Universidade do Minho
Escola de Engenharia
Departamento de Informática

Nuno Filipe Mendes

Delay Tolerant Networks with Traffic Differentiation Capabilities

December 2016

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Nuno Filipe Mendes

Delay Tolerant Networks with Traffic Differentiation Capabilities

Master dissertation
Master Degree in Engineering of Computer Networks and Telematic Services

Dissertation supervised by
Professor Doutor Pedro Nuno Sousa
Professor Doutor António Duarte Costa

December 2016

ACKNOWLEDGEMENTS

This thesis was accomplished through hard-work, dedication and perseveration. It would not be possible to successfully conclude without the help and advice given through the entire process.

I want to thank first, the guidance, support and academic help to my supervisors Professor Doutor Pedro Nuno Sousa and Professor Doutor António Duarte Costa.

Additionally I want to thank my current company Nabia Solutions, lead by Eng. Rui Lopes and Dr. Simone Pinto, and my previous company Spark Agency, lead by Miguel Gonçalves and Tânia Delalande, for all the patient, advice and time flexibility, that allowed me to reach this stage in my Masters Degree and conclude this thesis.

Thirdly, I want to thank Fábio Cerqueira for our teamwork and focus in the masters degree journey, in order to accomplish the evaluations and projects, and Ricardo Coelho for the help in design and graphic aspects of this thesis.

I would like to thank my partner, Paula, for all the support, advice and help she gave, which without, this thesis would not be possible.

Lastly I would like to thank my parents, for the guidance and focus in all my academic path.

ABSTRACT

In the last few decades, an increasing growth of Internet usage was witnessed worldwide. However, infrastructures do not always allow the existence of Internet connectivity everywhere. Therefore, to address this issue, the concept of Delay Tolerant Networks (DTNs) was developed. DTNs purpose is to provide a different level of intermittent connectivity, dissimulating connection problems that arise in complex connectivity scenarios. Examples of such scenarios are, for instance, cities, where cars exchange information about their location; in underdeveloped countries, where Internet is inexistent; in freeways, where is not viable to provide infrastructures for a continuous connectivity, but cars, tolls, and services need to be aware of each other. Thus, DTNs constitute a possible solution for all the aforementioned communication environments.

However, DTNs still faces some obstacles in terms of delivering a service with quality as it lacks specific mechanisms, such as traffic differentiation. Traffic differentiation is essential to provide different levels of service quality regarding delivering of messages. Current proposals to improve service delivery through traffic differentiation on DTNs are still under development or lack the proper testing and simulation. The main focus of these proposals is on buffer management mechanisms at each DTN node, instead of message prioritisation mechanisms. Message prioritisation allows some messages to be prioritised over others, improving the delivery rate and, therefore, increasing the probability of a message being correctly delivered.

The present thesis implements traffic differentiation in DTNs based on prioritisation strategies, assuming a clear alternative to other buffer management proposals and message prioritisation. Using The One simulation tool, three popular DTNs routing protocols (Epidemic, Spray & Wait, and PRoPHET) are adapted to comply with traffic differentiation. The DTNs traffic prioritisation objective is achieved by designing, implementing and testing four distinct algorithms that classify and order messages according to their priority levels. These algorithms are based and extend some traditional traffic differentiation mechanisms, namely the well-known Priority Queuing and Weighted Round Robin strategies.

Results from the simulation tests corroborate that the delivery rate of the messages is affected according to their priorities. Specifically, the simulation shows an increase in the delivery rate of high priority messages, with low impact on the total number of messages delivered, comparatively to the same scenario without differentiation capabilities. To conclude, DTNs can effectively benefit from traffic differentiation based on message prioritisation techniques, being a promising approach to improve service quality levels in such scenarios.

RESUMO

Nas últimas décadas assistiu-se a um aumento crescente no uso da internet. Contudo, as infra-estruturas nem sempre permitem uma ligação à internet. Assim, para enfrentar este desafio, o conceito de Delay Tolerant Networks (DTN) foi desenvolvido. O objetivo das DTN é providenciar diferentes níveis de ligação intermitente, atenuando os problemas de ligação que surgem em cenários de conectividade complexa. Exemplos de tais cenários incluem, cidades, onde carros trocam informação da sua localização; países em vias de desenvolvimento, onde a internet é inexistente; em auto-estradas, onde não é viável conceber infra-estruturas que permitam uma conectividade permanente, mas onde carros, portagens e serviços necessitam de comunicar. Deste modo, as DTNs constituem uma solução possível para os ambientes indicados.

Contudo, as DTNs ainda enfrentam alguns obstáculos na prestação de um serviço de qualidade, visto faltarem mecanismos específicos, como a diferenciação de tráfego. A diferenciação de tráfego é essencial para oferecer diferentes níveis de serviço de qualidade em termos de entrega de mensagens. As abordagens existentes para diferenciação de trafego em DTNs ainda estão em fase de desenvolvimento. Estas focam-se principalmente nos mecanismos de gestão do buffer a cada nodo da DTN, em vez de ao nível de mecanismo de priorização das mensagens. A priorização de mensagens permite que algumas recebam prioridade em detrimento de outras, melhorando a taxa de entrega, aumentando a probabilidade desta ser entregue corretamente.

Esta tese implementa diferenciação de trafego em DTNs baseando-se em estratégias de priorização, assumindo-se como uma alternativa a outras abordagens de gestão de buffer e priorização de mensagens. Usando a ferramenta de simulação “The One”, foram adaptados três protocolos de routing DTN (Epidemic, Spray & Wait, and PRoPHET) de modo a obedecerem à diferenciação de tráfego. Este objetivo é alcançado pelo desenho, implementação e experimentação de quatro algoritmos que classificam as mensagens de acordo com o seu nível de prioridade, baseando-se em mecanismos tradicionais de diferenciação de trafego, i.e. as estratégias de Priority Queuing e Weighted Round Robin.

Os resultados demonstram que a taxa de entrega de mensagens é influenciada de acordo com as prioridades. Nomeadamente, há um aumento na taxa de entrega de mensagens com prioridade alta, com pouco impacto no número total de mensagens entregues, comparativamente com o mesmo cenário sem mecanismos de diferenciação. Em suma, as DTN podem beneficiar da diferenciação de tráfego baseado em técnicas de priorização de mensagens, representando uma abordagem à melhoria da qualidade de serviço bastante promissora.

CONTENTS

LIST OF FIGURES

List of Figures

LIST OF TABLES

ACRONYMS

A

AF Assured Forwarding.

B

BE Best Effort.

D

.

DTN Delay Tolerant Network.

F

FIFO First In First Out.

H

HP High Priority.

I

IETF Internet Engineering Task Force.

L

LIFO Last In First Out.

LP Low Priority.

M

MERSTEL Engineering of Computer Networks and Telematic Services.

.

MP Medium Priority.

Acronyms

P

PQ Priority Queueing.

PROPHET Probabilistic Routing Protocol.

Q

QOS Quality of Service.

R

RTTL Return Time To Live.

S

SNC Sami Network Connectivity.

SW Spray and Wait.

T

TTL Time to Live.

U

UM Universidade do Minho.

V

VDTN Vehicle Delay Tolerant Network.

W

WFQ Weighted Fair Queueing.

.

1

INTRODUCTION

1.1 CONTEXT AND MOTIVATION

In Computer Networks, *Delay Tolerant Network (DTN)* (Fall, 2003) (Jones et al., 2007) can manage extremely complex scenarios regarding intermittent end-to-end connectivity. This complexity results in highly unstable routes due to the lack of a permanent connection between the different nodes. Examples of this type of networks can be found in vehicular, military, wild-life monitoring networks, and in under-developed countries where Internet is non existent, among others. In scenarios employing DTN, the nodes move in a variable pattern, with variable conditions, and can be sparsely allocated. Due to this situation, it is necessary to adopt strategies that can forward, and disseminate, traffic in the most effective possible way, without neglecting the context of delay-tolerance.

The present research project studies the viability of implementing traffic differentiation approaches. (Tschofenig et al., 2008), to provide a different traffic treatment between the nodes. Differentiation mechanisms can be applied to the already existing forwarding protocol solutions within DTN. The focus is on Epidemic (Vahdat and Becker, 2000), *Spray and Wait (SW)* (Spyropoulos et al., 2005), and *Probabilistic Routing Protocol (PRoPHET)* (Lindgren et al., 2003) protocols. With these protocols in mind, it is possible to achieve a wide range of different combinations of plausible traffic differentiation approaches. Conversely, differentiation required further development of some of the existing protocols. The research focuses mainly on traffic prioritisation aspects and studies the impact of such prioritisation approaches in DTN.

1.2 OBJECTIVES

The main purpose of the present project is to design and evaluate traffic differentiation mechanisms in DTN in order to obtain better performance in traffic behaviour, regarding the message delivery rate. To this end, the following objectives were considered:

- Literature review of the traffic differentiation and scheduling approaches, not only in modern IP networks, but also in DTNs;

- Full understanding of The One simulator tool;
- Viability analysis regarding the introduction of differentiation strategies and mechanisms to DTNs;
- Design and implementation of differentiation mechanisms in DTNs;
- Integration of the mechanisms in the already existing protocols;
- Experimentation and result analysis of the proposed traffic differentiation mechanisms.

The One (Keränen et al., 2009) simulation tool is used to obtain results regarding performance and network behaviour. These results are analysed, discussed and compared between the different routing strategies, that were enhanced with proposed differentiation mechanisms in the context of this thesis.

1.3 CONTRIBUTIONS

The main contribution of this work is the implementation of several traffic differentiation mechanisms in DTNs, in order to provide traffic prioritisation on this type of networks. The importance of traffic differentiation in non-delay tolerant networks provides us with enough evidence to support such mechanisms in DTNs (Litjens and Hendriks, 2014). For this reason, several optimisations and upgrades to some of the existing routing schemes were developed, providing enough tools and mechanisms to establish traffic differentiation in DTNs.

1.4 THESIS STRUCTURE

This dissertation has four main components, namely the literature review, the proposal and development of traffic differentiation in DTNs, the tests, and the conclusion. These components are divided in the following chapters:

- Chapter 2 - State of the Art
- Chapter 3 - Proposal for Traffic Differentiation on Delay Tolerant Networks
- Chapter 4 - Implementation
- Chapter 5 - Experiments and Results Analysis
- Chapter 6 - Conclusion

The literature review component implies a review of relevant literature in the topic to better comprehend the already existing solutions. This component is based on the search and integration of relevant scientific papers in the domain of traffic differentiation and DTNs. The state of the art is divided in three distinct subsections, delay tolerant networks, traffic differentiation and the combination of the two in traffic differentiation in delay tolerant networks. Each section describes an overview of the current state of the art, with the proper reference explanation and description. The third and fourth chapter present the base of the research and development process regarding the possible improvements that can be made when applying traffic Differentiation to delay Tolerant Networks. Those chapters describe the approaches, software usage, software implementations and simulation scenario, and structure. The fifth chapter approaches the experiments and results of all the simulations ran, where the focus is on the relevant variables and comparison between the several scenarios and routing strategies results. In this chapter the implementation of the proposed mechanism is evaluated. The goal is to verify its quality and performance behaviour, while comparing it to other mechanisms when no traffic differentiation capabilities are implemented in DTNs. Finally, in the conclusion, the sixth chapter, focus on discussion and conclusions about traffic differentiation in delay tolerant networks. These reflect the methods that were implemented, and their impact in traffic differentiation regarding DTNs.

2

DELAY TOLERANT NETWORKS AND TRAFFIC DIFFERENTIATION

This chapter presents the current state of the art on three topics: delay tolerant networks, traffic differentiation and traffic differentiation in delay tolerant networks. In the first section, regarding delay tolerant networks, a brief introduction is made to how they work and what are the possible routing approaches. The second section reviews traffic differentiation in non delay tolerant networks, describing the methods, algorithms and related approaches. Lastly, the third section, focus on the scientific research about implementing traffic differentiation in delay tolerant networks, overviewing the current results and achievements.

2.1 DELAY TOLERANT NETWORKS

Since their first appearance, Delay Tolerant Networks (DTNs) (Fall, 2003) (Cerf et al., 2007) (Jones et al., 2007) have become a viable option, in high delay environments, when compared to the commonly used non delay tolerant network connections.

In Figure 1, it is possible to see different approaches to establish a network connection in remote places. Some of these options are not always available, and perhaps the only possible way of communication is through a bike messenger, that can transport data in some manner, like a flash drive. This approach introduces low predictability of delivering a message and increases the message delivery delay. Most DTN scenarios accommodate the usage of city buses, cars, and pedestrians. The city buses act as mobile routers, which have store and forward capability (Tschofenig et al., 2008). Within such scenarios, the messages are delivered by being forward to the nodes that are closer and in-route of the bus. These message bundles are routed, through the bus network, based on scheduling tables and distance to the destination. However their contact time between the nodes is unknown and the message storage to hold on messages is limited. The cars act as nodes with faster capability of delivery, that follow randomised paths, adding a new way of delivering a message faster and with more storage. The last component are the pedestrians, which are characterised by having a low storage limit in their devices and a limited range of movement. These scenarios are common in underdeveloped countries. As an example, villages can use buses and other means of transportation to also transport network data and allow them to be in touch with the rest of the world. For the above reasons, several routing protocols were developed and implemented to augment the potential and effectiveness of DTNs (Jain et al., 2004).

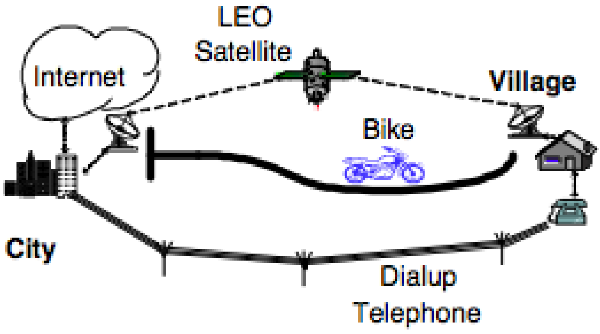


Figure 1.: Example of remote village (Jain et al., 2004)

2.1.1 *Architecture, Distribution and Bundle Protocol*

An alternative approach in message delivery with low delays, when comparing with other approaches such as using a satellite or dialup, which can can be unavailable at certain locations, is through the road, using vehicles, motorcycles or bicycles. This is represented in Figure 1 by a bike. In DTNs there are several routing approaches to improve message delivery, reduce latency or even reduce the used energy. Additionally, DTN can be considered as a layer on top of existing networks. This additional layer is often referred to as bundle, which applications use to communicate, become compatible and work with delay-tolerant scenarios. Figure 2 shows how the bundle protocol takes part, in the network stack. This overlaying protocol allows DTN to implement a store and forward message switching system. Bundles are just message agglomerators that contain application messages to be saved and distributed between the nodes (Scott and Burleigh, 2007).

2.1.2 *Routing in Delay Tolerant Networks*

Throughout the years, several routing protocols were introduced in DTNs. The most basic routing protocol is the Epidemic (Vahdat and Becker, 2000). Its implementation is quite simple. It behaves by sending everything to every contact. Spray and Wait (SW), although similar to Epidemic, limits the number of copies that are passed on, reducing the network message overflow and improving the delivery rate. PRoPHET, on the other hand, introduces a probabilistic table and only delivers messages to nodes that have a good probability of meeting the message final destination.

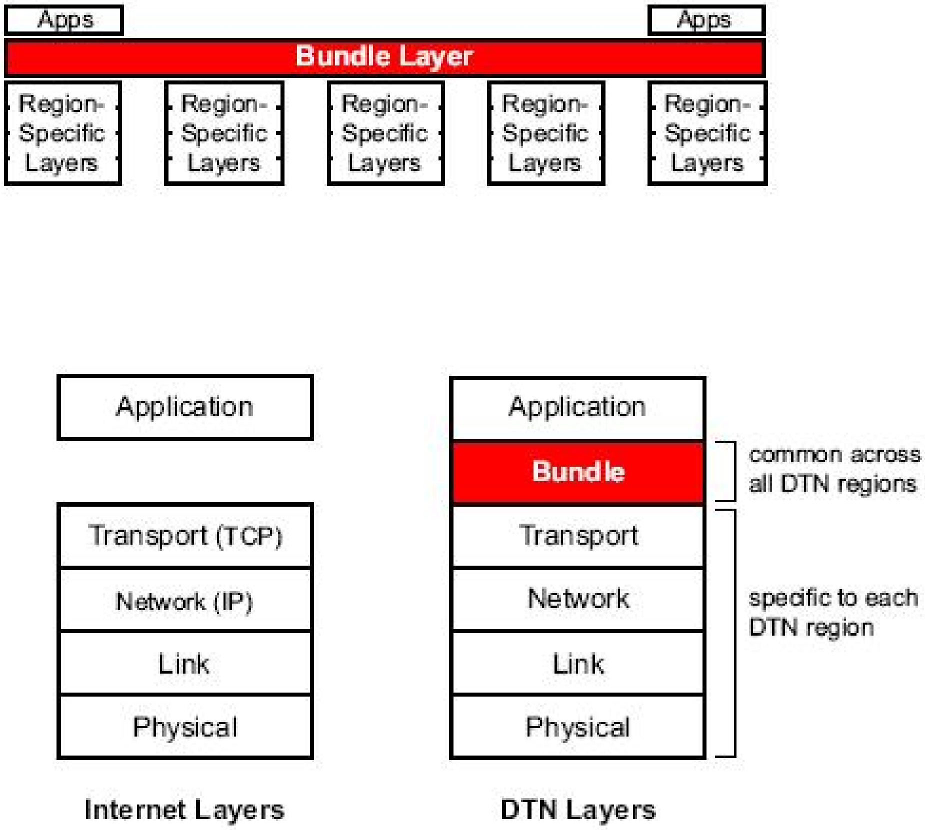


Figure 2.: Example of bundle protocol (Missouri-Rolla, 2006)

Direct Delivery

In Direct Delivery, the nodes behaves exactly like the routing name says, they only deliver the messages to another node if that is the final destination node, the recipient of that message (Keränen et al., 2009) . This means that for each contact that is established between the nodes they just verify the messages stored on the buffer. That verification can prove useless, if the node is indeed the recipient of any of the messages present on that node buffer.

First Contact

First Contact routing distributes messages based on an encounter way. They deliver the message to the first node they encounter (Keränen et al., 2009), and that node can only distribute to its destination node, allowing an extra jump when comparing to Direct Delivery. This basic approach avoids cluttering the network with multiple messages, but also limits the probability of message delivery due to the low message count, that is distributed.

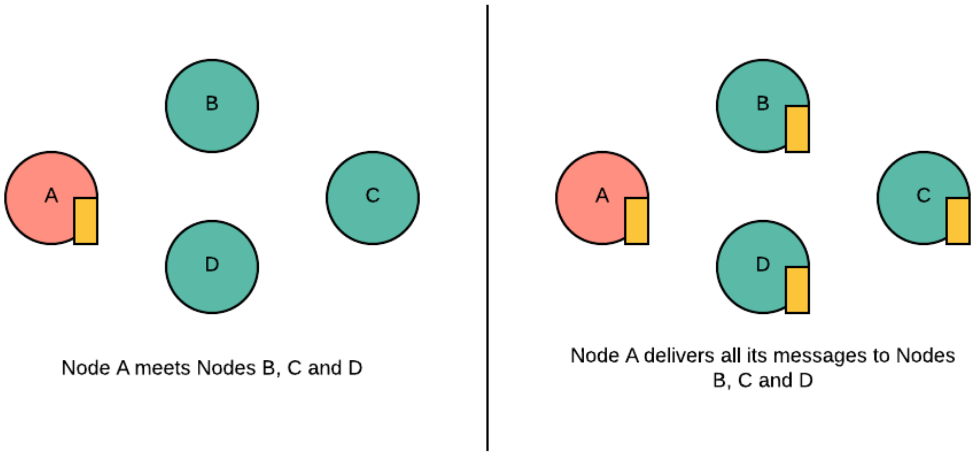


Figure 3.: Epidemic routing protocol (Vahdat and Becker, 2000)

Epidemic Routing

Epidemic routing is similar to a real epidemic outbreak, where a disease propagates from a sick person to the nearest personal, contaminating them with the disease if not contained. Like a real epidemic, nodes deliver every message in their storage buffer, to every other node that they encounter along their path (Vahdat and Becker, 2000). Figure 3 exemplifies a meeting between node A and nodes B, C and D, where node A deliveries its messages to every node that encounters, so B, C and D also get the same messages that node A has.

Spray & Wait

SW, a protocol divided in two stages, is able to replicate part of the Epidemic routing protocol scheme through spraying the messages to a predefined number of nodes, a definition named n-copies, and then waiting for its delivery (Spyropoulos et al., 2005). The spraying done by SW has two variants, the normal mode and the binary mode (Spyropoulos et al., 2005) (Keränen et al., 2009). The normal mode sprays its n-copies horizontally. This means that only the original node can make multiple sprays. For each message that is delivered to a node, it is subtracted from the n-copies variable. Nodes that have obtained the message after can only deliver that message to its rightful recipient. Figure 4 represents the normal mode behaviour, where only the node A is allowed to distribute additional copies, and the remaining nodes are only allowed to make one more copy. This copy can only be delivered to the recipient of the message.

The binary mode sprays its n-copies horizontally and vertically, with certain conditions. The horizontal spray behaves exactly as the normal mode described before in Figure 4. The vertical spray allows the intermediary nodes to also distribute copies of the same message. The

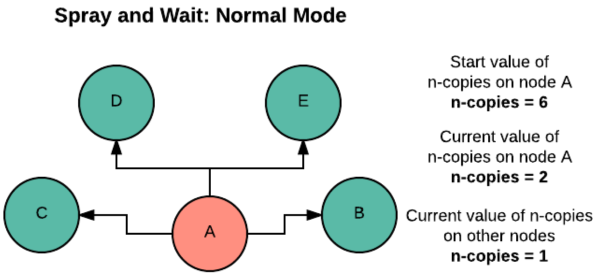


Figure 4.: Spray & Wait routing protocol behaviour on normal mode (Spyropoulos et al., 2005)

formula for calculating the number of copies, in binary mode vertical spray, is represented by Equation 1. It is important to note that the formula rounds up all the decimal cases.

$$nrcopies_{to} = \frac{nrcopies_{from}}{2} \tag{1}$$

This binary mode allows for more copies of a certain message to be distributed through the network, without completely overflowing the network (Spyropoulos et al., 2005). This process is also represented in Figure 5, where other nodes (e.g., node B) can be seen, with 3 copies, following the Equation 1, where half of the n-copies are passed to the new node. This also means that if node B decides to pass the message to a new node (e.g., node F), the new node would have to run the formula again and n-copies would be equal to 2, just like Equation 2 exemplifies.

$$\frac{3}{2} = 1.5 \cong 2 \tag{2}$$

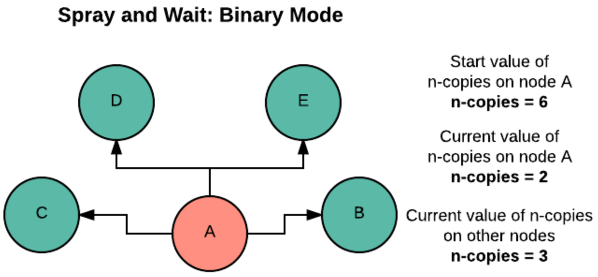


Figure 5.: Spray & Wait routing protocol behaviour on binary mode (Spyropoulos et al., 2005)

In this protocol, the nodes keep delivering and copying the message until they only have one copy left to distribute. When the n-copies variable reaches the value 1, it means that the message can only be delivered to its final destination. This approach has good results when comparing to Epidemic routing, as it achieves small delays while spending a small amount of energy. The approach used by this protocol can be observed in Figure 6, where increasing the number of allowed copies, improves the delivery delay of messages. This increase can help achieve a better application of the protocol. Authors obtained the optimal value through testing and calculations in a single copy routing intermittently connected mobile networks scenario (Spyropoulos et al., 2004).

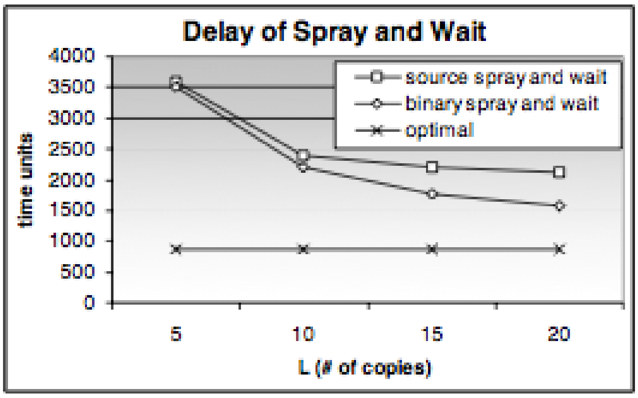
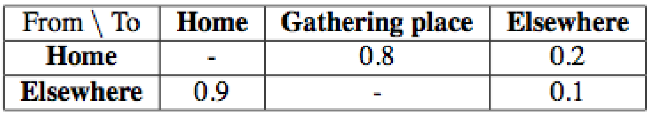


Figure 6.: Spray & Wait routing protocol (Spyropoulos et al., 2005)

PRoPHET

The PRoPHET routing protocol bases its functionality in the history of encounters between the nodes. Through a specific algorithm, PRoPHET attempts to explore the real probability of these encounters (Lindgren et al., 2003).



