

A Pub/Sub Message Distribution Architecture for Disruption Tolerant Networks

Sergio CARRILHO[†], *Nonmember* and Hiroshi ESAKI[†], *Member*

SUMMARY Access to information is taken for granted in urban areas covered by a robust communication infrastructure. Nevertheless most of the areas in the world, are not covered by such infrastructures. We propose a DTN publish and subscribe system called Hikari, which uses nodes' mobility in order to distribute messages without using a robust infrastructure. The area of Disruption/Delay Tolerant Networks (DTN) focuses on providing connectivity to locations separated by networks with disruptions and delays. The Hikari system does not use node identifiers for message forwarding thus eliminating the complexity of routing associated with many forwarding schemes in DTN. Hikari uses nodes paths' information, advertised by special nodes in the system or predicted by the system itself, for optimizing the message dissemination process. We have used the Paris subway system, due to it's complexity, to validate Hikari and to analyze it's performance. We have shown that Hikari achieves a superior deliver rate while keeping redundant messages in the system low, which is ideal when using devices with limited resources for message dissemination.

key words: *Delay Tolerant Networks, Publish and Subscribe, ad-hoc networking, content based routing*

1. Introduction

Information is critical for social and industrial activities on Earth, and is the innovation driver of our society, today. Nevertheless, the majority of the world has limited or no access to information because most of the populated areas are not covered by a communication infrastructure. There are many applications that do not require real time information delivery such as educational material for remote villages in developing countries, blog information, newspapers, weather forecasts and newsgroup subscriptions.

The area of Disruption/Delay Tolerant Networks addresses the problem of communication in an environment characterized by partitions, delay and errors in the network. In such environments, existing protocols do not work properly. In DTN, messages are forwarded using opportunistic contacts between nodes in the network. Since the DTN network model is usually represented by a time varying multigraph [1] where path information is not known beforehand, one of the great challenges in this area is knowing how to select paths for message delivery in an environment characterized by uncertainty of paths, nodes to be encountered and network status. In such environments, epidemic style algorithms [2] are the ones that achieve the best delivery ratios but consume resources, such as buffer space,

and add redundancy and noise to the system.

In a publish and subscribe system [3], a client node expresses its interest in a certain topic or a list of topics, which the node wants to receive, to the network. When there is any event related to the topics specified/requested by the subscriber node, the publisher node(s) is/are notified by the system so as to deliver the information to the subscriber node(s) when the information is available to be delivered.

For designing forwarding and routing protocols in DTN, many works assume that there is no infrastructure in the system. Therefore, nodes depend only on information of previous contacts and probabilistic methods to forward messages. This brings the problem of having to store and manage node identifiers, which makes routing complex in DTN. There are scenarios though, where information of node's movement paths can be known beforehand. Having such information decreases considerably the complexity of message forwarding. Imagine for example, a subway system or a regional bus system, where stations are fixed and bus/train movement can be known. Buses, trains or people travelling in them, can be used to distribute messages to other locations.

We propose the **Hikari** system, which delivers information to remote places only based on its content. The novelty of our approach is that no identifiers have to be assigned to nodes or managed in the system, and information is forwarded using a pub/sub concept, where all nodes are publishers and/or subscribers. We show in this paper that such approach decreases considerably the complexity of routing and achieves a better performance than legacy DTN schemes. For evaluating the system, we built tools to model the real Paris subway system, because of its complexity, and used a simulator built by us, called MOBINET.

Our main contributions are:

The Hikari System: A DTN message distribution system that uses nodes' mobility and message content for message dissemination in a public transport scenario.

The modular MOBINET simulator: A modular generic simulator used for simulating message distribution in a DTN environment for public transport systems.

[†]The University of Tokyo

duces the Hikari system design including the message semantics. Section 3 then describes the simulation and methodology. Finally Section 4 presents the evaluation results and findings. Section 5 summarizes the related work in DTN and the conclusions are presented in Section 6.

2. The Hikari System

The Hikari system aims at achieving information distribution in disconnected environments, using minimum infrastructure and node mobility for disseminating information.

The main assumption in the Hikari system is that most of the locations are disconnected from the Internet and there is limited communication infrastructure between most locations (e.g. antennas, radio links or satellites). Each location has a small infrastructure point called the *Location Master* (LM). In this architecture, information and queries are carried and distributed by mobile nodes that move between different locations containing LMs. We consider that nodes have enough storage to carry some amount of information to other nodes or on their behalf. This assumption comes from the observation that storage devices are becoming increasingly cheaper, powerful and small enough, that portable devices can have a decent storage capacity.

