Performance Evaluation of DTN Implementations on a Large-scale Network Emulation Testbed

Razvan Beuran
Hokuriku StarBED Technology Center,
National Institute of Information and Communications Technology,
Ishikawa, Japan
razvan@nict.go.jp

Shinsuke Miwa
Hokuriku StarBED Technology Center,
National Institute of Information and Communications Technology,
Ishikawa, Japan
danna@nict.go.jp

Yoichi Shinoda
Research Center For Advanced Computing Infrastructure,
Japan Advanced Institute of Science and Technology,
Ishikawa, Japan
shinoda@jaist.ac.jp

ABSTRACT

In this paper we present a series of experiments that evaluate the performance of two DTN implementations, DTN2 and IBR-DTN, in urban mobility scenarios. The experiments were carried out on the wireless network emulation testbed named QOMB, which was extended to support such DTN evaluations. Our quantitative assessment verified the basic behavior of the DTN implementations, but also identified scalability issues for DTN2 in scenarios with as few as 26 nodes. These results emphasize the need for more extensive large-scale experiments with DTN applications and protocols for comprehensive evaluations in view of functionality validation and performance optimization. This can be readily achieved through the use of emulation testbeds such as the one that we have developed.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques

Keywords

Performance evaluation, DTN implementation, network emulation

1. INTRODUCTION

The concept of Delay Tolerant Networks (DTN) was first put forward in connection with deep-space communications and the so called "Interplanetary Internet" [2]. However, delay is not the only type of impairment that can occur, and the terms "Disruption Tolerant Networks" (using the same acronym, DTN) or "challenged networks" started to be used as well [6]. This made it clear that the DTN paradigm can be applied to a wide range of network categories, such as mobile networks, ad hoc networks, or sensor networks.

CHANTS'12, August 22, 2012, Istanbul, Turkey.

We estimate that in the near future the concepts related to DTN will find more and more applications to everyday scenarios. However, a wide-scale deployment of DTN protocols and applications cannot be made without a proper evaluation of the corresponding implementations in realistic scenarios, so as to ensure that they perform as expected under various circumstances. In this paper we evaluate the DTN2 reference implementation [4] and also the IBR-DTN implementation [10], focusing on the quantitative assessment of their performance characteristics in urban mobility scenarios, in which nodes move in a realistic manner in a virtual city-like environment.

For experiments we used the wireless network emulation testbed named QOMB, developed at the Hokuriku StarBED Technology Center, National Institute of Information and Communications Technology, located in Ishikawa, Japan [1]. QOMB functionality was further extended as required for DTN-related experiments. The main advantage of using QOMB when compared to previous approaches is that it makes possible realistic experiments at large scale, including aspects such as node mobility, while using real implementations of the DTN protocol. Such experiments are not possible otherwise, neither by using existing (fixed) network testbeds nor through simulation.

The main contributions of this paper are:

- We evaluated for the first time through emulation the DTN2 reference implementation and the IBR-DTN implementation in several urban mobility scenarios;
- We identified scalability issues of the DTN2 reference implementation in scenarios with as few as 26 nodes, thus demonstrating the need to do more extensive experiments for functionality and performance assessment.

The remainder of this paper is organized as follows. In Section 2 we present a brief overview of QOMB and its extension for DTN experiments; we also summarize the main features of the DTN implementations that we evaluated. Then, in Section 3, we discuss a series of experiments that assess the performance of the DTN2 and IBR-DTN implementations through emulation in several realistic scenarios. In Section 4 we introduce some related work that was carried out in the context of DTN. The paper ends with conclusions, acknowledgments and references.

2. DTN EMULATION TESTBED

2.1 Overview

QOMB is a wireless network emulation testbed initially created for IEEE 802.11 network emulation [1]; its main components are StarBED and QOMET. StarBED is a large-scale wired-network testbed of the National Institute of Information and Communications Technology, Japan. With over 1100 interconnected PCs, users can perform a wide range of network experiments on StarBED, which is the physical infrastructure of QOMB.