From \ To	Home	Gathering place	Elsewhere
Home	-	0.8	0.2
Elsewhere	0.9	-	0.1

Figure 7.: PRoPHET routing protocol probability table (Lindgren et al., 2003)

This protocol maintains a database with the delivery odds to each known destination, and only delivers a replica of the intended message to a certain node if it has a good chance of meeting its goal. Like the SW routing protocol, PRoPHET outperforms Epidemic routing in terms of delivery rate and delay and also avoids compromising the networks resources (Cao and Sun, 2012). In (Lindgren et al., 2003), the scenario area was divided into twelve subareas and a gathering place. Each node had a home area. Figure 7 is a clear example of a probability table used in the PRoPHET protocol, in which it is possible to see different probabilities for different destinations. The probability is calculated according to the possibility of the node visiting that place. In Figure 7 shows that a node away from its home area, is highly likely to visit it.

This shows also shows that the number of distributed messages is not directly related with the number of delivery messages, otherwise Epidemic routing would had an higher delivery rate than SW or PRoPHET. This could be related to the fact that network resources are not so crowded in these routing alternatives and the short contact times between nodes (Cao and Sun, 2012) (Spyropoulos et al., 2005).

Max Prop

MaxProp behaves similarly to Epidemic routing, but specifies when a copy of a message is delivered (Burgess et al., 2006) (Keränen et al., 2009). These very small acknowledgements are flooded into the network, informing nodes that a certain message is already delivered and so to delete it. This means that, although it tries to send to every node that encounters, it also means that once the message is delivered, it no longer floods the network with the already delivered message. Comparing to Epidemic, this reduces the network message overflow, allowing other messages to be easily delivered (Burgess et al., 2006). The router structures its list of messages to be delivered in each encounter, calculating the hop counts of that message, and, similarly to PRoPHET, through delivery probabilities. These probabilities are based on previous node encounters.

2.1.3 *Real life Application example*

There are real case scenarios where DTN were deployed. In the region of Laponia, in northern Sweden, a project named *Sami Network Connectivity (SNC)* (Lindgren and Doria, 2007), allowed Sami population to be connected to the Internet. The Figure 8 displays a map of the target geographic area. The chosen routing approach was PRoPHET, previously addressed in this chapter. This SNC DTN implementation was actually tested against three different applications, the e-mail, web browsing, and instant messaging (Lindgren and Doria, 2007).

Although some services got less instant, such as the instant message application, which becomes more of a normal message application, they were able to make them work properly, considering the DTN inherent limitations. The problems arised mainly due to power supply, which the author suggests solar panels for the summer months and an interesting mechanical energy conversion approach to the other months of the year (Lindgren and Doria, 2007). These type of research allows the possibility to overcome the challenges related to DTN, creating new possibilities to remote areas, regarding Internet connection.

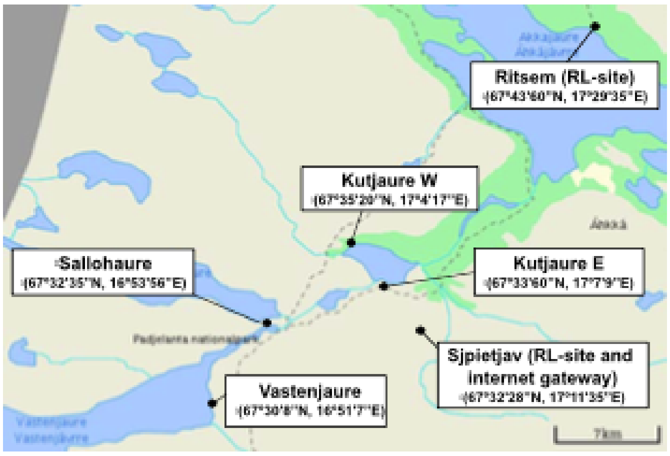


Figure 8.: Map of test area in Laponia, Sweden (Lindgren and Doria, 2007)

2.2 TRAFFIC DIFFERENTIATION

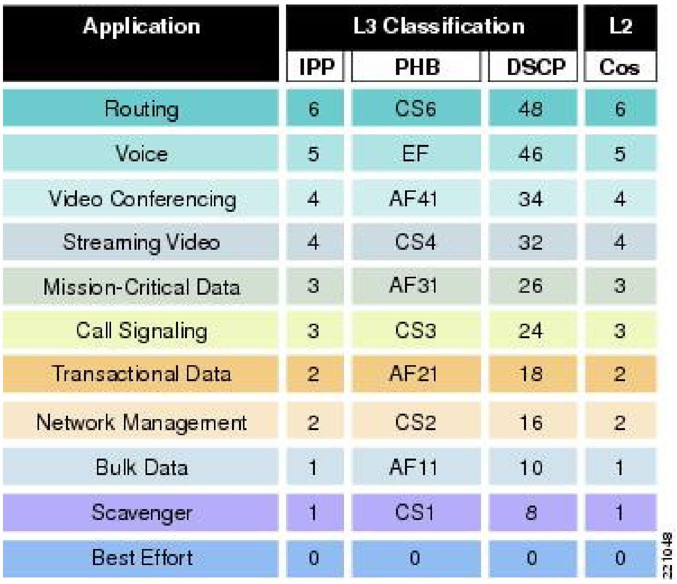
As networks become increasingly more complex and carry different types of data, with different delivery requirements, including Voice over IP, video, raw data, web pages, and e-mail, the need to organise and meet delivery demand of these services has increased. To fulfil this need, the topic of traffic differentiation has become progressively more attractive and several related mechanisms were implemented in real network scenarios, to meet the delivery demands of these services. In the following subsections some traffic differentiation related topics are overviewed.

2.2.1 *Traffic Classification*

One of the first stops in traffic differentiation happens in December of 1998, with the *Internet Engineering Task Force (IETF)* publishing a *Request for Comment (RFC)* (Nichols et al., 1998) that defined *Differentiated Service Fields (DS Field)* for both IP headers (v4 and v6), replacing the obsolete *Type of Service Field (TOS)*.

These fields allow for traffic to be classified, so that subsequent traffic control mechanisms can be applied correctly.

In Figure 9, the application stands for the service classification. The voice application has an higher priority when compared to best effort classified applications, at the end of the list, which means that voice services will be prioritised over best effort. It is also possible to see in the Per Hop Behaviour (PHB) column, which shows that voice applications are Expedite Forwarding (EF). Following the Differentiated Services Code Point (DSCP), this assures that any traffic class with this PHB classificaiton is not queued and is given the highest priority.



Application	L3 Classification			L2
	IPP	PHB	DSCP	Cos
Routing	6	CS6	48	6
Voice	5	EF	46	5
Video Conferencing	4	AF41	34	4
Streaming Video	4	CS4	32	4
Mission-Critical Data	3	AF31	26	3
Call Signaling	3	CS3	24	3
Transactional Data	2	AF21	18	2
Network Management	2	CS2	16	2
Bulk Data	1	AF11	10	1
Scavenger	1	CS1	8	1
Best Effort	0	0	0	0

Figure 9.: Classification of Application Traffic to DSCP Marking (Cisco, 2014)

2.2.2 *Queueing and Scheduling Mechanisms*

The differentiated service fields allow the traffic to be prioritised according to a pre-established table. By looking at Figure 9, voice related information is considered the one of the highest priority information data to be transmitted between the routers, followed by other sensitive-data traffic, such as multimedia-streaming, all the way to the best-effort traffic. Even though these classifications are a standard in the industry, their configuration and attribution is customisable, and can be different from implementation to implementation. Several approaches, regarding *Quality of Service (QoS)* and differentiated traffic, were proposed, in hope of supporting a fair and equal bandwidth distribution (Nichols et al., 1998). Queue and Scheduling mechanism provides complementary approaches and strategies to traffic differentiation. On these approaches, messages are sorted into a queue, by following a set of pre-established rules. One of these approaches is *Priority Queueing (PQ)* (Semeria, 2001). The following subsections are just an example of the possible approaches and strategies regarding the Queue and Scheduling mechanism.

Queuing: First In First Out

The most basic queuing strategy is *First In First Out (FIFO)*, which means the first message to get in the queue, is the first message to get out of the queue. This message transmission follows the queue order, which is exactly the same order as they come into the queue.

Figure 10 represents this behaviour, where message labelled **1** is at the head of the queue, which means it was the first message to be placed in queue and it will be first message to

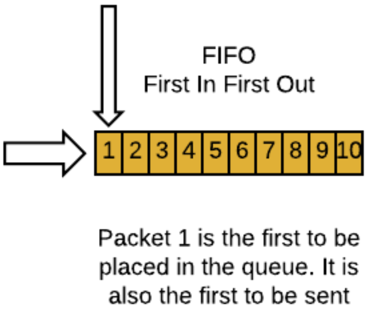


Figure 10.: FIFO - First In First Out Example

be picked up for transmission. Following the same pattern, message **2** is the next one to be transmitted.

Scheduling: Priority Queueing

Priority Queueing (PQ) (Semeria, 2001) scheme structures the messages according to their classified priority. Each classified message is placed in a specific queue corresponding to its classification. These queues are prioritised according to the requirements imposed by the network in which they are implemented. Within each priority queue, the FIFO scheme is used for organising the messages.

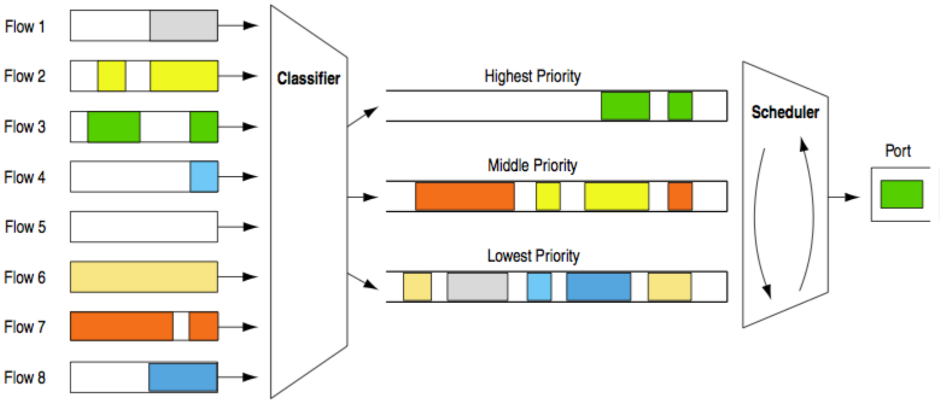


Figure 11.: Priority Queueing scheme (Semeria, 2001)

In Figure 11 several flows are being passed through a classifier, which, according to pre-established contracts or rules, classifies the traffic to the different configured priorities, in this case high, medium or low. Each time the scheduler needs a new message to be transmitted, it picks up from the highest to the lowest priority queue and transmits. One of the important characteristics of PQ is its low computational requirements, when compared to other more complex queuing approaches. However, this application can lead to starvation of lower and middle queues, if a default PQ is implemented. Several studies have shown the advantage of using PQ in traffic differentiation, even when messages are classified without any association with the types of traffic they contain (Semeria, 2001). When proposed to non delay tolerant networks, differentiating the traffic between delay-sensitive (e.g., voice and video) and best-effort, the typical standard data, proved to be effective. In this case, delay-sensitive data had delivery rate improved when compared to the best-effort approach. Priority queueing can also be applied to wireless sensor networks, to emphasise real-time traffic. The results demonstrated the capabilities of low end-to-end delay, with a high-energy efficiency.

Queueing and Scheduling: Weighted Fair Queueing with Weighted Round Robin Ordering

Weighted Fair Queueing (WFQ) (Demers et al., 1989) (Georges et al., 2005) is a data message scheduling algorithm that structures messages through specifically set weights. This algorithm was proposed as an alternative to FIFO, in order to provide a fair allocation of bandwidth and isolation from high bandwidth consumption sources (Demers et al., 1989). The main difference when comparing to PQ, is that it allows the bandwidth to be more fairly distributed on the network. This avoids an unmanaged consumption of the network resources. An example of WFQ is visible in Figure 12 where messages are divided into queues, and then classified according to the their finish time. In the example, the finishing time is the weight that each message has. WFQ picks each message according to their finishing time first, rotating through all the existing queues, allowing messages on each queue to be selected to be transmitted.

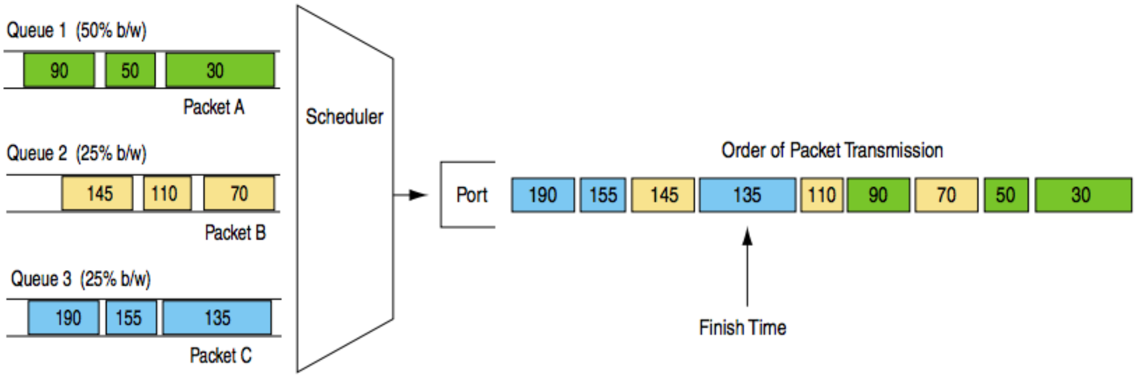


Figure 12.: Weighted Fair Queueing scheme (Semeria, 2001)

2.2.3 *Traffic Policing*

Traffic Policing is a mechanism that monitors traffic flow on a network, checking its compliance with a traffic contract (Blake et al., 1998). The traffic, if does not follow the traffic contract, can be marked for drop or remarked in terms of priority or traffic type. Traffic Policing is usually followed by Traffic Shaping, to control traffic so that it complies with the established contract. Figure 13 shows how traffic policing is applied in a network, monitoring network traffic for any non-compliant behaviour and taking action if needed, before the traffic enters or leaves the network.

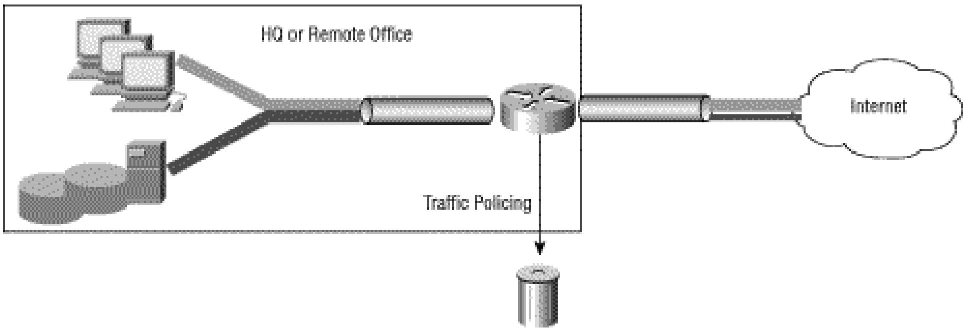


Figure 13.: Traffic Policing example (Networks, 2014)

2.2.4 *Traffic Shaping*

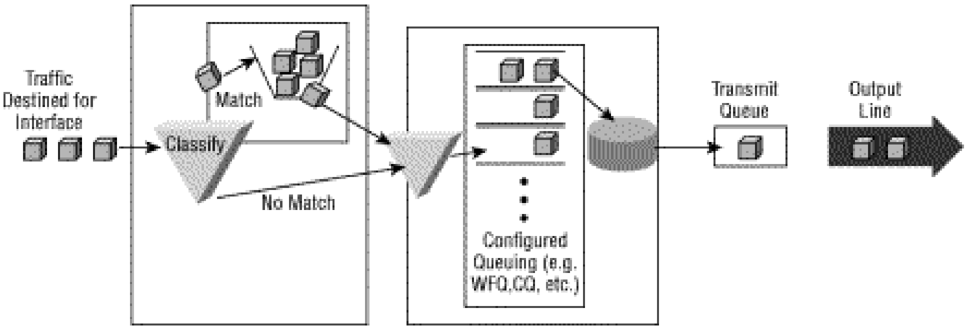


Figure 14.: Traffic Shaping example (Networks, 2014)

Another mechanism to control traffic is Traffic Shaping (Blake et al., 1998) (Recommendation, 1992). It alters the characteristics of a traffic stream to meet a certain traffic predefined requirement. This approach is expected to achieve better network efficiency and ensure conformity with the network QoS objectives. The basic function of Traffic Shaping is to delay, remark or, at a last resort, discard specific messages, to maintain network operability and bandwidth. In 2007, Comcast *Internet Service Provider (ISP)* reportedly applied traffic shaping to torrent files traffic to maintain a balanced bandwidth through its networks (Soghoian, 2007). Figure 14 illustrates what happens to messages that exceed the pre-configured shaping limits. In this case messages are classified accordingly to their compliance with traffic rules, it is then divided into different queues which have different priorities. These queues are organised by importance, and can follow a WFQ approach. The scheduler will then pick a message from the queues, according to the queueing and scheduling mechanism. Traffic Shaping is used alongside Traffic Policing. They complement each other.

2.3 TRAFFIC DIFFERENTIATION IN DELAY TOLERANT NETWORKS

There have been advances in trying to combine DTN and Traffic Differentiation to improve DTN performance in terms of message routing and efficiency in message delivery. For that reason, some proposals using different traffic differentiation strategies are analysed in the following sections.

2.3.1 *Buffer Management Policy*

Each node in DTN has a buffer to store messages. Since this buffer has a limited storage space, messages are eventually discarded to free storage space. Usually a FIFO mechanism is applied, which means that the first package to be placed in the buffer is also the first message to be discarded, if the buffer runs out of storage space. But it can be done in other ways, in order to achieve differentiation. Some of these approaches only apply to a certain type of routing protocols, according to their authors.

Management policy using TTL/RTTL

(Garpal et al., 2012) presents a buffer management policy *Time to Live (TTL)* and *Return Time To Live (RTTL)* approach, in which managing message drop based on the message TTL to ensure a high delivery probability. The proposed buffer management policy aims to improve the delivery probability through a new algorithm that takes into account two proprieties - the message TTL and the number of replicas -, combining them into one and using the result of that combination as a sorting mechanism. TTL and number of replicas are of relevance when applying buffer management to DTN. This approach points to an important aspect of buffering limitation in DTN. Eventually, messages that are in a certain node will need to be removed to provide space to new arriving messages, which will possibly decrease the chances of delivering the dropped messages (Garpal et al., 2012). In the following algorithm, it is observable the combination of the aforementioned proprieties, TTL and number of replicas:

```
/* New Message M (new) Arrives at node N. Available Buffer Size will be
   checked                                                              */
if Buffer Available (BA) $\geq$ M (new) then
    Put M in buffer;
else
    Find the two message that has highest (MH) & second highest replication (MSH) at
    node N;
    Let the message be MH and MSH;
    Compare TTL;
    if TTL (MH) $\geq$ TTL (MSH) then
        Discard MH & add M (new) in buffer;
    else
        Discard MSH & Add M (new);
    end
    ;
end
```

Algorithm 1: DTM Algorithm (Garpal et al., 2012)

Therefore, the combination of TTL and the number of replicas as an algorithm brings new possibilities on how messages are dropped, having a potential impact in delivery probability.

Another approach to buffer drop by message age was introduced, in which messages with higher priority, the *Assured Forwarding (AF)*, are catalogued with a younger age, when comparing with the rest of messages, tagged as low priority, the *Best Effort (BE)* (Park et al., 2014). The proposed algorithms focused on two approaches, approach ”TTL Change” and approach ”AF Priority”. ”TTL Change” allows AF messages that are created at the same time as BE messages, to have a younger age, being, therefore, the last to be dropped from the buffer. Conversely, the approach ”AF Priority” only drops AF messages if there are no BE in the buffer. With this approach, the author (Park et al., 2014) achieved the equal improvement in the delivery ratio with both algorithms, BE and AF, when compared to the traditional scheme (without differentiation when dropping messages).

Message proprieties

(Sulma Rashid, 2014) presented a weighted buffer scheme for dropping messages. This scheme was based on messages proprieties, such as size, RTTL, hop count, replication count, and the time that the message stays in queue. Those proprieties would then be ranked through a weight criteria, such as the size of the message or the number of hops. Results showed that weighted buffer scheme outperformed the existing buffer dropping policies, such as FIFO and *Last In First Out (LIFO)*, in terms of transmissions, overhead, and enhanced delivery.

A different approach is the level of prioritisation can be increased to support traffic differentiation in DTN. When proposed in a a sensor DTN scenario, in which sensors are used to distribute information, the use of several levels of prioritisation (Liu et al., 2010), categorising messages in terms of importance could be useful when deciding the messages to be discarded. For instance, if applied to an animal control scenario, it is more important to know the health condition of an animal, than the weather precipitation level at that same time.

Figure 15.: Melange state transition graph when transmitting messages (Liu et al., 2010)

An example of the Melange generic process, labelled by the authors, is described in Figure 15 (Liu et al., 2010). In this case, the authors created levels of prioritisation that are coordinated through network observation on the behaviour of the application, increasing the system performance in terms of delay, delivery, and reliability. These levels suit the two types of messages defined, those that must be delivered quickly and those that must be delivered reliably. The nine levels of prioritisation, in which the authors categorised the packages for buffer dropping, are the following (Liu et al., 2010):

1. New reliable messages generated locally
2. Old reliable messages generated locally
3. New reliable messages generated by other nodes
4. Old reliable messages generated by other nodes
5. New quick messages generated locally
6. Old quick messages generated locally
7. New quick messages generated by other nodes
8. Old quick messages generated by other nodes
9. Old reliable or quick messages that are known to have reached the base station

Due to these levels, messages considered reliable are prioritised over messages that need to be delivered quickly, since these do not have any reliability attached. The usage of multiple level of priorities outperform most current solutions or the widely used Epidemic routing protocol. This is due to the difference classification in messages, which allows quick messages to have more bandwidth and reliable messages to be persistent in buffer storage. Therefore, this mixture of priorities levels can be considered a credible and solid approach to differentiation, despite the complex system that is implied in the multiple levels of prioritisation.

2.3.2 *Queue Management*

Queue-Management is also a possibility when applying traffic differentiation to DTN. One of the proposed approaches (Lenas et al., 2011) was to divide the queues into connectivity and without connectivity, in which messages would be prioritised according to an application potential to run smoothly on current network conditions. The division is accomplished through three steps:

- Connectivity Buffer: Messages are only placed in this queue when there is a connection available. In that case they are forwarded to the next node. If the connection ceases to exist and the connectivity buffer contains messages, these will be transferred to the Persistent storage.
- Persistent Storage: Messages are placed here only when there is no connection or if there is a connection but the buffer space is full, or if the contact graph, which is known before hand is too low for it to be forward successfully.
- Non-Connectivity Buffer: Messages of high priority are place here alongside messages that did not fit the connectivity buffer, while there was a connection active, for opportunistic reasons.

