# Nuntifix-Modeling of Delay Tolerant Networks

A Technical Report

Muhammad Arshad Islam
arshad.islam@uni-konstanz.de

Department of Informatics and Information Engineering,
Universitaet Konstanz

**Supervisor**

Prof. Dr. Marcel Waldvogel

*Konstanzer Online-Publikations-System (KOPS)*
*URL: http://www.ub.uni-konstanz.de/kops/volltexte/2008/5174/*
*URN: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-51749*

# Contents

**1 Introduction 2**

**2 Need for new protocols 3**

**3 Why Protocol Comparison? 3**

**4 Simulation Setup 4**

4.1 Trace Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

4.1.1 Bluetooth (“MITBT”) . . . . . . . . . . . . . . . . . . . . . 6

4.1.2 Access Points (“IBM”) . . . . . . . . . . . . . . . . . . . . . 6

4.1.3 Cell Towers (“MIT”) . . . . . . . . . . . . . . . . . . . . . . 6

4.2 Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

4.3 Simulation Issues . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

4.3.1 Basics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

4.3.2 Connection . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

4.3.3 Cleanup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

4.3.4 Variations and confidence . . . . . . . . . . . . . . . . . . . 9

4.3.5 Timing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

4.3.6 Link sharing. . . . . . . . . . . . . . . . . . . . . . . . . . . 10

4.3.7 Shortcut . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

4.3.8 Imperfect Oracle . . . . . . . . . . . . . . . . . . . . . . . . 10

4.3.9 Simulated Algorithms . . . . . . . . . . . . . . . . . . . . . 11

**5 Results 15**

5.1 Reasoning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

**6 Conclusion 24**

# List of Figures

1 Comparison of all the traces: Rank of total meetings vs. distinct meeting count . . . 5
2 Plots for No. of Messages delivered with High Bandwidth from Access point and Bluetooth traces . . . 16
3 Plots for No. of Messages delivered with High Bandwidth from Cell Tower traces . . . 17
4 Plots for No. of Messages delivered with Med Bandwidth from Access point and Bluetooth traces . . . 18
5 Plots for No. of Messages delivered with Med Bandwidth from Cell Tower traces . . . 19
6 Performance Behaviour of Routing Algorithms w.r.t. Amount of Data (left) and Number of Messages (right) Delivered . . . 20
7 Plots for No. of Messages delivered with low Bandwidth from Access point and Bluetooth traces . . . 22
8 Plots for No. of Messages delivered with low Bandwidth from Cell Tower traces . . . 23
9 Plots for Data delivered with High Bandwidth from Access point and Bluetooth traces . . . 25
10 Plots for Data delivered with High Bandwidth from Cell Tower traces . . . 26
11 Plots for Data delivered with Med Bandwidth from Access point and Bluetooth traces . . . 28
12 Plots for Data delivered with Med Bandwidth from Cell Tower traces . . . 29
13 Plots for Data delivered with low Bandwidth from Access point and Bluetooth traces . . . 30
14 Plots for Data delivered with low Bandwidth from Cell Tower traces . . . 31

# List of Tables

1 Simulation parameters . . . 9

2 Algorithm Characterization . . . . . . . . . . . . . . . . . . . . . . 12

## Abstract

In the past years, many promising routing algorithms for delay-tolerant networks have been described as well as simulated or even implemented, and also evaluated. Anyone wishing to implement routing into delay-tolerant network can select from a wide variation of options, but the choice is hard, as there is no strong comparative evidence to the relative performance of the algorithms. While each algorithm by itself is exciting and adds to our world's knowledge, one important aspect of research is to understand when to do what. As the algorithm evaluations lack a common basis, comparison between the algorithms is currently impossible: First, most evaluations restrict themselves to comparing against the two extremes, namely direct-contact-only forwarding and flooding; second, each attempt uses a completely different choice of scenario and simulation parameters. In this report, we analyze and evaluate the currently existing algorithm under the common basis in an effort to grasp the strong and weak points of each of them and to see whether its possible to design a hybrid technique that may take advantage of strengths of many techniques.

# 1 Introduction

In past decade, communication technology has revolutionized the way we used to communicate. The spread of communication technologies is bringing more and more people into this cocoon, which is becoming increasingly global in nature. The internet in particular is global in design  it jumps territorial boundaries. It is creating an e-world of e-commerce, e-friendship, e-government and e-mail. [1]. Internet users leapt from nearly half a billion to slightly over one billion between 2001 and 2005. In 2004, 38% of all internet subscribers worldwide had access to broadband [2]. The recent trend is going a leap further than that as previously we had to sit in front of a PC thats is connected to network. Today, we can also use mobile phones and portable laptops. That can be next logical step in this technological revolution (connecting people anytime, anywhere) is to connect inanimate objects a communication network. Moreover, Falling costs allow more and more people across the planet to stay closer in touch by phone. Mobile phone subscribers more than doubled from just under a billion in 2001 to 2.1 billion in 2005. There has been considerable attention given to mobile technology as the development of wireless capable handeld devices such as PDAs, mobile phones, and light weight laptops has been revolutionized in the past decade.

In ad hoc wireless networks, where simultaneous links in the network are not possible, growing number of studies are exploring techniques for moving network traffic over asynchronous paths. Such networks, usually known as Disruption/Delay Tolerant Networks(DTNs) [3, 4]. Delay-tolerant networking (DTN) is an attempt to extend the reach of networks. It promises to enable communication between challenged networks, which includes deep space networks, sensor networks, mobile ad-hoc networks, and low-cost networks. [5] The core idea is that these networks can be connected if protocols are designed to accommodate disconnection. These networks have variety of applications in usually extra ordinary situations. that include crisis environments like emergency response in case of a catastrophe, military operations, vehicular communication and non-interactive Internet access in rural areas [3, 6].

## 2 Need for new protocols

In DTN paradigm, end-to-end contemporaneous path between two nodes cannot be guaranteed due to long duration partitioning. The network can be partitioned either due to the movement or due to un-availability of peers. DTNs can be considered as a classical case for asynchronous communication like email and person to person communication. With the advent of VOIP and rapid spread of high bandwidth media, there is a shift towards synchronous technologies but asynchronous communication has its own advantage. e.g. It still works even if both the parties are not available simultaneously and it is less sensitive to link failure. These fundamental difference in nature of DTNs, existing protocols for synchronous networks are not anymore valid and there is requirement to develop new routing protocols for moving devices with either pedestrians or vehicles. In order to route messages, these protocols have to predict the network topology. Therefore, network modeling plays an important role in increasing the delivery ratio of the protocol [7]. Moreover, as wireless device may have limited storage space as well as limited access to power, researchers have to place bounds on time and resources needed by the protocol to make the routing decision.

## 3 Why Protocol Comparison?

Several attempts have been made to devise efficient routing mechanism, using different experimental or simulated data and different claims have been made in favour of several techniques. We have observed that, most of the attempts have made several over-simplistic assumptions which either favor them for routing or give them access to unlimited resources and hence there is no way to verify the claims made by the authors in unbiased manner. We have made effort to see the effect of these several protocols on the same realistic data so that their effectiveness can be compared.Moreover we want to explore that what kind of factors have been considered properly and which ones are ignored that may have considerable effect on the performance of those mechanism. it is obvious that employing some techniques gives alot more inner understanding as compared to merely reading and analyzing the results.

To understand the nature and extracting the model of mobile ad hoc networks, simulation has proven its value as a tool to assist us the variety of phenomenon like multiple paths, movement of devices, variable bandwidth, obstacles, effects of atmosphere etc. In a real world scenario, where repetition of events is very unlikely, scientist apply simulations techniques on data that is either created artificially or is obtained from a real word phenomena. This helps a great deal to improve and refine the protocols, specially in the the study of ad hoc wireless networks.

# 4 Simulation Setup

For this purpose we embarked on developing a routing simulator that will help us to see in depth, the issues involved with existing solutions. We selected 11 different routing algorithm and tried to simulate them under different environments of bandwidth, number of messages and sizes of messages. These algorithms involved variety of strategies ranging from very computation intensive and intelligent to brute force. Most of the intelligent algorithms used history to find the next best hop for the message to be carried to destinations. Some algorithms exploited replication to enhance the probability of message delivery. To extent replication is done is primarily dependent on how intelligent is the algorithm. More intelligence means it need less replication as replica generation and transmissions has its own obvious overhead.

## 4.1 Trace Description

It can be easily argued that user traces, that include information of cell tower IDs and the duration for which they have been connected to these towers may be used for proximity discovery. Similarly with access points, snmp protocol maybe utilized for similar purposes. A recent idea is to use Bluetooth traces. Bluetooth is a wireless protocol supported by most of the PDAs, mobiles and Laptops. Bluetooth gives wireless connectivity in the range of 20-50 meters depending on the device. All these economic and technological trends support the direction of our research. As already stated, we decided to utilize data available from ready resources from Internet, this data was not in the form as

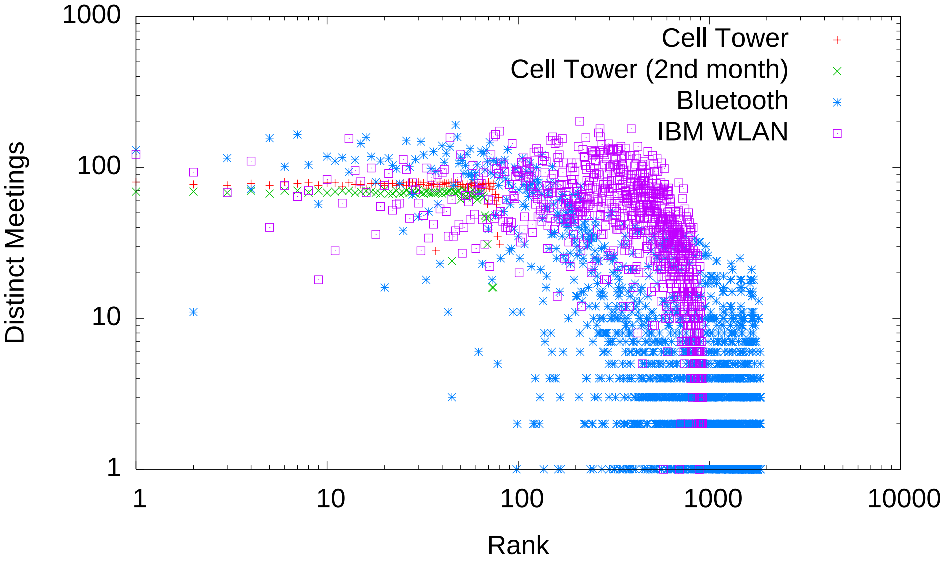


Figure 1: Comparison of all the traces: Rank of total meetings vs. distinct meeting count

we have preferred to be in one, and therefore we developed some tools to extract the useful information from these huge amounts of data. So far we have been able to touch three data sets, one provided by IBM and the other two been created by MIT reality mining project.