There are 2 micro-environments in this system:

- **Back-bone Distribution System (BDS):** Message dissemination for remote regions.
- **Local Message Distribution System (LMD):** Nodes pass information between them using peer to peer communication.

In LMD nodes belong to a group, limited by distance, and they can exchange information between them. Take as an example a small village near a train station. Each person with a mobile device is considered a node in the LMD. In such scenario, nodes make queries and information is distributed in an epidemic way using pairwise contacts.

Although we have built and tested a proof of concept for LMD, in this paper we will focus on the BDS part of the Hikari system.

2.1 Message semantics

Message management is important in such a system, since if there are too many replicas or there are loops, important messages will be lost. We consider two features that contribute to eliminate redundancy and loops in the system:

Time To Live (TTL): Messages have a TTL counter that represents the lifetime of the message. This is the time when the information in the message is important. For example, weather information for a given day, would have the counter set for 24 hours, and

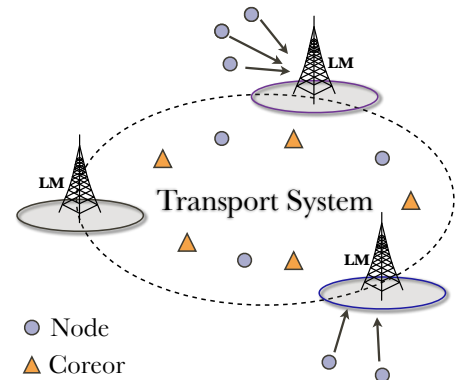


Fig. 1 The Hikari system's back-bone distribution system

would eventually die after 24 hours have passed. If a counter is set for a certain time, the counter will decrease using the clock in the host the message is located in.

Popularity rate (PR): Only popular messages are allowed to exist in the system. If a message is requested by many nodes, the PR in the LM will increase and it will be kept on the top of a stack. These stacks exist in the LMs and coreors. Messages with high PR stay on top of the stack, and the ones with low PR stay in the bottom of the stack. For a number of total requests, if the topic is not requested, its PR will decrease. If the PR of the message is low, it will be eligible for purging even if its TTL is high.

2.2 The Back-bone Distribution System (BDS)

The BDS is the part of Hikari system that deals with message distribution among remote areas. We consider remote areas as being spacial areas separated by a certain physical distance. There is some sort of physical communication between these areas, for example railways or roads, which allow nodes to travel from one location to another. These nodes are the ones that will make the queries, request, create, distribute and carry data between locations.

Figure 1 shows the BDS abstraction. There are 3 main components in the BDS:

Location Masters (LMs): Fixed points that store, retrieve and distribute messages and queries from/to nearby nodes. Since they are fixed, LMs do not have power constraints so much as mobile nodes, and they are located in strategic points like bus stops or train stations. They have full knowledge of the LM topology, that is, they know about the existence of the other LMs and know the virtual graph that represents the LM's locations. This information is used for the efficient message distribution between LMs.

Nodes: Units that query and provide information in the system but do not participate in the process of information dissemination. Usually we consider peo-

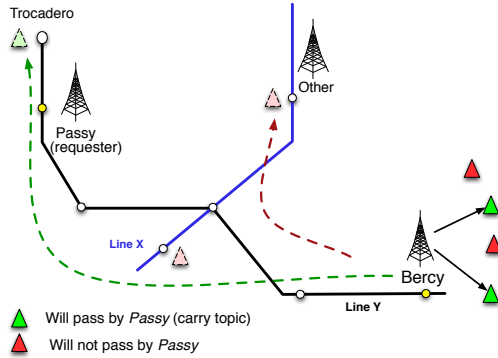


Fig. 2 Hikari in BDS

ple or mobile devices carried by people as being nodes. Nodes are not willing to use their resources in order to carry queries or information on behalf of other nodes.

Coreor: Special node that takes part in the dissemination of queries or data between LMs. Any node can choose to become a coreor. There are *static coreors*, which have a predefined movement, e.g. (buses or trains), and the *normal coreors* which chose to be coreors and do not have fixed movement patterns, as for example, people or mobile devices carried by people.

In BDS external nodes aggregate in *stations*, which are places where nodes gather in order to travel to remote locations. These nodes can be nodes that are going to use the stations to go to some other location or nodes that go the the stations to get messages. Each station has at least one LM.

The distribution system is composed by nodes and coreors that transit between stations. This is usually done using means of transportation such as trains, buses, cars and airplanes. An example of such a system is a metro system where metro stations are represented by ellipses around the LMs in Figure 1, and each station has LMs. Nodes can be either people or trains, which distribute messages in the system.