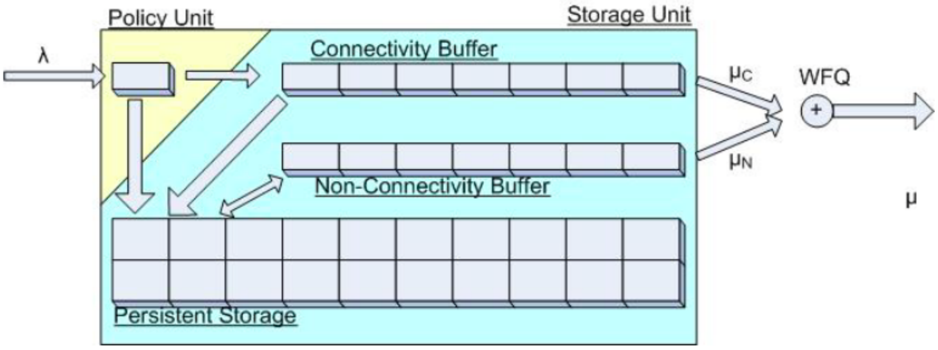


Figure 16.: Connectivity/Non-Connectivity Queuing and prioritization (Lenas et al., 2011)

This process is illustrated in Figure 16, in which the Policy unit distributes, according to the network state, the messages in the different type of queues. Afterwards, the messages from both connectivity buffer and non-connectivity buffer are picked up, using a WFQ Model (Semeria, 2001), and delivered into the network. In case there is no connectivity in the node, the message is moved to the persistent storage. If the buffer is maxed out or there is not enough time to transmit the message, it is also moved to the persistent storage. This approach also considers several points in traffic differentiation, including the contact graph. The contact graph defines, from previous encounters, the contact time available for the message transmission. That contact time is useful for the policy manager to decide whether there is enough time to transmit, avoiding incomplete transmissions and, therefore, failure to deliver other messages that would fit the time-frame.

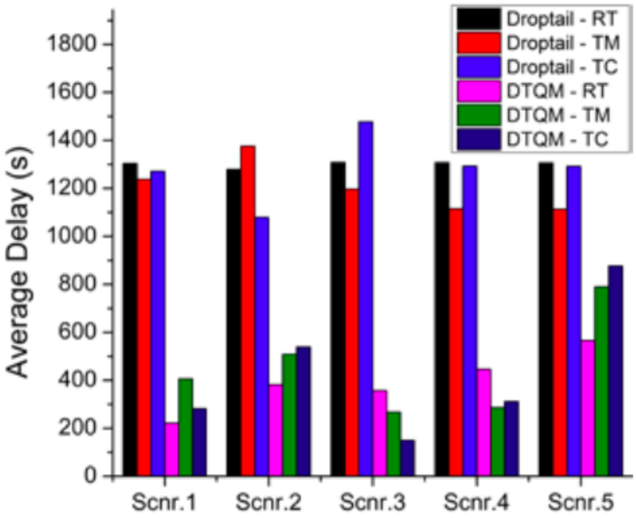


Figure 17.: Connectivity/Non-Connectivity Queuing simulation results (Lenas et al., 2011)

Considering different simulation scenarios, in Figure 17 is possible to see that the proposed Delay Tolerant Queue Management Model outperforms the DropTail approach by more than 600 seconds of difference. This model of differentiation can, apparently, achieve better results when compared to Priority Queueing (Semeria, 2001) and FIFO models when applied to DTN. Such results provide another clear path, and a new approach, on how traffic differentiation can be useful and applied to DTN.

2.3.3 *Buffer and Queue Management*

In some *Vehicle Delay Tolerant Network (VDTN)* works, it is possible to observe the application of two types of traffic differentiation, queueing and buffering management, simultaneously. This attempt is apparently improving the behaviour, performance and efficiency wise, of traffic in DTN (Soares et al., 2010). Therefore this leads to the application of priority queueing, in which messages are classified according to three levels of priority such as high, medium, and low (Soares et al., 2010). Additionally, buffer management is also considered in terms of prioritisation, as it attributes more importance to newly generated messages rather than old generated messages or messages with low time-to-live (Soares et al., 2010).

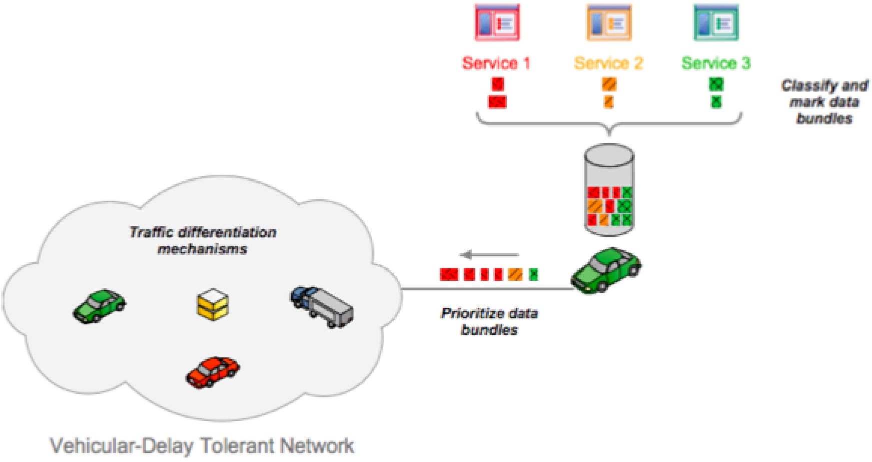


Figure 18.: Traffic differentiation in Vehicle Delay Tolerant Networks (Soares et al., 2010)

Figure 18 demonstrates how traffic differentiation can be applied to vehicular DTN. The approach is to apply different priorities only to message groups, ordering messages while respecting their priority, and later transmit them to other nodes in the correct prioritised order. This situation provides a great opportunity for new messages to be promptly delivered, as well as messages that are for too long in the buffer, or have a low time-to-leave, to be rapidly discarded. The results confirm the use of these approaches with better performance, as well as coordinated network resource allocations, when compared to classical protocol approaches.

Although buffer management and queueing techniques are the standard approaches, there are other solutions in the literature. For example, trusted based encounter solutions, in which nodes are classified, instead of messages being classified, from trustworthy to untrustworthy (Chen et al., 2010). These categories are stored and traded between the nodes, and are used to make decisions about through which nodes to pass the messages and the information. Moreover, the trust management for encounter-based approach assumes the existence of an *Intrusion Detection System (IDS)*, which is able to behave with high performance probability to detect nodes considered malicious, which are, therefore, labelled untrustworthy. Furthermore, the IDS classifies selfish nodes as the ones that prioritise locally generated messages over remotely generated ones. Nodes can be demoted from trustworthy to untrustworthy or promoted from untrustworthy to trustworthy depending on their behaviour. Nodes that are considered malicious, are permanently classified as untrustworthy, and cannot be promoted to trustworthy again (Chen et al., 2010).

In simulation scenarios, using Epidemic routing protocol, the trust-based mechanism obtain an ideal performance level in delivery ratio and message delay, without generating a high message overhead that is common when using the Epidemic protocol. In Figure 19, using the Epidemic routing approach, it can be observed the number of messages being copied and

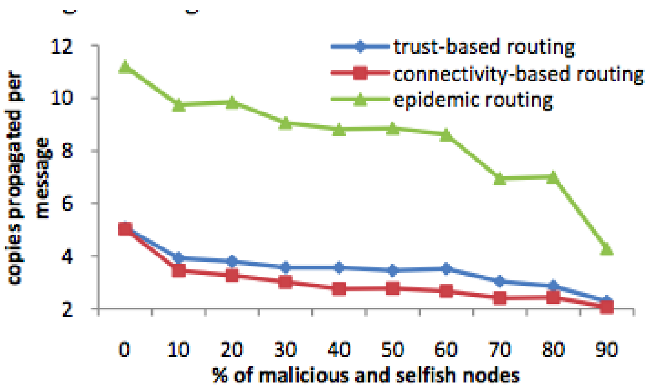


Figure 19.: Epidemic routing message propagation development with trust-based (Chen et al., 2010)

transmitted to other nodes. The lower the percentage of malicious nodes, the higher the number of messages that are copied and transmitted. Conversely, when the percentage of malicious nodes increases, the number of copied messages decreased, meaning less network overhead. Since the algorithm predicts that the malicious node will not deliver the message to its intended recipient. Therefore, the algorithm chooses not to deliver the message to that node (Chen et al., 2010).

2.3.4 *Summary*

Table 1.: Some approaches of Traffic Differentiation in DTN

Proposal	Buffer Management	Queue Management
(Garpal et al., 2012)	X	
(Park et al., 2014)	X	
(Sulma Rashid, 2014)	X	
(Liu et al., 2010)	X	
(Lenas et al., 2011)		X
(Soares et al., 2010)	X	X
(Chen et al., 2010)	X	X

The goal in this section was to present an overview of the current approaches on Traffic Differentiation in DTN. Most of the research was focused solely on Buffer Management (Garpal et al., 2012) (Park et al., 2014) (Sulma Rashid, 2014) (Liu et al., 2010), either through messages proprieties, such as size, hop count, and replication count, or the observing the TTL/RTTL of the message. Queue Management was also a safe bet, but with major focus on the connectivity between nodes (Lenas et al., 2011) (Semeria, 2001).

Lastly, some work has been done in combining the two, taking into account how the buffer discards the message and how it will forward (Soares et al., 2010) (Chen et al., 2010). In Table

1 it is possible to observe how each work presented in this section focused its differentiation efforts.

2.4 SUMMARY

In this chapter both DTN and traffic differentiation were overviewed, to better understand the current existing methods and approaches, regarding these strategies. Traffic differentiation in DTN combines both, introducing strategies to improve or enhance DTN, in terms of message delivery. The following chapter will overview different strategies, proposed in this thesis, that also apply traffic differentiation in DTN, with more focus in traffic prioritisation than buffer drop management.

3

PROPOSAL FOR TRAFFIC DIFFERENTIATION ON DELAY TOLERANT NETWORKS

This chapter presents new strategies proposed for traffic differentiation on Delay Tolerant Networks (DTN), as well as the steps and design required to introduce improvements to the delivery rate. First, an algorithm that was conceived and designed focusing mainly on PQ is presented. Second, the necessary classification and message scheduling are discussed, in order to achieve a good message ordering scheme. Using existing routing strategies in The One simulator as a base model, an analyse is made that includes all the algorithm particularities. In the last section of this chapter, the full algorithm, adapted to the various routers and their characteristics, is exposed and explained.

3.1 ALGORITHM SPECIFICATION

Most of the approaches, presented in previous chapter, are based on a buffer drop differentiation. These approaches create a set of directives that allow to select which message, present in the buffer of a node, is going to be discarded. This means that, it does not create a direct chance of delivery to certain messages, it just avoids them to be discarded later than sooner. The purpose of this thesis is to integrate traffic differentiation in delay tolerant networks, but using a different approach. The goal is to achieve differentiation, in a form similar to the currently existing one on non delay tolerant network connections that is through pre-selection of which messages are the ones to have priority on sending. This pre-selection directly influences the delivery. This chapter focus on the algorithms required to improve the delivery rate through traffic differentiation in DTN, by prioritising messages. The base algorithm will then serve as a guide for existing routing protocols, namely Epidemic, Spray and Wait, and PRoPHET. In the following sections it is also presented how the existing routing protocol implementations could benefit from the designed algorithm. There was also the need to introduce different variations to the designed algorithm, to better fit and influence some of the selected routing protocols.

3.1.1 *Traffic prioritisation in Delay Tolerant Networks*

PQ is a simple approach already used in traffic differentiation (Semeria, 2001), when considering existing non delay tolerant network connections. It was also implemented by a various number of authors (Soares et al., 2010) (Chen et al., 2010) (Lenas et al., 2011). The goal is to create a mechanism of priority that allows the node to decide, in which order and which messages, should be delivered to the next node, calculated on each encounter and based on a previous message classification.

The following draft algorithm describes the three step process regarding the approach taken to differentiate traffic in Delay Tolerant Networks scenarios:

1. Classify messages according to their priority
2. For each encounter, calculate the message order according to the defined scheduling approach and message priorities
3. Send the messages

In this thesis, the calculation time of message ordering, is not taken into account.

3.1.2 *Classification*

The priority levels selected to classify the messages were three. This is the most common way to classify messages (Semeria, 2001), usually under the names of High, Normal and BE Priority. For the sake of clarification, the selected names for the priorities were:

- High Priority (HP) - Messages that need to be delivered with the most importance
- Medium Priority (MP) - Messages that have a normal delivery intent
- Low Priority (LP) - The remaining messages

3.1.3 *Message Scheduling*

The second part of the algorithm, and the most important one, is the message ordering. Through the design, the algorithm starts with a simple ordering mechanism and moves on to more complex mechanisms adapted to the existing routing protocols that also influence how messages are ordered and filtered.

Strict Priority Ordering

The first basic approach is to simply order the messages by looking at their priority. This means that messages with higher priorities would always be place up front of the lower priorities. In any case, depending on the selected routing strategy to deliver messages, it uses a simple premise, there will be a considerable number of higher priorities messages flowing on each node transaction that is greater than other priority messages, which may introduce starvation for the remaining priorities. In theory, this could lead to a higher delivery rate of those messages than when not using any type of differentiation.

Weighted Round Robin Ordering

The obvious problem regarding just using a strict priority ordering approach is starvation. Although it theoretically increases the chances of delivery of messages classified as *High Priority (HP)*, it can have a side-cause effect by completely obliterating the chances of delivery to any messages that are not classified as HP, but as *Medium Priority (MP)* or *Low Priority (LP)*. Hence, to address that possibility, a weighted priority approach was designed. This approach of the algorithm is called *Weighted Round Robin Ordering (WRRO)*. WRRO consists in giving weights to each priority and allowing messages to follow those weight rules. These weights apply to a predefined number of message on each iteration, called scale. The WRRO consists in a algorithm for ordering messages, on each node encounter, according to their priorities. This algorithm can be referred in Algorithm 2.

3.1.4 *Routing Protocol Improvements for Differentiation*

After drawing the first draft of the algorithm, it was required to introduce the algorithm on each approach of the selected routing strategies to verify the algorithms behaviour, when faced with their characteristics.

Epidemic Routing Protocol

The Epidemic Routing Protocol prepares a list of messages to be sent, with all the messages at the nodes buffer (Vahdat and Becker, 2000) (Keränen et al., 2009). For that reason, applying the differentiation algorithms directly, did not bring any changes to the core behaviour of the epidemic routing approach. The algorithms just re-order the messages to be sent, which by definition are not ordered by the epidemic approach. This means the algorithms implementation in Epidemic Router is simple and straightforward.

```
/* This algorithm needs a scale and weights to be configured.           */
/* As an example a scale of 10 will be used.                            */
/* As an example of weights, these distribution will be used:           */
/* 70% for high priority messages, 20% for medium priority messages and
   10% for low priority messages                                        */
scale = 10;
HighQueue = getAllMessagesInNodeThatAreHighPriority();
MediumQueue = getAllMessagesInNodeThatAreMediumPriority();
LowQueue = getAllMessagesInNodeThatAreLowPriority();
messagesToSend = new List();
while messagesToSend.length < totalMessagesInNode.length do
    nrHigh = scale * 0.7; nrMedium = scale * 0.2; nrLow = scale * 0.1;
    while nrHigh > 0 && !HighQueue.Empty() do
        messagesToSend.add(HighQueue.PollMessage());
        nrHigh = nrHigh - 1 ;
    end
    while nrMedium > 0 && !MediumQueue.Empty() do
        messagesToSend.add(MediumQueue.PollMessage());
        nrMedium = nrMedium - 1 ;
    end
    while nrLow > 0 && !LowQueue.Empty() do
        messagesToSend.add(LowQueue.PollMessage());
        nrLow = nrLow - 1 ;
    end
end
return messagesToSend;
```

Algorithm 2: WRR Algorithm;

Spray & Wait Routing Protocol

SW Routing Protocol basis its implementation on Epidemic Routing Protocol, but creates a fixed number of copies of messages, limiting the network congestion that Epidemic Router inherently creates (Spyropoulos et al., 2005) (Keränen et al., 2009). For that reason, applying some differentiation algorithms can bring some limitations, due to this copy limit. In view of this possible limitation, the WRRO algorithm needs to evolve to work properly on this router, also taking in account the copy limit.

The WRRO with this specific evolution is called *Distinct Copies Weighted Round Robin Ordering (DCWRRO)*. See algorithm 3 and 4.

```
/* This algorithm needs a scale, weights and number of copies to be
   configured                                                          */
/* As an example, the following allowed copies for each priority will be
   used:                                                               */
/* 12 copies for high priority messages, 6 copies for medium priority
   messages and 1 copy for low priority messages                       */
/* As an example a scale of 10 will be used.                           */
/* As an example of weights, these distribution will be used:          */
/* 70% for high priority messages, 20% for medium priority messages and
   10% for low priority messages                                       */
/*                                                                     */
scale = 10;
HighList = getAllMessagesInNodeThatAreHighPriority();
MediumList = getAllMessagesInNodeThatAreMediumPriority();
LowList = getAllMessagesInNodeThatAreLowPriority();
HighQueueWithAvailableCopies = new List();
MediumQueueWithAvailableCopies = new List();
LowQueueWithAvailableCopies = new List();
foreach message in the HighList do
    if message.availableCopies > 1 || (message.destinyNode() == getTargetedNode()
    then
        HighQueueWithAvailableCopies.add(message);
    end
end
foreach message in the MediumList do
    if message.availableCopies > 1 || (message.destinyNode() == getTargetedNode()
    then
        MediumQueueWithAvailableCopies.add(message);
    end
end
foreach message in the LowList do
    if message.availableCopies > 1 || (message.destinyNode() == getTargetedNode()
    then
        LowQueueWithAvailableCopies.add(message);
    end
end
messagesToSend = new List();
```

Algorithm 3: Distinct Copies Weighted Round Robin Variation Algorithm - Part 1;

```
while messagesToSend.length < totalMessagesInNode.length do
    nrHigh = scale * 0.7; nrMedium = scale * 0.2; nrLow = scale * 0.1;
    while nrHigh > 0 && !HighQueueWithAvailableCopies.Empty() do
        messagesToSend.add(HighQueueWithAvailableCopies.PollMessage());
        nrHigh = nrHigh - 1 ;
    end
    while nrMedium > 0 && !MediumQueueWithAvailableCopies.Empty() do
        messagesToSend.add(MediumQueueWithAvailableCopies.PollMessage());
        nrMedium = nrMedium - 1 ;
    end
    while nrLow > 0 && !LowQueueWithAvailableCopies.Empty() do
        messagesToSend.add(LowQueueWithAvailableCopies.PollMessage());
        nrLow = nrLow - 1 ;
    end
end
return messagesToSend;
```

Algorithm 4: Distinct Copies Weighted Round Robin Variation Algorithm - Part 2;

PRoPHET Routing Protocol

SW was not the only routing strategy that forced to some modifications on the WRRO algorithm. PRoPHET had a twist, the probability table. To improve possibilities in delivering messages marked as HP, the differentiation should take advantage of the probability table that PRoPHET had implemented. For that reason, there was the need to regulate which messages should keep their priority status, even if their delivery probability is low.

This WRRO add-on was named *Hybrid Weighted Round Robin Ordering (HYBWRRO)*. This improvement was specially added because the algorithm actually checks the messages that are ready to be sent. There are two versions, both taking advantage of the existing message probability calculated by PRoPHET. Version A discards the message, which can be seen at algorithm 5 and version B decreases the message priority, which can be seen at algorithm 6. The message discard version (Version A), allows messages that have a low probability of delivery, according to the PRoPHET routing protocol, to not clutter the queues. In the case of a high priority messages that has a low probability to delivery, it would be prioritised several times where other high priority messages or even medium priority messages could have a higher delivery probability, and therefore, increase the delivery rate. The message priority decrease version (Version B), decreases the messages priority by one level, if it has a low probability to delivery. High priority messages are decreased to medium priority messages and medium priority messages are decrease to low priority messages. Contrary to the discard version, this version allows the message to still be queued up for a chance to be transmitted to the node. The probability of acceptance of a message is defined before using the HYBWRRO algorithm. This is value used as a reference against the message priority, to then choose the action over it.

```
/* This algorithm needs a scale, weights and a probability acceptance to
   be configured                                                          */
/* As an example a scale of 10 will be used.                             */
/* As an example a probability acceptance rate of 70% will be used.      */
/* As an example of weights, these distribution will be used:            */
/* 70% for high priority messages, 20% for medium priority messages and
   10% for low priority messages                                          */
/*                                                                        */
scale = 10;
HighQueueUnprepared = getAllMessagesInNodeThatAreHighPriority();
MediumQueueUnprepared = getAllMessagesInNodeThatAreMediumPriority();
LowQueueUnprepared = getAllMessagesInNodeThatAreLowPriority();
messagesToSend = new List();
foreach message in the HighQueueUnprepared do
    if $message.getProbability() >= 0.7$ then
        HighQueue.add(message);
    end
end
foreach message in the MediumQueueUnprepared do
    if $message.getProbability() >= 0.7$ then
        MediumQueue.add(message);
    end
end
foreach message in the LowQueueUnprepared do
    if $message.getProbability() >= 0.7$ then
        LowQueue.add(message);
    end
end
while $messagesToSend.length < totalMessagesInNode.length$ do
    The remaining code matches the original algorithm. See algorithm 2.
end
return messagesToSend;
```

Algorithm 5: HYBWRR Algorithm with Discard Policy;

In terms of priority decreasing, it is only decreased by one level, which means that HP messages, when decreased, become MP messages and MP become LP messages. Only LP messages do not get affected by the priority decrease decision.

Regarding discarding messages when they do not meet the requirement, the message is discarded from the current node message exchange. This means that the message is not actually discarded from the node, but only ignored for the current node message exchange. As an example, Node A is exchanging messages with Node B, and message A is discarded from the current exchange because it does not meet the necessary requirements, namely probability rate. If Node A starts exchanging messages with Node C, message A priority is evaluated again. The decision of message is discard is confined to the node transaction, following the PRoPHET probability table, which acts in the same way.

```
/* This algorithm needs a scale, weights and a probability acceptance to
   be configured                                                        */
/* As an example a scale of 10 will be used.  As an example a probability
   acceptance rate of 70% will be used.                                 */
/* As an example of weights, these distribution will be used:           */
/* 70% for high priority messages, 20% for medium priority messages and
   10% for low priority messages                                        */
/*                                                                      */
scale = 10;
HighQueueUnprepared = getAllMessagesInNodeThatAreHighPriority();
MediumQueueUnprepared = getAllMessagesInNodeThatAreMediumPriority();
LowQueueUnprepared = getAllMessagesInNodeThatAreLowPriority();
messagesToSend = new List();
foreach message in the HighQueueUnprepared do
    if message.getProbability() >= 0.7 then
        HighQueue.add(message);
    else
        MediumQueue.add(message);
    end
end
foreach message in the MediumQueueUnprepared do
    if message.getProbability() >= 0.7 then
        MediumQueue.add(message);
    else
        LowQueue.add(message);
    end
end
foreach message in the LowQueueUnprepared do
    if message.getProbability() >= 0.7 then
        LowQueue.add(message);
    end
end
while messagesToSend.length < totalMessagesInNode.length do
    The remaining code matches the original algorithm. See algorithm 2.
end
return messagesToSend;
```

Algorithm 6: HYBWRR Algorithm with Decrease Policy;

3.2 ALGORITHM ILLUSTRATION

3.2.1 *Message Scheduling: Strict Priority Ordering*

The basic algorithm, previously entitled *Strict Priority Ordering (SPO)*, ordered the messages to be delivered by that node by that message priority. Figure 20 illustrates this aspect correctly, following a message list order left to right, up to down. The algorithm proceeds the following way:

1. Two nodes meet, Node A and Node B.
2. One of them, Node A, has 20 messages that wants to pass to Node B.
3. Initially, the message list is completely unordered.
4. Strict Priority Ordering is applied to the message list and the messages are ordered by their priority, from High to Low.
5. The final message list represents the order by which the messages are sent to Node B.

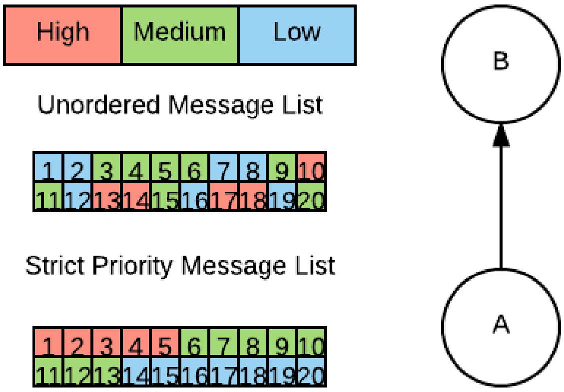


Figure 20.: How Strict Priority Ordering works.

It is important to note that, even though the list in this example contains 20 messages, all of them may not be delivered. The reasons behind this were previously mentioned and they are related to buffering limitations, time of contact, and transmission speed parameters.

