

Analysis of Buffer Management Policies for Opportunistic Networks

Christopher Probst

Email: christopher.probst@hhu.de

University of Düsseldorf, HHU - 2015 SS - Opportunistic and Peer-to-Peer Networks

Abstract—In Opportunistic Networks, which are inherently influenced by churn, messages are routed on a best effort basis. If nodes cannot forward messages due to missing connectivity, the messages are buffered according to the queue policy. In case of congestion, nodes drop messages based on a drop policy. Both policies form the buffer management policy, which has an impact on routing performance. Although a multitude of different policies exist, there has not yet been a comprehensive study regarding routing performance, which compares all possible policies against each other. Therefore, this paper analyzes epidemic routing with 121 different policies and compares them in terms of delivery ratio, overhead and latency. Evaluation shows, that only a few buffer management policies are qualified for common use cases.

I. INTRODUCTION

The increasing amount of mobile devices like phones, notebooks or wearables requires an ever-growing coverage of mobile web. While cities often provide a good mobile web infrastructure, there are still certain areas with weak network coverage. However, mobile devices usually support some form of wireless adhoc communication like WiFi or Bluetooth. Therefore, even without a mobile web infrastructure, nearby devices can communicate and provide certain services to form an opportunistic network, where messages can be routed hop-by-hop through other devices to their destination.

Since all wireless techniques have a maximum range, mobile devices usually connect and disconnect to each other very frequently. Therefore, the network topology is constantly changing, which makes deterministic routing almost impossible. To overcome this issue, the store-carry-forward pattern is applied, whereby nodes buffer messages in case of missing connectivity or forward them, if nodes are nearby. If the message buffer is exhausted, messages are dropped. Message forwarding is controlled by the routing protocol, whereas buffering and dropping of messages is considered buffer management.

Therefore, opportunistic routing protocols usually cannot guarantee message delivery, because the message receiver might be offline or is not reachable by the sender. Also, due to physical closeness, opportunistic networks usually transport messages with high delays.

Routing protocols are often part of active research. However, buffer management is often unattended in research or too specific. Therefore, this paper provides a broad study of different buffer management policies and compares them to each other.

II. ROUTING IN OPPORTUNISTIC NETWORKS

When a node *A* wants to send a message to another node *B*, *A* checks for connectivity first. If both are connected, *A* transfers the message directly. Otherwise, *A* will look for other nodes nearby, which could help to route the message to its destination. The routing behaviour is composed of the buffer management policy and the routing protocol. The buffer management policy itself is composed of the queue policy and drop policy.

The routing protocol decides which message is forwarded to which node, when multiple nodes are physically close to each other. For this study, only the Epidemic [1] routing protocol is considered, which is presented in Section II-A. The queue policy determines the order of messages stored in the send queue, whereas the drop policy determines the order in which messages are dropped, if the send queue is exhausted. Therefore, the performance of an opportunistic network largely depends on the buffer management policy and the routing protocol.

An efficient and effective routing protocol can be arbitrary complex, because it has to predict the future network topology, in order to route messages reliably. In comparison, for the buffer management policy often only local parameters are considered, like message arrival time, message size or remaining time-to-live.

Also, depending on the routing protocol, the queue policy can be overridden. Usually, the routing protocol takes a message from the head of the send queue and decides, which nearby node is best suited for forwarding this message. But it can also decide to ignore the queue policy completely and choose a message by itself, which is best suited for forwarding.

While the impact of the queue policy inherently depends on the routing protocol, the drop policy directly affects the routing performance, if congestion occurs. Some routing protocols, like Epidemic, even encourage congestion, by forwarding the same message multiple times to achieve a high chance of delivery. However, the network could lack enough bandwidth to cope with all requested messages, so these routing protocols have to apply the drop policy frequently.

This study evaluates the impact of both, the queue and drop policy. Therefore, a multitude of different buffer management policies are part of this study, which are presented in Section II-B.

A. Routing Protocols

This study uses the Epidemic routing protocol, which does not make any assumptions about the network topology. When two nodes are nearby, both exchange so called *summary vectors*, which contain hashes of all messages stored in the send queue of each node. Based on the difference of the summary vectors, both nodes exchange new messages, whose order are determined by the queue policy. If congestion occurs, it drops messages according to the drop policy. Since the routing performance is inherently influenced by the buffer management policy, the Epidemic routing protocol is perfectly suited for comparing different buffer management policies.

Another routing protocol is PROPHET [2], which is widely accepted as very efficient in terms of bandwidth usage and delivery probability. However, it is not part of this study, because it overrides the queue policy by a probability-based policy. It only falls back to the queue policy, if two messages have the same probability. Therefore, the impact of the queue policy is hard to quantify and cannot easily be compared to the Epidemic routing protocol. A detailed study of buffer management policies using PROPHET and further routing protocols is part of future work.

B. Buffer Management Policies

The buffer management policy consists of the queue policy, which determines the order of stored messages in the send queue, and the drop policy, which specifies the order of dropped messages, if congestion occurs. Usually, the sorting order is only based on local parameters. A well known sorting order, which is called FIFO (first-in-first-out), sorts messages according to their arrival times in ascending order. While most sorting orders have abbreviated names, we describe them by their sorting parameter and the sorting direction, so we describe FIFO as Arrival-time (ascending) and LIFO (last-in-first-out) as Arrival-time (descending).