Figure 2 shows an example of the operation of Hikari in the BDS.

There are two phases in this system:

2.2.1 The query phase:

Nodes that are near a LM can communicate with the LM and they can:

- Request certain topics or messages.
- Become coreors and carry topics' requests or messages from the LM to other LMs in the location the coreors are moving to or locations in the path of their movement.
- In the case of coreors, they inform the LM their destination, so that the LM can compute which messages to send to various locations.

Looking at Figure 2, imagine that a node goes to

station *Passy*. We will consider the LMs to have the same name as the stations that they belong to. The node arriving in *Passy* asks *Passy* (the LM) for the topic *slashdot*, which is an electronic tech magazine. If *Passy* does not have the topic, it will query for the topic to a given number of coreors, for redundancy, that use *Passy* to go to other stations (preferably to different destinations) in the train system. Besides TTL and PR, the query has the topic name and the requester information (e.g. *slashdot:Passy*), so that when the destination LMs find the content, they will know where to send it to. Imagine that one of these coreors goes to station *Bercy*. This coreor will deliver the query to *Bercy*.

2.2.2 Message distribution phase:

If *Bercy* does not have *slashdot*, it will query nodes that are using the station and search for the topic. If the TTL of the message expires, it will be purged from the system. Otherwise, if the topic is encountered with one of the nodes, *Bercy* will try to find coreors going to *Passy* to deliver the message. Using the Hikari algorithm, for coreors that use *Bercy*, *Bercy* will have information about where coreors are going. *Bercy* will choose a certain number of coreors that go directly to *Passy* or that pass by *Passy* in their paths (if they go for example to *Trocaadero*). One of these coreors will deliver the message to *Passy*. *Passy* will store the content until the message's TTL expires or the PR decreases to a low value. *Bercy* also has the option to keep the message in case other nodes using this station ask for the same content. Depending on the buffer management policy, the message can be purged after a while or if it has low popularity or very small TTL, it will be erased.

The objective of the system is not to deliver a message to the original node who requested it. Since no identifiers are kept for the nodes that come and go, the system does not even know which node asked for a certain message, but it knows in which LM the request was originated. The idea here is that if one node asks for a certain message, maybe some other nodes are also interested in such message. Therefore, the message is retrieved from other LMs and sent to the originator LM. If nodes that use the station ask for the same message (including the original node that requested the topic), the topic will be available, otherwise the message will be purged from the system.

2.3 How does the system know where coreors are going?

One of the most important parts of the BDS is the information of where coreors are going. Messages are given to coreors for distribution in the system, according to their paths and destinations. Depending on the type

of coreors (*static* or *normal*), there are 3 ways for the system to know where the coreors are going, namely: Static lists, coreors' announcements and prediction systems.

2.3.1 Static lists

If coreors are *static coreors*, they have fixed traveling routes that are known beforehand. The paths these coreors will take can be fed to the system as static lists. Examples of such coreors are buses and trains, which have a fixed and known path between stations.

2.3.2 Coreors' announcements

This method can be divided into 2 sub-methods:

Ability to learn: Devices can have the ability to learn about their own paths. For example, a device carried by a person, can monitor which stations this person uses on a specific day of the week, for example, Monday, for 4 weeks. After this learning period, if the person usually follows the usual path, the device can predict with a high probability where the person will go on Mondays. It can then try to compare with the real movement on that day and adjust accordingly the probability value for that path on that specific day of the week.

User configures destination: Consider a person carrying a mobile device as a node that decides to become a coreor. The person can pre-configure the path she is taking on a specific time. This can be tricky though, because the system needs to trust the person, and there is a possibility that the person lies, misconfigures her path or changes her mind in the middle of the way.

2.3.3 Prediction systems

Similar to the device's ability to learn, but in this case the system keeps track of nodes and their movement. Cong et al. [4] and Boc et al. [5] proved that nodes follow a cyclic movement path. An agenda based system like Otiy [5], which predicts where nodes are located at a given time, can be used to predict coreors' paths based on an agenda recorded by the system.

2.4 Publish and subscribe in BDS

In BDS if a node is interested in a topic, for example an electronic newspaper, it places the query to a LM. This node can be considered as a subscriber (probably one of a group of subscribers for the same topic). The LM looks for that information in publishers, which can be other nodes or content providers. If the topic is found, it is made available for any subsequent subscribers. Popular topics can be automatically fetched from publishers and made available before any request

is received. Also, publishers can choose to push topics to the LMs. A regular publisher, like a newspaper company, can place daily information on LM so it will be available to subscribers of that specific newspaper.