3.2.2 *Message Scheduling: Weighted Round Robin Ordering*

The SPO algorithm, orders messages by priority. This means that medium and low priority messages could suffer from starvation without having any chances of ever being delivered, if a great amount of high priority messages exists in the buffer. For that reason, an additional differentiating mechanism was developed, that uses a simple weighted round-robin to avoid starvation of lower priority messages. This scheduler follows a percent weight order. An example of a possible weight distribution is:

- High Priority = 70%
- Medium Priority = 20%
- Low Priority = 10%

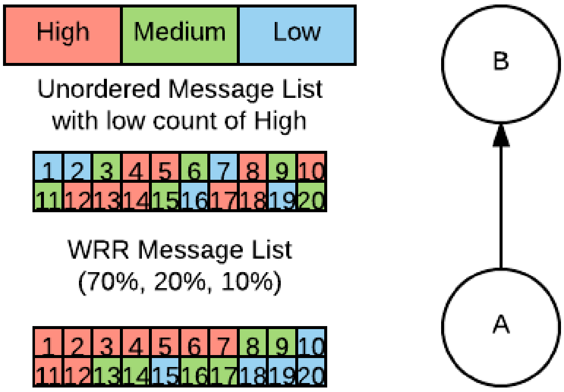


Figure 21.: How Weighted Round Robin Ordering works

There was a need to define the granularity, in order for the structure of the queue to remain constant. For that reason it was defined a value, which it is referred to it as scale. As an example, and for a easy conversion, a constant scale of 10 was applied to these percentages. In a total of ten messages, 70% are of High Priority, 20% of Medium Priority, and, finally, 10% of Low Priority. Unless there are no messages of a certain priority, these will scale correctly and accompany the list of messages growth. Figure 21 illustrates the adopted WRRO implementation. The algorithm proceeds in the following way:

1. Two nodes meet, Node A and Node B.
2. One of them, Node A, has 20 messages that wants to pass to Node B.
3. Initially, the message list is completely unordered.
4. WRRO is applied, initially gathering a total 10 messages that follow the percentages weight rule as previously described. This process is cyclically applied until all the messages on the unordered list are now ordered.

5. The final list message list represents the order by which the messages are sent to Node B.

As illustrated in Figure 21, for every 7 HP messages, 2 of MP and 1 of LP follow. This repeats until there is no HP, MP or LP marked messages to be processed.

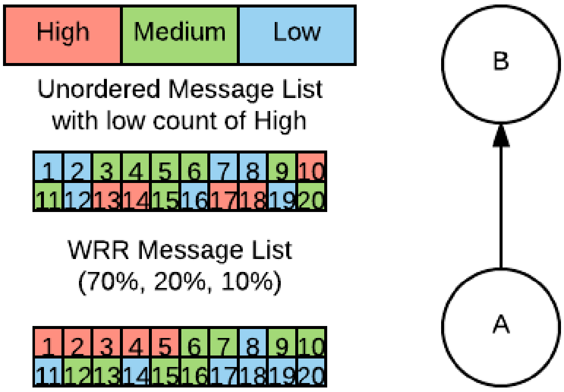


Figure 22.: How WRRO works with a low count of messages marked as HP

In case of the list has a reduced number of messages from a certain priority, WRRO still respects the scale, that was previously defined, of messages to be included. This means that if there are no high or medium priority messages, the tail of the list will have all the remaining low priority messages. Figure 22 represents a case where there are not enough High priority and so the remaining messages, with priorities medium and low, fill the remaining list space.

3.2.3 *Epidemic Routing Protocol Improvements*

The approach to Epidemic Routing, did not require any variations to the initial algorithm. Both SPO and WRRO were applied directly to the Epidemic router.

3.2.4 *Spray & Wait Routing Protocol Improvments*

SW, based on Epidemic routing, has a limitation on the number of copies of messages that are allowed to be transferred. This limited copy propagation, that exists in SW, prevents differentiation to have impact on the message delivery. This forced a variation to the WRRO algorithm, that allowed a more customisation and flexibility regarding the number of copies for each message. Table 2 and Table 3, as an example, represent the WRRO algorithm variation applied to both SW modes, normal and binary.

Table 2.: Spray & Wait using Normal mode - Distinct Copies WRRO

Mode/Priority	High	Medium	Low
Base	6	6	6
Distinct Copies Weighted Round Robin Ordering A	12	6	1
Distinct Copies Weighted Round Robin Ordering B	18	12	3

Table 3.: Spray & Wait using Binary mode - Distinct Copies WRRO

Mode/Priority	High	Medium	Low
Base	6	6	6
Distinct Copies Weighted Round Robin Ordering A	9	3	1
Distinct Copies Weighted Round Robin Ordering B	12	9	3

Figure 23 shows how DCWRRO behaves with a simple distribution of distinct copies, as seen in Table 2. Since LP messages only have one allowed copy, they are only delivered to its final destination. In Figure 23, every encounter is never the final, and so the messages that only have one copy left, are not listed for delivery. This means that by the 3rd encounter, there are only HP messages left that can be transmitted to nodes that are not, that message final destination. The values in Table 2 are higher than in Table 3 because of the way that SW normal version works (horizontal spread) when comparing to the SW binary mode (both vertical and horizontal), since the binary mode generates more messages. These variants difference is focused solely on the number of copies. They apply the same DCWRRO algorithm.

3.2.5 *PRoPHET Routing Protocol Improvements*

PRoPHET organises its list of messages to be sent by each mode according to a generated probability table. For that reason, an additional scheduling variant was created, entitled Man In the Middle, that takes advantage of the generated table. Hybrid Weighted Round Robin Ordering has a simple approach, with two possible final actions. The method takes into account the probability of delivery of a certain message to a certain node. As an example, using a 70% minimum probability for accepting a message, if the probability of delivery through that node to reach its destination is less than 70%, then it is considered that the message does not meet the minimum requirements, and will be discard or its priority level decreased. It is important to know that this analysis is done at each point of contact between nodes. This means that a message A exchanged between nodes Y and Z, can meet the requirements, and the same message in a different node encounter (e.g., between Y and W), can fail to meet the same requirements. For this matter each message is evaluated on each contact assuming, we have time to run the algorithm. There are, therefore, two ways to deal with a message that does not meet the requirements.

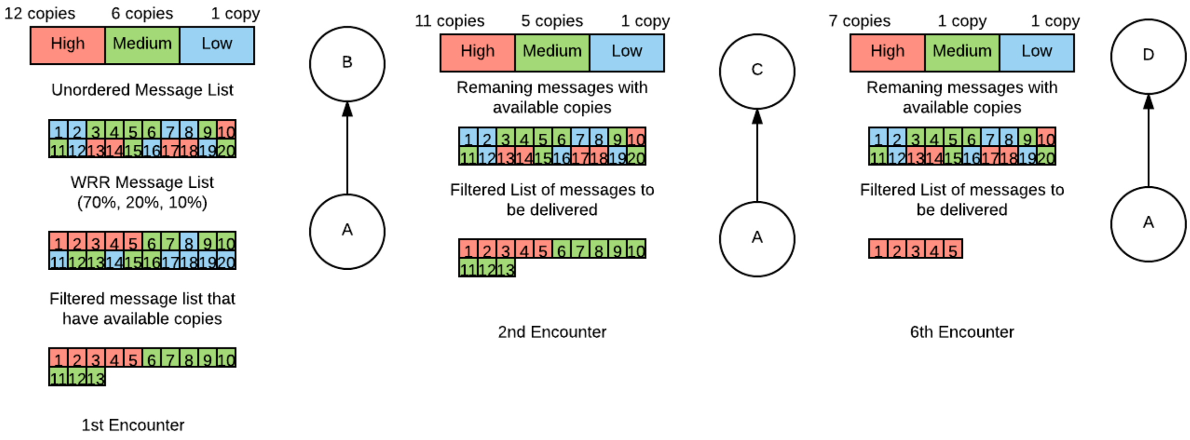


Figure 23.: How Spray & Wait Distinct Copies WRRO works

Hybrid Weighted Round Robin Ordering - Discard

The first way is through discarding. During a node meet, if a message that is about to be sent has less probability than 70% to reach its destination, it is discarded for that transaction and it is not sent to the other node. The idea behind this approach is to reduce the number of messages to be sent if they have a low chance of being delivered to the final node. On a HP message, if the probability of delivery value is lower than the acceptance value, then the message will not be considered for this node message meeting. The goal of this discard, is to favour messages that have good probability odds, and maintain the central objective of this routing protocol.

Even if this is a high priority message, if it has a low probability of delivery than it is better to have other messages of same or lower priority to receive more attention, as long as they have higher probability of being delivered.

Figure 24 shows how HYBWRRO behaves in Discard mode. Each message is carefully analysed regarding its probability of delivery, following PRoPHET design base. The message is then discarded if it does not meet the probability minimum, that in Figure 24 is defined as 70%. If the message has more than 70% to be delivered to its destination, that it is added just as the normal WRRO implementation suggests.

Hybrid Weighted Round Robin Ordering - Decrease Priority

The Decrease Priority does exactly what the name suggests. In case of failing to comply with the minimum of 70% delivery probability, the priority of that messages will be decreased. This means that a HP message with a probability of 60% to be successfully delivered, will be remarked as medium priority message. The Table 4 shows what is the behaviour for all the

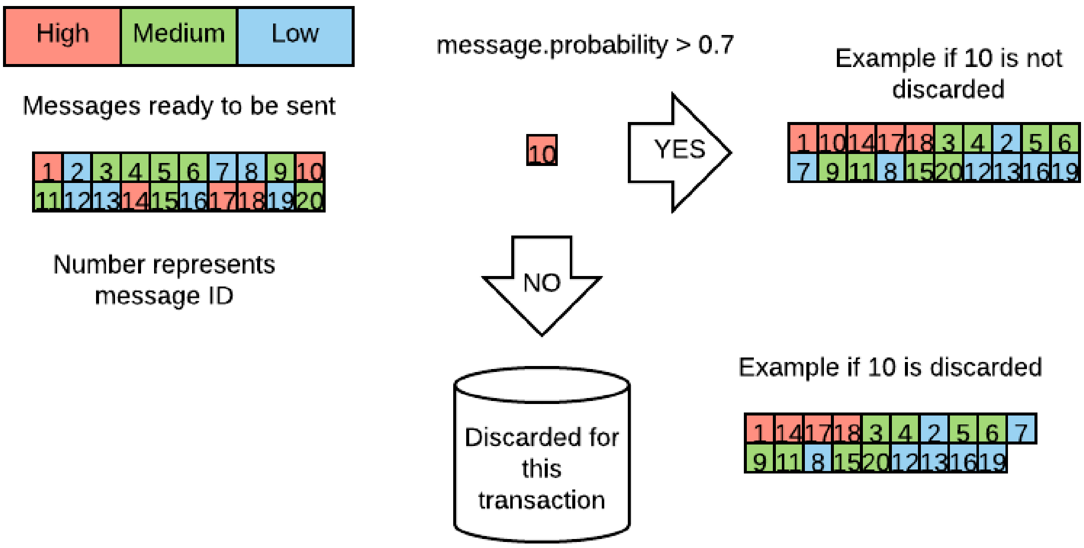


Figure 24.: How PRoPHET Hybrid WRRO (Discard) works

existing priorities. The goal of this approach is to reduce the importance of messages that have a lower probability of delivery, giving space for other high priority messages, that have a higher probability of delivery.

Figure 25 shows how HYBWRRO behaves in Decrease mode. Each message is carefully analysed regarding its probability of delivery, following PRoPHET design base. The message has its priority decreased if it does not meet the probability minimum, that in Figure 25 is defined as 70%. If the message has more than 70% to be delivered to its destination, that it is added just as the normal WRRO implementation suggests. It is possible to see that message marked with identification 10, is marked with the green colour when its priority is decreased. This means that the priority is decreased by one, from HP to MP

Table 4.: Hybrid WRRO Decrease Priority probability conversion

Priority	Decrease Priority if failed to comply with a 70% probability value
High	Medium
Medium	Low
Low	Low

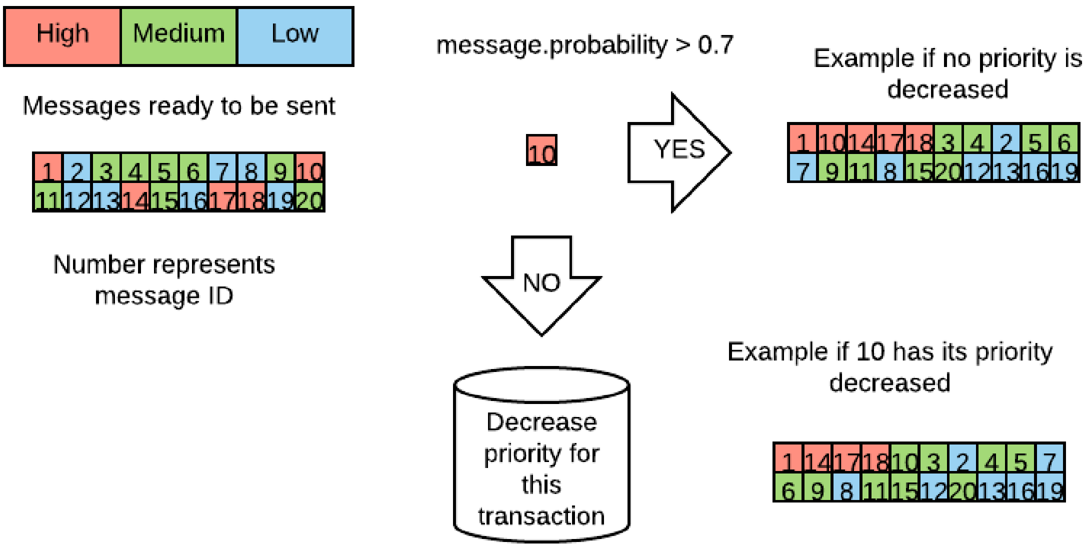


Figure 25.: How PRoPHET Hybrid WRR (Decrease) works

Table 5.: Summary of all the modes of each router

Schedule Mode/Router	Epidemic	Spray & Wait	PRoPHET
Normal	X	X	X
Strict Priority Ordering	X	X	X
Weighted Round Robin Ordering	X	X	X
Distinct Copies WRR Ordering		X	
Hybrid WRR Ordering Discard			X
Hybrid WRR Ordering Decrease			X

3.2.6 *Summary*

Table 5 overviews all the combinations of scheduling approaches and routing modes that were proposed and will be tested in this work, and which are now summarised.

The normal mode that any router currently has, represents the base version, with no differentiation objectives.

SPO approach messages are ordered according to their priority with no regards at whether there is starvation of other priorities. Basically is a no-toll high-way for HP messages.

In order to avoid possible starvation of the remaining priorities, a WRRO algorithm is implemented. This way, through weights, it is possible to create a fair distribution queueing system, while still following the message priorities.

In light of the particularity of SW regarding the number of copies it allows per message, the basic WRRO algorithm is adapted to have this propriety in consideration. DCWRRO variant allows the differentiation to be extended to the number of copies, giving HP messages more copies than MP, and giving MP more copies than LP.

PRoPHET also had different characteristics when dealing with messages. Since it is based on a probability table, the algorithm needed to consider each delivery message probability and take advantage of it. HYBWRRO variant takes the message delivery probability into account, by discarding or decreasing the priority of messages that have a low probability of delivery. This probability will affect the effect of differentiation, but it should be high enough to balance both differentiation and probabilities. Lowering the probability requirement, improves the effect of the differentiation and reduces the influence of the probabilities table.

4

IMPLEMENTATION

The simulator used for the development and testing of the devised differentiation mechanisms was The One Simulator (Keränen et al., 2009), which is well suited for Delay Tolerant Networks (DTN). Since it does not support differentiation out of the box, it need development to adapt certain existing features to work with traffic differentiation, such as the messages, the generators, and the reports. Additionally, the presented algorithms need to be implemented and integrated in the existing router implementations. In this chapter, it will be described the simulation tool, code development and algorithm implementation.

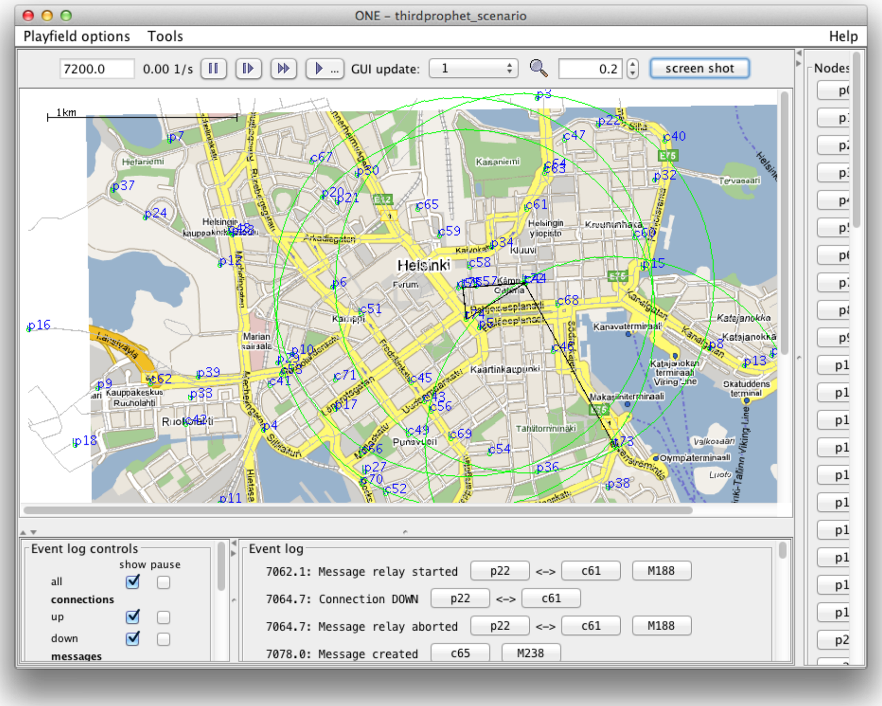
4.1 THE ONE SIMULATOR

The One Simulator is one of the most used simulators for DTN (Keränen et al., 2009). It is developed using JAVA programming language and is highly customisable. It also has several class implementations that fit on the current study's objectives Figure 26. It contains a report generator that informs about the delivery rate of each message, how many are created, how many are repeated, hop-counts, and many other important measurement variables. Additionally, it includes implementations of Epidemic, SW and PRoPHET routers, allowing to test and trial how the network behaves with and without modifications. For those reasons, The One Simulator source code was used for an easier manipulation.

This simulator is compatible with several node types, such as buses, people, and cars. The nodes move around the selected map exchanging messages with each other node that encounters.

The present approach to this problem is focused on a specific scenario. When necessary, protocols were adapted to accommodate the traffic differentiation proposals that the present thesis provides. While using The One Simulator, the default scenario of Helsinki was adopted, as can be seen in Figure 26. Since it is one of the default scenarios used in many experiments and simulations (Keränen et al., 2009), it was chosen as the main scenario for the experiments of this thesis.

The One simulator is structured by based classes, each serving as model to further customisation and development, as seen at Figure 27. The routing implementation in The One, using



7200.0 0.00 1/s GUI update: 1 0.2

	show	pause
all	☑	☐
connections		
up	☑	☐
down	☑	☐
messages		

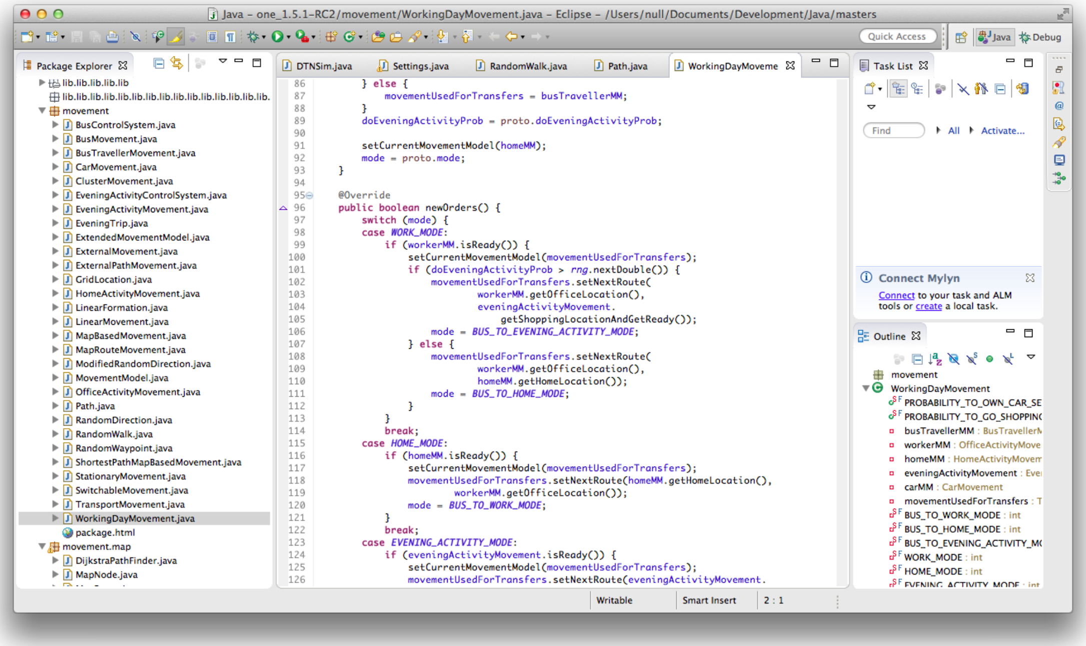
```
7062.1: Message relay started  p22 <-> c61  M188
7064.7: Connection DOWN  p22 <-> c61
7064.7: Message relay aborted  p22 <-> c61  M188
7078.0: Message created  c65  M238
```

Figure 26.: Example of The One simulator graphical interface (Keränen et al., 2009)

Epidemic router as an example, is done through the ActiveRouter class, which contains the base interface to message delivery and node encounter. Epidemic router is build by extending the ActiveRouter and manipulating according it its characteristics. All the routers, that are included in The One source code, are build on top of the ActiveRouter. The development made in this thesis, regarding routing protocol improvements, was done by building the router traffic differentiation versions on top of the existing base routers. As an example, the Differentiated Epidemic Router was built, by extending the Epidemic Router existing class. This also applies to the reports that the simulation generate and the messages class that is exchanged between the nodes.

4.2 DEVELOPED MODULES

Several classes required new methods and proprieties, to work with traffic differentiation. Figure 28 shows the most important classes that, are referred in this section, with new added new functionalities.



```java
        } else {
            movementUsedForTransfers = busTravellerMM;
        }
        doEveningActivityProb = proto.doEveningActivityProb;

        setCurrentMovementModel(homeMM);
        mode = proto.mode;
    }

    @Override
    public boolean newOrders() {
        switch (mode) {
        case WORK_MODE:
            if (workerMM.isReady()) {
                setCurrentMovementModel(movementUsedForTransfers);
                if (doEveningActivityProb > rng.nextDouble()) {
                    movementUsedForTransfers.setNextRoute(
                            workerMM.getOfficeLocation(),
                            eveningActivityMovement.
                                getShoppingLocationAndGetReady());
                    mode = BUS_TO_EVENING_ACTIVITY_MODE;
                } else {
                    movementUsedForTransfers.setNextRoute(
                            workerMM.getOfficeLocation(),
                            homeMM.getHomeLocation());
                    mode = BUS_TO_HOME_MODE;
                }
            }
            break;
        case HOME_MODE:
            if (homeMM.isReady()) {
                setCurrentMovementModel(movementUsedForTransfers);
                movementUsedForTransfers.setNextRoute(homeMM.getHomeLocation(),
                        workerMM.getOfficeLocation());
                mode = BUS_TO_WORK_MODE;
            }
            break;
        case EVENING_ACTIVITY_MODE:
            if (eveningActivityMovement.isReady()) {
                setCurrentMovementModel(movementUsedForTransfers);
                movementUsedForTransfers.setNextRoute(eveningActivityMovement.
```

Figure 27.: Example of The One source code (Keränen et al., 2009)

4.2.1 *Message and Message Generator*

To enable traffic differentiation, certain classes related to messages, needed to be extended. The One Simulator uses an object called Message and another one called Message Generator. These classes allow the simulator to create messages and inject them into the simulation scenario, and to control all the flow and information around them. Due to this important role they play, those classes had to be extended to fulfil the purposes of the present study.

4.2.2 *DiffMessage*

The first thing to be implemented was a new message compatible with traffic differentiation called DiffMessage. The message had to be compatible with the differentiation mechanisms that were integrated in the DTN environments. For that reason, a new propriety to the object, entitled DiffMessagePriority, was added.

