

Summer Research Report

Description

This report documents the results of research on congestion issues in Delay Tolerant Networks carried out by Durga Prasad Pandey during the summer of 2003. It consists of a proposal for an experiment to test the effectiveness of a cognition based congestion detection program. The experiment is proposed to be carried out during the academic year 2003-2004 as a final year research project at the Department of Electronics and Instrumentation, Indian School of Mines, Dhanbad.

Supervisors

Scott C. Burleigh (Principal Supervisor)
NASA Jet Propulsion Laboratory
Pasadena, CA, USA
Scott.C.Burleigh@jpl.nasa.gov

Adrian J. Hooke
NASA Jet Propulsion Laboratory
Pasadena, CA, USA
Adrian.J.Hooke@jpl.nasa.gov

Dr Vinton G. Cerf,
MCI
Ashburn, VA, USA
Vinton.G.Cerf@mci.com

Submitted by

Durga Prasad Pandey
Department of Electronics and Instrumentation
Indian School of Mines
Dhanbad 826004
India
dpsmiles@acm.org

Acknowledgements

I would like to express my deepest sense of gratitude to Scott Burleigh, who was patient with my innumerable queries, and always answered even the most basic questions. He gave me full freedom in pursuing new ideas, and provided an ideal working environment by being constantly supporting and challenging. Working with him has been a great learning experience, and I look forward to an exciting year of work ahead.

I am indebted to Dr Vint Cerf, without whose kind support and encouragement, this project couldn't have taken off in the first place. His humility is inspiring, and I am honored to work with him.

Adrian Hooke's constant support, right from my early days with the IPN, and his unspoken faith in my abilities has been a strong motivation, and it helped me immensely during the first crucial months of this project. Thank you, Adrian.

Dr Keith Scott, Pat Feighery, Leigh Torgerson, Dr Kevin Fall and Stephen Farrell were always keen to help in understanding the finer elements of Delay Tolerant Networking. Prof Len Kleinrock gave valuable tips on conducting research, and provided a deeper perspective on the project. I thank them all.

I dedicate this project to my family, and my friends, particularly Shulabh, Yagnick, Himanshu, Kuber, Manish, Deepa, and PJ sir, who were always there.

Durga Prasad Pandey

Proposal for Experiment

1 Details of the proposed experiment

- 1.1 Aim:** To test the hypothesis that a machine-learning program can detect congestion in a Delay Tolerant Network (DTN).
- 1.2 Methodology:** Use a publicly available machine learning tool kit to develop a system (congestion detection daemon) that would recognize congestion in a DTN.
- 1.3 Input to congestion detection daemon:** Periodic reports from all nodes of the network on:
- Rates at which they are receiving data from other nodes,
 - Rates at which they are sending data to other nodes, and
 - Rate at which local free storage is increasing or decreasing.
- 1.4 Output from congestion detection daemon:** A numeric figure that indicates the likelihood of congestion at each node.

2 Background

Delay Tolerant Networks (DTNs) [1] are characterized by large latency, intermittent connectivity, and high and/or variable error rates. Research on DTNs originally started with an attempt by Vint Cerf, a co-author of the TCP/IP protocol suite, and a team of scientists at JPL led by Adrian Hooke to build the framework for an InterPlanetary Internet (IPN) [2] which would extend internetworking to space. Thus the InterPlanetary Internet was the precursor to DTNs in much the same way as the ARPANET was to the Internet. The IPN team developed a message switched architecture [3] based on bundling [4], which was later generalized into Delay Tolerant Networking [5] to include networks whose operational characteristics make traditional approaches either unworkable or impractical. Examples [6] of such networks include terrestrial mobile networks, with a commuter bus acting as a store and forward message switch [7]; exotic media networks such as undersea and free-space optical communication networks; military ad-hoc networks; and sensor and sensor/actuator networks. As an illustration, we will use the example of the InterPlanetary Internet for our discussions.

The IPN architecture proposes a new layer called the bundling layer that lies just below the application layer. At interplanetary gateways, transport protocols hand over packets to the upper bundling layer, which 'consumes' them to 'produce' bundles. The bundles are routed through a deep space network that consists of gateways at the edge of different interplanetary regions. The IPN architecture defines a region as an Internet that shares the same networking protocols. The IPN is thus defined as a network of Internets.