3. Simulation

In order to evaluate the Hikari system we used the MOBINET simulator and built a tool to create the Paris subway system based on a line information file. The Paris subway system has thousands of trains, millions of nodes per day, and hundreds of stations. We consider that, though the movement of trains among stations is deterministic, the movement of nodes (i.e., people with mobile devices) has enough complex patterns. This environment, although complex, shares the same basic concepts with the disconnected environments in developing countries. Although in a different scale, in developing countries people travel between locations, usually using trains and buses that travel between stations. Therefore, we think that understanding the more complex case of the Paris metro system will help us to also model some parts of the system for developing countries.

We have carried out two big sets of simulations:

Algorithms' performance comparison: Since epidemic will achieve the best message delivery rate in such system, we used it as a reference for comparing with the Hikari algorithm. The drawback of epidemic though, is that it creates a lot of replicas in the network and we wanted to overcome this problem with Hikari, since when using portable devices, buffer space represents an important issue. Random algorithms work better for many cases where the information about intermediary nodes that can carry messages is not known.

Effect of arrival rate of coreors in the system: Since one of the main components of the system are the coreors, we analysed the effect of number of coreors in the system. For this we have varied the arrival rate of coreors in the stations. Higher arrival rate (larger Poisson λ) means that more coreors will be in the system.

3.1 Methodology

We have used different parameters for evaluating the two sets of simulations.

Algorithms' performance comparison: We have used the full Paris subway system, Poisson arrival rate in stations, of $\lambda = 4, 25,000$ coreors and a simulation time of 160 min. We used 10 unique messages that appear at time $t=0$ in the system and have an infinite TTL, so they will not expire during this simulation. The messages' requester LMs and originator LMs were generated randomly. In this scenario, in a different set of simulations, we also evaluated the system for unpopular stations, which are stations situated in far extremities of a line.

Effect of arrival rate of coreors in the system:

For this set of simulations we have used 50,000 coreors, 100 unique messages, randomly distributed in the stations and we have used λ values from 2 to 10. The simulation ran for 180 minutes.

3.1.1 Node movement and simulation parameters

The intervals of time considered in the simulations, where chosen based on the number of coreors that we analysed. For example, for the scenario with $\lambda = 4$ and 25,000 coreors, after around 160 minutes, almost all the coreors have arrived in their destinations. After this time there are almost no coreors in the system. The same model was applied to the second set of simulations where 180 was used. For a larger number of coreors and small arrival rate, the analysis interval should be larger.

The simulation results presented in this paper, are the average results of around 50 simulations with the same basic characteristics (e.g. number of coreors and duration), and we changed basically the source and destination stations for the nodes. The movement paths of the nodes and the graph scenario were created before the simulation had began, and saved in lists, to save machine resources and time. The simulations ran for around 50 minutes at a time, on a Pentium Xeon computer with 2 GB of RAM. The process of generating different movement path lists and running all the simulations, took around 3 weeks.

For both simulation sets we have considered the case of coreors being people with mobile devices, and not static coreors like trains. People arrive in stations, chosen randomly, following a Poisson distribution with a mean value of λ . Therefore, λ represents the expected number of coreors (persons) that will arrive in each iteration (the interval between 2 subsequent trains). For example, $\lambda = 4$ means that an average of 4 coreors (persons) will arrive in a station in the time interval between the departure of a train, and the arrival of the next train. The exact number of coreors arriving depends on the iteration but varies around the value of λ . We considered this interval as being 5 minutes, which represents the average delay between trains in the Paris subway system. We have also considered a constant commuting time between lines in a station of 5 minutes. The destination stations have been chosen randomly.

People that arrive in stations board the next train that arrives in that station and leave the train in their destination stations. When they leave the train, they disappear from the system. While people travel from source to destination, the train stops in intermediate stations, where people inside the trains exchange information with the LMs of those stations. People commuting from one line to another, in the same station, do not use trains. They are represented as "walking" to go from one line to the other, much like in the real

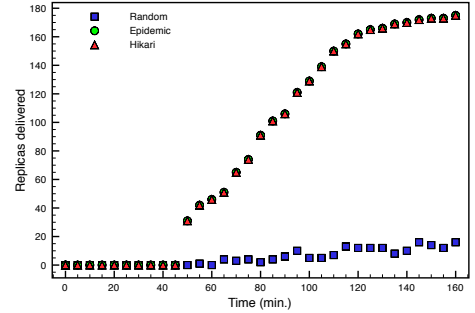


Fig. 3 Replicas delivered vs time

world.

