols and application protocols take asynchronous communication mode in this architecture. We have carried out an experiment with our prototype DTN framework and confirmed that applications could practically communicate well with each other even with large-delay IP packet delivery.**

I. INTRODUCTION

In delay or disruption tolerant networks (DTN), it is commonly believed that message delivery should be performed by an overlay manner: e.g., RFC4838[1]. They have introduced the Bundle layer to develop a DTN, which has its own ID space and transportation schemes. This approach certainly allows performance optimization at the networks of large-delay or intermittent connectivity, including interplanetary networks[2], village-to-village networks in developing regions[3] and mobile ad-hoc networks[4]. However, this overlay approach forces us to deploy a new global network infrastructure over the global Internet, which certainly needs great effort. The feasibility is not clear as well.

In this paper, we propose a new DTN architecture – IP over DTN, which is more practically and easily deployable in the Internet space. In the architecture, DTN message delivery framework works as a data link and delivers IP packets by encapsulating them with DTN frames. The upper layer protocols must take asynchronous communication mode, or in other words, a postal communication mode. The widely-studied DTN related works, including PEAR[5], MaxProp[6], DTLSR[3] and Spray and Focus[7], can be implemented at the data link layer.

The term *DTN* used in most part of this paper, is different from the well-known DTN: i.e., RFC4838[1] and DTNRG[8]. We use the term *well-known* to distinguish them from our DTN framework. Our DTN framework has only to deliver typical-size (i.e., about 1500 octet) IP packets, which simplifies the design and the implementation of DTN frameworks. To send a large message, it should be divided into multiple IP packets. In IP over DTN architecture, the transport layer or application layer should care about the delivery of application protocol data units (APDUs) (see Section 5).

IP networks deliver packets in the best effort manner. Basically, any latency is allowed and any packet loss is allowed (of course, not preferable). By taking a DTN framework as a data link, IP packets can be stored, physically carried, and forwarded to isolated networks over the challenged environment even though it takes times. Then, we will witness large-delay IP packet delivery; e.g., a receiver gets IP packets sent minutes, hours or days before.

There are two communication mode that end-to-end applications take: *synchronous* and *asynchronous* mode [9]. In the synchronous mode, both communication ends are online at the same time and make interaction between them: i.e., hand-shaking. In the asynchronous mode, both communication ends do not always need to be online at the same time. The asynchronous mode is also known as a postal model of communication. The sender can transmit their messages even when the receiver is not online. The receiver gets the messages from the network even after the sender has left it.

The communication failure and performance reduction at large-delay networks [10] comes only when applications try to take synchronous communication mode, for example, by TCP. Generally, in large-delay networks, the communication between end-to-end applications must shift from synchronous mode to asynchronous mode. The well-known DTN avoids end-to-end synchronization by developing an overlay and dividing an end-to-end session into hop-by-hop sessions. Our approach also avoids end-to-end synchronization by taking and enhancing the asynchronous delivery capability that IP networks originally have. Therefore, IP over DTN architecture only uses asynchronous communication protocols. As IP multicast practically works by taking asynchronous transport protocols like UDP instead of TCP, we believe that, in IP over DTN, applications can be practically developed over the network even with this constraint.

This paper is organized as follows. In section 2, we discuss related works. In section 3, we propose IP over DTN architecture. Then, we present our prototype implementation and experiment in section 4. In section 5, we provide a discussion. Finally, we conclude this paper in section 6.

II. RELATED WORK

The concept of IP over DTN itself is simple; however, we could not find this approach in literature. A DTN architecture