We have considered three different kinds of data sets, all of which have been obtained from CRAWDAD. The first and third data are the observations of communication, proximity, location, and activity information of 100 subjects at MIT over the course of the 2004/05 academic year [8], accurate to a few seconds. It represents a total of over 350,000 hours (≈40 years) of continuous data on human behaviour. Two different forms of connectivity were logged, namely continuos Bluetooth neighbours and the cell tower the mobile phone was registered to. The other data set consists of WLAN access point records for a corporate research laboratory (IBM Watson research center) at five minute resolution over several weeks [9].

### 4.1.1 Bluetooth ("MITBT")

The MIT Bluetooth trace spans around 16 months, i.e., from February 2004 to August 2005. Each device scanned every five minutes for active Bluetooth neighbours. We limited ourselves to one month of connectivity trace, where any visible Bluetooth device was considered a candidate connection. We selected the month with the largest activity, November 2004, for our simulation. Even then, the resulting connectivity remains sparse.

### 4.1.2 Access Points ("IBM")

In the case of IBM Access Point trace, SNMP was used to to poll access points every 5 minutes, from July 20, 2002 through August 17, 2002. A total of 1366 devices have been polled over 172 different access points during approximately 4 weeks. To turn these samples into continuous data, we assume that the snapshot data will remain constant for the next 5 minutes. In the rare cases where this would cause an overlap with another snapshot from another access point, we assume that the transition happens halfway between the two snapshots.

### 4.1.3 Cell Towers ("MIT")

This trace again holds exact timings, relieving us from such plays. Due to several lapses in data gathering, mentioned by the creators of the data, only 89 of 100 devices are included, which see 32768 different cell towers. As expected, November 2004 came out to be the month with maximum activity here as well. We observed 81 devices and 12592 different cell towers. Our connectivity model assumes that nodes registered at the same tower (actually, antenna) could communicate with each other. The resulting connectivity is the most dense of the three.

Other analyses we performed with the second-most active month, October 2004 with 79 devices and 11784 antennae, resulted in similar performance for the algorithms, supporting our thesis that November is nothing special.

## 4.2 Comparison

Another way to visualise the contact density is shown in Figure 1. It shows that in the Cell Tower case (both for the selected month of November as well as the preceding month), essentially all active nodes could directly communicate with every other node at least once during a month. In the Access Point case, the majority of 928 devices were able to contact 50. . . 100 devices, many of them up to 200.

In the Bluetooth scenario, the 1858 nodes get in touch with at most 100. . . 200 nodes, the majority only with 5. . . 50. Interestingly, some of the highest-ranking devices were not one of 89 participant nodes, but showed up more frequently in the communications range of trace group members than other members. We have to expect that these non-participants had contact with many other non-members; unfortunately, there is no way to tell given these traces.

We can see that the wider the communication range, we increase the likelihood to communicate among peers and thus make the network more dense, creating an increasing density from Bluetooth to Cell Towers. On the other hand, the data communication rate is mostly lower for long-distance networks (e.g., Access Points and Cell Towers; not so much for Bluetooth vs. Access Points). For current networks, the effects of the density (or lack thereof) seem to dominate the bandwidth effects.

## 4.3 Simulation Issues

The simulation for delay tolerant networks posed another big challenge for us as its a new developing technology and there are no simulator available for this purpose. Therefore, we had to start from scratch. As mentioned in our previous report, Kevin fall group is also working on similar problem and there simulator is under development. Rather than to go through the manual and details of that partially completed tool, we thought we can achieve our goal better if we do it on our own and futuristically we can open our simulator for the use of research community depending on its flexibility and usability.

The simulation of DTN is different from other network simulation as there is no constant connectivity among the devices i.e. underlying graph of the devices is changing very rapidly. The graph characteristics like depth, width, average fan out and topology, all are variables. Moreover the devices are not online all the time. Even if a device is turned on but there is no near by neighboring device, we have to take this device as offline. Due to all such issues, we faced some initial hiccups in our effort to make a simulator but in the end we were able to find a stable model for simulating different algorithms.

### 4.3.1 Basics

The aim of our simulator is to help us find the delays incurred by messages during execution of different routing algorithms. The output is analysed on the basis of both number of messages as well as amount of data delivered. As already mentioned, three different traces have been used that significantly differ in the number of devices involved as well as the number, frequency, and distinctness of meetings that were taking place among the participants. As the span time of Access Point trace is approximately one month whereas for Cell Tower and Bluetooth traces is more than one year, we have chosen one month from Cell Tower and Bluetooth data on the basis of highest activity, so that the results can be compared. We observed that November 2004 had the highest activity[1] among all the month for which Cell Tower data has been recorded. Furthermore to see the correlation with other months, for Cell Tower trace, we decided to include the month that ranked second, i.e., October 2004.

### 4.3.2 Connection

In Access Point and Cell Tower traces, we define a transfer opportunity between two peers if they are connected for overlapping times to the same access point or cell tower. This setup is obviously not needed in the case of Bluetooth trace. Devices that are included in 30th percentile with respect to their online time are eligible to be source and destination; all devices may help out as intermediate nodes. The simulation parameters are summarised in Table 1.[2]

---

[1] Activity is defined as time spent "online" by devices, i.e., being connected to either cell towers or other neighbouring devices.

[2] As our simulations do consider control traffic to take no time at all, only the distribution of transfer times is significant: For low bandwidth, it takes ≈1.5…15000 s to transfer messages, medium results in ≈0.15…1500 s, and high corresponds to ≈0.015…150 s.

| | |
|---|---|
| Message count | 100 |
| Message size | 1.6E3. . . 1.6E7 B |
| Size distribution | Power law |
| Replication | $r = 4$ |
| Erasure coding | $k = 4$ |
| Bandwidth (low) | 100 kiB/s |
| Bandwidth (med) | 1000 kiB/s |
| Bandwidth (high) | 10,000 kiB/s |

Table 1: Simulation parameters

Bluetooth trace has 81 participant devices but several non-participant devices had more online time than participant devices. Therefore, we decided to include them in our analysis as well, resulting the device count of 1858. Due to the ordinary performance by all the algorithms, we intensified the selection criteria for source destination pairs by setting it to 70th percentile of the devices w.r.t online time, we have simulated two different source-destination configurations.

### 4.3.3 Cleanup

In Access Point trace, we removed 7 devices that had no connectivity to other device due to spatial or temporal locality. We have found 3 clusters of access points as also suggested by the creators that the trace is collected from 3 different buildings. We have used the network present in the largest building in which, we found 129 access points with 928 devices connected at different time intervals to them. We believe the disagreement between our access point count of 129 vs. 131 provided by creators is due to the absence of 7 devices that we removed from trace at the beginning.

### 4.3.4 Variations and confidence

We have simulated three different configurations of source-destination pairs for Cell Tower scenario. We have constructed two different source-/destination pair configurations for November 2004 to confirm the behaviour of protocols. This way we have tried to avoid the coincidental bias to any protocol that may have been there due to nature of input. Furthermore, to our interest, we have created one more configuration for another month that has the second highest activity among the traces i.e., October 2004. This helps us to see the correlation between different month depending on the nature of activity for several protocols. Given the very long simulation times for many algorithm/trace combinations, it has not been possible for us to run enough independent simulations to obtain formal confidence intervals. Visual comparison among the data sets show a very high consistency in both shape and values.

### 4.3.5 Timing

As the simulated time span is approximately one month, each message has a lifetime of one week with birth time for all the messages evenly distributed throughout the third week, while the first two weeks are used as the history for the protocols that depend on it.

### 4.3.6 Link sharing.

Each device can only participate in one communication with another device. There are enough independent channels available that any number of node pairs can communicate at the same time with full bandwidth, independent of their proximity to other pairs.

### 4.3.7 Shortcut

We have tried to treat all the algorithms nicely by delivering the message to destination in case of direct contact of the current carrier of the message with the destination ultimate destination, even when the predetermined route requests to transmit to another node first.

### 4.3.8 Imperfect Oracle

In analogy to the observations of Apostolopoulos et al. [10] for frequent link QoS updates, frequent transmission of connectivity changes or history will severely reduce the network bandwidth, even more so in a mobile environment where dynamics are high and bandwidth scarce. The simulator assumes that these topology exchanges happen out-of-bound, clearly unrealistic, especially for Earliest Delivery. We therefore wanted to examine the impact of small prediction errors on the routing performance. To do this, we created two versions of Earliest Delivery: Perfect Oracle, which corresponds to the scheme presented by Jain et al. [11], and Imperfect Oracle, described below. To bring Perfect Oracle at par with real world scenarios, we have created an imperfect contact oracle that share most of the properties of contact oracle but its accuracy for predicting the future is 1/3 of the contact oracle. We have introduced 2 different types of weak errors in the contact oracle through the following mechanism, each responsible for modifying the predictions of 1/3 of the devices.

**Mistiming:** Assume that the start and end times of contacts may be off. This is done by randomly moving the start or end point while maintaining the middle of the active period as active and the middle of the idle period as idle. This ensures that two devices will still meet, but maybe somewhat earlier, later, longer, or shorter.

**Systematic errors:** Exchange the timelines of two similar devices, namely the ones that have seen each other the most number of times. This is comparable to someone changing his habits and, while still a weak modification, stronger than mistiming.

We have, in this way, tried to introduce the alterations, that are in line with the system and may be somewhat closer to what a realistic contact oracle can achieve. Here again, whenever the system realises that a packet has failed to take a hop or the hop was not available at the predicted time, then a new route to destination according to imperfect oracle is computed.