In order to model the Paris subway system, a graph object with 380 nodes representing stations, and 943 edges representing connections between stations (train lines as well as commuting paths, to represent walking people commuting between stations) was created. The graph object has an average degree of 4.96, representing the number of connections that each station has with other stations.

3.2 Message distribution algorithms

The main algorithms for message distribution in the station chosen, were the Epidemic Message Distribution algorithm (ED) and the Random Message Distribution algorithm (RD), since they are the basis for many DTN algorithms. These algorithms refer to the way the messages are distributed in each station to coreors, and not the way the coreors interact between themselves.

Many of the algorithms proposed in the field of DTN, are aimed at node to node exchange of messages. Since our system does not require complex routing, we have used the basic algorithms that are used in DTNs for analysis and for building other algorithms.

ED: Messages are distributed to coreors in an epidemic way. LMs that have a message requested by other LMs, distribute the messages to every coreor in range (both in the station and inside passing trains).

RD: Given an integer constant N , LMs with messages for distribution, select a N number of coreors randomly and distribute messages to them.

Hikari algorithm: Uses coreors' path information for message distribution. For coreors using the station, the LM asks for their paths and if the coreor passes for a station where a message has to be delivered, a copy of the message will be pushed to this coreor for delivery. For the other coreors that go to different directions, no replicas are pushed.

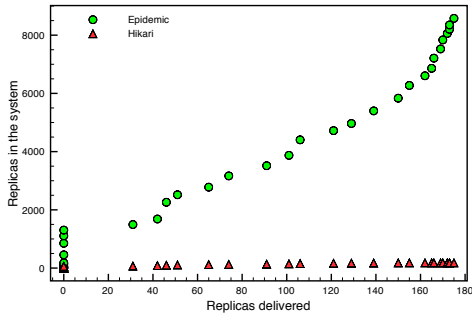


Fig. 4 Replicas in the system vs replicas delivered

4. Evaluation results

4.1 Algorithm evaluation

Figure 3 shows the number of replicas delivered to a requester LM. The total number of messages to be delivered, calculated as the number of coreors that transit between a message creator LM and a requester LM, is 177. The *Delivery rate* is defined as the relation between the number of replicas of messages to be delivered, created in the system, and the number of replicas that were actually delivered. The figure shows that Hikari has the same delivery rate as Epidemic, which is 175, representing 98.9% of the total number of replicas that are possible to deliver. This is an expected result since Hikari delivers the messages to coreors that it knows will go to the requester LM. The result is not 100% because the simulation did not run for long enough so that all coreors that have replicas could deliver them to the LMs. In the case of loss in the system, or of coreors changing their routes in the system, it is expected that epidemic might have a slight better performance than Hikari, since some nodes that change routes may go to requester LMs that they did not expect to find in the path announcement/discovery phase.

Random has a poor delivery rate, since only 16 replicas are delivered, which corresponds to around 9% of total replicas delivered. Therefore, for the relation of replicas *created vs replicas delivered* we did not consider the effect of the Random message delivery. Figure 4 shows this relation. Note that, due to the scale used in the figure, for the Hikari algorithm, it looks like the number of replicas in the system is a constant number close to zero. In fact, the replicas in the system increase from 0 to 177 replicas in the case of Hikari, and in the case of Epidemic, it increases from 0 to 8,577 messages. The figure shows that for 175 replicas delivered, Hikari creates around 177 replicas, while Epidemic creates 8,577 replicas. This is an encouraging result because it shows that Hikari can achieve a result close to Epidemic for message delivery, while creating fewer redundant replicas.

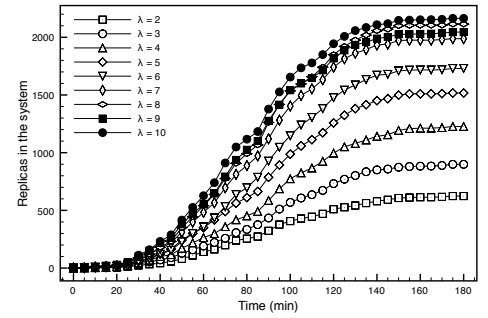


Fig. 5 Replicas in the system for different arrival rates

4.1.1 Algorithm evaluation in unpopular stations

For the evaluation of message distribution in unpopular stations, we have used the same simulation model and node movement, with the difference that messages are originated in stations located in the extremities of different lines, and destined to requester LMs in the other extremity of the same line. When coreors are trains or buses there is no problem, since they always pass by all stations on the line. When they are people, the number of replicas created in the system is: Hikari: 17, Epidemic: 2,145, and Random: 388, but only 10 replicas are actually delivered. This means that when mobile nodes or persons are used as coreors, the performance is sub-optimal and some improvement should be introduced.