QOMET (Quality Observation and Mobility Experiment Tools) is a set of tools for wireless network emulation that can be run on StarBED. QOMET allows the definition of various complex scenarios, including node mobility and urban settings. Several new features and components were required in order to make DTN emulation experiments possible on QOMB, but they will not be discussed here because of lack of space.

QOMB experiment hosts have different characteristics depending on their "age". In the experiments presented in this paper we used machines with Intel Pentium 4 3.2 GHz CPUs with 1 Gbps Ethernet network interfaces; each emulated node was run on a single physical machine.

2.2 DTN Implementations

We used two DTN implementations on our testbed. One is DTN2, the reference implementation by the Delay Tolerant Networking Research Group (DTNRG) of the Internet Research Task Force (IRTF) [4]. The other is IBR-DTN, which is "a lightweight, modular and portable bundle protocol implementation and DTN daemon" [10].

In our evaluation we used the applications called *dtnping* and *dtnperf* in DTN2. We also used the routing protocols called *flood* and *dtlsr*. The dtnping application is also available in IBR-DTN, but the dtnperf application is missing. The authors of IBR-DTN reportedly relied on the commands "dtnsend" and "dtnrecv" plus some custom code for this purpose, which does not allow the reproduction of the results by independent parties. Although IBR-DTN lacks the dtlsr protocol implementation from DTN2, it does provide an equivalent to flood routing. For both implementations links were established using a discovery protocol.

3. EXPERIMENTAL RESULTS

In this section we present several results obtained by running DTN2 and IBR-DTN on our emulation testbed. The experiments that will be presented next were preceded by a series of preliminary experiments that confirmed the basic operation of the two implementations, including in simple 2-3 node emulation scenarios. Due to the lack of space we omit the detailed presentation of the results. However, we want to note that during these experiments we have discovered that, depending on conditions, the RTT shown by the dtnping implementation in DTN2 is at best of several tens of ms, which is about 10 times larger than the RTT shown by the dtnping in IBR-DTN, and even 40 times larger than the RTT shown by the typical ping command on those experiment hosts. The dtnperf results were nevertheless close to expectations, which makes us suspect the existence of a bottleneck in DTN2 that has a significant effect for short transactions, such as those in dtnping, but becomes concealed in long ones, such as the dtnperf tests. We also noticed that it takes around 30 s for dtlsr routing in DTN2 to reestablish a link in multi-hop communication with mobility.

3.1 Scenario overview

Following the basic experiments presented so far, we proceeded to larger-scale experiments aimed at determining how the DTN implementations performs in realistic scenarios with multiple mobile nodes in an urban environment.

Our scenario included 25 mobile nodes plus 1 fixed communication gateway. The mobile nodes started from an initial position near the gateway, and moved towards individual destinations within a 400 x 300 m area. The main parameters for wireless communication were: transmit power 10 dBm for IEEE 802.11b, and propagation attenuation 3.32. Traffic was generated using dtnping from all nodes with the gateway as destination. The interval between dtnping requests was 10 s for DTN2 (to lower network load), and about 1 s for IBR-DTN, which does not allow changing this parameter. Each experiment run lasted for 10 minutes.

These initial experiments showed a very poor performance of DTN2 when all the nodes were sending traffic, with dtnping success rates of only 6% for flooding and 28% for dtlsr, compared to 50% in the case of IBR-DTN (which was generating 10 times more traffic per node).

We assumed that high mobility is the cause of this low performance of DTN2, and we simplified the above scenario so as to only include 5 mobile nodes, namely #1, #3, #6, #18 and #22, whereas all the other nodes are placed from the beginning at their corresponding destinations. The destinations are shown in Figure 1, which presents a snapshot from this scenario at time 250 s. To reduce the network load, we also reduced the number of senders (from the initial 26), and we run several variations of this urban mobility scenarios with 5 mobile nodes out of 26, as follows:

- *5 mobile*: All the 5 mobile nodes and the gateway send dtnping towards the gateway GW0;
- *1 mobile*: Only the mobile node #1 and the gateway send dtnping towards the gateway GW0;
- *5 fixed*: A total of 5 fixed nodes (#8, #11, #15, #17 and #20) and the gateway send dtnping towards GW0;
- *1 fixed*: Only fixed node #8 and the gateway send dtnping towards the gateway GW0.