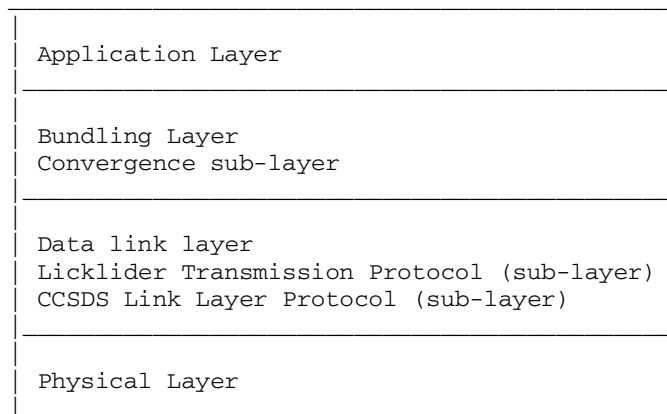


Figure 1: Bundling architecture in the IPN backbone

The IPN uses an addressing scheme based on a 2-entity tuple, which consists of a region ID (e.g. mars.sol) and a region-specific entity ID (e.g. durga.org). The entity ID of a tuple is not interpreted (e.g., bound to an address) outside its local region: a concept called late binding. The IPN backbone is concerned only with delivering the bundle to the destination region, without worrying about who the intended recipient is. The IPN gateway in the destination region 'consumes' the bundles, and hands over the packets to the Internet in the destination region, which routes the packet(s) to the recipient defined by the entity ID.

Bundling eliminates the chattiness that is characteristic of TCP/IP based protocols by including all the control information (metadata) needed at intermediate/destination nodes in the bundle itself. This enables connections to be set up without waiting for multiple round trip times exchanging SYN/ACK messages. The expectation that a bundle will be delivered is very high, and recipients are not expected to send acknowledgements, except in special cases. Mechanisms at the data link layer provide reliability through the use of strong Forward Error Correction (FEC) techniques along with a limited amount of retransmission. The point of retransmission is progressively moved forward towards the destination region by a mechanism called custodial transfer; when a node accepts custody of a bundle, it stores a copy of the bundle till the next custodian accepts custody and informs it, upon which it discards the bundle to free up its storage space.

When reasoning about what is desirable and possible in a DTN, it is often helpful to consider the analogy[8] of living in England in the mid-19th century, before telephones or even widely available telegraphy. The only way to communicate with friends or business associates is by letters, conveyed through the postal system. The postal system is very good - there are three or four mail deliveries per day in central London - but one never has absolutely up-to-the-minute information about *anything*. One is always reasoning from past experience and the most recent information, which is always hours or days old. This is exactly the condition in which delay-tolerant networks may often (though not always) need to function.

3 Congestion in DTNs

Congestion has been a key issue for the Internet community, right from the ARPANET days. It is often difficult to define congestion precisely, since it depends on a network's particular conditions, as well as users' expectations and perceptions. The following symptoms [9] are generally associated with congestion in the Internet:

- The queuing delay of the data packets increases.
- There may be packet losses.
- Traffic is dominated by retransmissions, so that the effective data rate decreases.

These parameters are however inadequate to characterize congestion in a delay tolerant network. In a DTN, queuing delays are *expected*, because of the high latencies and intermittent connectivity. Further, paths are very lossy, so losses do not necessarily indicate congestion. Finally, bundling is designed to eliminate chattiness, so there is no question of traffic being dominated by retransmissions.

We believe that congestion in the InterPlanetary Internet is likely to be due to unexpected link failures, or data bursts from opportunistic contacts. Keshav has given a definition [9] of congestion based on economic theory, which appears to be suitable to characterize congestion in a DTN. According to this definition, a network is said to be congested from the perspective of user *i* if the utility of the network to *i* decreases due to an increase in network load. 'Utility' here refers to a user's preference for a resource, or a set of resources. Strictly speaking, the utility to a user is a number that represents the relative preference of that user for a resource or set

of resources, so that if a user prefers A to B, the utility of A is greater than the utility of B.

In a DTN, nodes could **learn** to expect a particular level of utility(performance) from the network. The utility will have to be defined/inferred by the network. A reduction in the utility might be a possible indication of impending congestion.