For our study, we use the below mentioned sorting orders, used by both, the queue and drop policy. To get a comprehensive set of results, we evaluate all combinations for both, the queue and drop policy. We consider eleven sorting orders (policies), which yield $11 * 11 = 121$ different results per scenario. The simulation setup and evaluation are presented in Section III and V respectively.

1) *Random*: The order of the messages is chosen randomly, so the sorting direction is not important. While this sorting order does not have a specific strategy, it often outperforms other sorting orders.

2) *Arrival-time (ascending)*: The messages are sorted by their arrival time, in ascending order. So, the head of the queue contains the oldest message, and the tail contains the latest message.

3) *Replications (ascending)*: The messages are sorted by the number of their replications, in ascending order. A message is replicated each time, when it is forwarded to another node. So, the head of the queue contains the message, which was

forwarded least, while the tail contains the message, which was forwarded most.

4) *Relayed-nodes (ascending)*: The messages are sorted by the number hops from the origin to the current node, in ascending order. So, the head of the queue contains the message, which has the fewest hops from the origin, while the tail contains the message, which has the most hops from the origin.

5) *Time-to-live (ascending)*: The messages are sorted by their remaining time-to-live, in ascending order. Each message has an initial time-to-live parameter, which determines the maximum duration, in which this message can be forwarded. So, the head of the queue contains the message with the lowest remaining time-to-live, and the tail contains the message, with the highest remaining time-to-live.

6) *Message-size (ascending)*: The messages are sorted by their message size, in ascending order. So, the head of the queue contains the smallest message, while the tail contains the largest message.

7) *Arrival-time (descending)*: The messages are sorted by their arrival time, in descending order. So, the head of the queue contains the latest message, and the tail contains the oldest message.

8) *Replications (descending)*: The messages are sorted by the number of their replications, in descending order. A message is replicated each time, when it is forwarded to another node. So, the head of the queue contains the message, which was forwarded most, while the tail contains the message, which was forwarded least.

9) *Relayed-nodes (descending)*: The messages are sorted by the number hops from the origin to the current node, in descending order. So, the head of the queue contains the message, which has the most hops from the origin, while the tail contains the message, which has the fewest hops from the origin.

10) *Time-to-live (descending)*: The messages are sorted by their remaining time-to-live, in descending order. Each message has an initial time-to-live parameter, which determines the maximum duration, in which this message can be forwarded. So, the head of the queue contains the message with the highest remaining time-to-live, and the tail contains the message, with the lowest remaining time-to-live.

11) *Message-size (descending)*: The messages are sorted by their message size, in descending order. So, the head of the queue contains the largest message, while the tail contains the smallest message.

III. SIMULATION SETUP

Evaluating different routing protocols or buffer management policies in Opportunistic Networks is not trivial. With real devices, the evaluation results will certainly differ to some

extent, since there are always factors, which cannot be controlled properly. Also, a realistic evaluation requires devices to move physically, which is inherently slower than simulating. A simulator can gather and store the results of all participating nodes at one place, while real devices can only store their own results individually. Therefore, the Opportunistic Network Environment Simulator (ONE Simulator) [3] is used to evaluate different buffer management policies reliably.

However, in order to evaluate all buffer management policies listed in Section II-B, the simulator was modified in several ways, which is explained in Section III-A. The methodology, scenarios and evaluation are presented in Section III-B, IV and V respectively.

A. Simulator Changes

The latest version of the ONE simulator, which is 1.5.1.RC2 at the time of writing, does only support two queue policies, which are Random and Arrival-time (ascending) and a single drop policy, which is also Arrival-time (ascending). Furthermore, the simulator does not support parallel execution of simulations, which is impractical, because every scenario requires 121 simulations to evaluate all possible policies.

Therefore, the contribution to the simulator contains implementations for all buffer management policies, listed in Section II-B. Unfortunately, it was not trivial to introduce parallelism by modifying the simulator directly, because the sequential execution model is integrated too deeply into the core.

Therefore, the parallelization is realized by running several simulator instances at the same time through Python scripts, which also merge the results afterwards. All changes are publicly accessible on GitHub [4].

B. Methodology

The evaluation consists of three scenarios and a composite scenario evaluation, which are presented in Section IV. A scenario is represented by a set of settings, which specify the routing protocol, buffer management policy and the movement model, type and network interfaces of each node. For this study, all scenarios use the Epidemic routing protocol, which is explained in Section II-A.

For each scenario, all possible buffer management policies are simulated, which makes 121 simulations per scenario, since a buffer management policy consists of both, the queue and drop policy and each policy has eleven possible sorting orders, which are listed in Section II-B.

Each simulation creates a report, which contains multiple metrics to rate the settings used for the simulations. For this study, the following metrics were considered. Including more metrics is part of future work.

1) *Delivery Ratio*: This metric represents the ratio of received messages over sent messages. Therefore, a high delivery ratio means more successfully delivered messages.

2) *Overhead Factor*: The overhead factor metric represents the amount of relayed messages for each sent message. Hence, a low overhead factor means less relayed messages per sent message.

3) *Average Delay*: The average delay metric measures (in minutes) the average time it takes to send a message from origin to destination.

4) *Composite*: For each cell of each table the percentage relative to the minimum and maximum of the respective table is computed. The final result is the average of the percentage values of all three tables. Therefore, the composite metric is the equally-weighted average of the delivery ratio, overhead factor and average delay metric, expressed as percentage. It can identify combinations, which work well for all other metrics on average.