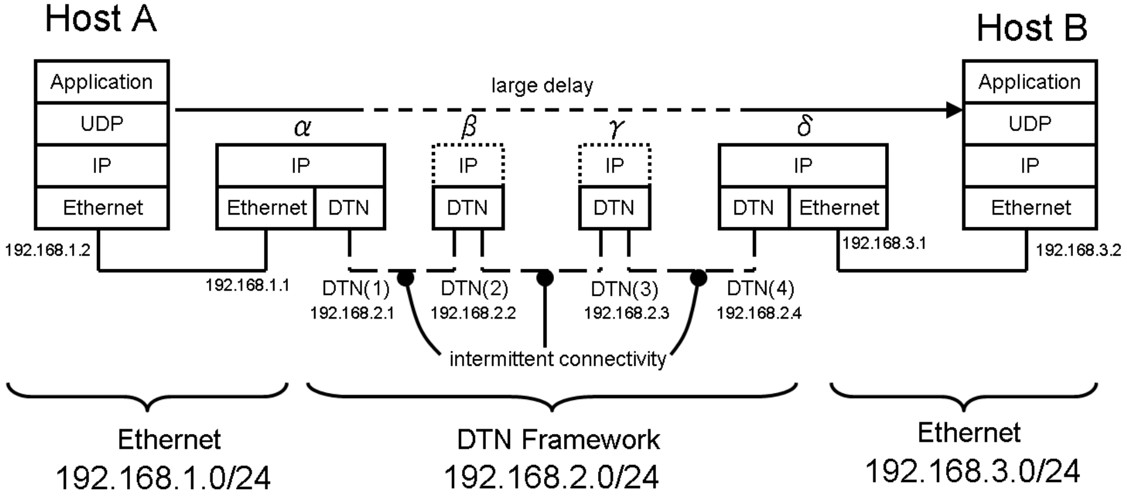

Fig. 1. IP networking over a DTN framework

was proposed by Burleigh et. al. [2] and Kevin Fall[11], and the internet engineering task force(IETF) has published the architecture by RFC4838[1]. The architecture has introduced the Bundle layer, which deploys DTN framework as an overlay network and delivers messages by APDU. This approach has come from the study of TCP performance reduction and failure over large delay and high packet loss networks[12]. Robert et. al.[10] has mentioned that in deep space communication, TCP fails because of the large packet propagation delay between the two sites. They proposed with the architecture to divide an end-to-end communication session into hop-by-hop sessions and presented that it should improve the performance of end-to-end throughput.

However, there can be another point of view. In large delay or highly topology disruptive networks, the mode of communication must be shifted from synchronous to asynchronous mode. IP over DTN, our approach, takes asynchronous communication mode from the start by using DTN framework as a data link, and enhance asynchronousness that IP networks originally have. IP over DTN architecture uses UDP-based or UDP-like asynchronous communication protocols in the upper layer. Burleigh et. al.[2] has discussed the possibility of UDP-based communication in interplanetary scenarios, however, they basically suppose synchronous communication (e.g., data acknowledgement and re-transmissions) at the application level in their paper; although in most of the time of discussion with regard to DTN, they suppose a postal communication model.

IP multicast has given up using TCP-based reliable communication protocol, because synchronization of multi-point states is a quite heavy task[13]. In this regard, our approach is the same; i.e., IP over DTN can give up using TCP, because synchronization of end-to-end point states is impossible. However, IP over DTN will be more deployable than IP multicast. The difficulty in IP multicast dwells in management issues over the global Internet[14], whereas in IP over DTN, a DTN framework can be locally managed as a data link.

III. IP OVER DTN

The TCP/IP is originally designed under a wired environment, where packet loss and delivery latency is small enough to establish synchronized communication, for example, by TCP. Most of the communication in the current Internet is practically implemented with such synchronized manner. However, the delay or disruptive properties of the network that we targeted at in this paper potentially forces the mode of communication to be asynchronous. This means that end-to-end TCP-based communication is no longer a proper communication style. The well-known DTN has achieved the shift to asynchronous by an overlay called bundling. However, we emphasize that IP itself basically provides asynchronous packet delivery mechanism. IP packet delivery can be delayed if the applications assume large-delay and asynchronous communication mode.

This has motivated us to use a DTN framework as a data link in IP networking. Figure 1 shows an overview of IP over DTN architecture. In this example, two Ethernets are connected via a DTN framework, which delivers IP packets from one gateway to the other with large delay. Host A sends a message to host B by an asynchronous protocol on IP (e.g., UDP-based asynchronous application protocol). We describe here how the message can be delivered to host B.

1) The application at host A divides the message into a sequence of datagrams (i.e., a sequence of IP packets).
2) With referring to its routing table, host A forwards the packets to node α (192.168.1.1), the gateway to the DTN framework.
3) The gateway α, which is named ”1” in DTN framework, looks up its IP routing table and finds that to relay the packets to host B, it should forward them to 192.168.2.4 and to do so, the packet should be encapsulated by DTN frames and sent to DTN(4) – node δ, the gateway to host

B. Here, IP to DTN-address mapping (i.e., functionally equivalent to ARP table) is managed and shared in the DTN framework.

4) The DTN framework sends the frames to DTN(4) by its own message delivery scheme.
5) Finally, the gateway δ gets IP packets from the DTN framework, and forwards them to host B via the Ethernet.

In this way, DTN framework can be implemented into IP networks.