4 Our solution: Structural flow control

Scott Burleigh has proposed a mechanism called structural flow control [8] to deal with congestion in the DTN. To understand the mechanism, let us consider the following analogy. Suppose all the links in some small neighborhood ad-hoc network are dial-up telephone links using cheap modems. The owner of one node of the network, noticing that data are arriving too fast for his node to keep up with processing, might phone all his friends and ask them to switch their modems' transmission rates from 56 Kbps to 28.8 Kbps. If they all complied, then data would flow more slowly in the network even though there would be no additional TCP or IP flow control. In effect, aggregate transmission speed would have been "structurally" degraded - at the physical level, by human agency - rather than operationally or logically by protocol activity.

The notion of structural flow control is built on feedback loops similar to flow control built into TCP and IP. There are some important differences [8], however:

- The loops are potentially quite long and slow, not tight and fast as in TCP,
- The flow control information (commands and feedback) is carried at the bundling layer ("a higher level") rather than built into the transport protocol itself, and
- There may (or may not) be a human in the loop.

Structural flow control is based on the following principles:

- Autonomous cognition: Each node takes decisions independently of others, depending only on its own input and output rates, and the condition of its buffers. A cognitive model helps the node take accurate decisions based on past experience.
- Decentralized operation: There is no central authority that regulates the network or influences its decisions. The network operates in a decentralized fashion like the Internet.
- Tolerance of imperfect information/mistakes: Because the information available about the network at a node is generally imperfect at any given moment, mistakes are inevitable. Nodes learn from past mistakes and try to minimize their occurrence. At the same time, they must be resilient to mistakes, by being prepared to deal with them.

Congestion and flow control are delegated to the bundling layer or the convergence sub-layer. The convergence layer can force flow control by refusing to read on the input socket, and sending NACKs to the source; while the bundling layer can send quench packets to the source to ask it to reduce its sending rate. The aim is to decrease the amount of data entering the IPN backbone. In the event of congestion occurring in the DTN, nodes would have to start discarding bundles - resulting in NACKs - as an immediate step, and thereafter ask senders to reduce their sending rates after considering the impact of reduction in sending rates by individual senders and by different factors.

Nodes can ask certain senders to reduce their rates, while letting others continue sending at the previous rates. To accomplish this, they would have to run AI programs that search the set of possible solutions, and select the best, after analyzing the effect of various combinations. Simultaneously, they could use online machine learning programs that suggest which option(s) is better in the light of past experience. In this context, we wouldn't want hardware limitations to constrain our thinking unduly, because although

space-qualified processors will likely always be slower than their terrestrial counterparts, they too are getting faster and more capable.

Cascaded buffer exhaustion, caused by backward propagation of hop by hop flow control, can possibly be avoided by using some form of Random Early Decision[11], which will be an integral part of the structural flow control mechanism.

Currently, all interplanetary communication devices are radios either mounted on spacecraft (including rovers, etc.) or installed in the tracking stations of the Deep Space Network and other tracking systems. Links are highly directional, so these devices - analogous to the hosts and routers of the Internet - have to be actively managed: the antennae of the spacecraft and tracking stations must be physically moved (pointed) and the radios powered on and off on an intricate and detailed schedule in order for data to flow. The transmission capacity of each such radio is known in advance, so each individual communication opportunity has a known duration and data rate; hence capacity. The data to be transmitted in each opportunity are carefully selected so as not to exceed that capacity. The aggregate effect of all of this planning and management is, in effect, structural flow control[8].

The future growth of capacity in the Interplanetary Internet will make some of this management infeasible: pointing and scheduling are unavoidable, but today's manual negotiation of schedule and selection of the data to be transmitted in each opportunity will have to be supplanted by automation. That automated management will similarly be "structural flow control[8].

4.1 Rationale: Why a cognition-based approach?

Traditional algorithmic approaches fail in the face of imperfect information, because the complete information that they need to take correct decisions is not available. On the other hand, machine learning programs often provide results with a high degree of accuracy once they have been trained adequately, even when the input data is incomplete.

We propose to utilize this property of machine learning programs of taking accurate decisions even with imperfect information to detect the likelihood of congestion in a DTN. We believe that this approach is especially relevant to DTNs where information is always delayed and often incomplete. To keep the network running at an acceptable efficiency, nodes in DTNs would need to assess the likelihood of congestion, by analyzing the typical data at the node such as input rates, output rates, and buffer availability, etc. The congestion detection daemon running at the node could suggest a numerical figure that indicates the likelihood of congestion. This would enable the node to take appropriate measures, such as discarding bundles, rejecting custody, or asking a sending node to reduce its sending rate.