The reason for having the gateway ping itself in our experiments was first of all in order to have a comparison element. We then realized that in the case of DTN2 even for these messages successful replies are not 100% received, demonstrating that poor dtnping performance characteristics are related to the CPU overload of the experiment hosts.

3.2 Results for dtnping

The dtnping results of these experiments are summarized in Figure 2 for DTN2 using flood and dtlsr (1 request per 10 s), and IBR-DTN using flood (1 request per 1 s). We first observe that DTN2 flood shows better performance that DTN2 dtlsr in all cases, which is counter-intuitive given that flooding should put a higher load on the network. Thus, for the experiment "5 mobile", flooding leads to almost 40% dtnping success rate, whereas dtlsr produced less than half of this. In the experiment "1 mobile", DTN2 performance increases both for flood and dtlsr, probably due to the reduced number of senders, hence the reduced network load.

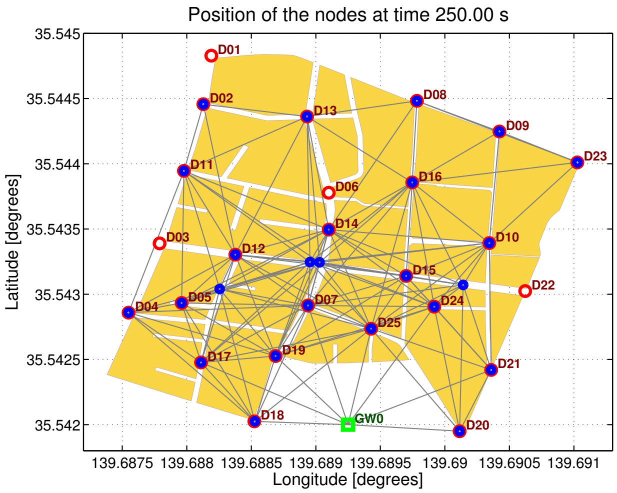

Figure 1: Urban mobility scenario for DTN experiments with 26 nodes.

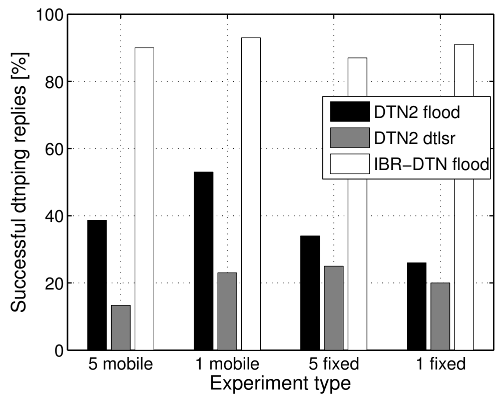

Figure 2: Successful dtnping replies for DTN2 and IBR-DTN experiments with 26 nodes.

The results from the DTN2 experiments "5 fixed" and "1 fixed" are less good compared to those of their counterparts "5 mobile" and "1 mobile", respectively. This is explained by the smaller number of connection opportunities of the fixed nodes compared to the mobile ones. The only exception is the fact that the success rate for dtlsr in experiment "5 mobile" (around 13%) is lower than the corresponding rate in experiment "5 fixed" (around 25%). This is probably justified by the fact that it is easier for dtlsr to manage the topology for the fixed nodes compared to the mobile ones.

The IBR-DTN results show a similar trend, with the results of the mobile experiments being better that those of the fixed ones, both for the "5" series and for the "1" series of experiments. However, the difference is only of a few percents, proving a better general performance of dtnping in IBR-DTN. Moreover, the absolute values around 90% are closer to the maximum value, despite the message rate of 1 request per second. On the contrary, DTN2 has less than half the same performance, even though the message rate per node is 10 times lower.