This enumeration property allowed to classify each message and associate it with a priority, namely:

- PriorityHigh - 0
- PriorityMedium - 1

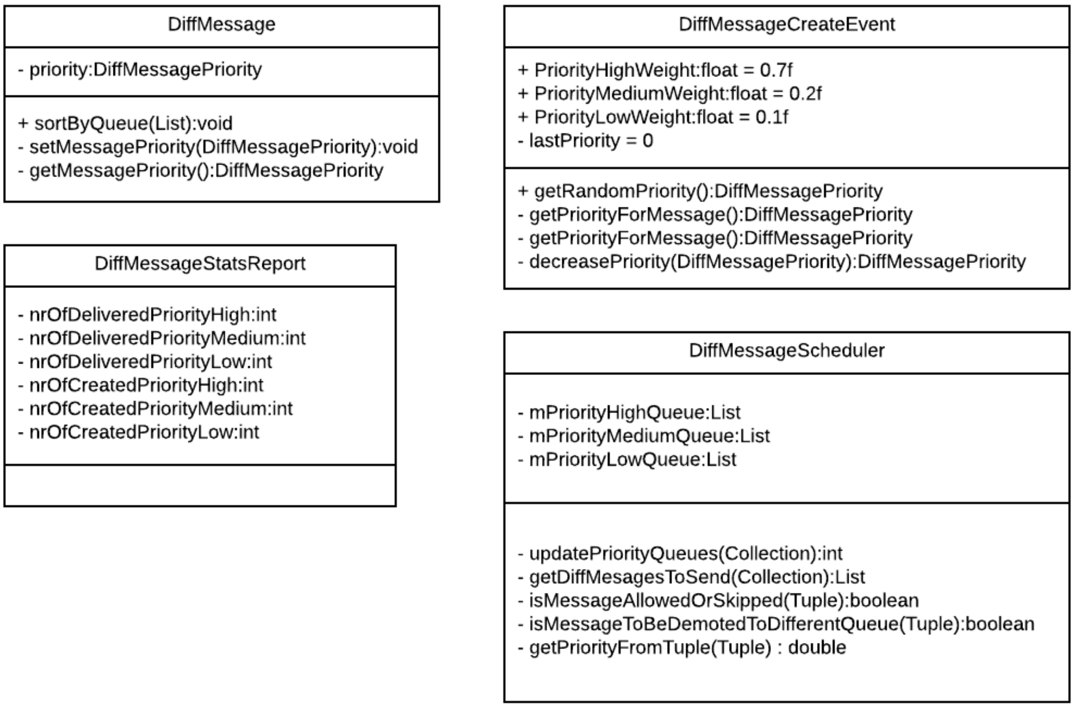


Figure 28.: Class diagram of classes that needed new methods and proprieties

- PriorityLow - 2
- PriorityDiscard - 3

This four priorities represent the importance that is ascribed to each message. The message that has a priority represented by a zero (PriorityHigh) is considered to be the highest priority of delivery, where number two is the lowest in terms of priority, and, lastly, number three represents a priority that should be ignored by the node. An example use of this priority is the discard of a message, instead of being relayed to the next node.

Additionally, DiffMessage has also an important method that uses this priority propriety, entitled *sortByQueueMode*. This method is a *Collection.Sort* JAVA type and implements a Comparator, which allows to compare two objects. In this particular case, the goal was to compare two messages in terms of priority. With this method a list of DiffMessages can be sorted, creating a new list of priority sorted DiffMessages. This object is represented in Figure 28.

4.2.3 *DiffMessageGenerator*

The DiffMessageGenerator handles how many messages are generated and with which properties. Following The One working flow, the existing default message generator had to be extended to fully take advantage of the new developed DiffMessage. Therefore, a new class was created (DiffMessageCreateEvent). This class randomly generates the messages for each node, also classifying each message with the intended scenario priority. Figure 28 represents this object. At the start of a simulation, the constructor for the DiffMessageGenerator is called to create every message. It is defined in the constructor the origin and destination of each message, its size and the time to live. Additionally, for the purpose of traffic differentiation in DTN, it is also here that the priority is defined, through the usage of random methods as explained in the beginning of this subsection.

4.2.4 *DiffMessagesStatsReport*

In order to generate reports for each simulation experiment, the existing stats report had to be adapted to work properly and measure correctly the new differentiated messages. For that reason the existing class MessageStatsReport was extended. The goal was to correctly analyse the new type of Message (DiffMessage) and have the necessary data to be compiled into tables and graphs describing the obtained differentiation results. For that reason twenty-one extra fields were added, and described in Table 6. Figure 28 contains a class of this object.

With these values being registered, it is possible to observe and analyse correctly the impact differentiation has on DTN and on the generated traffic.

4.2.5 *DiffRouters*

Every router needs to be adapted to comply with the considered traffic differentiation mechanisms. In The One Simulator implementations, the principal points are the messages constructors in each router. For that reason, each router has a corresponding differentiated version in which the methods to send and calculate the messages are override to use the newly created algorithm schemes. In order to better structure the algorithms, a new class was created (DiffMessageScheduler). Each router will calculate its messages through their own algorithms, like SW and PRoPHET, and then that list is passed through the DiffMessageScheduler for a final sending list. Figure 29 illustrate what methods are called in a node encounter, depending which algorithm is being executed. No new methods were added, the existing methods of each router that represent the process from which the router acts on each node encounter, where slightly modified to include the usage of the designed and developed algorithms. In SPO, the router fetches the messages from its original message list and passes through the

Table 6.: DiffMessageStatsReport generated values

Field		Description
created	High Medium Low	Number of created messages for each priority
delivered	High Medium Low	Number of created messages that were delivered to the final node for each priority
delivery probability	High Medium Low	The number of delivered messages over all the messages that were created for each priority
latency average	High Medium Low	The average latency measured between a message starting to be replicated and reaching the final done for each priority
latency median	High Medium Low	The median latency between a message starting to be replicated and reaching the final done for each priority
hopcount average	High Medium Low	The average hopcount between a message starting to be replicated and reaching the final done for each priority
hopcount median	High Medium Low	The median hopcount between a message starting to be replicated and reaching the final done for each priority

method that sorts messages according to their priority. A JAVA Comparator was used to sort all the messages. In WRRO, the router fetches the messages by feeding the original message list to the DiffScheduler class. This class will run the methods shown in Figure 28, to obtain an structured list, respecting the algorithm, which can be seen in Algorithm 2. In the WRRO variants, DCWRRO in SW and HYBWRRO in PRoPHET, the scheduler acts according to each algorithm, respecting each defined settings.

A routing protocol is embedded on each node, and follows a flow of methods to enable the message exchange. On each encounter, the node calls its own router class, depending on the selected router on the configuration, through the method Update. This method allows for calculations to take place, in order to prepare the messages to be delivered. The last method to be called is to send the messages to the node of current encounter. Each routing protocol, varies in this last method call. This last method call allows the node to send to specific nodes that are around it, or to all the nodes. It also allows the node to send all its messages, or a filtered list such as the ones provided by the algorithms defined in this thesis. This can be seen in Figure 30.

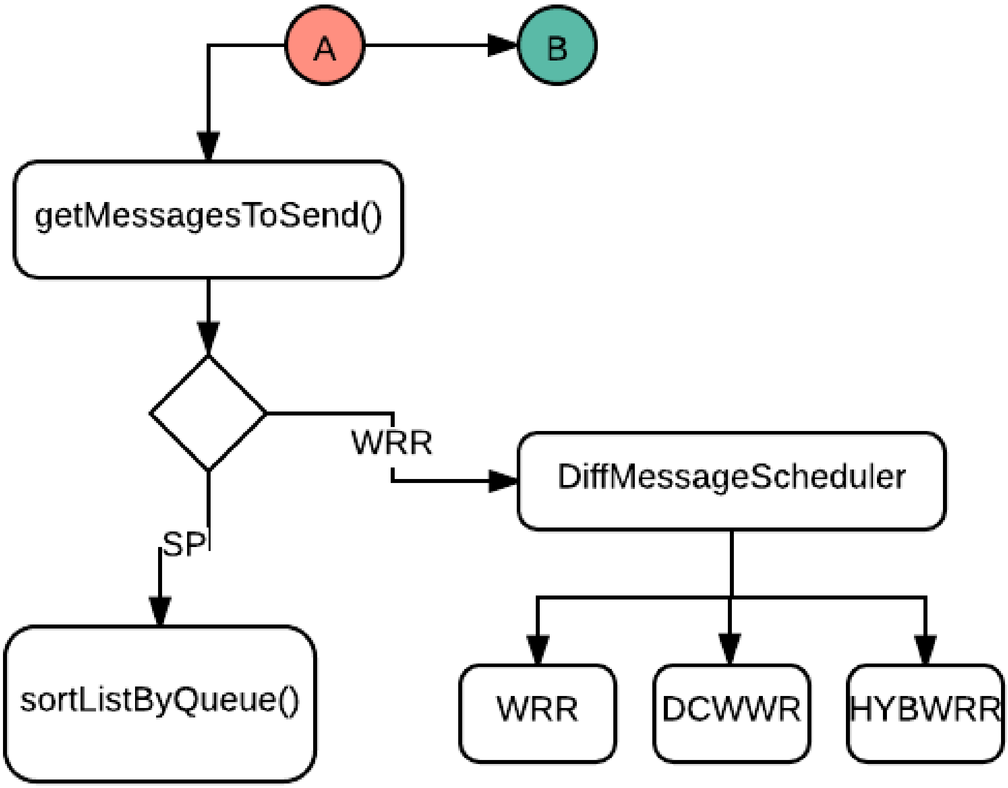


Figure 29.: Node encounter and methods called by routers in The One

4.2.6 *DiffMessageScheduler*

This class handles all the message sorting and organisation. It is here where the traffic scheduling alternatives are implemented. In the following subsections, the approaches and strategies used by the DiffMessagesScheduler, according to which router, are explained.

Epidemic

The differentiated version of Epidemic Router is the most basic of them all. Essentially, it just re-organises the randomly generated messages that the original Epidemic Router creates on each node, and applies the selected priority algorithms. The only two extensions needed are the SPO algorithm and the WRRO algorithm.

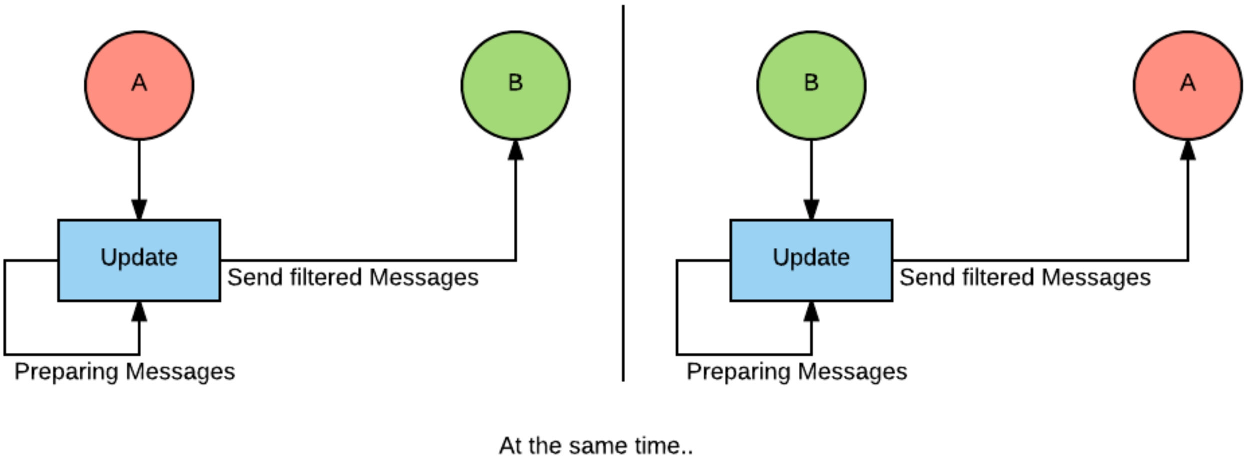


Figure 30.: Update logic on a node encounter in The One

Spray & Wait

Differently from Epidemic Router, SW needs more configuration and attention regarding the scheduler. Besides the common algorithms, such as the SPO and the WRRO, it needs to implement the DCWRRO algorithm. Each priority has a number of allowed copies, which are defined in DiffRouter class. Figure 31 shows the flow and actions of the differentiation version of SW. If the message has no more allowed copies, it ignores and does not send it. It is similar to a permanent discard for that simulation. That node will not send any more of those messages, until the end of the simulation.

PRoPHET

Lastly, PRoPHET also needs more customisation to confirm to the HYBWRRO algorithm implementation. Basically, it works with two filters: the first is obtaining the list of messages that actually have good probability to be delivered, which is inherent of its own base implementation, the second is applying the HYBWRRO algorithm, discarding messages or decreasing its priority. The mode used, discard or decrease, is previously configured in the configuration file of The One Simulator. Figure 32 shows how the implementation of a differentiated version of PRoPHET behaves in The One simulator. On each encounter, before sending the messages to the other node, it checks each message probability and priority, correcting or discarding according to the algorithm that is being used. Contrary to the SW version, which discard means no more copies of that message will be distributed, in PRoPHET the discard is encounter based, allowing the message to be sent on a different encounter, if the probability satisfies the requirements defined in the configuration.

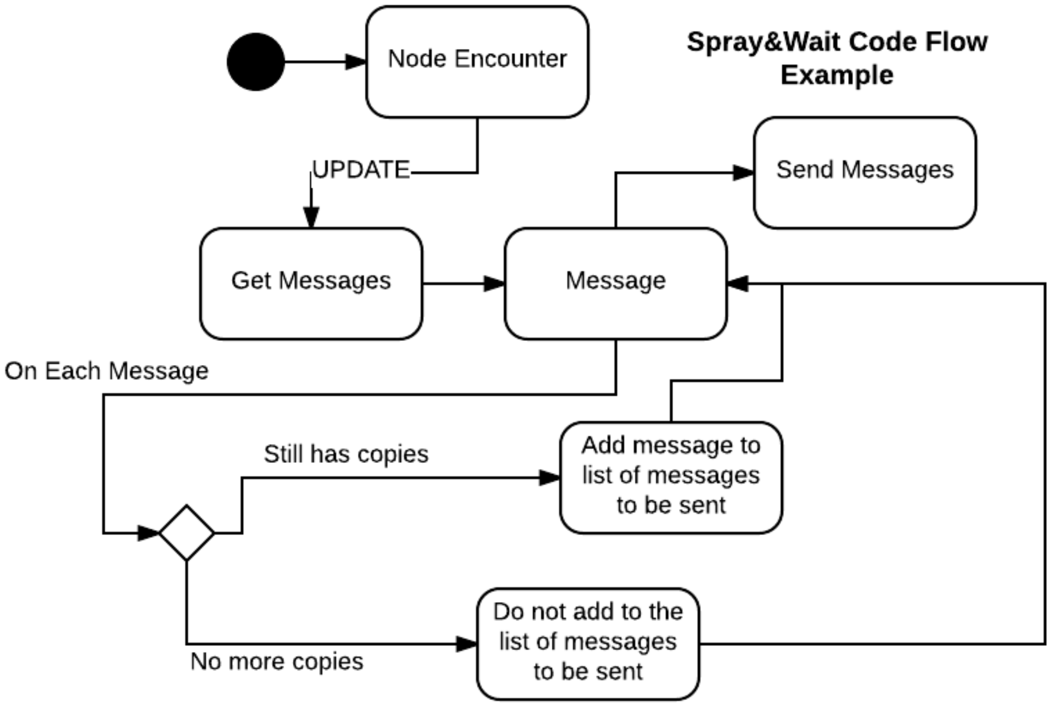


Figure 31.: Spray & Wait with differentiation behaviour in The One

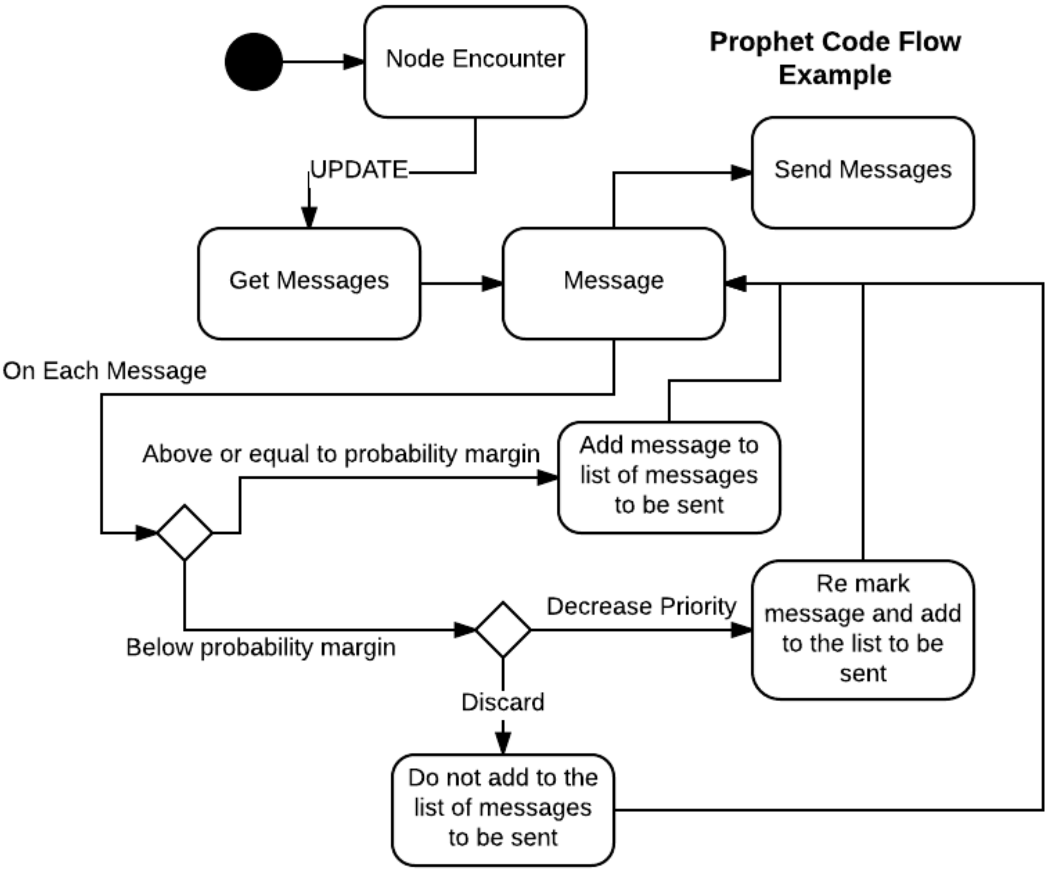


Figure 32.: PRoPHET with differentiation behaviour in The One

5

EXPERIMENTS AND RESULTS ANALYSIS

In this chapter the results of simulations and experiments are exposed and described. The simulator used is The One simulator, with some source modifications regarding three of its routers, Epidemic routing, SW routing and PRoPHET routing. The simulation scenario configuration and settings are based in Helsinki scenario of The One. This simulation scenario has two additional variants. These variations test the algorithms by increasing the number of generated messages and generated nodes. Every scenario variation, including the base scenario, has a corresponding chart. Each chart represents the delivered messages in one routing protocol, and it is divided in three or five panels, depending on the number of variations that a routing protocol has, when using differentiation. Each set of bars represents the average of 10 simulation runs combined, using different seeds in The One simulator.

5.1 THE PRIORITIES

The purpose of these priorities is to consider the different organisation messages, already existent in traffic differentiation of non delay tolerant networks, and find the possible applications of those priorities in DTNs. The selected group of priorities deals with different types of messages assigned with different types of priorities.

- High Priority Traffic - Considered the most important traffic flowing through the network;
- Medium Priority Traffic - Considered important information that must be distributed in a short period of time to remain useful;
- Low Priority Traffic - All the rest.

The messages were always marked with the respective priority, even thought there are scenarios where differentiation was not being applied to the traffic. This was used so results could be easily interpreted and discerned, when comparing to the differentiated messages. In every scenario the number of messages generated are equally divided between the existing

Table 7.: Modified values of The One Helsinki Config

Variable Name	Value
Scenario.endTime	43200
Events1.class	DiffMessageEventGenerator
Events1.prefix	[A;B;C;D;E;F;G;H;I;J]
MovementModel.rngSeed	[1;2;3;4;5;6;7;8;9;10]

priorities, so for the default scenario with 1440 messages, there are around 480 HP, 480 MP and 480 LP.

5.2 SCENARIOS

Additional scenarios for simulation purposes were also devised. These scenarios change certain variables, that will allow to understand how the differentiation implementation behaves in different conditions and if there is a differentiation at all. Each of these variables was individually manipulated.

The initial values of all this variables are based on the default settings of The One Simulator Helsinki scenario.

5.2.1 *Normal Scenario*

The normal scenario uses the stock values of The One simulator. Regarding number of messages, the original value generated is 1440. The original value of generated nodes can be checked in Table 8.

5.2.2 *Increased messages scenario*

In order to analyse how the differentiation algorithm implementations behave when scaling up the number of messages, a sub-scenario was created that doubles the number of generated messages. The norma scenario generated around 1440, equally divided between the three existing priorities. In this sub-scenario, 2880 messages are generated.

5.2.3 *Increased nodes scenario*

Following the manipulation of the total messages generated, a sub-scenario increased the number of nodes inside the system, could also shed some light regarding the algorithms behaviour when scaling up the number of nodes. Table 8 shows how this new sub-scenario node increase is distributed.

Table 8.: Number of nodes variation

Variable Name	**Original**	**Variation 1**
Group 1 (Pedestrians)	40	80
Group 2 (Cars)	40	80
Group 3 (More Pedestrians)	40	80
Group 4 (Trams/Buses)	2	4
Group 5 (Trams/Buses)	2	4
Group 6 (Trams/Buses)	2	4

Additional Scenarios

Additional scenarios were tested, where the number of increased messages and nodes were three times and four times greater. These results can be seen at the Appendix, in the last pages of this thesis.

5.2.4 *Router Specific scenarios*

As previously explained in the specification chapter, there was a need to introduce some variations to the newly developed SPO and WRRO traffic differentiation algorithms in DTN. Those were DCWRRO for SW and HYBWRRO for PRoPHET. These also spawn two different approaches each, that are relevant for testing purposes.

Spray & Wait Distinct Copies Weighted Round Robin Ordering

DCWRRO was the approached used in SW in order to better improve the traffic differentiation influence. This created a series of different variable scenarios while using this approach. The tables 2 and 3 in chapter 4, reflect these values and approaches.

PRoPHET Hybrid Weighted Round Robin Ordering

PRoPHET present a probability table when exchanging messages with nodes. In order to better use that existing table, HYBWRRO was introduced, creating a filter that could improve the delivery rate. This filter discards, or decreases priority of, messages if they do not meet a minimum probability value. The baseline for this filter to engage was 70%.

5.3 EPIDEMIC ROUTING

Epidemic routing is one of the standard routing implementations in The One. It uses the basic principle of distribution on every encounter. When running the simulation with the default version of Epidemic routing, messages were marked with priorities even thought traffic was not being differentiated. Each chart represents a scenario, and contains three panels, a normal default version, the SPO algorithm (Figure 20) algorithm and the WRRO algorithm (Figure 21). Each variation contains three bars that represent the priorities, defined by a single colour.

1. High Priority is represented by the colour red
2. Medium Priority is represented by the colour green
3. Low Priority is represented by the colour blue

5.3.1 *Normal scenario*

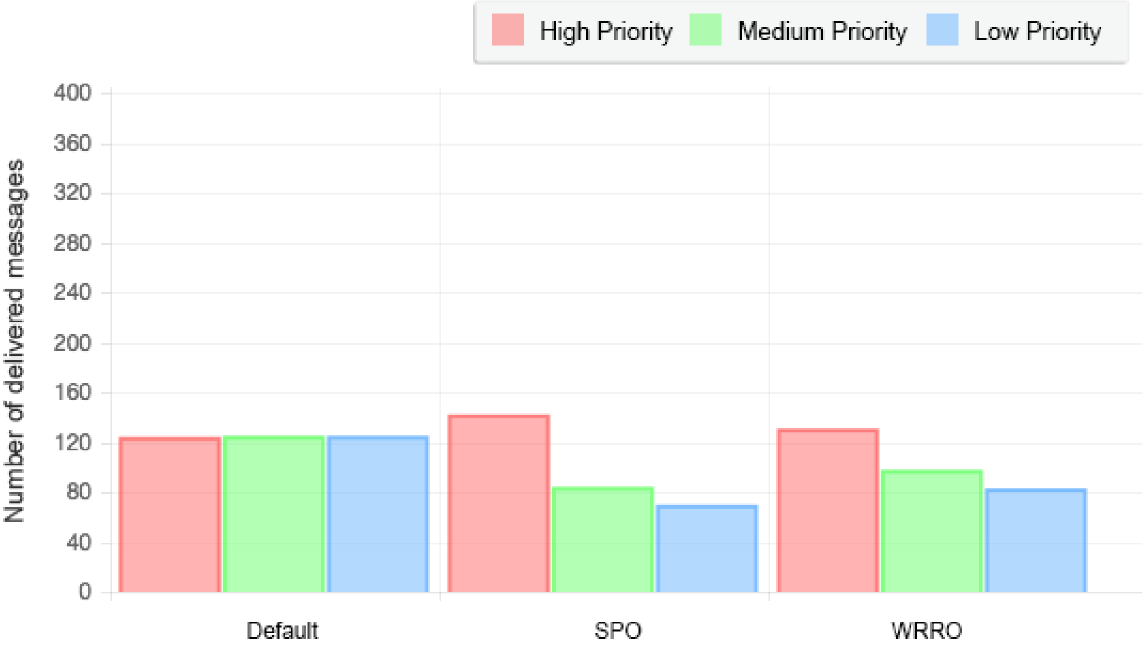


Figure 33.: Results for Epidemic Routing: Normal Scenario

In Figure 33 it is possible to see three panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node.