We here summarize the properties that DTN framework we discussed above should have. They are basically the same as the well-known DTN framework except regarding to the maximum length of the protocol data unit. This difference simplifies the design and implementation of the framework.

- The maximum length of the protocol data unit is around 1500 octet – enough length to encapsulate a typical IP packet.
- When the next hop of a DTN frame not found in the transmission range, the node stores it in its local buffer and try re-transmission when it encounter the next hop node.
- RTT between any two nodes is basically much larger than typical data links.
- The DTN framework has its own ID space for data delivery in the network.

IV. Implementation and Deployment

We have developed a prototype implementation into Linux. The DTN framework we used in the prototype is PEAR[5]. PEAR (1) computes routing tables at the DTN level with potential-based schemes, and (2) replicates messages in the network in order to increase delivery probability. PEAR carries out these two procedures autonomously based on the movements or contact patterns of DTN nodes. In order to manage and share IP to DTN-address mapping over the DTN framework, we have made some modifications to the PEAR source code.

With our prototype, we carried out an experiment of integration of a DTN framework into an IP network. Figure 2 shows the detailed configuration of the experiment.

We prepared five network segments; four were deployed on Ethernet and one was deployed on PEAR. Each network was connected by IP routers as Figure 2 shows. We used three DTN nodes named α, β and γ. These three nodes have 802.11g radio interface with ad-hoc mode through which they exchange messages in DTN manner. Node α and γ were also networked by Ethernet and worked as IP routers. They were deployed in separate so that they cannot directly communicate with each other. In the experiment, they stored and forwarded IP packets, between Ethernet and the DTN framework. We physically carried node β, an embedded node with battery powered (Figure 3), between the two stations around α and γ in 10 minutes per cycle.

We setup two hosts A and B in the network as an Internet application. Host B provides an image repository service, and host A tries to retrieve an image from host B. The communication protocol is implemented over UDP, and the application is designed assuming large-delay message delivery. In the experiment, host A sent an image request query to host B, and host B returned the image by five to nine UDP segments depending on the image size. Though the overall communication style is synchronous, the delivery of each APDU (i.e., query and image data) was done in asynchronous manner.

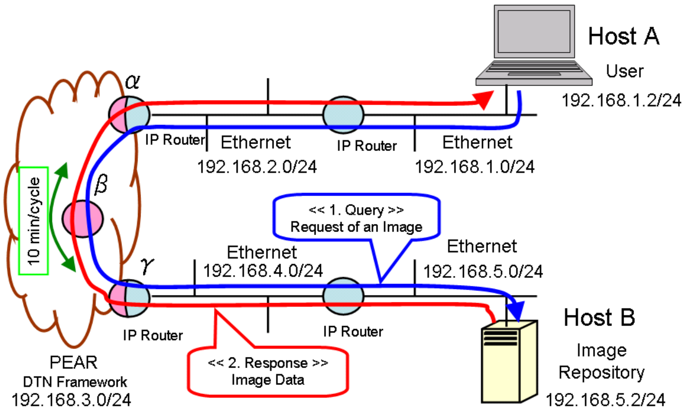

Fig. 2. The detailed configuration of the experiment

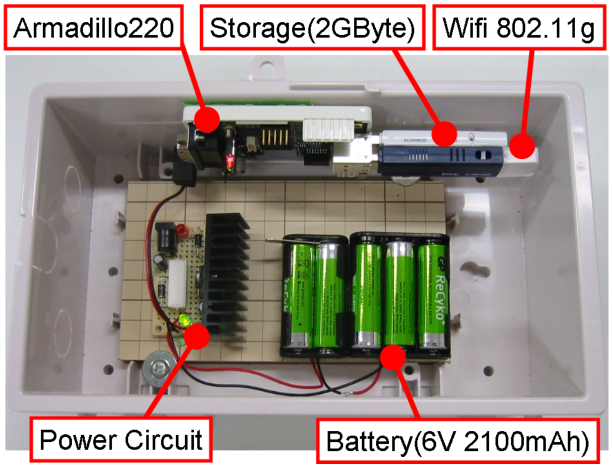

Fig. 3. The DTN node – embedded implementation

Figure 4 illustrates the IP packet delivery pattern we observed in this experiment for one request-to-response transaction.