For each scenario, four gray scale tables are created, each containing 121 cells, one for each metric, with cell colors ranging between white and gray, where white represents the best and gray the worst results measured in the scenario. The x/y-axes show the used drop and queue policy respectively. The best cell value per table is displayed in bold font.

Finally, a composite scenario evaluation is created, where each table contains the average of the respective metrics of all scenarios. This identifies buffer management policies, which worked well for all three scenarios. While this does not yield the best solution for each scenario, it identifies policies, which are preferable on average. This is exactly what this study is looking for.

The final result gives some indication of the impact of the buffer management policy. However, the results gathered in this study cannot easily be transferred to any conceivable scenario, although it is likely, that buffer management policies, which yielded good results for all three scenarios, might also behave well in other scenarios. Evaluating more scenarios is part of future work.

IV. SCENARIOS

A scenario is a set of settings, processed by the ONE Simulator. While the simulator is extensively configurable, it also provides default settings, so it can be started without further changes. Since the default settings represent a common use case and are often used for testing and research, these settings are used for the base scenario, which is extended by each scenario by adjusting the message generation. The message generation is a part of the simulator, which can randomly schedule messages between nodes. While modifying other settings would certainly influence the results, this study only evaluates changes of the message generation. Evaluating the impact of other settings is part of future work.

The base scenario, which is based on the default settings, but without message generation, is shown in Table I. It runs for twelve hours and simulates 126 nodes. Nodes belong to groups, which specify the movement model and network interfaces of their members. Nodes can only move on fixed routes, which simulates streets in cities. There are two types of

Parameter	Value
Simulation time	12 hours
Total Number of Nodes	126
Total Number of Node Groups	6
Routing Protocol	Epidemic
Buffer Management Policies	11 * 11 = 121
2x Pedestrian Group [Count; Speed]	40; 0.5-1.5 m/s
1x Car Group [Count; Speed]	40; 2.7-13.9 m/s
3x Tram Group [Count; Speed]	2; 7-10 m/s
Movement Model 1 (M1)	Shortest Path Map based Movement
Movement Model 2 (M2)	Map Route Movement
Movement Model Groups [M1; M2]	3; 3
Movement Model Buffers [M1; M2]	5 MB; 50 MB
Interface types	Low-Speed, High-Speed
Low-Speed [Range; Bandwidth]	10 m; 250 KB/s
High-Speed [Range; Bandwidth]	1 km; 10 MB/s
Groups using Low-Speed	All
Groups using High-Speed	1x Tram Group

TABLE I
BASE SCENARIO

Parameter	Value
Message Generators	1
Message Generator 1 [Interval; Size]	25-35 s; 0.5-1 MB

TABLE II
SCENARIO 1

Parameter	Value
Message Generators	2
Message Generator 1 [Interval; Size]	1-5 s; 0.5-2 KB
Message Generator 2 [Interval; Size]	25-35 s; 64-512 KB

TABLE III
SCENARIO 2

Parameter	Value
Message Generators	3
Message Generator 1 [Interval; Size]	1-5 s; 0.5-2 KB
Message Generator 2 [Interval; Size]	25-35 s; 64-512 KB
Message Generator 3 [Interval; Size]	60-120 s; 1-5 MB

TABLE IV
SCENARIO 3

interfaces, which are Low-Speed and High-Speed. The Low-Speed interface has a short range and low bandwidth, whereas the High-Speed interface has a long range and high bandwidth. Only one of the three tram groups, which consist of two trams each, has the High-Speed interface. This group acts as a fast moving, long-range message gateway. The Low-Speed interface is supported by each node. Therefore, most nodes can only communicate with other nearby nodes. All scenarios are described below.

A. Scenario 1

The first scenario, shown in Table II, generates moderate traffic. Only one message each 25-35 seconds with a random

size of 0.5-1 MB is scheduled. This scenario simulates rare communication, with moderately large messages, which remotely models web browsing.

B. Scenario 2

The second scenario, presented in Table III, modifies the first scenario by decreasing the size of the moderate sized messages, which now varies between 64-512KB. It also adds small messages, which have a random size of 0.5-2 KB and are scheduled each 1-5 seconds. This scenario simulates frequent communication with relatively small messages, which is comparable to instant messaging, and periodically scheduled moderate sized messages, to evaluate mutual influence.

C. Scenario 3

The last scenario extends the second scenario, as shown in Table IV, by adding large messages, which are scheduled each 60-120 seconds, each with a random size of 1-5 MB. These messages remotely model file sharing traffic. Frequent and moderate traffic is the same as in the second scenario. Thereby, the interaction of messages, which significantly differ in size, can be analysed.

D. Composite Scenario Evaluation

The composite scenario evaluation merges the tables of all three scenarios, by computing the average of the respective metrics of all scenarios. These tables identify preferable buffer management policies per metric.

V. EVALUATION

The numerical results of scenario IV-A, IV-B, IV-C and the composite scenario evaluation IV-D are presented in Figure 1, 2, 3 and 4 respectively. These results are analysed in the following.

