Optimizing Routing in Delay-Tolerant Network (DTNS)

Lalit Kumar¹, Khem Kumar Nagar², Deepak Garg³

¹ Department of ECE, USIT, GGSIPU Delhi, Meerut, U.P. 250001, India
² Department of ECE, MTU Noida, IIMT Engineering College, Meerut, U.P. 250001, India
³ Department of ECE, USIT, GGSIPU Delhi, Meerut, U.P. 250001, India

¹ lalitkr12@yahoo.com
² khemmagar@gmail.com
³ deepak.vlsi@rediffmail.com

Abstract: - We try to formulate the delay-tolerant networking routing problem, where messages are to be moved end-to-end across a connectivity graph that is time-varying but whose dynamics may be known in advance. The problem has the added constraints of finite buffers at each node and the general property that no contemporaneous end-to-end path may ever exist. This situation limits the applicability of traditional routing approaches that tend to treat outages as failures and seek to find an existing end-to-end path. We propose a framework for evaluating routing algorithms in such environments. We then develop two algorithms and use simulations to compare their performance with respect to the amount of knowledge they require about network topology. We find that with additional knowledge, far less than complete global knowledge; efficient algorithms can be constructed for routing in such environments.

Keywords-Delay Tolerant Networks, Omnet++, Inet framework

I. INTRODUCTION

Delay Tolerant Networks (DTNs) are assumed to experience frequent, long-duration partitioning and may never have an end-to-end contemporaneous path. This problem contrasts with routing in conventional data networks, which typically selects a shortest policy-compliant path in a connected graph without considering availability of intermediate buffering and bandwidth capacity.

In graph theoretic terms, the problem is a form of the “quickest transshipment problem” in which both edge capacities and transit delays along an edge can vary (down to zero) as a function of time and nodes have finite buffers [12]. In practical terms, DTNs arise in networks with known connectivity patterns such as Low-Earth Orbiting Satellites (LEO) or those with unpredicted, opportunistic connectivity (e.g., communication among PDAs when brought into close proximity [5]). Here, we focus on the latter case. The routing problem in a DTN may at first appear as the standard problem of dynamic routing but with extended link failure times. For the standard dynamic routing problem, the objective of the routing algorithm is to find the best currently-available path to move traffic end-to-end.

In a DTN, however, an end-to-end path may be unavailable at all times; routing is performed over time to achieve eventual delivery by employing long-term storage at the intermediate nodes. The DTN routing problem amounts to a constrained optimization problem where edges may be unavailable for extended periods of time and a storage constraint exists at each node. This formulation reveals DTN routing to be a considerably different and more challenging problem. In this paper, we formulate the DTN routing problem when the connectivity patterns are known, then provide a framework for evaluating various routing algorithms, and finally show a simulation based comparison of several of our algorithms.
We also include an optimal algorithm based on a linear programming approach to serve as a basis for comparison with the simulations. Finally, we outline the future work to be accomplished in the area [10].

Delay tolerant networks (DTNs) are steadily gaining popularity in the research community for their ability to provide connectivity, or a semblance of connectivity, in “challenged” networking environments. Examples of these environments include [3]

1. Urban networks in which opportunistic meetings between cars and buses can be used to transfer messages from disconnected portions of the network to areas with Internet access.
2. Rural networks in which villages have reliable connectivity between local hosts but have unreliable connections to the wider world.
3. Networks of sensors that collect and share information about animal movement and behavior and
4. Networks in which roving autonomous robots provide connectivity or message ferrying capabilities in disruption-tolerant environments.

We provide OMNeT++ implementations of two popular routing models. Additionally we create a statistical analysis package for OMNeT++ that greatly extends the capabilities of the simulator and provides insight into the characteristics of the routing models independent of the overlying communication protocols. These statistics can provide insight into how different routing models affect ad-hoc or DTN routing algorithms and other higher layer protocols. Due to the modular nature of OMNeT++, authors of new routing models do not have to make any changes to their code to use our package and benefit from the information it provides.