#### 4.3.9 Simulated Algorithms

Before presenting the descriptive summary of all the simulated algorithms, qualitative summary in Table 2 shows, Flooding has high performance on the cost of storage as well as communication in contrast to Perfect Oracle, that achieves high performance on the cost of processing and communication overhead. MaxProp on the other hand, has an impressive performance just with processing

| **Algorithm** | **Knowledge** | **Processing** | **Storage** | **Communication** | **Performance** |
|---|---|---|---|---|---|
| Direct Delivery | + | + | + | + | + |
| Flooding | + | + | +++ | +++ | +++ |
| First Contact | + | + | + | +++ | + |
| Simple Replication | + | + | ++ | + | ++ |
| History-based Replication | ++ | ++ | ++ | + | ++ |
| History-based Erasure Coding | ++ | + | ++ | ++ | ++ |
| Estimation-based Erasure Coding | ++ | +++ | ++ | ++ | +++ |
| Mobile Vehicle | ++ | ++ | ++ | + | + |
| MaxProp | ++ | +++ | + | + | +++ |
| Perfect Oracle | +++ | +++ | + | +++ | +++ |
| Imperfect Oracle | ++ | +++ | + | ++ | + |

Table 2: Algorithm Characterization

cost. Estimated Erasure Encoding is successful in reducing storage and communication overhead due to inherent advantages of erasure encoding. removePlease find a quick description of the algorithms taken from the literature that we simulated. For more details, we refer to the actual publications.

**Direct Delivery.** The source holds the data until it comes in contact with the destination. Direct Delivery uses minimal resources since each message is transmitted at most once. However, it may incur long delays [12] and frequently shows poor performance (Table 2).

**Flooding/Epidemic Routing.** Each node forwards all the non-duplicated messages (including messages received on behalf of other nodes) to any other node that it encounters. Flooding has the potential to deliver messages with the minimum delay if there are no resource constraints, such as link bandwidth or node storage [12, 13]. In our implementation, flooding avoids transmitting a message to a devices which already has a copy using the ihave/sendme model [14].

**First Contact Routing.** Messages in this scheme follow a seemingly random path determined by a hot-potato algorithm. The next hop is chosen randomly from the available neighbours, if any. Otherwise, it is handed off to the node coming into proximity first [11]. This phenomenon can cause the message to hop among a group of two or more peers for a long time until the one having the message leaves the group. To reduce this overhead, our simulation prevents returning the message to one of the previous 10% of the hops the message has travelled. The choice of a next hop does not try to make progress towards the destination; therefore, messages may aimlessly propagate through the network.

**Simple Replication.** This is a simple replication strategy in which identical copies of the message are sent over the first $r$ contacts, with $r$ known as the replication factor. Only the source of the message sends multiple copies, the relay nodes are allowed to send only to the destination; they cannot forward it to another relay. This makes it a mixture between direct delivery and flooding [12]. This algorithm has medium consumption of bandwidth and storage.

**History-Based Simple Replication.** In this technique, the source creates $r$ identical copies of a message, who are then delivered to the "best" $r$ nodes, where quality is determined by history. The intermediate nodes will then each perform Direct Delivery. Our simulation follows the ZebraNet model of relying on the frequency at which a node has encountered the destination [12, 15].

**History-Based Erasure Coding.** This mechanism works very similar to history-based simple replication, but $kr$ fragments totalling $r$ times the message size are generated and sent to the best $kr$ intermediate nodes. The intermediate nodes will deliver only to the final destination, where any $k$ fragments can reconstruct the message. This has the same performance as Simple Replication when the path failure model is Bernoulli and the contact volume is sufficient for an entire message [12].

**Estimation-Based Erasure Coding.** History-Based Erasure Coding is an all-or-nothing function: The nodes with highest probability get all the data and path length is limited to two hops. EBEC [16] is more adaptive, as the two communicating intermediate nodes exchange data until the number of fragments for a given destination is proportional to the nodes' probability of meeting the destination. To accelerate the simulation, history is calculated at intervals of 5 minutes and history oracle is used.

**Mobile Vehicle Routing.** The routing decision is based on finding a peer that has the highest probability of visiting the region of the destination. In our simulation, we allocate each peer to a home access point or cell tower, depending on the time they have spent with different access points or cell towers. Then we try to find out the peer that is most probable to visit the home region of destination [4]. Both the source and the selected node try to perform Direct Delivery to the destination, which results in a slightly higher resource consumption than Direct Delivery alone.

**MaxProp Routing.** MaxProp attempts to forward the message to any device that has the greater probability to deliver the message to destination. MaxProp involves calculating the path for each message at each transfer opportunity using a modified Dijkstra algorithm with history as pivotal criterion. MaxProp defines its own way of computing history to dictate the path computation but it is assumed that topology information does not consume bandwidth. It also incorporates a fancy mechanism of message queuing at peer level that prefers the newly born messages and degrades the priority of messages based on the number of hops they have travelled and the delivery probability [17]. Even without the computational complexity of erasure coding, MaxProp is hungry for processing resources as the maintenance of the local queue is expensive for mobile devices under high message counts.

**Earliest Delivery a.k.a. Perfect Oracle.** The path of a message is computed using a modified Dijkstra algorithm [11], where the link costs represent the waiting time for the next contact between the vertices. It assumes a contact oracle which has perfect foresight of future node encounters, equivalent to knowing the time-varying DTN multi-graph. This algorithm is bound to perform better than all of the others because it has the unrealistic knowledge of the future. A message may still fail to reach the destination due to complete lack of a path to destination or congestion.

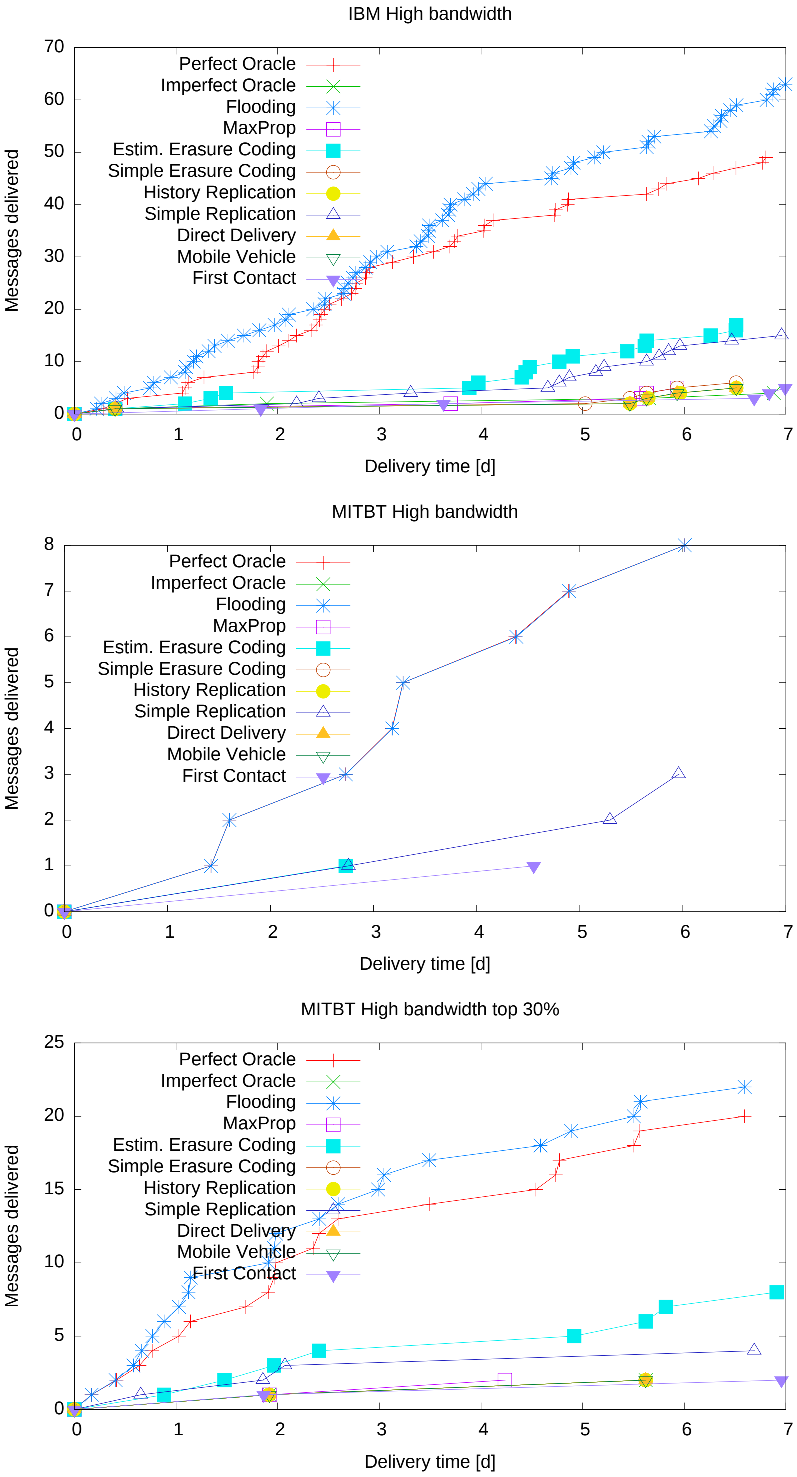
Jain et al. [11] have also proposed more advance version of Earliest Delivery which include knowledge of the local queue, or using an even less realistic oracle, a global overview over all messages in the system at any given time in the future. With local queue knowledge (as in our simulation), a new path is generated when a node realises that the packet has been unable to reach the next hop in time. This results in consumption of lot of processing power if the bandwidth is low or the path is a repetitive failure (e.g., for bottleneck paths).

Table 2 summarises the relative performance. For storage, note that we assume the sum of message sizes in the network dominates the amount of node information.

# 5 Results

To get good amount of confidence in our readings we have made multiple runs for Cell Tower as well as Bluetooth traces. As already discussed, we observed maximum activity in the month of November 2004 and after that in October 2004. We performed 2 runs on November and one run on October whereas in Bluetooth case, the performance of all the algorithms was very ordinary due to lack of connectivity among the nodes and size of the network. To get more meaningful output from bluetooth traces, we intensified the selection criteria of source and destination to top 30% of online nodes and as predicted some algorithms showed considerable performance with this setup. We have plotted 2 different kinds of graphs against time i.e. No. of messages and amount of data delivered.