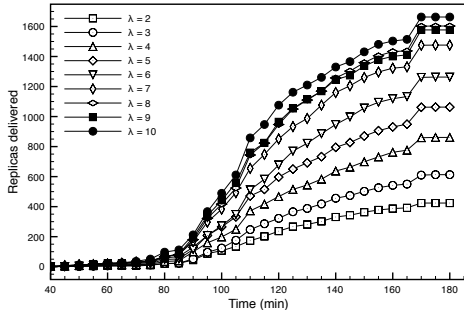
4.2 Effect of node arrival rate in stations

For this set of simulations we used the same basic settings described in subsection 3.1.1, but we varied the arrival rate of coreors in the stations. For each Poisson λ value, we ran a set of simulations corresponding to the time interval of 180 minutes, and extracted the average values of the simulations. In this set of simulations, a certain number of coreors, given by the Poisson λ value, arrive in stations for each iteration. A low λ value means that the number of coreors arriving in the stations will be small, thus implying that the number of coreors in the system will increase slowly. A larger λ value means that more coreors will be arriving in each interval, thus increasing faster the total number of coreors that use the system.

Figure 5 shows the variation of the replicas in the system, caused by different arrival rates. As we can see from the Figure, for higher λ values (more coreors in the system) the number of replicas also increases. Since in Hikari, the number of replicas is proportional to the delivery rate, this means that more messages will be delivered, as Figure 6 shows. Nevertheless, we can see that the number of replicas created in the system is also proportional to the number of coreors that

Table 1 Coreor arrivals vs replicas in the system

time (min)	$\lambda = 2$	$\lambda = 10$
20	5	25
25	11	49
30	23	112
35	33	159

**Fig. 6** Replicas delivered for different arrival rates

go or pass by the message destination station. When there are too many coreors going to the message destination station, too many replicas are created. Having too much redundancy in the system is not good, since it consumes resources in the LMs side and in coreors, who have to carry more messages. In order to address this problem, an idea is to find an optimal number of coreors to carry a certain message, in a way that not too many replicas are created, but enough redundancy is created in order to assure the delivery of messages. We want to have enough redundancy to cope with loss in the system (e.g. coreors changing routes, messages not being delivered, etc.).

We also noticed something interesting from the plots. Figure 5 shows that for small numbers of λ , replicas take longer to be delivered to a large number of nodes, while for larger λ the contrary happens. This result shows that more resources in the LM can be saved if there is a higher arrival rate of coreors in the stations.

To illustrate this, consider that we find that the optimal number of coreors to deliver replicas in order to compensate for losses is 25 coreors. We want to deliver a message to a certain LM. Table 1, which represents the values of Figure 5, shows us that for $\lambda = 2$ it would take the LM 35 min to distribute the replicas to, at least, 25 coreors. In contrast, for $\lambda = 10$ it would take only 20 min to distribute the replicas to 25 coreors. This would save resources in the LM, such as memory, processing power and energy, for 15 minutes for the case of $\lambda = 10$, since the LM can be released after 20 min. Also this would increase the probability of messages being delivered faster as Figure 6 shows.

The main findings of this set of simulations, is that the number of coreors in the system is crucial for the performance and the speed of the system. Therefore,

we plan, as future work, to study incentive mechanisms to have more nodes willing to become coreors. These experiments also rose our interest on how a different number of coreors in different stations affects the system. For example, the question of if having a larger number of λ in popular stations helps with the information dissemination. We plan to analyse these and other issues in future work, in order to make the system more efficient and fast.

5. Related work

Although the DTN area is quite new, many research work has been done in areas like overlays, village communications, routing and content distribution. We briefly present some of the works.

The DTN Research Group [6] has been working on an architecture that implements a store-and-forward message switching by overlaying a layer, the bundle layer, on top of heterogeneous region-specific lower layers [1], [7], [8]. The DTN architecture is a general platform aimed at providing support for different types of networks and protocols, by adding a layer of communication.

Many routing schemes, like MaxProp [9] and Spray&Wait [10], optimize epidemic/flooding style algorithms in a way that less redundancy is created in the network. These algorithms use direct node communication for message dissemination, and do not rely on infrastructure. Node identifiers are important here, since the concept of source and destination is binded with the way forwarding takes place.