A. Scenario 1

In the first scenario, which generates only moderate traffic, the highest delivery ratio is reached using Relayed-nodes (ascending) as queue policy and Time-to-live (ascending) as drop policy, which is shown by the bold font style in the first table. So messages with the lowest hop count from the origin to a respective node are forwarded first, while messages with the lowest remaining time-to-live are dropped first. While this is the best combination for this scenario, there are other good combinations, shown by light gray and white cells. Interestingly, most of these combinations are using Time-to-live (ascending) or Relayed-nodes (descending) as drop policy.

Therefore, the drop policy seems to have a greater impact on the delivery ratio than the queue policy, which implies congestion. Otherwise, the drop policy would have never been applied. Conversely, the queue policy only has minor impact on the delivery ratio, if congestion never occurs.

The best combination regarding the overhead factor, is Relayed-nodes (ascending) and Arrival-time (ascending) as queue and drop policy respectively. However, Arrival-time

(ascending) used as drop policy seems to yield the best results, independent of the queue policy, although the Random drop policy also generates good results. As with the delivery ratio, the drop policy seems to have greater impact on average than the queue policy, at least in case of congestion.

Regarding the average delay, Time-to-live (descending) and Time-to-live (ascending), used as queue and drop policy respectively, yield the best result. While the average delay seems to be more influenced by the queue policy than the other two metrics, the drop policies Replications (ascending), Arrival-time (descending) and Relayed-nodes (descending) still dominate over the respective queue policies, by producing inferior results on average.

Unfortunately, there is not a best combination regarding all metrics, because each metric has its own characteristics. However, we can identify combinations, which work well on average for all three metrics. Those combinations are shown by the composite metric, which is the equally-weighted average of all other metrics, expressed as percentage. Regarding the composite metric, the best three queue / drop policy combinations are shown in the following list.

- 1) Time-to-live (descending) / Arrival-time (ascending) (89.4 %)
- 2) Relayed-Nodes (ascending) / Arrival-time (ascending) (88.5 %)
- 3) Relayed-nodes (ascending) / Time-to-live (ascending) (82.2 %)

While these combinations are preferable on average, regarding the first scenario, the Time-to-live (ascending), Arrival-time (ascending) and even Random drop policies seem to have a greater impact on average, since these drop policies produce consistently satisfying results, independent of the queue policy.

B. Scenario 2

The second scenario modifies the first scenario, by decreasing the size of the moderate traffic and adding frequent traffic using small messages. Figure 2 shows the results of this scenario.

Regarding the delivery ratio, the best combination is Replications (ascending) and Relayed-nodes (descending) used as queue and drop policy respectively. The best combination suited for a high delivery ratio in the first scenario, which is Relayed-nodes (ascending) and Time-to-live(ascending), also yields a satisfying result for this scenario.

Interestingly, in addition to the drop policies Time-to-live (ascending) and Relayed-nodes (descending), the drop policy Message-size (descending) also produces very good results on average, independent of the queue policy. This is different to the first scenario. However, it can be explained by the small and frequently sent messages, because the delivery ratio does not differentiate between small and moderate sized messages. Therefore, dropping a larger message makes room for multiple smaller messages, which in turn increases the delivery ratio.

Regarding the overhead factor, the best combination is Message-size (descending) and Arrival-time (descending) as

queue and drop policy respectively. This time, the queue policy Message-size (descending) dominates other combinations, which is also related to different message sizes. So forwarding larger messages first reduces the overhead factor in general.

The higher influence of the queue policy is also noticeable regarding the average delay. The best combination is Time-to-live (descending) and Time-to-live (ascending) as queue and drop policy respectively, which are also the best policies for the average delay in the first scenario. As with the overhead factor, the queue policy seems to dominate the quality of the results.

The composite metric shows the best combination on average, which is Time-to-live (descending) and Message-size (descending) as queue and drop policy. But as with the first scenario, the combination Time-to-live (descending) and Arrival-time (ascending) also yields good results on average. The three best queue / drop policy combinations for the second scenario are listed below.

- 1) Time-to-live (descending) / Message-size (descending) (88.9 %)
- 2) Time-to-live (descending) / Arrival-time (ascending) (87.4 %)
- 3) Time-to-live (descending) / Time-to-live (ascending) (87.1 %)

C. Scenario 3

The third scenario extends the second scenario, by adding demanding traffic using large but rarely sent messages. The results of the third scenario, which are shown in Figure 3, are very close to the results of the second scenario, although there are minor differences regarding the best combination per metric.

The highest delivery ratio is reached using Time-to-live (descending) and Message-size (descending) as queue and drop policy. Like in the second scenario, the drop policies Time-to-live (ascending), Relayed-nodes (descending) and Message-size (descending) produce satisfying results on average, independent of the queue policy.

The lowest overhead factor is reached using Message-size (descending) and Replications (descending) as queue and drop policy. While this is different to the second scenario, the overall distribution of good and bad results is very similar.

The same is true for the average delay metric, whose results are comparable to those of the second scenario. However, the best combination is Time-to-live (descending) and Message-size (ascending) as queue and drop policy.

Consequently, the composite metric is almost identical to the second scenario. It even has the same three best queue / drop policy combinations, which are listed below.

- 1) Time-to-live (descending) / Message-size (descending) (89.2 %)
- 2) Time-to-live (descending) / Arrival-time (ascending) (86.6 %)
- 3) Time-to-live (descending) / Time-to-live (ascending) (85.6 %)

D. Composite Scenario Evaluation

The composite scenario evaluation, which is shown in Figure 4, merges all results, by computing for each metric the average of all scenarios. These results are suited to identify combinations, which work well for common use cases like mixing instant messaging, web browsing and file sharing, where the messages often differ significantly in size and are sent at different time intervals. Based on these results, the following list shows the three queue / drop policy combinations of the first three metrics, which are best on average.