II. TODAY´S INTERNET

The Internet has been a great success at interconnecting communication devices across the globe. It has done this by using the TCP/IP protocol suite. All devices on the hundreds of thousands of subnets that make up the Internet use these protocols for routing data and insuring the reliability of message exchanges. Today Internet widely works on these protocols. These protocols provides end-to-end service (reliable or otherwise), with consideration that changes in the underlying link-layer technology does not affect the basic working of these protocols. [3]

A. Evolving Wireless Networks outside the Internet

Communication outside of the Internet—where power-limited mobile wireless, satellite, and interplanetary communications are developing—is accomplished on independent networks, each supporting specialized communication requirements. These networks do not use Internet protocols and they are mutually incompatible—each is good at passing messages within its network, but not able to exchange messages between networks each network is adapted to a particular communication region in which communication characteristics are relatively homogeneous. The boundaries between regions are defined by such things as link delay, link connectivity, data-rate asymmetry, error rates, addressing and reliability mechanisms, quality-of-service provisions, and trust boundaries. Unlike the Internet, these wireless networks support long and variable delays, arbitrarily long periods of link disconnection, high error rates, and large bi-directional data-rate asymmetries. Examples of such networks are:

(i) Terrestrial wireless networks: Such networks may connect mobile wireless devices, or devices in areas without much infrastructure to support continuous connectivity. Examples include networks in rural/remote areas, vehicular networks, etc.
Optimizing Routing in Delay- Tolerant Network (DTNS)

(ii) **Military networks:** These networks operate under hostile conditions and involve large numbers of nodes (troops, vehicles, aircraft, satellites, sensors) where mobility is high and the environment may cause signals to be disconnected or jammed frequently.

(iii) **A typical media networks:** These include networks that operate in deep-space or underwater, and are usually characterized by high, but predictable latencies with periodic service, such as orbit of satellites, passage of ships, etc.[10]

Such networks require protocols that take into consideration the specific communication needs of such networks, including link connectivity, delay, data rate asymmetry, addressing, reliability mechanisms, quality of service, etc. These parameters not only vary with those of the protocols used on the Internet, but also with each other. That is, such networks are typically mutually incompatible - within a network, communication is fluid, but between different networks, message exchange is not possible directly due to varying characteristics. Therefore, it is necessary to have a means to translate between such networks with mismatched delays [3].

**B. The Concepts of Delay Tolerant Networks (DTNs)**

A DTN is a general-purpose overlay network that operates on top of varying regional networks which allow regional networks with varying delay characteristics to interoperate by providing mechanisms to translate between their respective network parameters. Therefore, the underlying protocols and technologies for these regional networks may differ considerably, but the flexibility of the DTN architecture allows them to be connected to each other. DTNs support interoperability of regional networks by accommodating long delays between and within regional networks, and by translating between regional network communications characteristics. In providing these functions, DTNs accommodate the mobility and limited power of evolving wireless communication devices.

The wireless DTN technologies may be diverse, including radio frequency (RF), ultra-wide band (UWB), free-space optical, and acoustic technologies. As shown in figure 2 the concept of DTN in Interplanetary Internet paper.

Why a Delay-Tolerant Network (DTN)?

Many evolving and potential networks do not conform to the Internet’s underlying assumptions. These networks are characterized by:

1. **Intermittent Connectivity:** If there is no end-to-end path between source and destination—called *network partitioning*—end-to-end communication using the TCP/IP protocols does not work. Other protocols are required.

2. **Long or Variable Delay:** In addition to intermittent connectivity, long propagation delays between nodes and variable queuing delays at nodes contribute to end-to-end path delays that can defeat Internet protocols and applications that rely on quick return of acknowledgements or data.

3. **Asymmetric Data Rates:** The Internet supports moderate asymmetries of bi-directional data rate for users with cable TV or asymmetric DSL access. But if asymmetries are large, they defeat conversational protocols.

4. **High Error Rates:** Bit errors on links require correction (which requires more bits and more processing) or retransmission.
of the entire packet (which results in more network traffic). For a given link-error rate, fewer retransmissions are needed for hop-by-hop than for end-to-end retransmission (linear increase vs. exponential increase, per hop).

III. OMNET++

The OMNeT++ simulation IDE is based on the Eclipse platform and extends it with new editors, views, wizards, and other functionality. OMNeT++ adds functionality for creating and configuring models (NED and INI files), performing batch executions and analyzing the simulation results, while Eclipse provides C++ editing, SVN/GIT integration and other optional features (UML modeling, bug-tracker integration, database access, etc.) via various open-source and commercial plug-ins. The environment will be instantly recognizable to those at home with the Eclipse platform [11]. OMNeT++ is an object-oriented modular discrete event simulator. The name itself stands for Objective Modular Network Test bed in C++. OMNeT++ has its distant roots in OMNeT, a simulator written in Object Pascal by Dr. György Pongor. The simulator can be used for:

Traffic modeling of telecommunication networks, Protocol modeling, modeling queuing networks, modeling multiprocessors and other distributed hardware systems, validating hardware architectures, evaluating performance aspects of complex software systems, modeling any other system where the discrete event approaches is suitable.