Daknet [11] and Kiosknet [12] use a mechanical back-haul to ferry messages between villages and to/from Internet gateways/hubs. In these works, fixed points called "Kiosks", located in villages, are used to exchange data with buses, motorcycles or other transportation means. In order to route packets, Kiosknet, for example, relies on unique addresses and IDs, that have to be managed in databases.

Works like Delay Tolerant Broadcasting [13] and PodNet [14] aim at extending the coverage of networks in an area by providing means of content dissemination using one hop transfers. Identifiers are not assigned to nodes but only to contents. These are receiver-driven systems in where the content is transferred only when a node requests for it. Contents are divided in channels. In PodNet, these channels are identified by unique IDs, and a permanent identifier is assigned to the creator of the channel(s).

Infostations [15], [16] are small, fixed islands of connectivity, with a high bandwidth available of radio transmission aimed at mobile wireless terminals. Infostations are separated from each other, and the main goal is to provide good data and messenger access to mobile nodes. Mobile nodes that travel between Infostations have intermittent connectivity, since there is no

connectivity in areas between Infostations.

Hui et al. [17] and Leguay et al. [18] realized measurements of contact patterns in DTNs. Traces and analysis from these works helped us to understand certain behaviour of nodes in DTN.

Cong et al. [4] and Boc et al. [5] analyse the patterns of node mobility and propose some routing schemes. One interesting insight of these works, was the verification that mobile nodes have a cyclic movement pattern. In other words, people don't have random movement paths, instead they tend to go to the same places using the same way, frequently.

For information dissemination in Hikari, LMs resemble Infostations, although the aim is different in terms of goals. The LMD in Hikari uses a concept very similar to PodNet, being the main difference, the way nodes handle messages. In Hikari, nodes also store queries and messages on behalf of other nodes, and although topics are organized in categories, they do not have unique identifiers associated to them. User are free to create and subscribe to topics as they like.

Regarding routing, since we focus on a specific set of scenarios where some information is known, a general system like the bundle system is not needed. The main difference of the Hikari system with most routing mechanisms, is that while most works use identifiers for forwarding (mostly because of the node-to-node forward schemes), our system decreases the forwarding complexity by using the pub/sub paradigm, which is the novelty of our approach. With only small information and weak infrastructure support, it is possible to decrease substantially the complexity of routing in such DTN scenarios.

6. Conclusions

We have presented the Hikari system, a publish/subscribe system for message distribution in DTNs. The Hikari system is aimed at distributing information created locally or in content servers, to a scenario that is disconnected from a communication infrastructure. The publish and subscribe paradigm is used in order to eliminate the complexity of routing inherent to DTNs. By using such approach there is no need to maintain and manage identifiers for the nodes that participate in the forwarding of information.

Simulations showed that Hikari achieves the same delivery rate as Epidemic message distribution (which is the one that achieves the best delivery rate in DTNs) while creating less redundant replicas. Therefore the efficiency of Hikari is higher, compared to Epidemic schemes in stations. The Random message distribution scheme achieves very poor performance and we found out that using random algorithms in such scenarios results in a sub-optimal performance.

We also found that it is important for the system to have a large number of coreors, arriving frequently

in the system. Having more coreors arriving with a higher frequency, results in a fast message distribution and in more efficiency in the resource management in LMs. Therefore, it is important to create some sort of incentive mechanism to have more nodes turning into coreors.

There are some areas of improvements that were visible in the simulation as well. From the results of simulations mentioned in the subsection 4.1.1, we could notice that the Hikari system does not have an optimal performance when requesting stations are unpopular stations. If there is some information in an unpopular station, it will be difficult to disseminate this information, especially to other small stations. We also verified that while with a small number of coreors, the system will not achieve a good delivery rate, having a large number of coreors creates unwanted redundancy. For future work, we plan to find an optimal number of coreors to distribute messages to in order to create only enough redundancy to compensate loss in the system. Also we plan to use popular stations and node to node communication to improve the performance in unpopular stations.

For different disconnected scenarios, where the LM topology changes frequently (e.g. disaster stricken areas), using only static topology lists is not realistic. Therefore, as future works we plan to also design and evaluate a dynamic topology update method for LMs for such scenarios.

As the simulations showed, the Hikari system achieves good performance and it provides suitable distribution of message service in a disconnected environment. Future works on top of the basic concept can make the system more effective, i.e., providing a better pseudoconnectivity service to various disconnected environments.