- Delivery Ratio
 - 1) Replications (ascending) / Relayed-nodes (descending) (70.6 %)
 - 2) Replications (ascending) / Time-to-live (ascending) (70.1 %)
 - 3) Relayed-nodes (ascending) / Time-to-live (ascending) (69.5 %)
- Overhead Factor
 - 1) Message-size (descending) / Arrival-time (ascending) (33.4 transmissions / message)
 - 2) Message-size (descending) / Replications (descending) (35.5 transmissions / message)
 - 3) Message-size (descending) / Random (38.3 transmissions / message)
- Average Delay
 - 1) Time-to-live (descending) / Time-to-live (ascending) (30.7 minutes)
 - 2) Time-to-live (descending) / Arrival-time (ascending) (33.5 minutes)
 - 3) Time-to-live (descending) / Random (36.0 minutes)

The averaged composite metric is particularly interesting, which is presented below, because it combines the equally-weighted results of all three averaged metrics. Therefore, we propose to consider one of the following buffer management policies for common use cases.

- Composite
 - 1) Time-to-live (descending) / Arrival-time (ascending) (87.8 %)
 - 2) Relayed-nodes (ascending) / Arrival-time (ascending) (85.8 %)
 - 3) Relayed-nodes (ascending) / Time-to-live (ascending) (83.1 %)

VI. CONCLUSION AND FUTURE WORK

In this paper, we analysed the impact of buffer management policies on epidemic routing in terms of different metrics like delivery ratio, overhead and latency, which are shown in Section III-B. The buffer management policy is composed of the queue policy, which defines the order of stored messages and the drop policy, which determines the order of dropped messages, in case of congestion. For both, the queue and drop policy, we defined eleven different sorting orders, which are defined in Section II-B.

Since a buffer management policy consists of both, the queue and drop policy, we have $11 * 11 = 121$ different combinations. To simulate different use cases, we defined three scenarios, which all extend the base scenario defined in Table I, and only differ in terms of message generation. The scenarios are presented in Section IV. In order to evaluate all combinations for each scenario, we parallelized and used the ONE Simulator. Based on all scenarios, whose results are analysed in Section V-A, V-B and V-C, we created a composite scenario evaluation, which is presented in Section V-D.

The composite scenario evaluation shows, that only a few buffer management policies are preferable on average. Therefore, we propose to consider the following queue / drop policy combinations for use cases, where messages differ significantly in size and are sent at different time intervals. Such traffic could be produced by mixing instant messaging, web browsing and file sharing.

- 1) Time-to-live (descending) / Arrival-time (ascending) (87.8 %)
- 2) Relayed-nodes (ascending) / Arrival-time (ascending) (85.8 %)
- 3) Relayed-nodes (ascending) / Time-to-live (ascending) (83.1 %)

This study only evaluated the impact of different buffer management policies and message generators, while the base scenario remained unchanged. Hence, increasing the amount of base scenarios is part of future work. However, routing performance is also inherently influenced by the choice of the routing protocol. Therefore, future work should also consider evaluating other routing protocols like PROPHET. Furthermore, besides the buffer management policies evaluated in this study, there are other policies, which should be considered for future work as well.

REFERENCES

- [1] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks, Duke University, Tech. Rep. CS-200006," 2000.
- [2] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in intermittently connected networks, Lecture Notes in Computer Science, Vol 3126, 239-254," 2004.
- [3] A. Keränen, T. Kärkkäinen, and J. Ott, "Simulating Mobility and DTNs with the ONE, Journal of Communications, Vol 5, No 2, 92-105," 2010.
- [4] C. Probst, "One simulator modifications (parallelization, buffer management policies)," May 2015. [Online]. Available: <https://github.com/chrisprobst/one-simulator>