Figures 2,3,4,5,7,8 show the graphs of Time Vs No. of messages with high, medium and low bandwidth respectively. These figures show how well messages are delivered for the three environments. For example, in IBM and MIT Bluetooth High bandwidth case (Figures 2,9) we can observe that perfect oracle and flooding are dominant among all the algorithms and as the bandwidth decreases (Figures 4,11,7,8 )the performance degrades and algorithm that don't employ replication show significant performance. In MIT Bluetooth low bandwidth case (Figures7,13) we observe that intensifying the criteria of source, destination pairs gives a significant delivery increase to flooding, perfect oracle and Estimation based Erasure coding.



IBM High bandwidth

MITBT High bandwidth

MITBT High bandwidth top 30%

Figure 2: Plots for No. of Messages delivered with High Bandwidth from Access point and Bluetooth traces

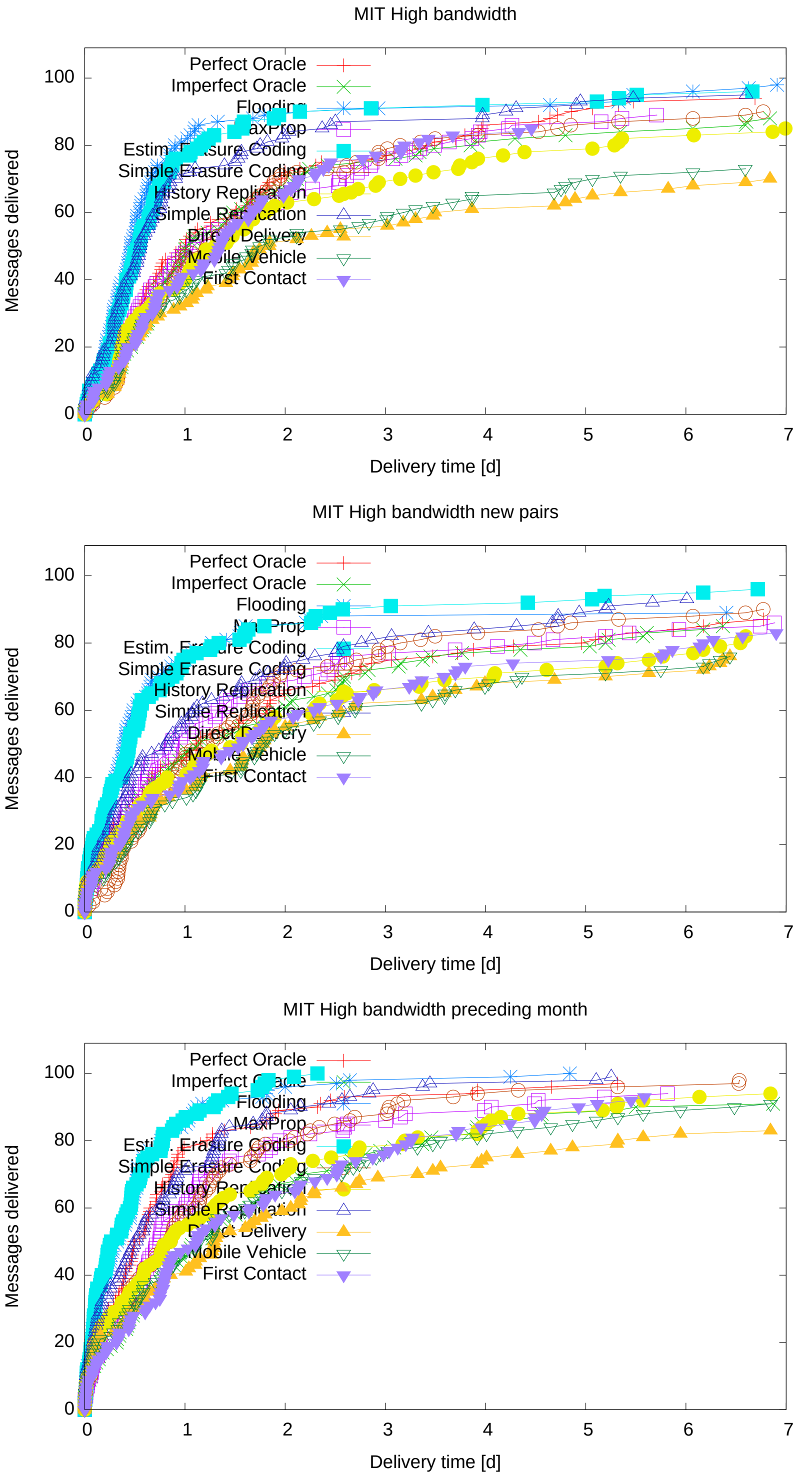


Figure 3: Plots for No. of Messages delivered with High Bandwidth from Cell Tower traces

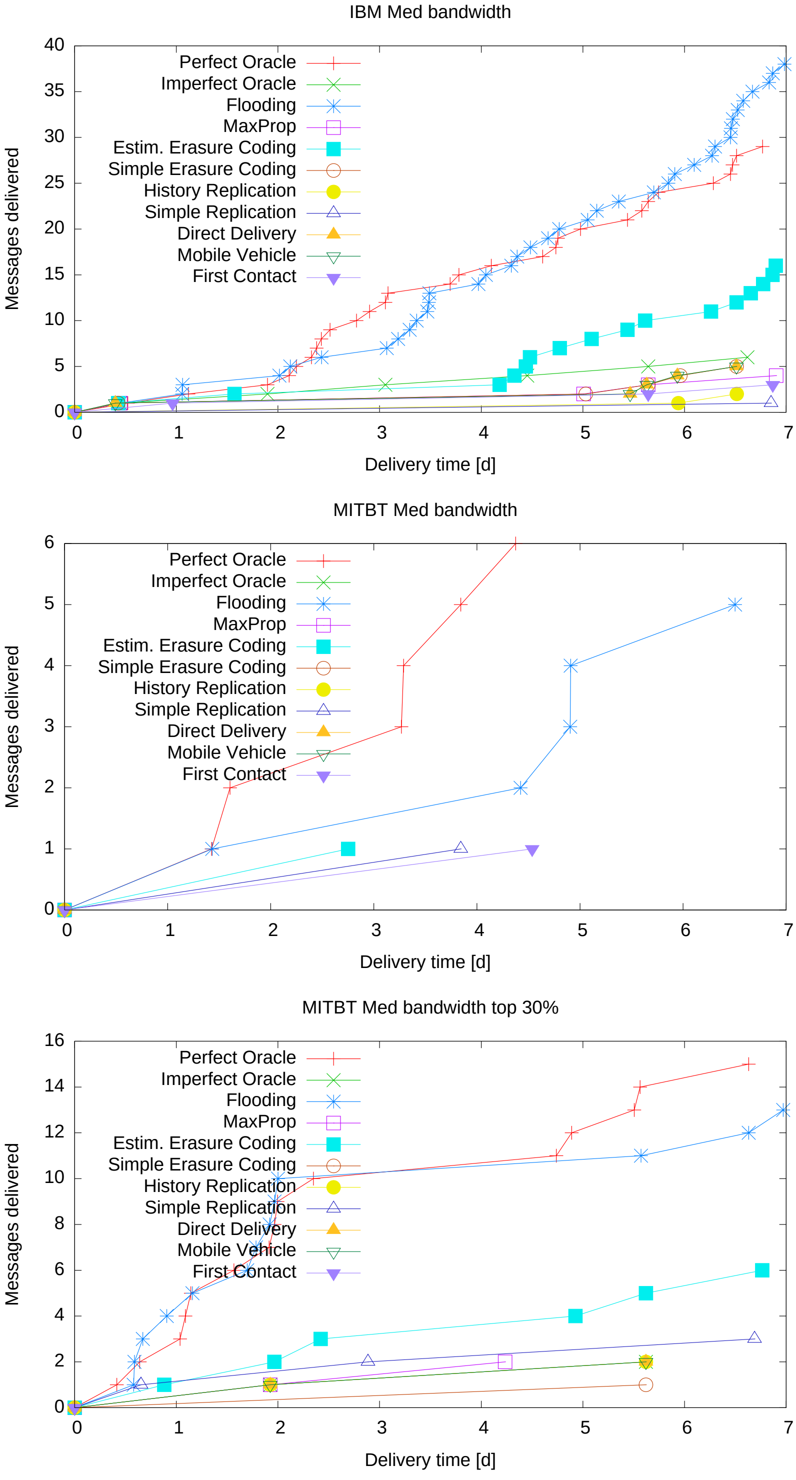


Figure 4: Plots for No. of Messages delivered with Med Bandwidth from Access point and Bluetooth traces