Panel - Epidemic

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - Epidemic Strict

This panel represents the number of delivered messages using SPO algorithm. This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. However, the total number of messages delivered has diminish, when comparing to the panel with no differentiation.

Panel - Epidemic WRRO

The last panel represents the number of delivered messages using WRRO algorithm. HP messages have a greater delivery rate when comparing to the MP and LP messages. However, following WRRO algorithm principle of avoiding starvation and allowing a more evenly distributed message delivery, both MP and LP have an increased number of messages delivered when comparing to the panel of the SPO algorithm. HP messages were also affected, with a reduced number of delivered messages, balancing the three priorities.

5.3.2 *Increased messages scenario*

Figure 34.: Results for Epidemic Routing: Increased number of messages

In Figure 34 it is possible to see three panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node. The number of generated messages was doubled, increasing from 1440 messages to 2880 messages.

Panel - Epidemic

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height, but have more delivered messages, which is a consequence of the increase in the number of created messages. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - Epidemic Strict

This panel represents the number of delivered messages using SPO algorithm. This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. However, the total number of messages delivered has diminish, when comparing to the panel with no differentiation.

Panel - Epidemic WRRO

The last panel represents the number of delivered messages using WRRO algorithm. HP messages have a greater delivery rate when comparing to the MP and LP messages. However, following WRRO algorithm principle of avoiding starvation and allowing a more evenly distributed message delivery, both MP and LP have an increased number of messages delivered when comparing to the panel of the SPO algorithm. HP messages were also affected, with a reduced number of delivered messages, balancing the three priorities.

5.3.3 *Increased nodes scenario*

In Figure 35 it is possible to see three panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node. The number of nodes in the scenario was doubled. The Table 8, in column Variation 1, specifies how the node number was increased and distributed.

Panel - Epidemic

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height, but have less delivered

Figure 35.: Results for Epidemic Routing: Increased number of nodes

messages, which is a consequence of the increase in the number of nodes, that do not create more messages but generate more copies flooding even more the network. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - Epidemic Strict

This panel represents the number of delivered messages using SPO algorithm. This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. However, the total number of messages delivered has diminish, when comparing to the panel with no differentiation.

Panel - Epidemic WRRO

The last panel represents the number of delivered messages using WRRO algorithm. HP messages have a greater delivery rate when comparing to the MP and LP messages. However, following WRRO algorithm principle of avoiding starvation and allowing a more evenly distributed message delivery, both MP and LP have an increased number of messages delivered when comparing to the panel of the SPO algorithm. HP messages were also affected, with a reduced number of delivered messages, balancing the three priorities.

5.3.4 *Summary*

The experiments and simulations began with Epidemic routing protocol, running all three scenario variants (normal, increased messages, increased nodes) combined with the normal, SPO algorithm and WRRO algorithm. In the normal scenario, see in Figure 33, it is possible to see the results of applying differentiation. In the first panel, with no differentiation algorithm, all three priorities have the same number of delivered messages. This panel serves as a reference when comparing with other panels that differentiation. Second and third panel in Figure 33, refer to SPO and WRRO algorithms approaches to this scenario in Epidemic routing. It is possible to see in both differentiation taking action, since HP messages are delivered in greater number when compared to MP and LP messages. Comparing between both algorithms, the WRRO algorithm actually shows greater balance between all the priorities, allowing MP and LP to be more equally delivered. What is noticeable in Figure 33 is that applying the differentiation algorithms, although increasing the delivery of HP which is the main goal of the algorithm, it decreases the total messages delivered, regardless of the priority. This result is reasonable, since HP messages are being tried to be delivery every time. This result was maintained after the increased messages and increased nodes, showing that the algorithm fulfils its function, in these conditions.

5.4 SPRAY & WAIT ROUTING

SW routing is based on Epidemic Routing. The difference is that SW limits the number of replication of a given message, avoid the network to be overflown with messages and replications. When running the simulation with the default version of SW routing, messages were marked with priorities even thought traffic was not being differentiated. Each chart represents a scenario, and contains five panels, a normal default version, the SPO algorithm (Figure 20), the WRRO algorithm (Figure 21) and two variations of the DCWRRO algorithm. Each variation contains three bars that represent the priorities, defined by a single colour. SW has two versions of operation, normal and binary, which will be divided in sections. The configuration used for SW can be verified in Table 3.

1. High Priority is represented by the colour red
2. Medium Priority is represented by the colour green
3. Low Priority is represented by the colour blue

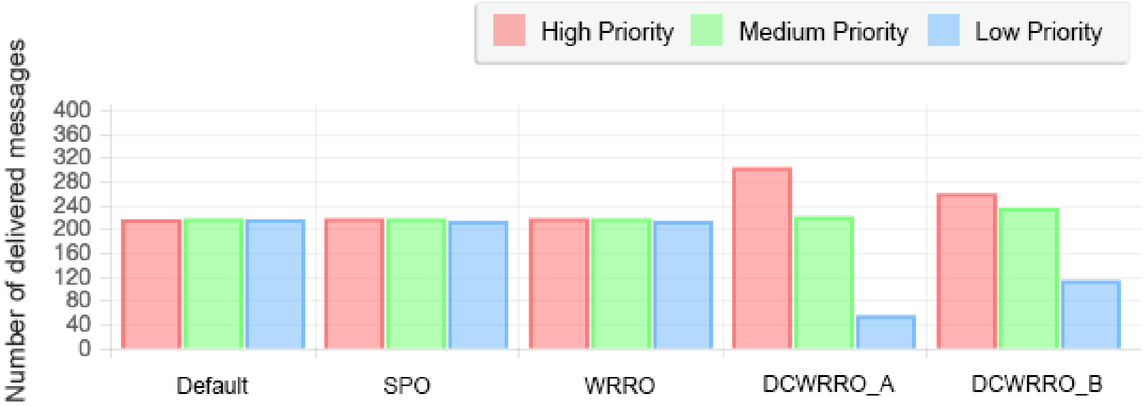


Figure 36.: Results for Spray & Wait Routing: Normal Scenario

5.4.1 *Normal scenario - Spray & Wait Normal Version*

In Figure 36 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node.

Panel - Spray & Wait

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - Spray & Wait Strict

This panel represents the number of delivered messages using SPO algorithm. This panel is similar to the version with no differentiation, since all the bars have approximately the same height.

Panel - Spray & Wait WRRO

The third panel represents the number of delivered messages using WRRO algorithm. Again, all the bars have the same height, which means that all the messages, regardless of their priority, are delivered in equal form.

Panel - Spray & Wait DCWRRO - A

The fourth panel represents the number of delivered messages using DCWRRO algorithm, version A. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The difference between the bars is more accentuated in the LP messages bar, when comparing to the HP and MP bars, represented by the red and green colour respectively. The total number of delivered messages is lower than the previous panels.

Panel - Spray & Wait DCWRRO - B

The fifth and last panel represents the number of delivered messages using DCWRRO algorithm, version B. In this panel, HP messages have a greater delivery rate than MP messages. MP messages also have a greater delivery rate when comparing to LP messages. The bars are more balanced between each other and the total delivered messages is higher than the fourth panel, but it is still lower than the first to third panels.

5.4.2 *Increased messages scenario - Spray & Wait Normal Version*

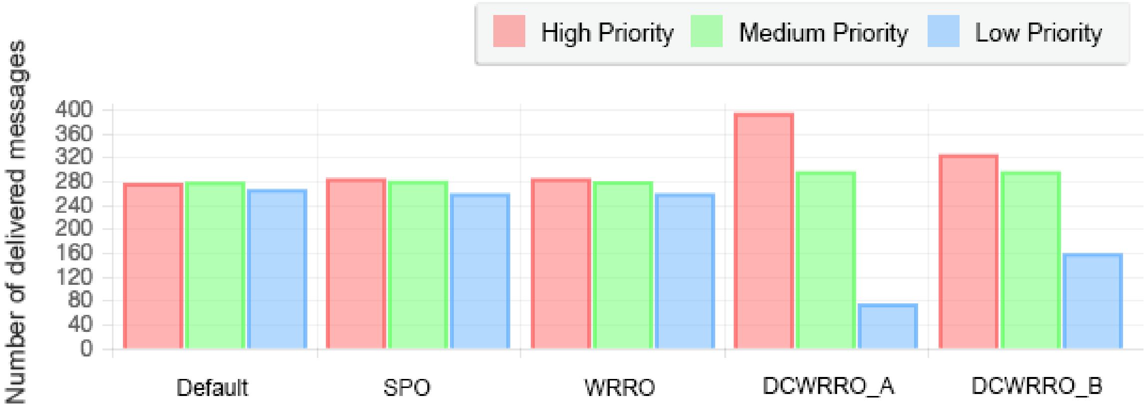


Figure 37.: Results for Spray & Wait Routing: Increased number of messages

In Figure 37 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node. The number of generated messages was doubled, increasing from 1440 messages to 2880 messages.

Panel - Spray & Wait

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms. The increase in generated messages is noticeable since the total number of delivered messages has also increased.

Panel - Spray & Wait Strict

This panel represents the number of delivered messages using SPO algorithm. This panel is similar to the version with no differentiation, since all the bars have approximately the same height.

Panel - Spray & Wait WRRO

The third panel represents the number of delivered messages using WRRO algorithm. Again, all the bars have the same height, which means that all the messages, regardless of their priority, are delivered in equal form.

Panel - Spray & Wait DCWRRO - A

The fourth panel represents the number of delivered messages using DCWRRO algorithm, version A. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The difference between the bars is more accentuated in the LP messages bar, when comparing to the HP and MP bars, represented by the red and green colour respectively. The total number of delivered messages is lower than the previous panels.

Panel - Spray & Wait DCWRRO - B

The fifth and last panel represents the number of delivered messages using DCWRRO algorithm, version B. In this panel, HP messages have a greater delivery rate than MP messages. MP messages also have a greater delivery rate when comparing to LP messages. The bars are more balanced between each other and the total delivered messages is very similar when comparing with the fourth panel, but it is still lower than the first to third panels.

5.4.3 *Increased nodes scenario - Spray & Wait Normal Version*

In Figure 38 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that

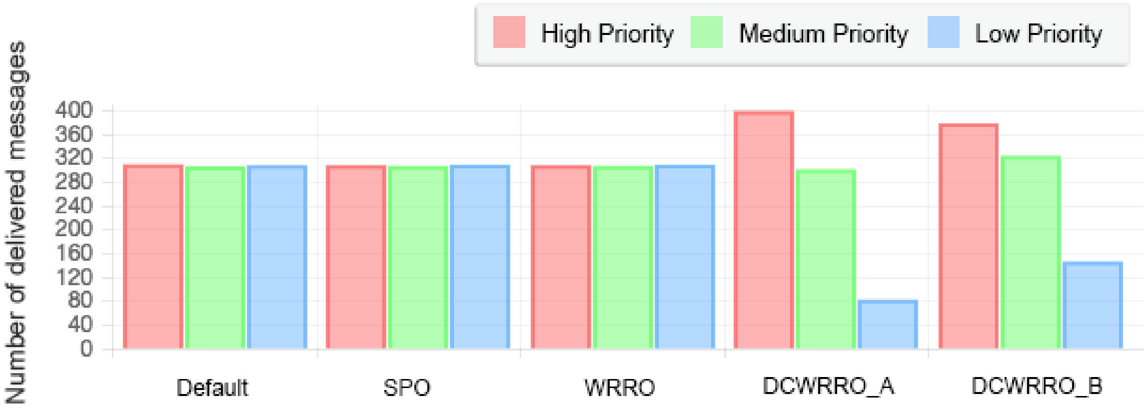


Figure 38.: Results for Spray & Wait Routing: Increased number of nodes

were successfully generated and delivered to its intended destination node. The number of nodes in the scenario was doubled. The Table 8, in column Variation 1, specifies how the node number was increased and distributed.

Panel - Spray & Wait

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - Spray & Wait Strict

This panel represents the number of delivered messages using SPO algorithm. This panel is similar to the version with no differentiation, since all the bars have approximately the same height.

Panel - Spray & Wait WRRO

The third panel represents the number of delivered messages using WRRO algorithm. Again, all the bars have the same height, which means that all the messages, regardless of their priority, are delivered in equal form.

Panel - Spray & Wait DCWRRO - A

The fourth panel represents the number of delivered messages using DCWRRO algorithm, version A. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The difference between the bars is more accentuated in the LP messages bar, when comparing to the HP and MP bars, represented by the red and green colour respectively. The total number of delivered messages is lower than the previous panels.

Panel - Spray & Wait DCWRRO - B

The fifth and last panel represents the number of delivered messages using DCWRRO algorithm, version B. In this panel, HP messages have a greater delivery rate than MP messages. MP messages also have a greater delivery rate when comparing to LP messages. The bars are more balanced between each other and the total delivered messages is higher when comparing with the fourth panel, but it is still lower than the first to third panels.

5.4.4 *Normal scenario - Spray & Wait Binary Version*

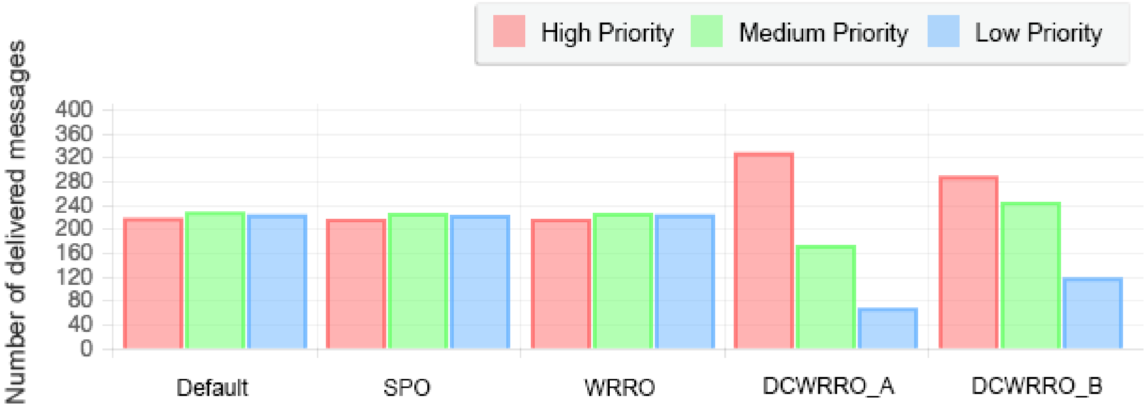


Figure 39.: Results for Spray & Wait Routing: Normal Scenario

In Figure 39 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node.

Panel - Spray & Wait

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - Spray & Wait Strict

This panel represents the number of delivered messages using SPO algorithm. This panel is similar to the version with no differentiation, since all the bars have approximately the same height.

Panel - Spray & Wait WRRO

The third panel represents the number of delivered messages using WRRO algorithm. Again, all the bars have the same height, which means that all the messages, regardless of their priority, are delivered in equal form.

Panel - Spray & Wait DCWRRO - A

The fourth panel represents the number of delivered messages using DCWRRO algorithm, version A. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The difference between the bars is more accentuated in the LP messages bar, when comparing to the HP and MP bars, represented by the red and green colour respectively. The total number of delivered messages is lower than the previous panels.

Panel - Spray & Wait DCWRRO - B

The fifth and last panel represents the number of delivered messages using DCWRRO algorithm, version B. In this panel, HP messages have a greater delivery rate than MP messages. MP messages also have a greater delivery rate when comparing to LP messages. The bars are more balanced between each other and the total delivered messages is higher than the fourth panel, but it is still lower than the first to third panels.

5.4.5 *Increased messages scenario - Spray & Wait Binary Version*

In Figure 40 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node. The number of generated messages was doubled, increasing from 1440 messages to 2880 messages.

Panel - Spray & Wait

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a

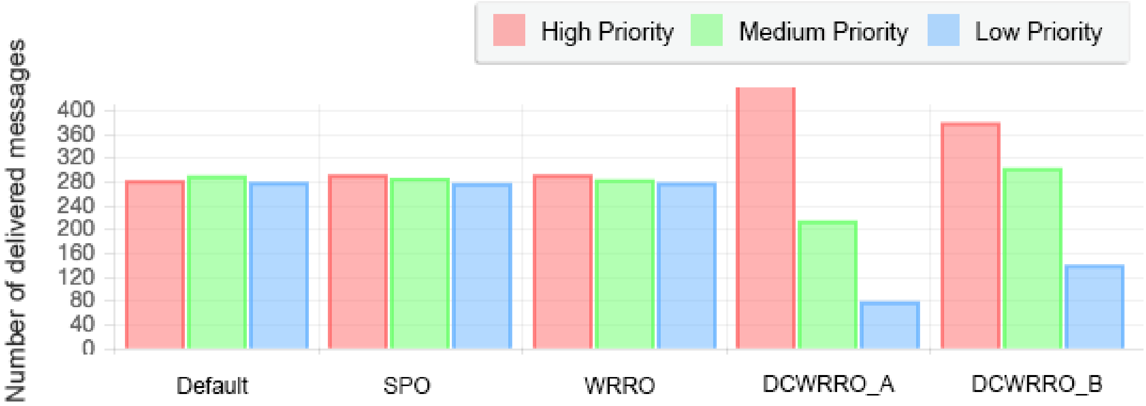


Figure 40.: Results for Spray & Wait Routing: Increased number of messages

base model to compare with the other panels that are using differentiation algorithms. The increase in generated messages is noticeable since the total number of delivered messages has also increased.

Panel - Spray & Wait Strict

This panel represents the number of delivered messages using SPO algorithm. This panel is similar to the version with no differentiation, since all the bars have approximately the same height.

Panel - Spray & Wait WRRO

The third panel represents the number of delivered messages using WRRO algorithm. Again, all the bars have the same height, which means that all the messages, regardless of their priority, are delivered in equal form.

Panel - Spray & Wait DCWRRO - A

The fourth panel represents the number of delivered messages using DCWRRO algorithm, version A. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The difference between the bars is more accentuated in the LP messages bar, when comparing to the HP and MP bars, represented by the red and green colour respectively. The total number of delivered messages is lower than the previous panels.

Panel - Spray & Wait DCWRRO - B

The fifth and last panel represents the number of delivered messages using DCWRRO algorithm, version B. In this panel, HP messages have a greater delivery rate than MP messages. MP messages also have a greater delivery rate when comparing to LP messages. The bars are more balanced between each other and the total delivered messages is very similar when comparing with the fourth panel, but it is still lower than the first to third panels.

5.4.6 *Increased nodes scenario - Spray & Wait Binary Version*

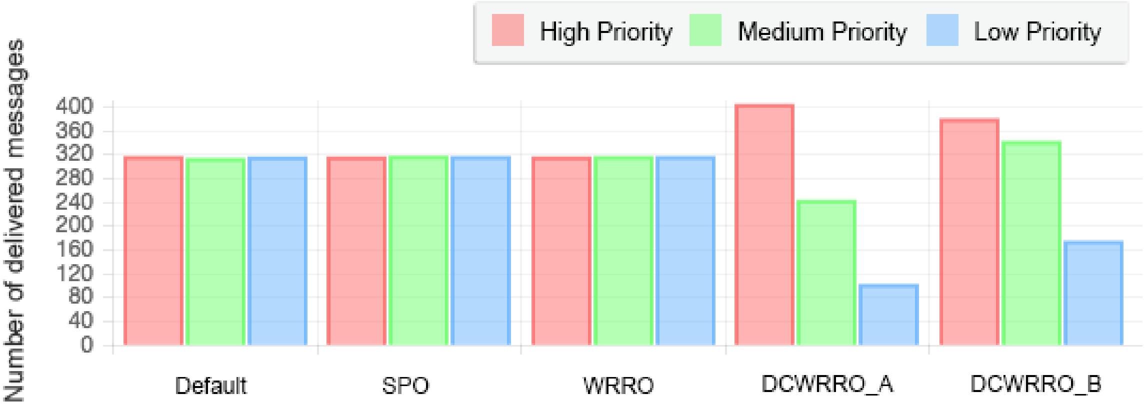


Figure 41.: Results for Spray & Wait Routing: Increased number of nodes)

In Figure 41 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node. The number of nodes in the scenario was doubled. The Table 8, in column Variation 1, specifies how the node number was increased and distributed.

Panel - Spray & Wait

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - Spray & Wait Strict

This panel represents the number of delivered messages using SPO algorithm. This panel is similar to the version with no differentiation, since all the bars have approximately the same height.

Panel - Spray & Wait WRRO

The third panel represents the number of delivered messages using WRRO algorithm. Again, all the bars have the same height, which means that all the messages, regardless of their priority, are delivered in equal form.

Panel - Spray & Wait DCWRRO - A

The fourth panel represents the number of delivered messages using DCWRRO algorithm, version A. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The difference between the bars is more accentuated in the LP messages bar, when comparing to the HP and MP bars, represented by the red and green colour respectively. The total number of delivered messages is lower than the previous panels.

Panel - Spray & Wait DCWRRO - B

The fifth and last panel represents the number of delivered messages using DCWRRO algorithm, version B. In this panel, HP messages have a greater delivery rate than MP messages. MP messages also have a greater delivery rate when comparing to LP messages. The bars are more balanced between each other and the total delivered messages is higher when comparing with the fourth panel, but it is still lower than the first to third panels.

5.4.7 *Summary*

The second group of simulations were with the SW routing protocol, and it was divided in the two SW modes, normal and binary. Following the same structure as the Epidemic routing experiments and variants, SW had one more algorithm to be applied, DCWRRO. Both normal and binary version behaved similarly. In Figure 36, the first panel represents a simulation with no differentiation, while the second and third panels have SPO and WRRO applied respectively. There is no impact, in terms of differentiation, in the first three panels. Although in the first panel this is a normal behaviour, the other two panels are affected by SW n-copies limit, not allowing the algorithms to influence the routing, just as predicted when the algorithms were devised in chapter 3. The last panels represent the WRRO algorithm taking in account these n-copies limit changes, labeled DCWRRO. In the fourth and fifth panels, due to the influence of DCWRRO in the number of copies allowed for each message, differentiation takes place and it is possible to visualise HP messages having an increased number of delivered messages. The number of copies allowed is actually very important, since in the fifth panel, MP and LP have a greater number of copies allowed than in the fourth panel, creating a more evenly message distribution. As in what happened in Epidemic routing, the scenarios with increased messages and nodes, follow the same results and interpretations.

5.5 PROPHET ROUTING

PRoPHET routing protocol uses a probabilistic encounter table, to known if a node has a good probability of delivering that message to its intended destination. e simulation with the default version of PRoPHET routing, messages were marked with priorities even thought traffic was not being differentiated. Each chart represents a scenario, and contains five panels, a normal default version, the SPO algorithm (Figure 20), the WRRO algorithm (Figure 21) and two variations of the HYBWRRO algorithm. Each variation contains three bars that represent the priorities, defined by a single colour.

1. High Priority is represented by the colour red
2. Medium Priority is represented by the colour green
3. Low Priority is represented by the colour blue

5.5.1 *Normal scenario - PRoPHET*

In Figure 42 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node.

Panel - PRoPHET

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - PRoPHET Strict

This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. The total number of delivered messages has decreased when compared to the first panel, with no differentiation.