			Drop Policy											
			X	Ascending						Descending				
			Random	Arrival-time	Replications	Relayed-nodes	Time-to-live	Message-size	Arrival-time	Replications	Relayed-nodes	Time-to-live	Message-size	
Metric			Delivery Ratio (percentage, more is better)											
Queue Policy	X	Random	21.9	25.2	15.1	8.8	39.4	11.4	16.2	23.9	38.3	11.2	10.1	
		Arrival-time	18.6	25.1	14.2	8.8	38.7	10.3	14.6	21.0	37.5	10.3	10.3	
		Replications	25.0	24.7	15.3	8.8	39.9	11.3	15.9	23.3	39.7	9.4	10.5	
		Relayed-nodes	30.4	36.8	22.7	10.2	40.9	10.6	17.5	26.7	39.1	11.1	10.3	
		Time-to-live	18.3	19.1	12.4	9.0	38.2	9.5	16.1	23.8	35.5	10.3	10.9	
	Descending	Message-size	17.0	18.9	11.6	7.3	33.0	12.0	14.6	23.0	38.7	8.8	9.7	
		Arrival-time	23.2	24.1	15.6	8.5	38.1	11.8	15.9	24.5	40.6	9.4	10.3	
		Replications	18.1	20.4	14.7	8.5	35.5	10.2	15.5	21.6	36.2	9.8	10.3	
		Relayed-nodes	15.9	17.5	11.8	8.6	32.1	9.6	14.2	23.0	37.4	9.8	10.7	
		Time-to-live	30.4	34.0	18.9	9.1	38.0	11.4	16.1	25.5	40.1	9.8	10.1	
Message-size	17.8	20.2	14.2	9.0	31.6	10.3	17.9	25.6	36.1	9.9	11.8			
Metric			Overhead Factor (transmissions per message, less is better)											
Queue Policy	X	Random	115.6	84.0	145.1	233.3	290.9	127.9	147.9	167.1	242.2	129.6	148.2	
		Arrival-time	116.6	81.2	144.8	203.0	227.1	134.1	139.8	112.9	161.6	125.8	147.7	
		Replications	117.9	89.3	153.6	245.1	329.5	173.4	188.2	148.3	258.5	136.4	140.5	
		Relayed-nodes	92.0	53.5	123.9	191.6	227.7	193.0	137.7	216.8	273.9	112.7	139.4	
		Time-to-live	102.2	91.2	166.2	178.1	212.1	100.0	126.1	88.5	157.9	113.7	138.0	
	Descending	Message-size	167.4	134.4	239.7	320.3	410.6	501.3	190.6	152.8	293.2	161.4	153.0	
		Arrival-time	136.3	93.7	159.2	252.1	346.4	104.9	156.4	178.4	288.6	139.6	143.9	
		Replications	124.0	95.7	140.3	217.5	258.4	105.3	131.1	156.3	149.0	131.0	146.0	
		Relayed-nodes	122.5	110.2	183.3	199.4	295.0	97.8	154.0	92.6	172.0	122.6	144.6	
		Time-to-live	104.8	62.7	145.0	201.8	350.2	178.0	201.0	284.6	313.4	133.7	143.9	
Message-size	73.4	59.8	105.2	137.2	246.3	91.6	86.1	72.3	137.2	97.9	248.6			
Metric			Average Delay (minutes, less is better)											
Queue Policy	X	Random	83.5	78.4	98.3	78.0	58.4	83.7	101.9	79.5	110.1	97.2	75.2	
		Arrival-time	97.2	92.2	96.6	78.1	67.9	96.8	101.2	95.8	125.4	101.0	75.7	
		Replications	68.2	65.4	97.8	67.6	47.6	76.6	94.1	60.9	101.0	81.0	62.2	
		Relayed-nodes	72.9	59.4	106.8	71.2	52.7	75.8	94.0	69.4	114.1	93.2	62.0	
		Time-to-live	97.6	93.0	96.6	85.5	84.6	92.7	114.1	100.4	124.4	101.9	76.4	
	Descending	Message-size	78.1	72.2	91.3	71.5	56.0	75.1	103.6	78.2	110.3	79.2	68.8	
		Arrival-time	71.2	55.8	98.3	74.2	45.3	74.8	98.6	64.1	98.0	78.7	61.8	
		Replications	97.4	81.8	102.6	82.8	66.1	86.5	107.3	92.8	120.9	96.8	75.8	
		Relayed-nodes	83.3	86.4	101.1	83.7	67.7	92.4	103.9	95.7	117.9	85.4	76.0	
		Time-to-live	51.6	48.0	90.8	61.4	40.4	72.2	90.5	60.8	96.9	71.7	61.8	
Message-size	86.4	73.8	104.5	77.2	61.8	84.2	111.6	87.4	113.7	92.1	74.6			
Metric			Composite (percentage, more is better)											
Queue Policy	X	Random	59.7	67.3	44.9	40.0	73.8	48.3	44.4	59.4	56.0	42.7	48.8	
		Arrival-time	50.9	61.9	44.7	42.3	74.1	41.5	43.7	54.1	55.2	40.5	48.8	
		Replications	68.5	71.5	44.7	43.3	75.6	47.6	44.2	67.5	59.8	46.7	54.8	
		Relayed-nodes	73.9	88.5	50.7	47.2	82.2	45.7	49.5	62.4	52.9	45.4	54.8	
		Time-to-live	51.6	55.0	41.4	41.4	68.2	44.9	41.1	56.9	53.9	41.1	49.9	
	Descending	Message-size	53.0	59.7	37.2	34.6	59.5	24.4	39.0	60.0	52.5	45.0	50.6	
		Arrival-time	64.2	74.3	44.4	39.9	73.5	53.9	44.8	65.2	59.6	47.3	54.5	
		Replications	49.8	60.3	43.2	39.1	69.3	47.6	42.8	52.7	56.7	41.3	48.9	
		Relayed-nodes	53.2	54.5	37.6	40.2	62.6	45.3	41.1	57.7	57.3	46.4	49.3	
		Time-to-live	81.4	89.4	51.6	49.2	75.0	49.0	44.8	59.5	57.6	51.0	54.4	
Message-size	57.6	65.9	44.6	47.7	68.0	49.6	46.9	65.0	60.2	45.7	43.3			