References

- [1] M. Demmer, E. Brewer, K. Fall, M. Ho, R. Patra, M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra, "Implementing delay tolerant networking," tech. rep., Intel, 2004.
- [2] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," tech. rep., Duke University, 2002.
- [3] P.T. Eugster, P.A. Felber, R. Guerraoui, and A.M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol.35, no.2, pp.114–131, 2003.
- [4] C. Liu and J. Wu, "Routing in a cyclic mobispace," *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, New York, NY, USA, pp.351–360, ACM, 2008.
- [5] M. Boc, A. Fladenmuller, and M.D. de Amorim, "Otiy: locators tracking nodes," *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, New York, NY, USA, pp.1–12, ACM, 2007.
- [6] D.R. Group, "Dtnrg web site." <http://www.dtnrg.org>.
- [7] F. Warthman, "Delay-tolerant networks (dtms): A tutorial

- v1.1," March 2003.
- [8] K. Fall, "A delay-tolerant network architecture for challenged internets," SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, pp.27–34, ACM, 2003.
 - [9] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, pp.1–11, April 2006.
 - [10] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, New York, NY, USA, pp.252–259, ACM, 2005.
 - [11] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: rethinking connectivity in developing nations," *Computer*, vol.37, no.1, pp.78–83, Jan. 2004.
 - [12] S. Guo, M.H. Falaki, E.A. Oliver, S.U. Rahman, A. Seth, M.A. Zaharia, and S. Keshav, "Very low-cost internet access using kiosknet," SIGCOMM Comput. Commun. Rev., vol.37, no.5, pp.95–100, 2007.
 - [13] G. Karlsson, V. Lenders, and M. May, "Delay-tolerant broadcasting," CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks, New York, NY, USA, pp.197–204, ACM, 2006.
 - [14] Wireless opportunistic podcasting: implementation and design tradeoffs, New York, NY, USA, ACM, 2007.
 - [15] D. Goodman, J. Borras, N. Mandayam, and R. Yates, "Infostations: a new system model for data and messaging services," Vehicular Technology Conference, 1997 IEEE 47th, vol.2, pp.969–973 vol.2, May 1997.
 - [16] R. Frenkiel, B. Badrinath, J. Borres, and R. Yates, "The infostations challenge: balancing cost and ubiquity in delivering wireless data," *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol.7, no.2, pp.66–71, Apr 2000.
 - [17] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, New York, NY, USA, pp.244–251, ACM, 2005.
 - [18] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic content distribution in an urban setting," CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks, New York, NY, USA, pp.205–212, ACM, 2006.



Sergio CARRILHO Was born in Cabo Delgado, Mozambique on September 09, 1979. He received his Licentiate diploma in Electronic Engineering from the University Eduardo Mondlane in Mozambique in 2003. He worked on Industrial Automation for BHPBilliton (MOZAL) from 2003 to 2005 in Mozambique. In 2005 he joined the Government Network in Mozambique, as a Network Security Specialist and worked as

the technical responsible for the network extension phase. He

was a member of the board of the Electrotechnic College of the Mozambican Order of Engineers and is member of the WIDE project. He is presently with the Graduate School of Information Science and Technology, University of Tokyo, where he finished the Masters degree. His main interest is the area of Delay Tolerant Networking, wireless networks and alternative network infrastructure for developing regions.



Hiroshi ESAKI Received the B.E. and M.E. degrees from Kyushu University, Fukuoka, Japan, in 1985 and 1987, respectively. And, he received Ph.D from University of Tokyo, Japan, in 1998. In 1987, he joined Research and Development Center, Toshiba Corporation, where he engaged in the research of ATM systems. From 1990 to 1991, he has been at Applied Research Laboratory of Bellcore Inc., New Jersey (USA), as a residen-

tial researcher. From 1994 to 1996, he has been at CTR (Center for Telecommunication Research) of Columbia University in New York (USA). During his staying at Columbia University, he has proposed the CSR architecture, that is one of the origin of MPLS (Multi-Protocol Label Switching), to the IETF and to the ATM Forum. From 1996 to 1998, he has conducted the CSR project in Toshiba, as a chief architect. From 1998, he has served as a professor at the University of Tokyo, and as a board member of WIDE Project (www.wide.ad.jp). From 1997, he has involved in the IPv6 research and development at the WIDE project. Currently, he is executive director of IPv6 promotion council, vice president of JPNIC (Japan Network Information Center), IPv6 Forum Fellow, board member of WIDE Project and Board of Trustee of ISOC (Internet Society).