The blue arrows show request delivery pattern, and the red arrows show image data delivery pattern. The number along with the arrows shows the time in second when the transmission event had happened. The number inside the bracket shows the number of packets transmitted at the same time. Host A submitted a query to host B at time 0[sec], which was stored at α. At time 162[sec], β has contacted to α and got the query from it. β has arrived at γ and handed the query to it, then the network droved the packet to host

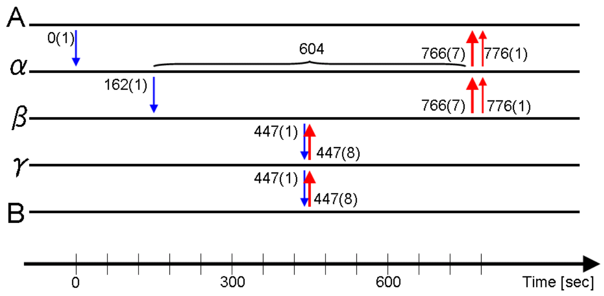

Fig. 4. IP packet delivery pattern

B at time 447[sec]. The requested image was sent by eight UDP segments at 447[sec] and stored at β via γ. β has re-contacted to α and send seven packets at 766[sec] and the last one packet at 776[sec]. Here, the last one packet was also transmitted at 766[sec] but failed to receive at α thus re-transmitted 10 seconds later. The total time to retrieve an image over the network was 776[sec].

V. DISCUSSION

As our experiment has shown IP over DTN applications must aware the delay of APDU delivery; the host A had to wait 776 seconds to receive the requested image data. This requires the design of application to be DTN-aware – the delay should be supposed at the early stage of its protocol design. However, this is not a problem because even in the well-known DTN, application protocols should be aware of the delay[15].

In the well-known DTN, to get reliability at message delivery, they proposed to use data acknowledgement and retransmission scheme[1], [2], [11]. This scheme can be used in some DTN application scenarios. However, it basically works effectively only at short round trip environment, because it is a kind of synchronous communication. Round trip in DTN is costly, thus, another scheme to get reliability should be also needed.

Forward error correction (FEC), which is known as a method to get reliability in poor data transport systems, can be alternatively used for this purpose. Large-delay asynchronous contents delivery media (e.g., compact disks and DVDs) and high-rate frame-loss media (e.g., WiMAX[16]) have adopted this kind of technology to improve reliability without data acknowledgement. FEC detects data error at the receiver side, and corrects them if the damage is recoverable. FEC approach for getting reliability might practically work well in many DTN scenarios.

In this work, we have not focused on name-to-address resolution over large-delay environment, which we have left as an open research issue. Sharing IP-address to DTN-address mapping (i.e., ARP table) could be easily implemented in our experiment platform. Sharing DNS-name to IP-address mapping should be studied in the future.

VI. CONCLUSION

We proposed a new DTN architecture – IP over DTN, which DTN framework works as a data link and manages to deliver IP packets over the challenged network environment. In this architecture, transport protocols and application protocols take asynchronous communication mode.

Compared to the well-known DTN architecture proposed by [1], [2], [11], our architecture is more practically and easily deployable in the Internet space.

We have developed a prototype system and carried out an experiment. The experiment has shown that IP packets were certainly delivered over isolated networks with large-delay but the applications succeeded to exchange their messages by taking asynchronous communication mode.

REFERENCES

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "RFC4838: delay tolerant networking architecture," apr 2007.

[2] 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, no. 6, pp. 128–136, jun 2003.

[3] D. Demmer and K. Fall, "DTLSR: Delay tolerant routing for developing regions," in *ACM NSDR*, 2007.

[4] J. Ott, D. Kutscher, and C. Dwertmann, "Integrating DTN and MANET routing," in *ACM SIGCOMM workshop*, 2006, pp. 221–228.

[5] H. Ochiai and H. Esaki, "Mobility entropy and message routing in community-structured delay tolerant networks," in *ACM AINTEC*, 2008, pp. 93–102.

[6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *IEEE INFOCOM*, 2006.

[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, feb 2008.

[8] "DTN research group," http://www.dtnrg.org/.

[9] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, jun 2003.

[10] R. C. Durst, P. D. Feighery, and K. L. Scott, "Why not use the standard internet suite for the interplanetary internet?" http://www.ipnsig.org/reports/TCP_IP.pdf.

[11] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, 2003, pp. 27–34.

[12] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, jun 1997.

[13] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784–803, dec 1997.

[14] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, pp. 78–88, jan 2000.

[15] J. Ott and D. Kutscher, "Bundling the Web: HTTP over DTN," in *In Proceedings of WNEPT*, 2006.

[16] "IEEE 802.16e," http://wirelessman.org/tge/.