Fig. 1. Scenario 1

			Drop Policy										
			X	Ascending					Descending				
				Random	Arrival-time	Replications	Relayed-nodes	Time-to-live	Message-size	Arrival-time	Replications	Relayed-nodes	Time-to-live
Metric			Delivery Ratio (percentage, more is better)										
Queue Policy	X	Random	61.3	69.4	46.8	29.4	84.6	24.1	51.0	71.2	84.0	33.7	82.7
		Arrival-time	48.5	61.3	43.0	22.3	82.7	22.4	48.3	60.7	77.6	27.6	79.4
		Replications	74.6	80.8	61.3	38.3	88.4	29.3	57.8	82.7	90.2	38.4	88.1
		Relayed-nodes	75.3	86.1	66.8	43.0	88.5	28.3	55.9	79.5	81.5	33.5	88.9
		Time-to-live	44.9	49.6	33.0	22.7	72.5	20.6	56.2	68.2	72.8	30.4	75.8
	Descending	Message-size	82.4	85.2	73.1	62.2	85.6	36.5	69.7	85.0	85.1	47.7	85.6
		Arrival-time	68.6	72.5	57.8	37.7	77.6	26.6	59.5	77.0	85.3	33.1	84.7
		Replications	47.4	52.9	40.1	21.9	65.0	22.0	44.7	59.8	68.6	26.9	73.4
		Relayed-nodes	44.0	50.2	30.8	26.8	67.3	19.7	47.5	67.7	71.9	24.6	77.0
		Time-to-live	77.5	83.5	66.0	37.6	83.6	31.0	58.6	79.8	88.3	43.4	89.7
Message-size	32.9	35.8	27.7	18.5	47.9	22.0	59.3	51.6	63.3	24.7	76.6		
Metric			Overhead Factor (transmissions per message, less is better)										
Queue Policy	X	Random	85.5	75.7	111.3	166.2	62.2	94.8	103.2	71.2	57.3	142.1	72.2
		Arrival-time	90.8	76.8	94.9	171.3	55.7	71.7	81.0	75.7	54.3	134.0	63.1
		Replications	79.3	72.0	96.1	142.8	65.9	99.6	100.3	70.4	58.8	134.1	72.3
		Relayed-nodes	68.2	56.9	79.9	115.4	60.3	91.9	85.6	64.9	67.9	137.8	62.7
		Time-to-live	79.5	73.2	90.6	151.6	64.3	87.5	67.0	57.2	58.6	113.0	60.0
	Descending	Message-size	83.1	80.2	93.0	105.2	80.5	124.0	93.3	80.0	81.0	131.7	74.5
		Arrival-time	84.7	75.6	99.1	146.9	75.0	108.1	98.9	73.9	66.2	150.5	66.1
		Replications	104.3	96.4	107.8	170.4	80.0	85.3	93.0	81.7	66.1	145.4	69.8
		Relayed-nodes	93.8	82.7	121.9	158.2	65.1	114.8	81.2	66.1	71.2	180.4	60.5
		Time-to-live	67.7	59.4	82.5	149.3	61.6	97.8	92.7	67.9	61.5	127.9	59.8
Message-size	17.9	17.5	21.4	35.9	31.1	35.8	12.2	13.8	15.9	75.5	42.2		
Metric			Average Delay (minutes, less is better)										
Queue Policy	X	Random	66.6	64.1	64.3	45.2	61.6	47.0	70.5	70.6	79.1	50.9	71.4
		Arrival-time	72.9	75.0	79.1	46.2	87.3	53.0	84.8	77.4	110.0	53.6	94.0
		Replications	47.8	46.0	48.4	35.5	43.1	39.3	50.6	52.7	49.8	39.4	44.8
		Relayed-nodes	40.9	35.8	62.8	39.7	40.6	46.3	53.3	39.8	73.4	41.4	41.4
		Time-to-live	72.4	69.7	60.9	47.0	109.1	52.5	91.3	95.3	108.7	52.9	98.9
	Descending	Message-size	59.8	59.1	53.4	42.8	62.7	49.3	55.3	65.6	66.8	45.4	55.6
		Arrival-time	36.8	33.8	44.0	33.6	33.7	41.3	49.5	48.0	49.5	38.0	42.0
		Replications	68.8	63.8	74.4	45.5	79.0	52.2	82.3	78.1	96.9	52.5	89.1
		Relayed-nodes	67.1	65.2	58.0	49.6	84.4	53.0	83.2	88.9	94.8	49.7	88.6
		Time-to-live	28.3	26.5	35.2	29.6	26.0	38.0	38.8	35.2	42.1	34.6	29.8
Message-size	62.2	59.0	64.8	49.4	57.2	51.5	89.3	75.6	96.9	51.6	85.9		
Metric			Composite (percentage, more is better)										
Queue Policy	X	Random	56.0	62.6	45.0	33.6	73.4	44.6	46.1	61.8	67.1	38.1	66.6
		Arrival-time	46.4	54.3	40.6	28.9	63.6	46.0	43.6	53.3	52.5	35.9	57.9
		Replications	70.8	75.9	61.1	46.3	81.7	49.1	57.8	74.4	81.3	46.5	79.6
		Relayed-nodes	76.0	85.3	61.1	52.2	83.9	47.4	58.7	79.1	66.1	42.6	83.3
		Time-to-live	47.2	51.7	44.0	32.7	48.5	42.2	47.4	53.4	49.9	41.6	54.9
	Descending	Message-size	68.9	71.0	65.2	61.9	69.8	43.7	62.8	68.4	67.8	48.9	73.7
		Arrival-time	71.3	76.1	60.6	45.9	78.7	45.4	59.2	72.9	77.7	41.3	80.4
		Replications	44.9	51.0	38.6	29.2	53.8	43.4	40.5	51.4	51.1	33.7	55.7
		Relayed-nodes	46.1	51.9	38.0	32.3	55.7	36.2	43.8	53.9	52.5	26.8	59.4
		Time-to-live	82.2	87.4	71.2	47.0	87.1	50.8	64.3	80.4	82.9	52.0	88.9
Message-size	57.9	60.6	53.8	52.7	64.2	53.5	60.5	62.1	58.7	46.9	64.0		