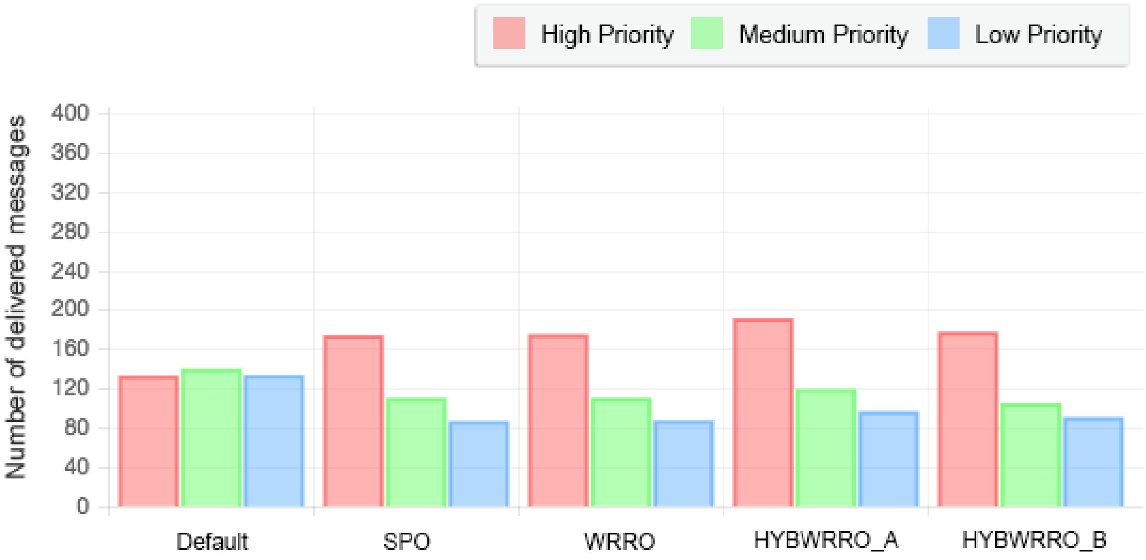


Figure 42.: Results for PRoPHET Routing: Normal Scenario

Panel - PRoPHET WRRO

This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. The total number of delivered messages is similar when compared to the second panel, using SPO algorithm.

Panel - PRoPHET HYBWRRO - Discard Priority

The fourth panel represents the number of delivered messages using HYBWRRO algorithm, discard version. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The total number of delivered messages has increased when comparing to the previous panels.

Panel - PRoPHETHYBWRRO - Decrease Priority

The fifth and last panel represents the number of delivered messages using HYBWRRO algorithm, decreased priority version. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The total number of delivered messages has decreased when comparing to the previous panels.

5.5.2 *Increased messages scenario - PRoPHET*

In Figure 43 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that

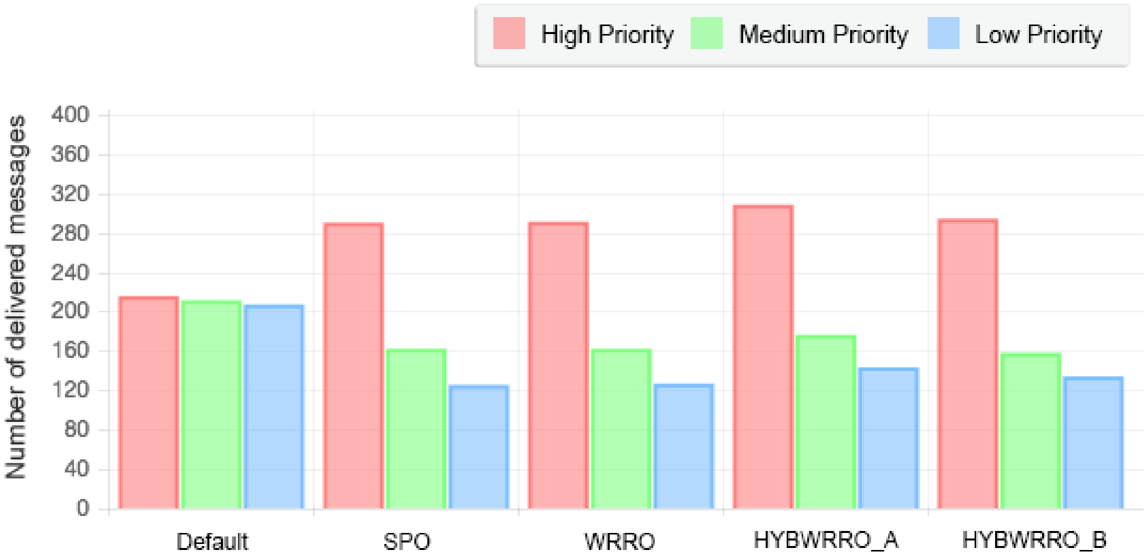


Figure 43.: Results for PRoPHET Routing: Increased number of messages

were successfully generated and delivered to its intended destination node. The number of generated messages was doubled, increasing from 1440 messages to 2880 messages.

Panel - PRoPHET

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - PRoPHET Strict

This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. The total number of delivered messages has decreased when compared to the first panel, with no differentiation.

Panel - PRoPHET WRRO

This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. The total number of delivered messages is similar when compared to the second panel, using SPO algorithm.

Panel - PRoPHET HYBWRRO - Discard Priority

The fourth panel represents the number of delivered messages using HYBWRRO algorithm, discard version. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The total number of delivered messages has increased when comparing to the second and third panels. The gap between the HP messages and the other priorities is almost twice more messages delivered.

Panel - PRoPHETHYBWRRO - Decrease Priority

The fifth and last panel represents the number of delivered messages using HYBWRRO algorithm, decreased priority version. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The total number of delivered messages has decreased when comparing to the previous panels. The gap between the HP messages and the other priorities is almost twice more messages delivered.

5.5.3 *Increased nodes scenario - PRoPHET*

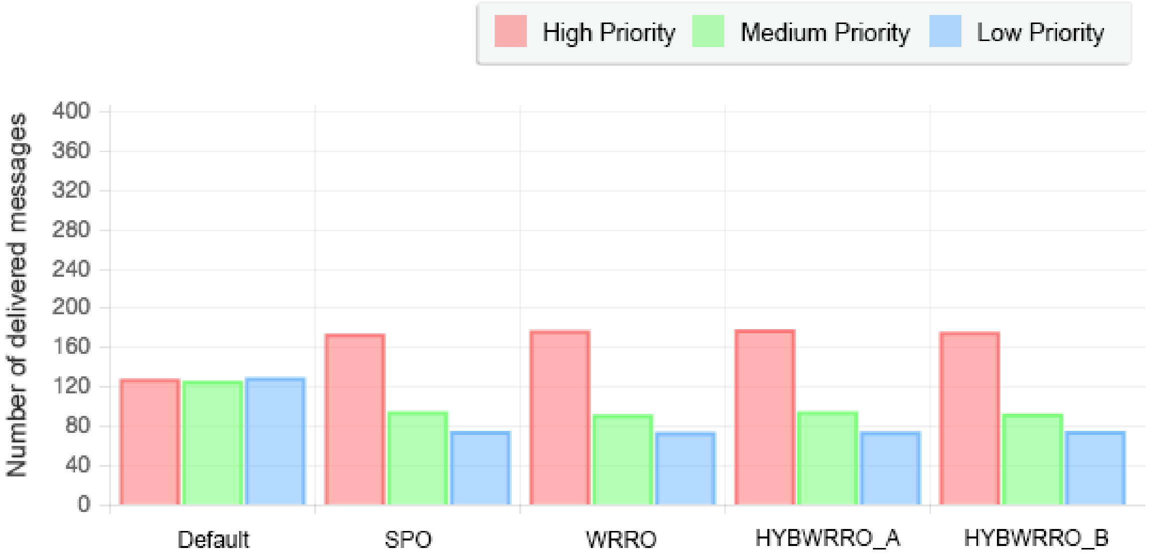


Figure 44.: Results for PRoPHET Routing: Increased number of nodes

In Figure 44 it is possible to see five panels, which will be described in the following sections. These charts represent the total number of delivered messages, which indicates messages that were successfully generated and delivered to its intended destination node. The number of nodes in the scenario was doubled. The Table 8, in column Variation 1, specifies how the node number was increased and distributed.

Panel - PRoPHET

This panel represents a simulation of the current scenario with no differentiation. This is confirmed by the three bars, that represent messages marked with a priority, have the same number of delivered messages. All the bars have the same height. This panel serves as a base model to compare with the other panels that are using differentiation algorithms.

Panel - PRoPHET Strict

This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. The total number of delivered messages has decreased when compared to the first panel, with no differentiation.

Panel - PRoPHET WRRO

This panel shows that the red bar, HP messages, has more clearly more messages delivered when comparing to the green, MP messages, and blue, LP messages. The total number of delivered messages is similar when compared to the second panel, using SPO algorithm.

Panel - PRoPHET HYBWRRO - Discard Priority

The fourth panel represents the number of delivered messages using HYBWRRO algorithm, discard version. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The total number of delivered messages has increased when comparing to the second and third panels.

Panel - PRoPHETHYBWRRO - Decrease Priority

The fifth and last panel represents the number of delivered messages using HYBWRRO algorithm, decreased priority version. In this panel, the red bar, HP messages, is higher than the green bar, MP messages, and green bar is higher than the blue bar, LP messages. The total number of delivered messages has decreased when comparing to the previous panels.

5.5.4 *Summary*

PRoPHET was the routing protocol used in the the last group of simulations. Following the same structure of the previous routing protocols, an additional algorithm was added to PRoPHET, that take account its probabilistic table, HYBWRRO. This algorithm has two variations when handling a message that does not meet the requirements regarding the minimum probability. Those messages are either discarded or have their priority decreased. In the first panel, of Figure 42, the simulation was executed with no differentiation algorithm, resulting in all the priorities have the same number of messages delivered. The next panel, represents the SPO algorithm, where differentiation takes place influencing the delivered messages. It is possible to see HP with a higher rate of delivered messages when compared to lower priorities. WRRO algorithm, in the third panel, allows a more evenly distribution of all the priorities. The fourth and fifth panels show how HYBWRRO acts, discarding (fourth panel) or decrease (fifth panel). HYBWRRO algortihm allows differentiation to take place, having HP message to have an increased delivery rate. The difference between discarding and decrease is clear, as discarding delivers a greater number of messages than when the message priority is decreased. When comparing to SPO and WRRO algorithms, HYBWRRO has more delivered messages in each priority, which means that it actually delivers more messages and while respecting the differentiation. Increasing the number of generated messages or increasing the number of nodes inside the simulation, yields similar results and conclusions.

6

CONCLUSION

The main objectives of this work were to explore and test traffic differentiation mechanisms in Delay Tolerant Networks (DTN). The main purpose was to obtain a better performance in traffic behaviour regarding message delivery rate. For that purpose a literature review was needed, in order to analyse the current state of both traffic differentiation, DTN and traffic differentiation in DTN. Along side the literature review followed a better understanding of the One simulator tool, for the simulations and experiments.

As the study through traffic differentiation in DTN progressed, the understanding of the routing protocols resulted in two distinct algorithms that worked into integrated those existing protocols, which spawned two distinct algorithms. These two distinct algorithms, Strict Priority Ordering (SPO) and Weighted Round Robin Ordering (WRRO), focused in message marking and forwarding to the next node according to its priorities. For the priorities, three levels were devised, High Priority (HP), Medium Priority (MP) and Low Priority (LP). The focus of these algorithms was to provide a great deal of importance to HP message, so they are delivered more than other messages with lower priorities. The routing protocols selected to implement the algorithms were the Epidemic routing protocol, Spray and Wait routing protocol (SW) and PRoPHET routing protocol. These protocols needed to be extended, in The One simulator, to work with traffic differentiation.

While studying these protocols, there were several predictable outcomes regarding their behaviour when following the algorithms. For that reason, two variants of WRRO were devised. One of those variants, Distinct Copies WRR (DCWRR), was extended from SW own distinct copies propriety, enabling a different count of allowed copies per priority. The other variant, Hybrid WRR (HYBWRR), was extended from PRoPHET existing probability tables, allowing those probabilities to take effect over the message priority. This variant, when a message probability is lower than a pre-established mark, discards the message from a node to node transaction or decreases the priority. The mode used, decreased priority or discard message, is pre-selected before the simulation.

With these algorithms in place, and using the base scenario of the The One simulator in Helsinki, the experiments and simulations were devised, giving light to three possible scenarios, the default with no changes to the original Helsinki configuration, an increased message variant and an increased node variant. The objective of these variants were to test the algorithms behaviour even with an increased message count or an increase node variant.

Traffic differentiation is possible in DTN (Garpal et al., 2012) (Park et al., 2014) (Sulma Rashid, 2014) (Liu et al., 2010) and in VDTN (Soares et al., 2010). Most approaches focus on buffer management and its size, while this approach focus on routing delivery and forwarding according to priority. The results obtained on the simulation show that traffic differentiation through prioritisation has an impact on message delivery rate. In all the simulations HP messages were delivery in greater number, while only adapting the current routing protocols to be compatible with differentiation. This adaptation does not change the architecture, structure or organisation of DTN, providing an easy and visible solution when implementing differentiation in DTN. For future work, the delays caused by differentiation should be tested, in order to provide a greater enlightenment in how fast the message is delivered when compared to non differentiation versions. Additionally, further testing the developed algorithms, SPO, WRRO and its variants, in different routers, would allow to see how they behave with differentiation and its viability.

BIBLIOGRAPHY

Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. Technical report, 1998. URL `https://tools.ietf.org/html/rfc2475#section-2.3.3.3`.

John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM*, volume 6, 2006.

Yue Cao and Zhili Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys & Tutorials, IEEE*, pages 654–677, 2012. doi: 10.1109/SURV.2012.042512.00053.

V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant networking architecture. RFC 4838, April 2007.

Ing-Ray Chen, Fenye Bao, Moonjeong Chang, and Jin-Hee Cho. Trust management for encounter-based routing in delay tolerant networks. *Global Telecommunications Conference (GLOBECOM 2010)*, pages 1–6, December 2010. doi: 10.1109/GLOCOM.2010.5683235. URL `http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5683235&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5682081%2F5683069%2F05683235.pdf%3Farnumber%3D5683235`.

Cisco. Cisco networks. http://www.cisco.com, Last visited on 11/12/2014 2014.

Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM Computer Communication Review*, volume 19, pages 1–12. ACM, 1989.

Kevin Fall. A delay-tolerant network architecture for challenged internets. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34, 2003. doi: 10.1145/863955.863960. URL `http://dl.acm.org/citation.cfm?id=863960`.

Animesha Garpal, Prof. Shweta Jain, and Dinesh Singh Baghel;. An analysis optimal buffer management policy to improve qos in dtn routing protocol. *International Journal of Computer Applications*, 48(5):11, June 2012.

J. P. Georges, T. Divoux, and E. Rondeau. Strict priority versus weighted fair queueing in switched ethernet networks for time critical applications. In *19th IEEE International*

Parallel and Distributed Processing Symposium, pages 141–141, April 2005. doi: 10.1109/IPDPS.2005.413.

Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. *SIGCOMM 04*, 34(4):145–158, August 2004.

Evan PC Jones, Lily Li, Jakub K Schmidtke, and Paul AS Ward. Practical routing in delay-tolerant networks. *Mobile Computing, IEEE Transactions on*, 6(8):943–959, 2007.

Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The one simulator for dtn protocol evaluation. *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 55:1 – 10, 2009. doi: 10.4108/ICST.SIMUTOOLS2009.5674. URL `http://dl.acm.org/citation.cfm?id=1537683`.

Sotirios-Angelos Lenas, Stylianos Dimitriou, Fani Tsapeli, and Vassilis Tsaoussidis. Queue-management architecture for delay tolerant networking. *WWIC'11 Proceedings of the 9th IFIP TC 6 international conference on Wired/wireless internet communications*, pages 470–482, 2011. URL `http://dl.acm.org/citation.cfm?id=2023141`.

A. Lindgren and A. Doria. Experiences from deploying a real-life dtn system. In *2007 4th IEEE Consumer Communications and Networking Conference*, Jan 2007. URL `10.1109/CCNC.2007.50`.

Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003. doi: 10.1145/961268.961272. URL `http://doi.acm.org/10.1145/961268.961272`.

R. Litjens and R. W. Hendriks. The impact of qos differentiation on service quality and system capacity in cellular networks. *Telecommunication Systems*, 55(2):281–298, 2014. doi: 10.1007/s11235-013-9781-3. URL `http://dx.doi.org/10.1007/s11235-013-9781-3`.

Hengchang Liu, Aravind Srinivasan, Kamin Whitehouse, and John A. Stankovic. Melange: Supporting heterogeneous qos requirements in delay tolerant sensor networks. *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 93–96, June 2010. doi: 10.1109/INSS.2010.5573277. URL `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5573277`.

University Missouri-Rolla. Delay tolerant networks. Website, May 2006. URL `http://web.mst.edu/~mobildat/DTN/index.html`.

Cisco Networks. Comparing traffic policing and traffic shaping for bandwidth limiting. http://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html, May 2014. URL `http://www.cisco.com/c/en/`

us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html#trafficshaping.

K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. RFC 2474, Network Working Group, December 1998.

Hanjin Park, Yusung Kim, Euiyul Ko, and Ikjun Yeom. Packet forwarding assurance in delay tolerant networks. *Wireless Personal Communications*, 82(1):353–376, 2014. ISSN 1572-834X. doi: 10.1007/s11277-014-2212-2. URL http://dx.doi.org/10.1007/s11277-014-2212-2.

CCITT Recommendation. I. 371: Traffic control and congestion control in b-isdn. *Geneva, Study Group*, 13, 1992. URL http://www.itu.int/rec/T-REC-I.371-200403-I/en.

K. Scott and S. Burleigh. Bundle protocol specification. RFC 5050, NASA Jet Propulsion Laboratory, November 2007.

Chuck Semeria. Supporting differentiated service classes: Queue scheduling disciplines. *White paper*, 2001. Available at: url(www.juniper.net/solutions/literature/white_papers/200020.pdf).

Vasco N. G. J. Soares, Farid Farahmand, and Joel J. P. C. Rodrigues. Traffic differentiation support in vehicular delay- tolerant networks. *Telecommunication Systems*, 48(1-2):151–162, October 2010. doi: 10.1007/s11235-010-9325-z. URL http://link.springer.com/article/10.1007%2Fs11235-010-9325-z.

Chris Soghoian. Is comcast's bittorrent filtering violating the law? https://www.cnet.com/news/is-comcasts-bittorrent-filtering-violating-the-law/, September 2007. URL https://www.cnet.com/news/is-comcasts-bittorrent-filtering-violating-the-law/.

T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, pages 235–244, October 2004. doi: 10.1109/SAHCN.2004.1381922.

Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. *SIGCOMM 05 Workshops*, August 2005.

Abdul Hanan Abdullah Sulma Rashid, Qaisar Ayub. Reactive weight based buffer management policy for dtn routing protocols. *Wireless Personal Communications*, 80(3): 993–1010, 2014. doi: 10.1007/s11277-014-2066-7. URL http://dx.doi.org/10.1007/s11277-014-2066-7. TBA.

Hannes Tschofenig, Franz Muenz, and Murugaraj Shanmugam. Taxonomy of dtn scenarios. Available at url(http://wiki.n4c.eu/wiki/images/2/21/DTN_scenarios.pdf), 2008.

Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. *Technical Report CS-200006*, April 2000. doi: 10.1.1.34.6151.

A

SUPPORT MATERIAL

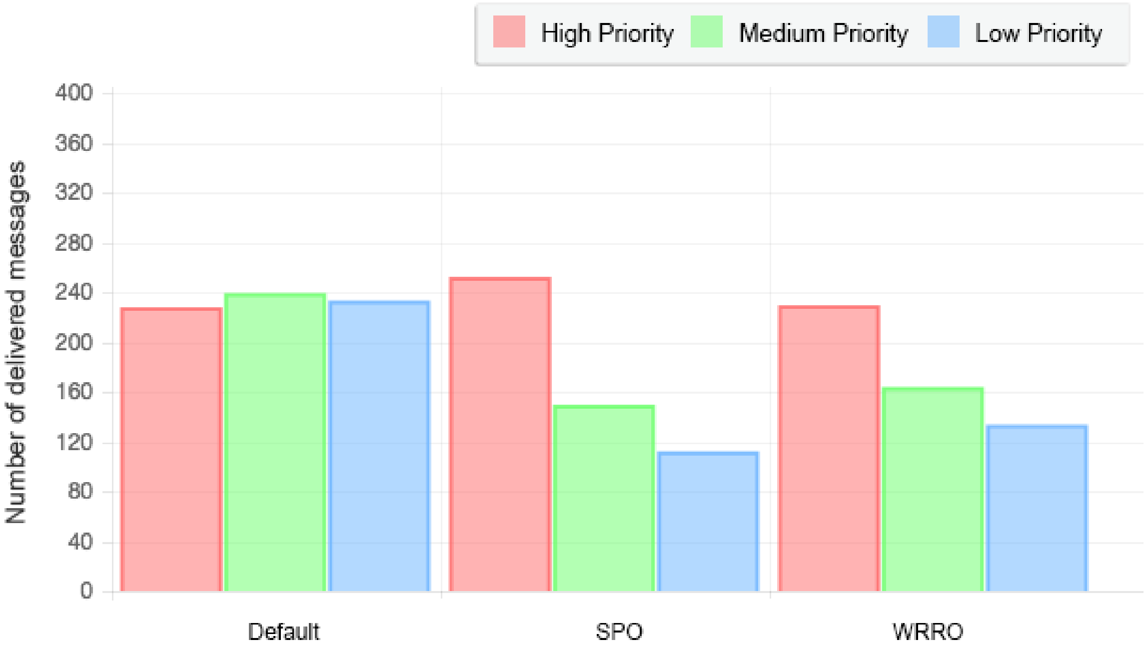


Figure 45.: Epidemic: Increased number of messages (4320)

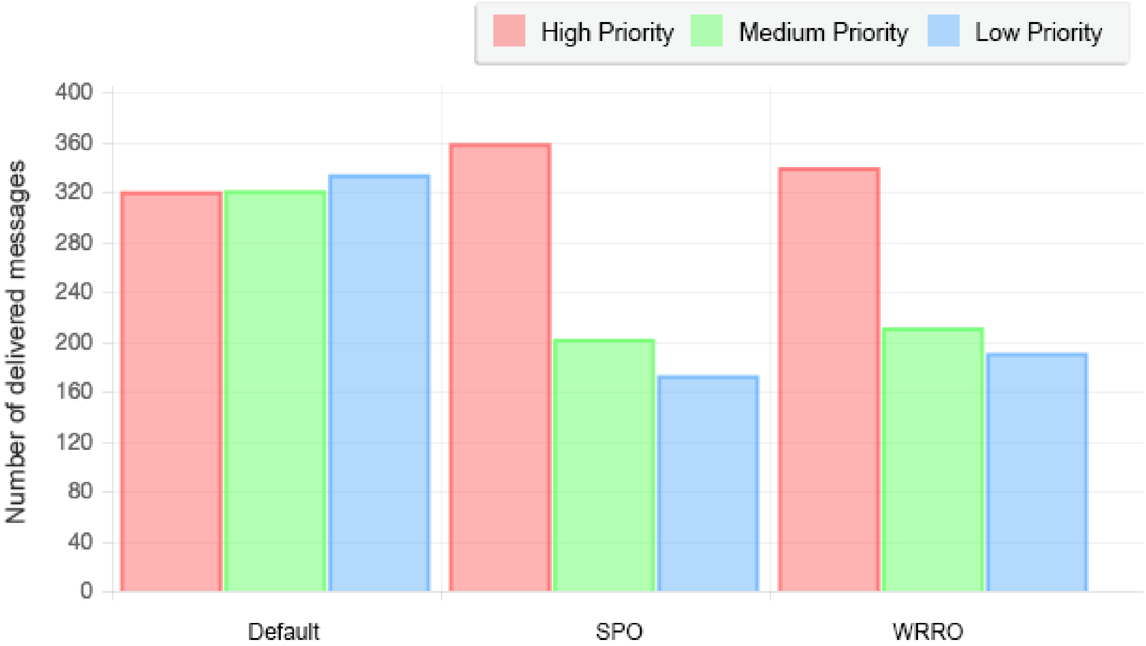


Figure 46.: Epidemic: Increased number of messages (8640)