Taking flow control decisions is especially tricky in DTNs because the decision is to be based on old information, and the effects of the decision will often not be visible until some significant amount of time in the future. This dual disadvantage in time implies that a node must have the ability to explore plausible scenarios - similar to a chess program - and suggest the best course of action in the light of available information. Furthermore, the need to minimize human intervention requires the ability to learn from both successes and failures, and evolve mechanisms to diagnose and fix problems in the network [12]. These considerations strongly support the choice for an AI based approach to congestion and flow control.

4.2 What is imperfect information?

Imperfect information refers to delayed, incomplete or conflicting information. In the latter case, we expect that the program will be able to detect if a certain piece of information that has come in is wrong, based on past experience.

5 Project Steps

5.1 Find a usable machine-learning tool kit. We propose using C5.0 [13], Quinlan's powerful inductive machine learning software. The choice of a machine learning method is dictated by the amount of information available about a system. The decision to select inductive machine learning is motivated by the fact that this method works even when very little information is available about the system.

5.1.1 *Due date:* 10th Oct, 2003.

5.1.2 *Status:* Completed

5.2 Learn how to use the tool kit.

5.2.1 *Due date:* 20th Oct 2003

5.2.2 *Status:* Currently in progress.

5.3 Design the input and output data (the reports).

5.3.1 *Due Date:* 30th October 2003.

5.4 Develop software that simulates running node models and can issue the requisite periodic reports to the congestion detection daemon.

5.4.1 *Due Date:* 30th November 2003.

5.5 Use the tool kit to develop and train the congestion detection daemon.

5.5.1 *Due Date:* 15th February 2004.

5.6 Run the experiment a number of times, varying the parameters that alter traffic flow, to evaluate the effectiveness of the daemon.

5.6.1 *Due Date:* 15th March 2004.

5.7 Write a detailed report of what worked and what didn't.

5.7.1 *Due Date:* 15th April 2004

6 Relevance: How does it fit in the bigger picture?

The proposed experiment is a first step towards achieving our vision of automated and intelligent delay tolerant networks. The key contribution of this effort is expected to be two-fold; it will test the effectiveness of cognitive models in combating congestion in DTNs, and in doing so, it will lead us to a better understanding of congestion issues, and possibly other issues of interest in Delay Tolerant Networking.

7 Related work

David Clark et al have proposed a concept called the Knowledge Plane [12] that aims to automate diagnosing and fixing problems in a network. But while Knowledge Plane advocates for global state information and a co-operative approach, structural flow control is based on node autonomy and local information. Maintaining global state is neither useful nor desirable in DTNs, since the information will be generally too old to be of any use, and it will also not be feasible from the point of view of scaling the network.

8 References

- [1] Delay Tolerant Networking Research Group, <http://www.dtnrg.org>.
- [2] InterPlanetary Internet Special Interest Group, <http://ipnsig.org>.
- [3] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Travis, E., and Weiss, H. Interplanetary Internet (IPN): Architectural Definition, draft-irtf-ipnrg-arch-00.txt, work in progress, May 2001.
- [4] Burleigh, S., Cerf, V., Durst, R., Fall, K., Hooke, A., Scott, K., Torgerson, L. and Weiss, H. Bundle Layer Protocol Specification v0.4, work in progress, Sept 2002.
- [5] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K. and Weiss, H. Delay- Tolerant Network Architecture: The evolving Interplanetary Internet, draft-irtf-ipnrg-arch-01.txt, work in progress, Feb 2002.
- [6] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets", IRB-TR-03-003, Feb 2003.
- [7] Daknet project, <http://www.daknet.org>.
- [8] Burleigh, S. Personal communication.
- [9] Keshav, S. Congestion Control in Computer Networks, PhD thesis, University of California, Berkeley, August 1991.
- [10] Burleigh, S. Flow control/congestion control/resource management in DTNs, dtn-interest mailing list, 16 March 2003.
- [11] Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance V.1 N.4, August 1993, p. 397-413.
- [12] Clark, D., Partridge, C., Ramming, J., and Wroclawski, J. A Knowledge Plane for the Internet, SIGCOMM 2003.
- [13] Rulequest, <http://www.rulequest.com>.