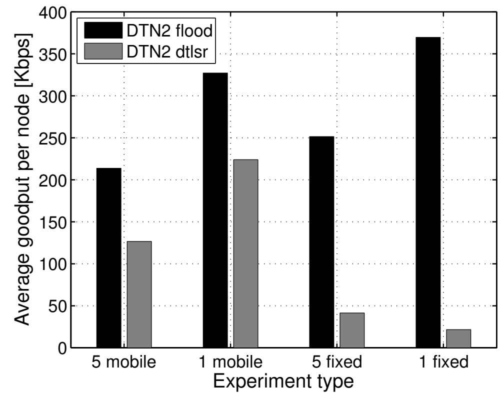

Figure 3: Average dtnperf goodput per node for DTN2 experiments with 26 nodes.

3.3 Results for dtnperf

We also conducted an evaluation using dtnperf in the same conditions as above, except that the gateway GW0 was not sending any traffic to itself. Only the DTN2 implementation could be used in this case, since IBR-DTN lacks an equivalent tool. The results in Figure 3 show again that the dtlsr performance is lower compared to flood in all cases.

We also observed an interesting effect that corroborates our previous hypothesis about the network load effect on performance for DTN2. Thus, the dtnperf results for flooding have an opposite trend to those for dtnping: "5 mobile" throughput is lower than "5 fixed" throughput, and "1 mobile" throughput is lower than "1 fixed" although the equivalent dtnping results were exactly the opposite. We interpret this as follows: high throughput as reported by dtnperf indicates good communication conditions; for dtnping this results initially in a higher number of requests being delivered to the destination, which leads to an increased network (and processing) load and then to a decrease in the dtnping success rate. Nevertheless, for dtlsr, which does not cause a significant network load, the dtnperf results mirror more closely those of dtnping.

The poor results of dtlsr compared to flood in all our experiments, while counter-intuitive, may be explained by the fact that the sparse placement of the fixed nodes decreases the network overload caused by flooding, whereas it increases the complexity of the topology that needs to be constructed by dtlsr. However, these results may also indicate an implementation issue for dtlsr, or at least some kind of misconfiguration (even though we have used its default parameters).

3.4 Reducing scenario scale

Because of the unsatisfactory results obtained in the 26 node experiment, we have not attempted to run any larger-scale scenarios, even though the testbed would have easily allowed it. Instead, we have performed a series of experiments in which we *decreased* the scale to a total of 10 nodes (out of which 3 are mobile) with between 3 and 10 traffic senders. Although we cannot present the results here due to lack of space, we note that the dtnping and dtnperf results were according to our expectations, both for DTN2 and IBR-DTN, with dtnping success rates close to 100% in most cases. An exception is the performance of the flood routing protocol in the case with 10 senders for DTN2, as the success rate was only around 40%. One may attribute this low performance to the overload caused by the packet flood as before, but this was not expected given the low rate of requests (one request per 10 s from each node, hence a total average of 1 request per second), especially when in the same circumstances IBR-DTN performance was over 90%, while having a 10 times higher request rate per node.

As a final note, we would like to stress the fact that preliminary investigations show that DTN2 and IBR-DTN performance can be improved by changing the default parameters. Such changes refer mainly to whether the processed data is being stored on disk or in memory. In this respect we plan to do an objective evaluation for various sets of parameters. Nevertheless, source code optimizations may also be required in order to achieve production-level performance characteristics.

4. RELATED WORK

Real DTN implementations have been previously evaluated, but only on a low scale. For instance, in [5] two scenarios with one sender, one receiver and up to 4 hops are compared through emulation experiments done on Emulab. The work in [9] uses 4 real wireless nodes for the evaluation, and includes a low-level performance analysis. The DTN2 and IBR-DTN implementations are compared in [10] from throughput point of view using a 2-node scenario.

On the other hand, there is a considerable number of works that do DTN evaluations at large-scale through *simulation*, such as [8], which used a realistic scenario with 600 vehicles. By contrast, our testbed allows to assess performance of *real* DTN implementations in large-scale scenarios, hence leads to results with a direct practical application.