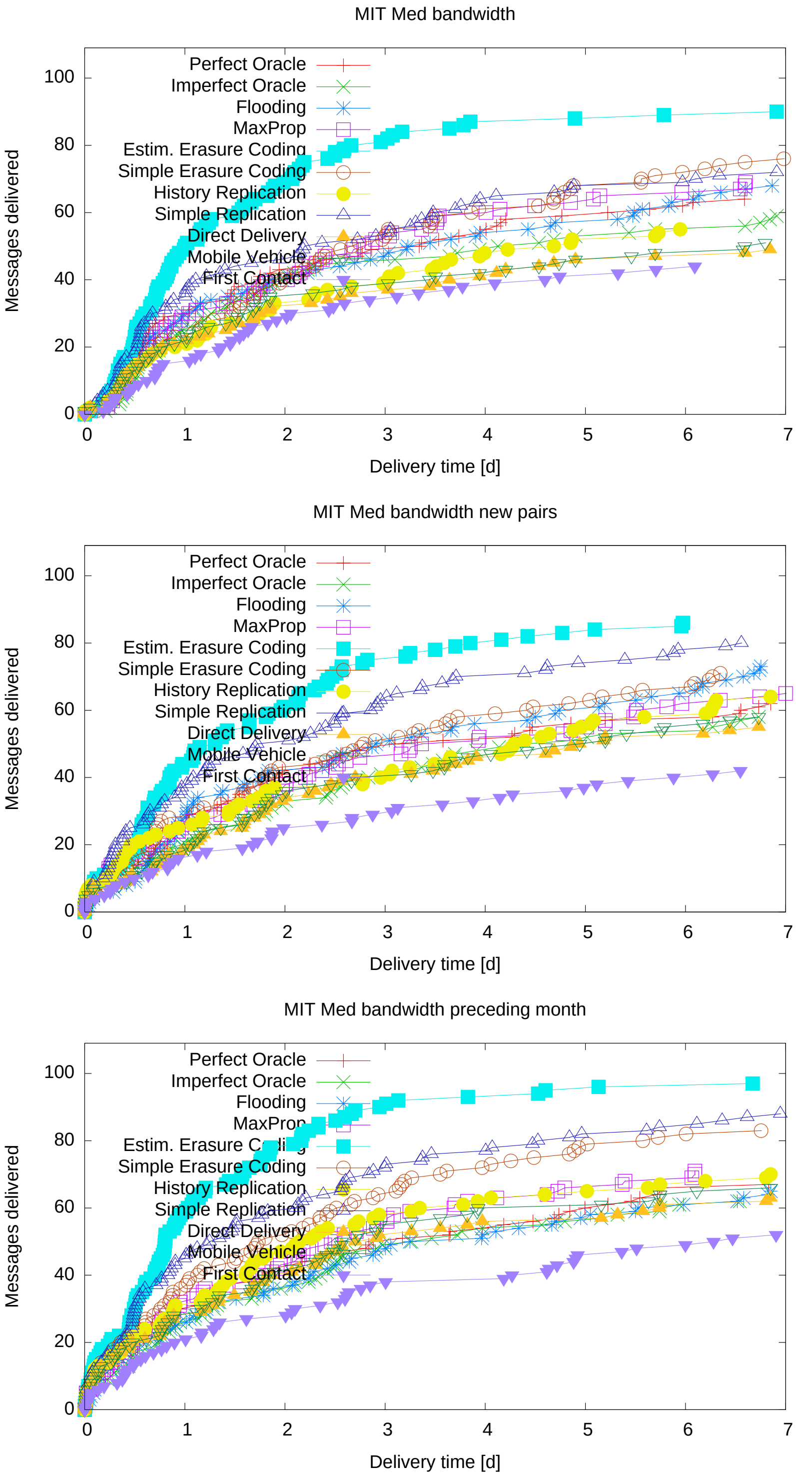


Figure 5: Plots for No. of Messages delivered with Med Bandwidth from Cell Tower traces

In the MIT Bluetooth case(Figures 9, 11, 13) we can see that Perfect Oracle early on starts delivering messages and after almost six days, it has delivered close to 50-120 MBytes, about 10%-25% of the total message load. The runner-up, Flooding, starts a little bit slower, but after 2 days, it has caught up and will remain close to the winner. In the IBM Access Points scenario (Figures11), however, Flooding, Direct Delivery, Mobile Vehicle, Imperfect Oracle, and EBEC all deliver their first big message half a day after it was injected into the network; Perfect Oracle requires about four times as long and four smaller messages until it reaches the same volume. In the meantime, its ugly stepbrother Imperfect Oracle has taken the performance lead for a day, but after seven days, Flooding has delivered roughly 25% of the data, collecting the Maillot Jaune.

In the MIT Cell Tower scenario (Figures 3,10), High bandwidth plots show the strong connectivity of the network as Direct delivery also shows considerable performance and its is not easily to rank the algorithms in this case. In low and medium cases (Figures 5,12, 8,14)) however, things look very different again: MaxProp immediately leads the pack with a huge margin, having delivered about 90% of the total data in just 2.5 days. EBEC and Simple Replication start catching up then, but remain without chances.

## 5.1 Reasoning

Why does this happen? Why does the Perfect Oracle behave so poorly despite its omniscience? The following paragraphs will try to answer these questions.

**Omniscience** . . . is not all. First of all, Perfect Oracle is *not* omniscient, it lacks knowledge about concurrent traffic so it cannot avoid bottlenecks. Even worse, it does not include message size into the calculation, which can result in the choice of a path which does not provide long enough connection times to transmit the message even in the absence of other messages. The latter could be avoided, but not the former.[3] It also seems that the perfection is bad, as the selection of the “best” path is predictable. Even in the Cell Tower case, connectivity seems to be sparse enough to create a few attractive bottleneck

---

[3] It would probably require a high-speed ubiquitous wireless network for topology/traffic information exchange. If you have such a network, why not use it for the actual data?!

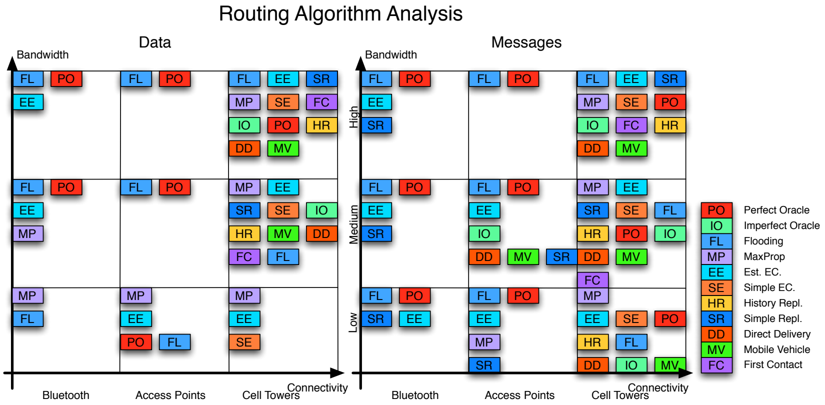


Figure 6: Performance Behaviour of Routing Algorithms w.r.t. Amount of Data (left) and Number of Messages (right) Delivered

links, through which large portions of the traffic should be funnelled. When this fails, the remaining messages need to be rerouted, probably again along similar routes, creating more bottlenecks.

This is where other algorithms including Flooding take control of the network. In a well-connected environment such as Cell Towers, Direct Delivery and its two-hop cousin Simple Replication perform well, the latter delivers about 80% of the data; even the hot potatoes from First Contact manage to reach 20%.

**Comparison.** To answer further questions, we compiled a figure of merit, Figure 6, summarising all the best algorithms in each of the nine bandwidth/ connectivity areas. In each square, the top row corresponds to the top performers, which are roughly on par with each other. The second row describes the second group, and so on. Algorithms not mentioned in a square perform very poorly.

What we can see is that only 5 of the 11 algorithms make it ever to the top: Flooding and Perfect Oracle dominate under weak connectivity, while MaxProp works well under high connectivity. Flooding, EBEC and Simple Replication can take advantage of “nice” networks (high bandwidth, high connectivity). As Simple Replication gives opportunity to each replica of the message to take at-least one hop, therefore, in a dense network like Cell Tower, its performance in

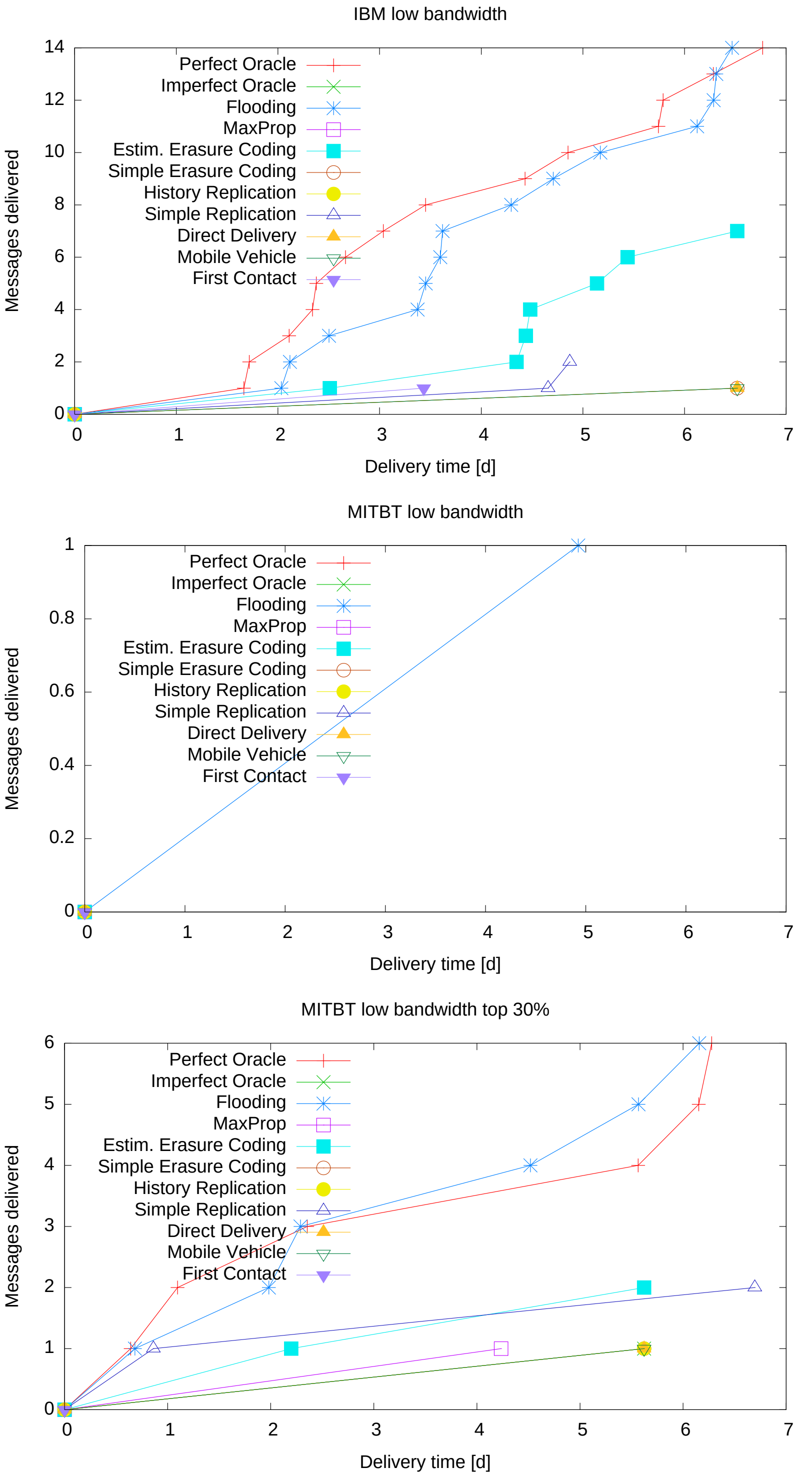
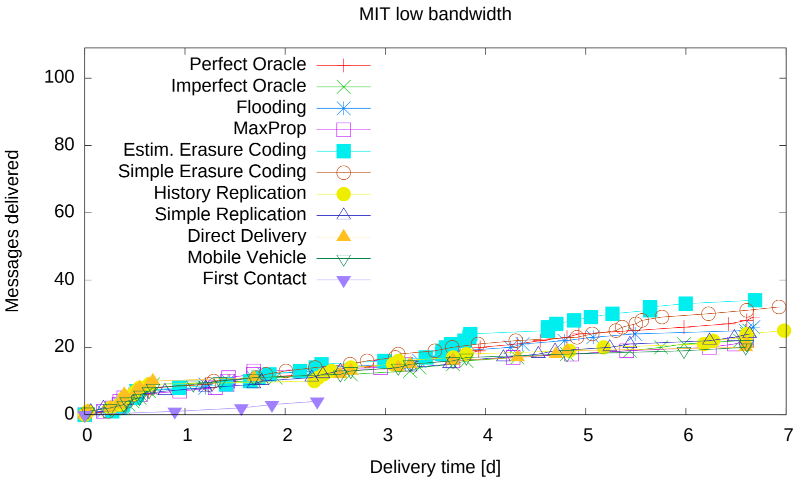
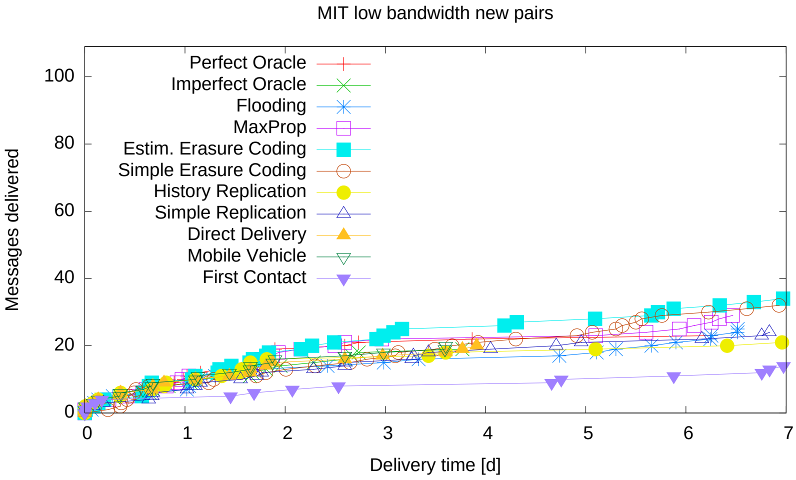