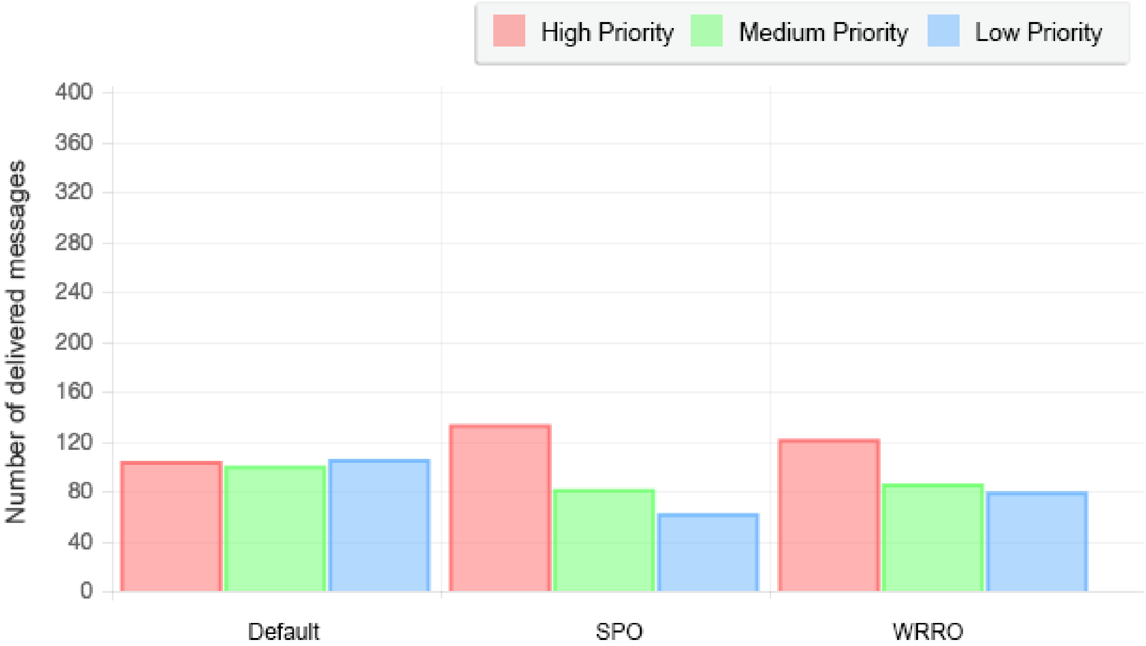


Figure 47.: Epidemic: Increased number of nodes (3x)

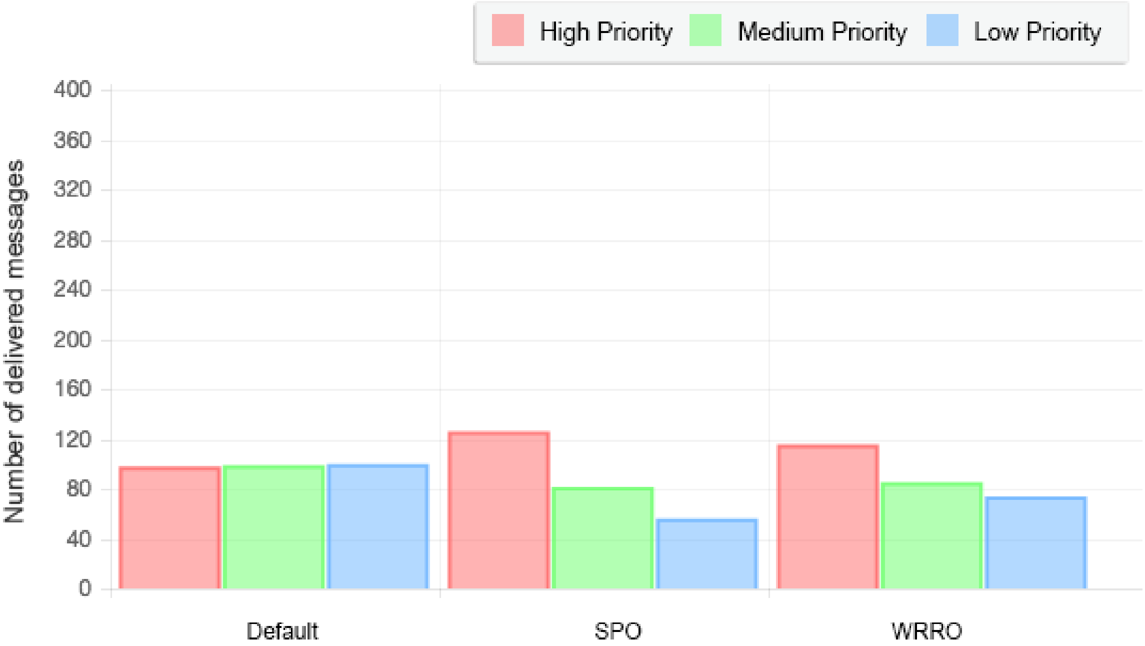


Figure 48.: Epidemic: Increased number of nodes (4x)

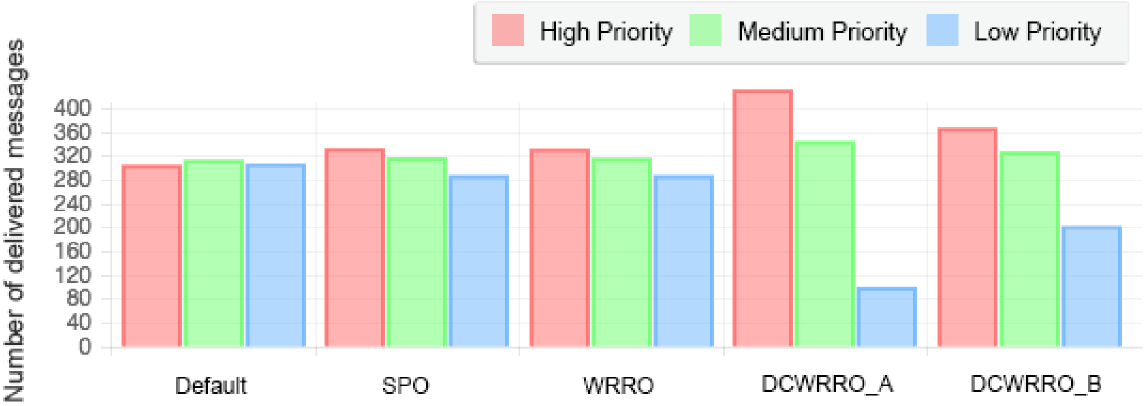


Figure 49.: SW: Increased number of messages (4320)

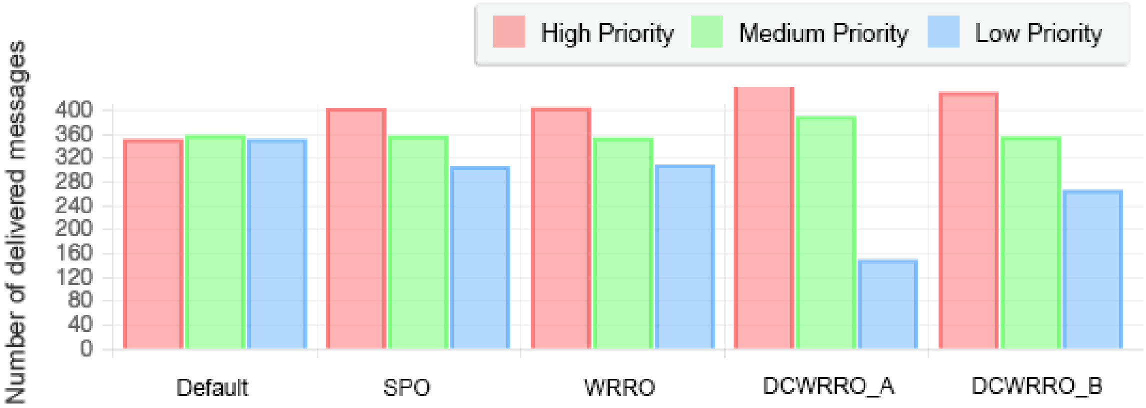


Figure 50.: SW: Increased number of messages (8640)

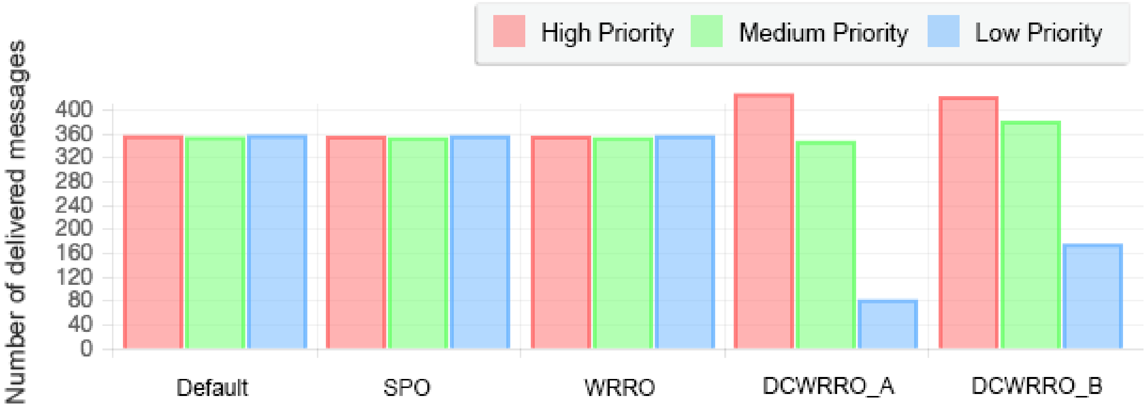


Figure 51.: SW: Increased number of nodes (3x)

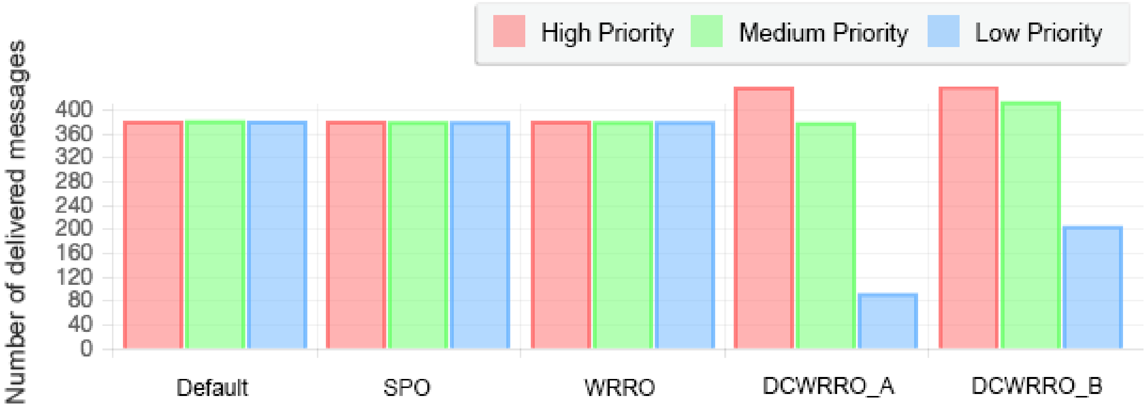


Figure 52.: SW: Increased number of nodes (4x)

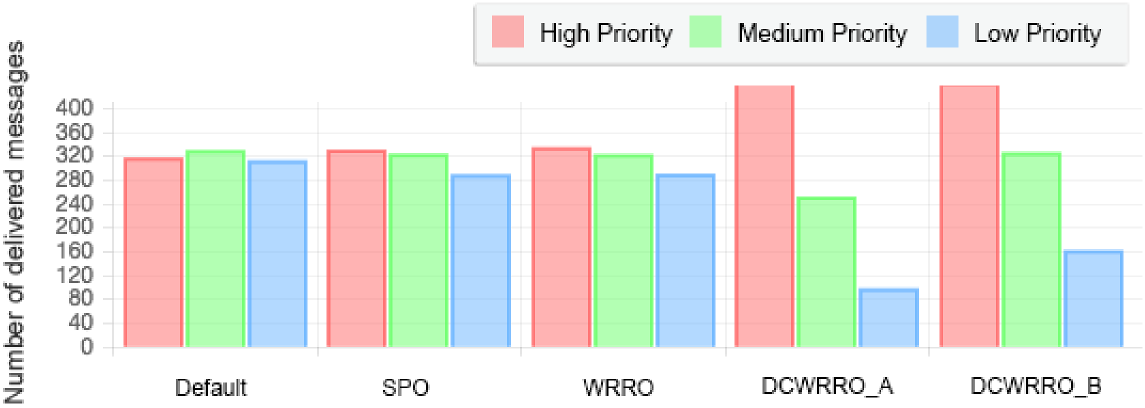


Figure 53.: SWBinary: Increased number of messages (4320)

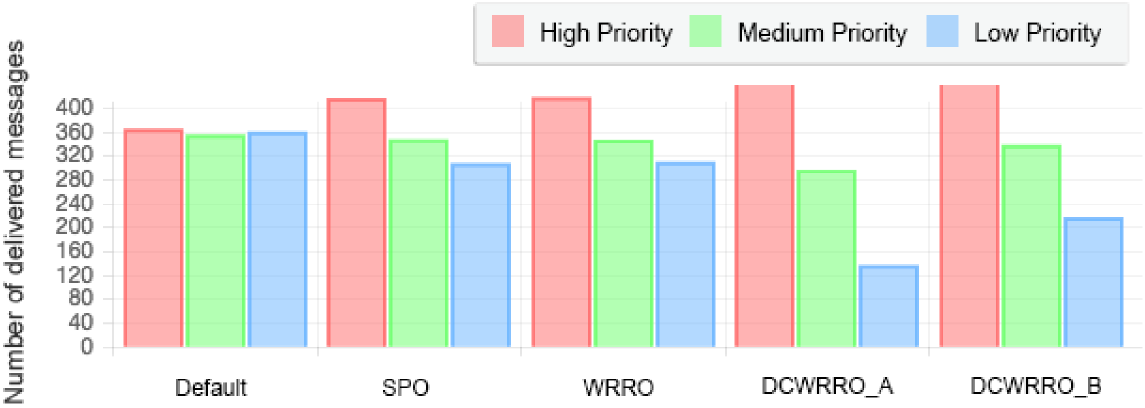


Figure 54.: SWBinary: Increased number of messages (8640)

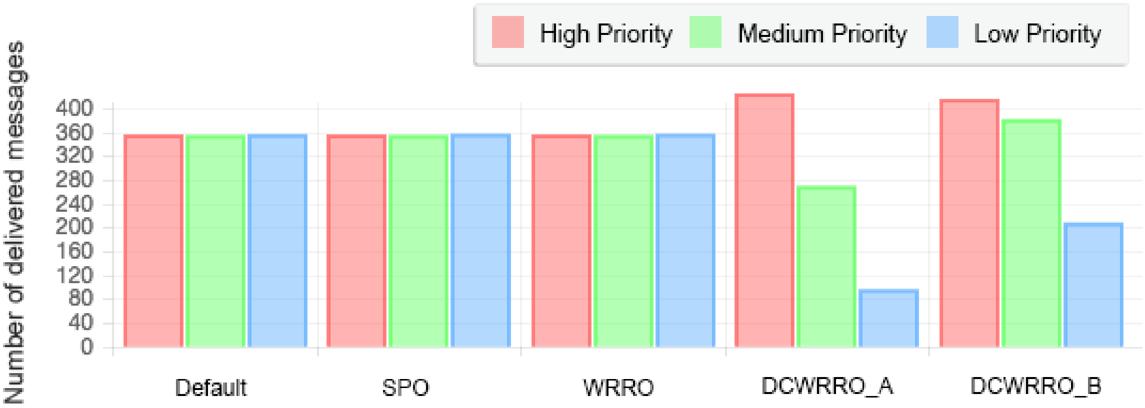


Figure 55.: SWBinary: Increased number of nodes (3x)

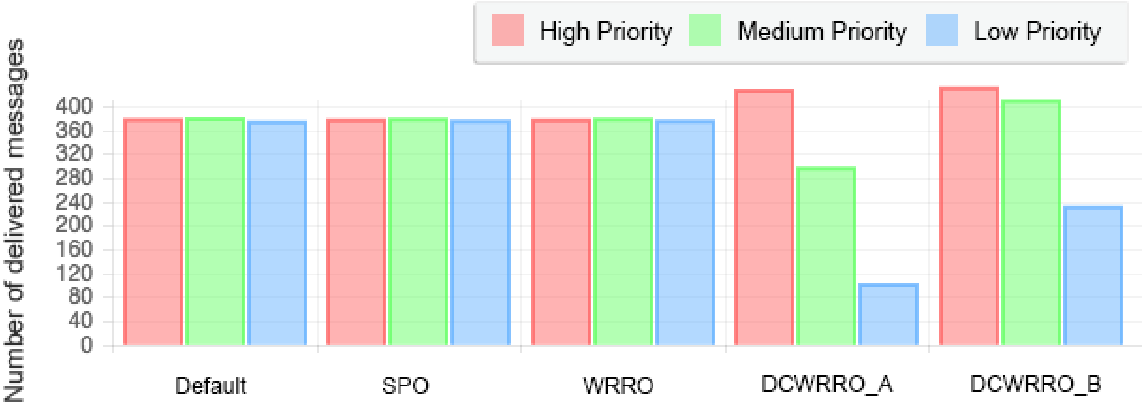


Figure 56.: SWBinary: Increased number of nodes (4x)

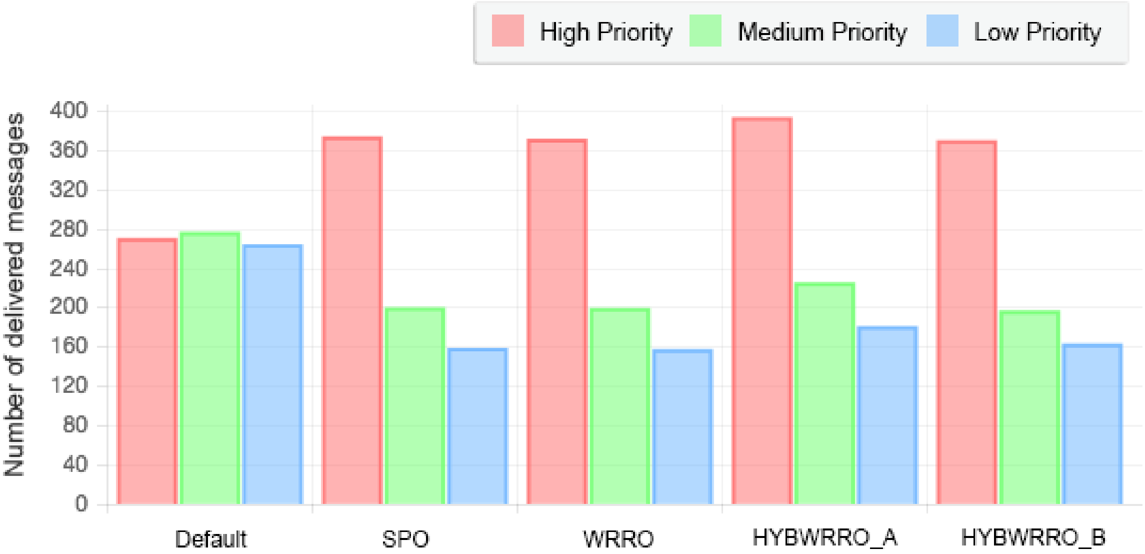


Figure 57.: PRoPHET: Increased number of messages (4320)

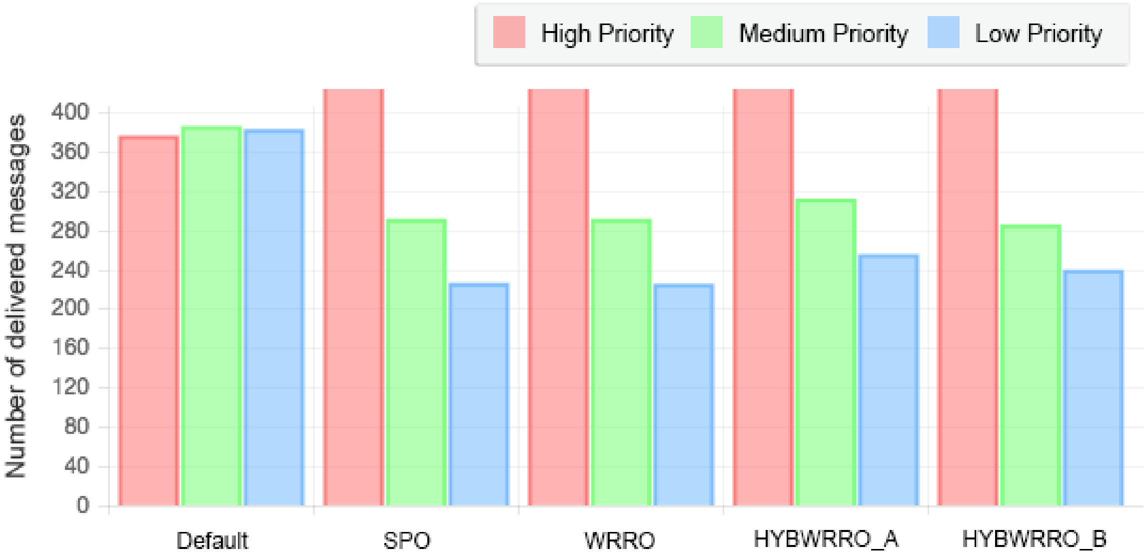


Figure 58.: PRoPHET: Increased number of messages (8640)

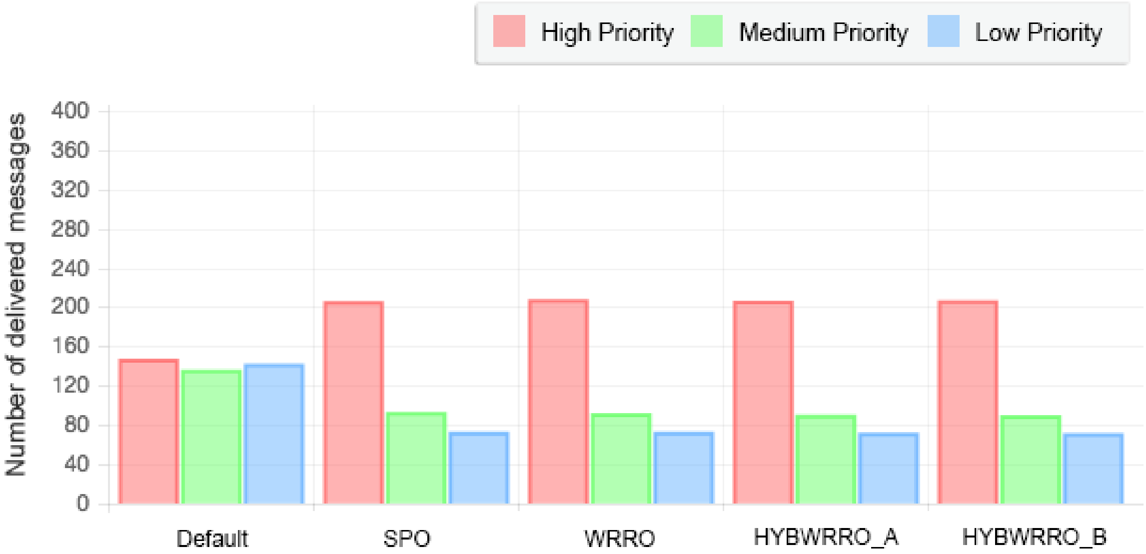


Figure 59.: PRoPHET: Increased number of nodes (3x)

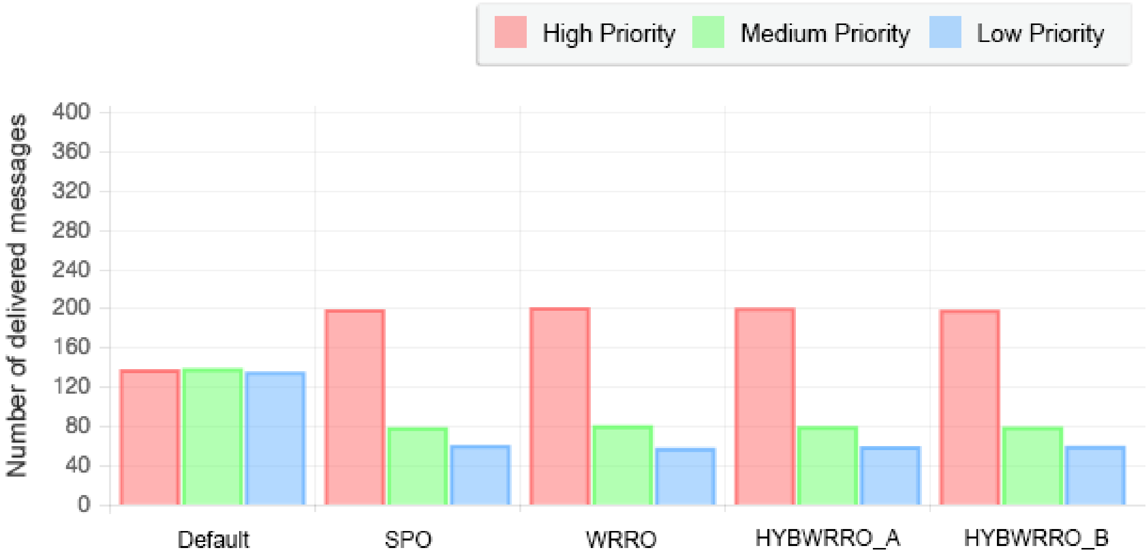


Figure 60.: PRoPHET: Increased number of nodes (4x)