A DTN testbed using real nodes is DTN-Bone, described as an "effort to establish a worldwide collection of nodes running DTN bundle agents and applications" [3]. This testbed currently connects around 9 institutions, but makes available only a little more than a dozen nodes. Hence, we position DTN-Bone more as an *inter-operability testbed*, given that 5 different implementations of DTN are being run on it, rather than as a testbed for DTN performance testing. Similarly, the DTN testbed presented in [7] includes only 12 geographically-spread nodes.

5. CONCLUSION

In this paper we employed a DTN emulation testbed for a series of experiments that assessed the performance characteristics of the DTN2 and IBR-DTN implementations. Our extensive evaluation in urban mobility scenarios, which we believe to be the first of its kind, helped identify several issues that demonstrate the need to perform extensive repeatable experiments with DTN applications and protocols.

With 2-3 nodes, both DTN2 and IBR-DTN functioned as expected. DTN2 performance degraded quickly with scale, leading to poor results for 26-node scenarios, even if only some of them are mobile. Results were somewhat better for goodput measurements using dtnperf. For 10 node scenarios performance was closer to the expected one; still, even in this case some issues were detected in high-mobility conditions. While dtlsr in DTN2 had relatively good performance in good connectivity conditions, the performance was lower than expected in sparse networks and in the presence of mobility, which leads us to believe that the dtlsr implementation itself is to be blamed in some of these circumstances. IBR-DTN behaved as expected in all the tested scenarios.

Our experiments have also illustrated how the emulation testbed that we designed and implemented can be used for a thorough assessment of the performance characteristics of DTN applications and protocols. This makes possible performance optimizations procedures, by allowing to identify performance bottlenecks through precise controlled experiments, followed by testing the improved implementation in exactly the same scenarios, so as to validate that the problems were fixed. We believe that such performance testing is mandatory should one wish to apply the DTN paradigm to everyday situations, in which node count is large and network conditions are difficult to predict.

Acknowledgments

The authors would like to thank Shinya Yamamura, Akira Nagata, and Prof. Masato Tsuru for their advices related to the DTN2 reference implementation.

6. REFERENCES

[1] R. Beuran, L. T. Nguyen, T. Miyachi, J. Nakata, K. Chinen, Y. Tan, Y. Shinoda, *QOMB: A Wireless Network Emulation Testbed*, IEEE GLOBECOM 2009, Honolulu, Hawaii, USA, Nov. 30-Dec. 4, 2009.

[2] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, *Delay-Tolerant Networking – An Approach to Interplanetary Internet*, IEEE Communications Magazine, June 2003.

[3] Delay-Tolerant Network Research Group, *DTN-Bone*, Internet Research Task Force, http://dtnrg.org/wiki/DtnBone.

[4] Delay-Tolerant Network Research Group, *DTNRG home page*, Internet Research Task Force, http://www.dtnrg.org/wiki.

[5] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, R. Patra, *Implementing Delay Tolerant Networking*, Intel Research Berkley Technical Report, IRB-TR-04-020, December 2004.

[6] K. Fall, *A Delay-Tolerant Network Architecture for Challenged Internets*, Intel Research Berkley Technical Report, IRB-TR-03-003, February 2003.

[7] E. Koutsogiannis, S. Diamantopoulos, G. Papastergiou, I. Komnios, A. Aggelis, N. Peccia, *Experiences from architecting a DTN Testbed*, Journal of Internet Engineering, Vol. 3, No.1, December 2009, pp. 219-229.

[8] P. Luo, H. Huang, W. Shu, M. Li, M. Wu, *Performance Evaluation of Vehicular DTN Routing under Realistic Mobility Models*, IEEE WCNC 2008, Las Vegas, Nevada, USA, March 31-April 3, 2008, pp. 2206 - 2211.

[9] E. Oliver, H. Falaki, *Performance Evaluation and Analysis of Delay Tolerant Networking*, ACM/USENIX MobiSys 2007, MobiEval Workshop, Puerto Rico, June 2007.

[10] S. Schildt, J. Morgenroth, W.-B. Pottner, L. Wolf, *IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation*, Electronic Communications of EASST, Vol. 37, 2011, pp. 1-10.