Figure 7: Plots for No. of Messages delivered with low Bandwidth from Access point and Bluetooth traces


MIT low bandwidth


MIT low bandwidth new pairs

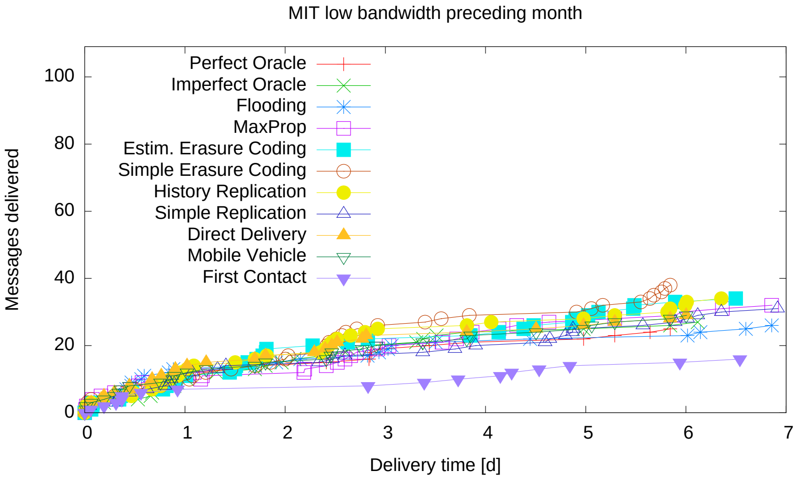

MIT low bandwidth preceding month

Figure 8: Plots for No. of Messages delivered with low Bandwidth from Cell Tower traces

high bandwidth case is very impressive.

MaxProp performs very well when bandwidth is low, but does not seem to be able to take advantage of higher bandwidths: It delivers not significantly more despite 10- or 100-fold increases of bandwidth. Its noteworthy that MaxProp is the only algorithm that defines the queue management mechanism for one device, that apparently gives it an edge. This technique gives priority to messages that are destined for next hop and sorting the remaining messages according to their age, i.e., younger messages are given priority for next transfer opportunity. We believe that this strategy provides quick delivery of fresh messages while being persistent about older messages. Together with the fine-grained proportional message sharing, this ensures that many pieces of the message follow the best path quickly.

MaxProp also employs a unique strategy to compute history information that in our view helps the algorithm to better determine the candidate next hops. We believe that the history normalisation employed by MaxProp helps reducing the chance that too many messages will be loaded onto a very mobile device, which does not stay long enough in the vicinity of the destination to actually deliver all messages.

If you wanted to implement only one algorithm, hoping it would perform reasonable under most regimes, Flooding and EBEC are the candidates to consider (Perfect Oracle can be excluded, as it seems impossible to implement the necessary oracle in real life).

# 6 Conclusion

To our knowledge, this is not only the first comprehensive comparison and analysis of DTN routing algorithms, but also the first to approach realistic communication models. To our own surprise, many of the simpler algorithms, under the broad leadership of Flooding, perform among the best in all classes.

To be among the best here does not imply to be good. For many scenarios, the performance even of the best is lousy, especially at low bandwidths or low connectivity. Therefore, a huge challenge lies before us, namely, *to make the world ready for DTNs* by creating higher densities and higher bandwidths. Even

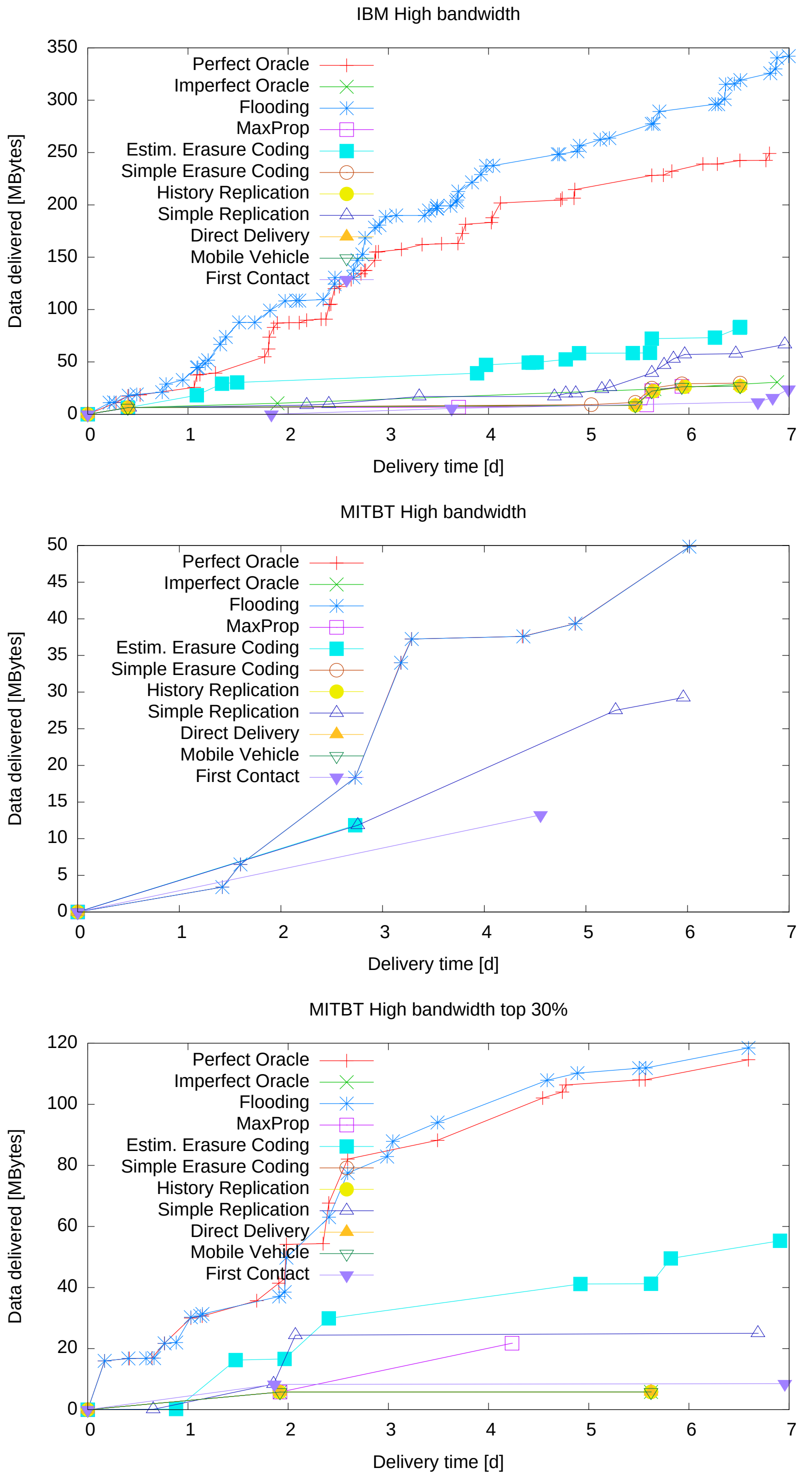


Figure 9: Plots for Data delivered with High Bandwidth from Access point and Bluetooth traces