Modules can have parameters which are used for three main purposes: to customize module behavior; to create flexible model topologies (where parameters can specify the number of modules, connection structure etc); and for module communication, as shared variables [12].

Modules at the lowest level of the module hierarchy are to be provided by the user, and they contain the algorithms in the model. During simulation execution, simple modules appear to run in parallel, since they are implemented as co routines (sometimes termed lightweight processes). To write simple modules, the user does not need to learn a new programming language, but he/she is assumed to have some knowledge of C++ programming.

What is INET Framework?

INET Framework contains IPv4, IPv6, TCP, SCTP, UDP protocol implementations, and several application models. The framework also includes an MPLS model with RSVP-TE and LDP signaling. Link-layer models are PPP, Ethernet and 802.11. Static routing can be set up using network auto configurations, or one can use routing protocol implementations. The INET Framework supports wireless and mobile simulations as well. Support for mobility and wireless communication has been derived from the Mobility Framework.

IV. METHODOLOGY

Here we propose to design a network of definite dimensions and propose to route the packets from the source node to destination node. For routing the packets we use following cases:-

A. Case 1 - Direct Transmission, which can also be considered a degenerate case of the forwarding family, where it always selects the direct path between the source and the destination. Here we have some common sink for the adjacent four sources.
nodes and each source nodes is in direct contact with these sink. Whenever a source has some packets to transmit, it directly forward the packets to the sink. Due to its simplicity, it does not consume many other resources, and it uses exactly the one message transmission. Here, we have a more numbers of sinks and each sink is attached to adjacent four sources as shown in figure below.

B. Case 2 - Flooding strategy, which rely primarily on replicating messages to enough nodes so the destination receives it. Routing algorithm used in this type of strategy is called Epidemic. Epidemic algorithms guarantee that provided a sufficient number of random exchanges of data, all nodes will eventually receive all messages. Thus, the destination node is guaranteed to have received the data, Epidemic Routing works as follows. When a message is sent, it is placed in the local buffer and tagged with a unique ID. When two nodes connect, they send each other the list of all the messages IDs they have in their buffers, called the summary vector. Using the summary vector, the nodes exchange the messages they do not have. When this operation completes, the nodes have the same messages in their buffers. Epidemic Routing represents the extreme end of the flooding family because it tries to send each message over all paths in the network. This provides a large amount of redundancy since all nodes receive every message, making this strategy extremely robust to node and network failures. Additionally, since it tries every path, it delivers each message in the minimum amount of time if there are sufficient resources. Epidemic Routing is relatively simple because it requires no knowledge about the network. For that reason, it has been proposed to use it as a fallback when no better method is available [19]. The disadvantage is that a huge amount of resources are consumed due to the large number of copies. This requires large amount of buffer space, bandwidth, and power. Many papers have studied ways to make Epidemic Routing consume fewer resources. Epidemic flooding can be understood as shown in figure below.
Now, we will try to simulate the above two cases. We will form our network and apply these two routing algorithms on that network. Figure 5.3 and 5.4 as given below show of our network on which, we will apply above said algorithms. For figure 5.3, we form a network for the flooding strategy in which we will use Epidemic routing algorithm. In figure 5.4 we make our network according to the first case i.e. “Direct Transmission” algorithm. After running the simulation, we will try to compare both the routing protocols by analyzing the graphs produce as a result of data transmission that has been taking place between Host and Servers for the above two routing cases.
Optimizing Routing in Delay-Tolerant Network (DTNS)

VI. CONCLUSION

In DTN, which is used in networks where intermittent connectivity, high error rates, long delays are common problems and we need a technique which can cope with these problems, taking into account that less power is required from the transmitter and mostly routing takes place between moving nodes, two criteria are major: (1) efficiency and (2) cost. By seeing the result of the paper, we can suggest that if efficiency is the main concern like in military networks so that maximum information is needed to take the decisions, Direct routing (i.e., Case I) is more robust and reliable. But if cost is the criterion and with an average efficiency like in rural networks, Multi Hop routing (i.e., Case II) provides a good solution. For further study and future work, the paper presents a comparison of two algorithms which can work as a feedback while comparing other protocols and in designing more new protocols.

Fig. 16 - Network Simulation file Snapshot for Case II

Fig. 17 – Packets and Time mean radio state
VII. REFERENCES