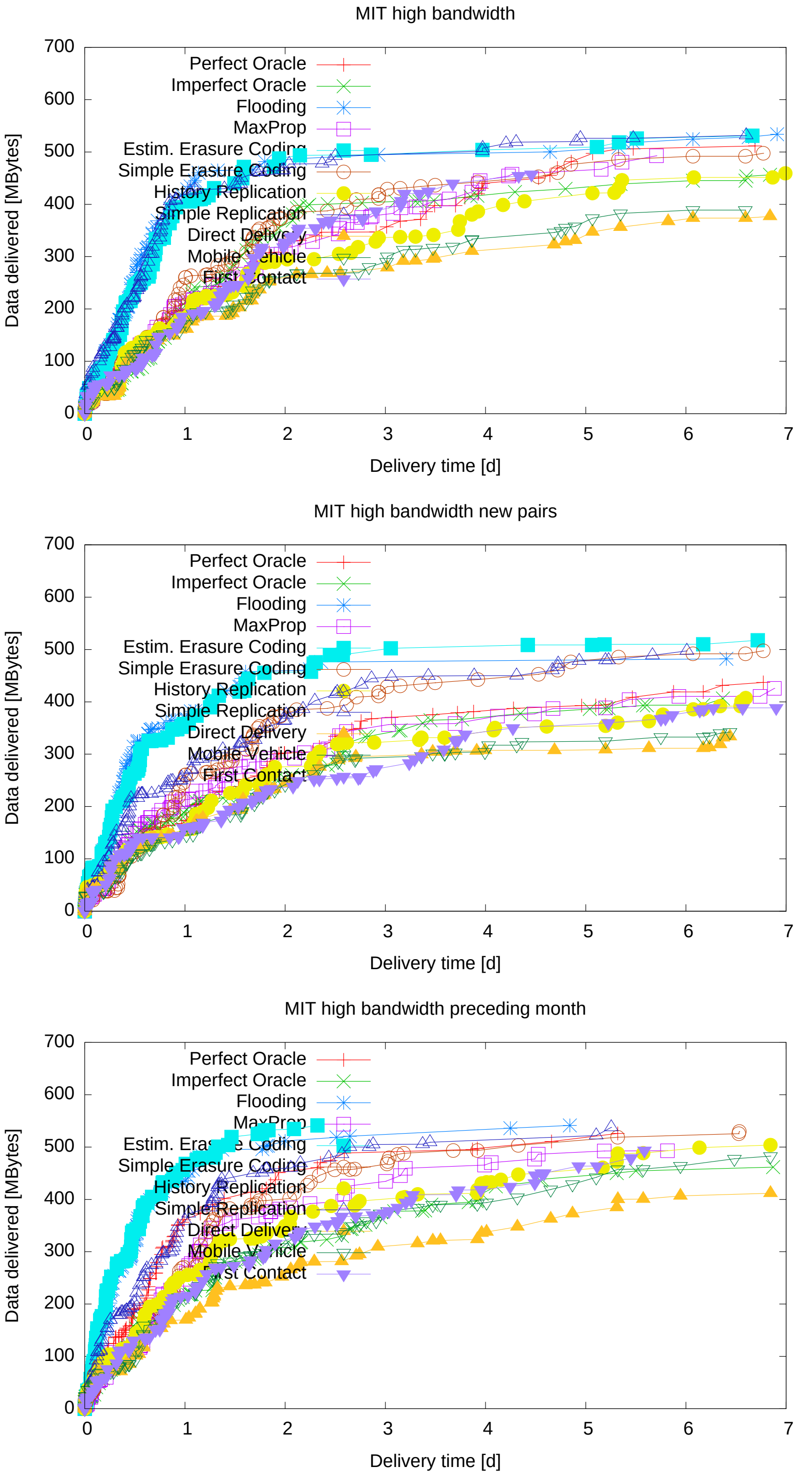


Figure 10: Plots for Data delivered with High Bandwidth from Cell Tower traces

with a tenfold increase, waiting times of around a day and success rates between 5 and 80% will still be common, which does not make it ready for prime-time yet.

So there is also a goal for DTN researchers, namely, *to make DTNs ready for the world.* This will require more research, combining the best mechanisms of the top candidates (Flooding, EBEC, MaxProp) and maybe invent some more. A further challenge will be to create protocols that will perform well in all domains; this possibly includes the need for adaptive protocols.

Social algorithms may have to be included in the list, where the device knows more about its human carrier and her friends than just statistics. Coupling with a calendar and other information resources may open new avenues, but retaining privacy under these circumstances will be a daunting task.

So far, we have no clear winner, but many good candidates. Each of them probably has a home turf on which it performs extremely well. Even with the DTN community's goal to find the all-encompassing solution, many niches will remain, where specialised algorithms will always excel.

In either case, the task remains to find the reason *why* algorithms perform the way they do and then apply these techniques to the appropriate domains and niches.

# References

[1] J. A. Scholte, "Globalization: A critical introduction,," *Basingstoke: Palgrave,*, vol. I, p. p. 68., 2005,.

[2] UNCTAD, "The information economy report the development perspective, new york," United Nations Conference on Trade and Development, Tech. Rep., 2006.

[3] kevin, "Delay tolerant network research group." [Online]. Available: http://www.dtnrg.org/

[4] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proceedings of IEEE Infocom*, 2005.

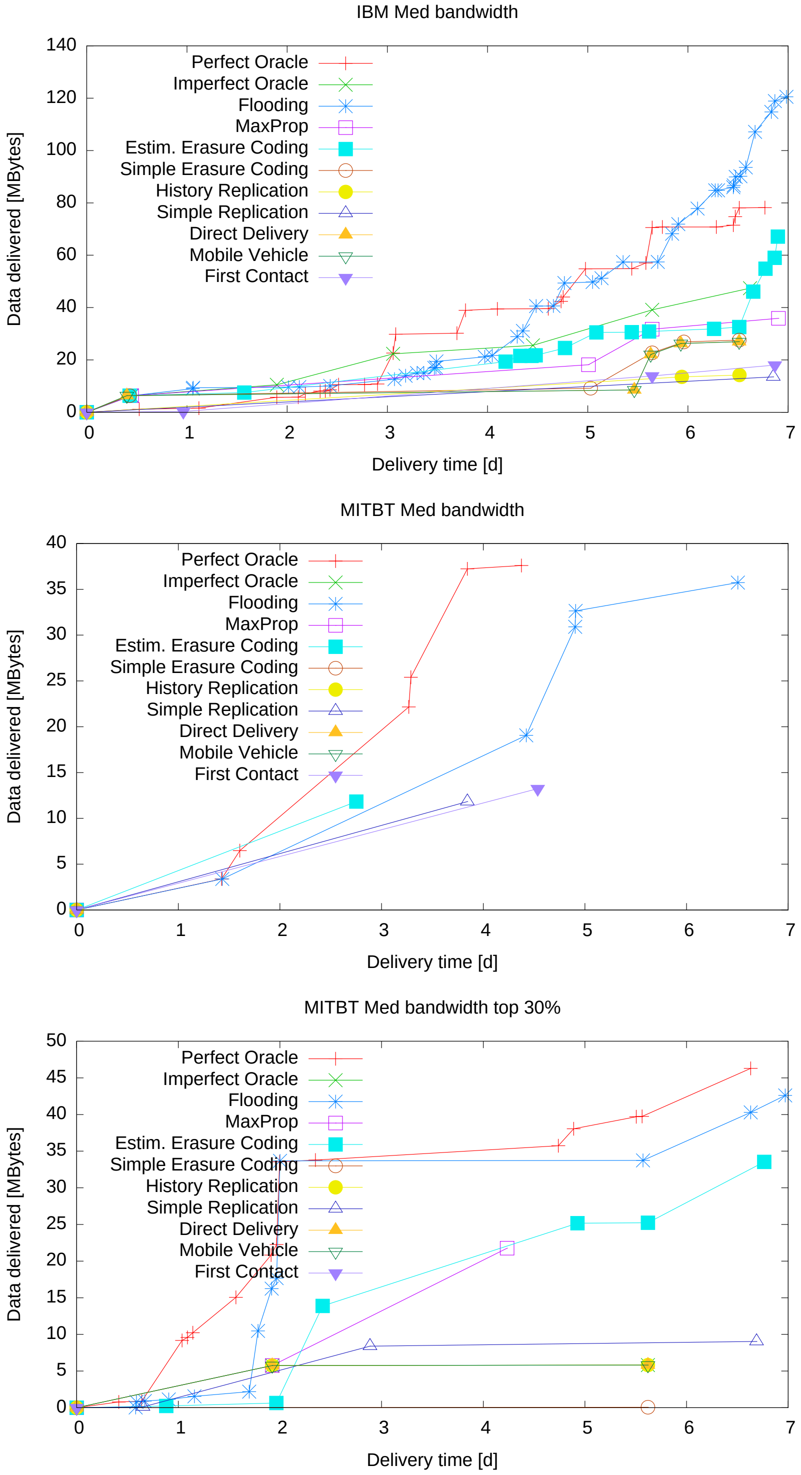
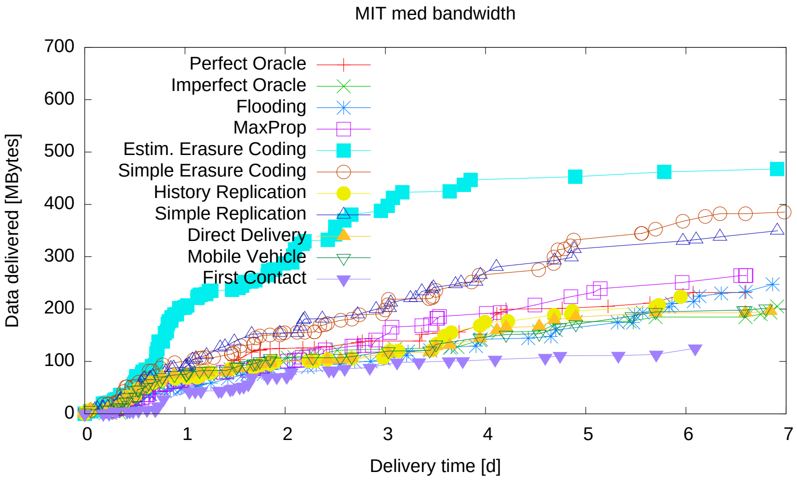
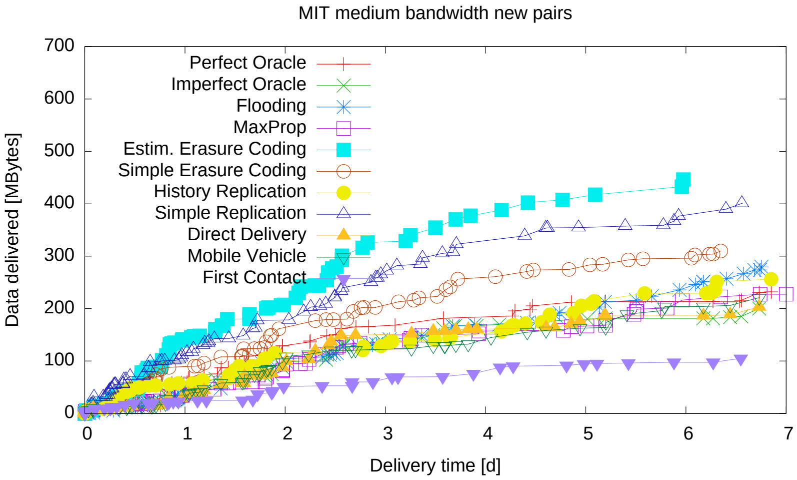


Figure 11: Plots for Data delivered with Med Bandwidth from Access point and Bluetooth traces





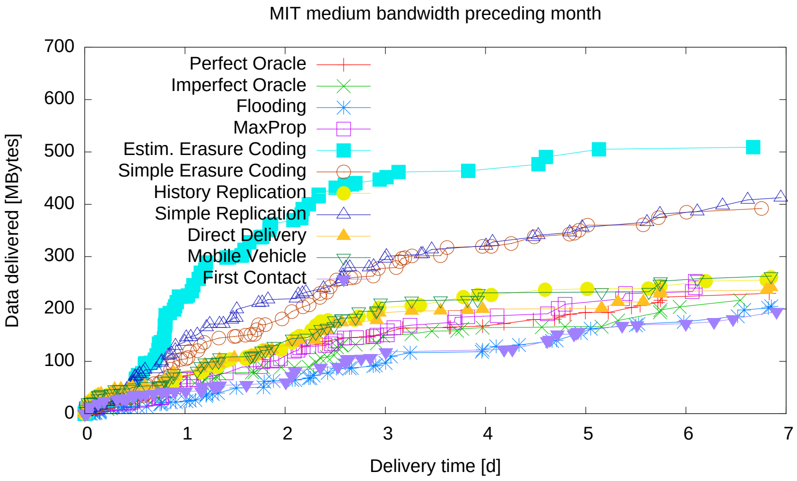


Figure 12: Plots for Data delivered with Med Bandwidth from Cell Tower traces

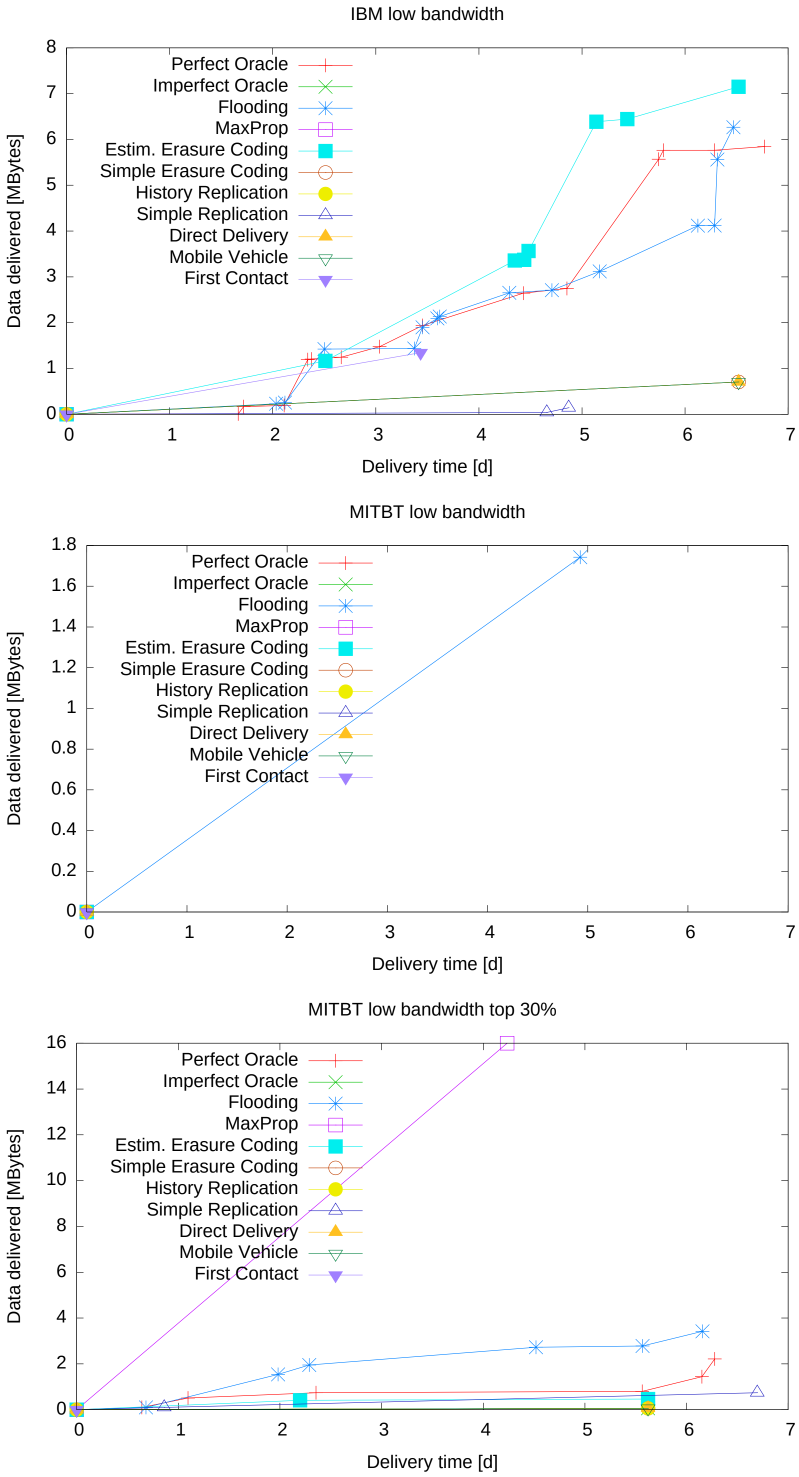
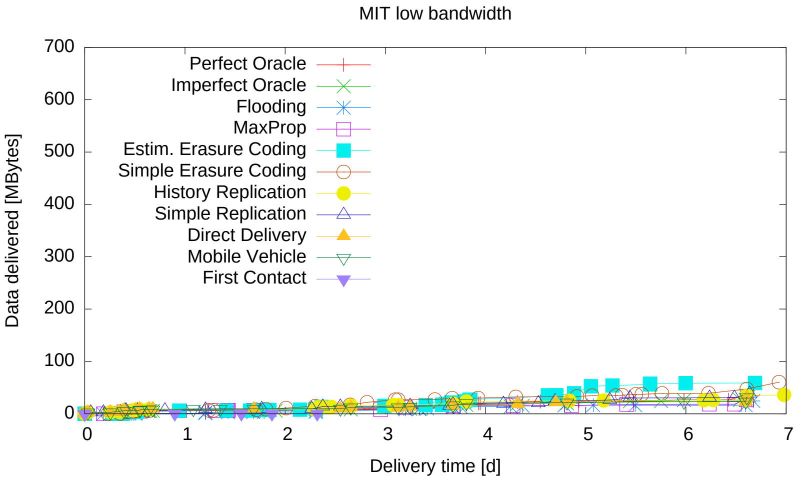


Figure 13: Plots for Data delivered with low Bandwidth from Access point and Bluetooth traces



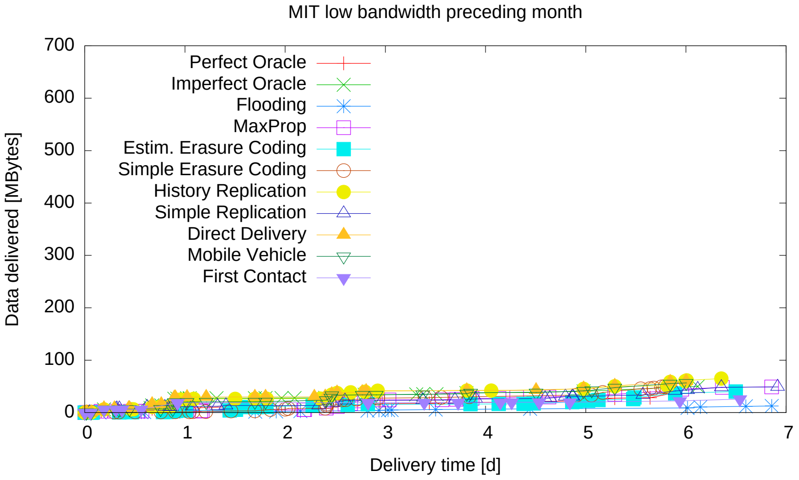


Figure 14: Plots for Data delivered with low Bandwidth from Cell Tower traces

[5] E. P. Jones and P. A. Ward, "Routing strategies for delay-tolerant networks," in *ACM Computer Communication Review (CCR).*, 2005.

[6] A. S. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking connectivity in developing nations," *IEEE Computer*, vol. 37, no. 1, pp. 78–83, Jan. 2004.

[7] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[8] N. Eagle and A. S. Pentland, "CRAWDAD data set mit/reality (v. 2005-07-01)," Downloaded from http://crawdad.cs.dartmouth.edu/mit/reality, July 2005.

[9] M. Balazinska and P. Castro, "CRAWDAD data set IBM/Watson (v. 2003-02-19)," Downloaded from http://crawdad.cs.dartmouth.edu/ibm/watson, Feb. 2003.

[10] G. Apostolopoulos, R. Guérin, S. Kamat, and S. K. Tripathi, "Quality of service based routing: a performance perspective," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 17–28, 1998.

[11] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of SIGCOMM 2004*. ACM Press, 2004, pp. 145–158. [Online]. Available: http://portal.acm.org/citation.cfm?id=1015484

[12] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 2005, pp. 229–236.

[13] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. CS-200006, April 2000.

[14] B. Kantor and P. Lapsley, "Network news transfer protocol," Internet Engineering Task Force, RFC 977, Feb. 1986.

[15] S. Jain, M. Demmer, R. Patra, and K. Fall, “Using redundancy to cope with failures in a delay tolerant network,” in *Proceedings of ACM SIGCOMM*. New York, NY, USA: ACM Press, 2005, pp. 109–120.

[16] Y. Liao, K. Tan, Z. Zhang, and L. Gao, “Estimation based erasure-coding routing in delay tolerant networks,” in *Proceedings of IWCMC*, June 2006.

[17] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, “Maxprop: Routing for vehicle-based disruption-tolerantnetworking,” in *Proceedings of IEEE Infocom*, Barcelona, Spain, Apr. 2006.