Fig. 2. Scenario 2

			Drop Policy										
			X	Ascending					Descending				
				Random	Arrival-time	Replications	Relayed-nodes	Time-to-live	Message-size	Arrival-time	Replications	Relayed-nodes	Time-to-live
Metric			Delivery Ratio (percentage, more is better)										
Queue Policy	X	Random	55.2	63.5	44.5	33.7	76.3	26.9	47.3	64.1	72.5	38.1	77.3
		Arrival-time	41.6	53.0	32.7	23.0	70.5	19.8	35.0	52.2	61.9	27.6	73.2
		Replications	69.3	73.9	59.1	47.3	81.9	33.9	60.3	75.6	81.8	46.5	83.3
		Relayed-nodes	68.7	77.8	60.5	53.6	79.2	33.5	55.1	76.6	74.9	46.0	85.0
		Time-to-live	41.5	46.8	30.2	22.5	61.2	21.0	36.0	58.0	58.6	29.1	71.7
	Descending	Message-size	64.0	70.9	62.2	60.8	74.4	43.0	57.9	69.2	71.4	55.4	75.3
		Arrival-time	63.8	66.9	55.5	42.3	72.1	32.9	57.1	68.3	75.5	43.0	79.2
		Replications	41.7	46.2	31.9	23.0	57.0	19.6	35.7	53.2	60.1	27.7	67.1
		Relayed-nodes	40.3	45.4	30.4	26.8	57.4	18.5	36.8	57.4	60.8	24.4	71.9
		Time-to-live	72.1	77.2	65.1	49.9	76.4	40.3	61.9	76.4	79.8	53.5	86.0
Message-size	29.3	34.0	22.4	20.7	45.2	20.3	27.0	42.5	49.5	21.2	67.3		
Metric			Overhead Factor (transmissions per message, less is better)										
Queue Policy	X	Random	94.2	83.1	113.5	141.4	65.2	87.5	104.6	76.8	67.7	123.0	74.3
		Arrival-time	104.3	85.0	114.6	160.3	59.2	69.5	95.7	87.6	72.1	130.5	69.8
		Replications	82.1	75.7	93.4	115.1	66.5	98.3	91.5	73.8	64.9	112.9	71.8
		Relayed-nodes	65.5	54.4	80.8	85.9	54.6	67.8	76.0	60.2	60.7	89.9	61.4
		Time-to-live	84.5	77.7	107.3	140.7	77.7	93.0	84.3	67.4	75.0	103.4	62.4
	Descending	Message-size	98.6	89.5	101.7	102.8	85.1	122.5	105.7	90.8	88.2	111.8	82.0
		Arrival-time	84.4	77.6	97.0	122.1	75.7	95.2	93.7	78.9	68.3	114.8	67.0
		Replications	112.6	105.8	134.5	156.3	84.9	80.4	109.3	89.3	77.3	137.1	76.1
		Relayed-nodes	97.1	85.9	118.9	142.4	82.5	138.2	106.0	74.4	82.3	168.0	61.4
		Time-to-live	67.8	57.2	77.0	100.1	60.9	86.4	78.0	64.9	59.5	96.1	55.8
Message-size	23.6	23.0	31.9	31.5	24.5	29.2	35.0	20.4	25.7	45.0	33.4		
Metric			Average Delay (minutes, less is better)										
Queue Policy	X	Random	63.9	62.1	63.7	50.5	55.2	46.9	66.2	67.6	68.0	54.9	71.9
		Arrival-time	68.6	68.7	65.8	50.7	62.7	51.7	70.6	72.0	86.2	55.9	90.1
		Replications	47.4	46.3	47.9	39.6	42.1	37.4	51.0	50.8	49.5	41.6	46.8
		Relayed-nodes	39.0	35.0	54.0	39.7	37.5	33.3	48.2	39.6	57.6	41.6	42.6
		Time-to-live	68.9	65.7	58.5	54.3	86.3	54.1	75.0	86.0	96.9	55.5	98.9
	Descending	Message-size	53.7	52.7	47.6	45.4	54.9	48.8	53.0	59.0	61.9	47.9	49.7
		Arrival-time	37.3	33.8	40.5	33.5	33.6	32.0	46.6	43.5	46.2	37.5	42.5
		Replications	65.8	59.7	63.9	51.7	59.4	49.0	70.5	74.9	84.5	54.6	85.2
		Relayed-nodes	63.9	61.1	56.8	50.6	68.7	51.7	66.8	80.7	86.2	52.2	87.8
		Time-to-live	28.1	26.1	31.7	28.4	25.7	24.5	35.8	33.0	36.5	34.2	30.8
Message-size	55.5	50.1	52.5	51.9	47.0	44.9	61.1	62.8	68.6	52.0	86.2		
Metric			Composite (percentage, more is better)										
Queue Policy	X	Random	50.5	57.9	40.9	35.2	71.4	45.7	43.2	57.2	63.2	39.6	62.4
		Arrival-time	39.4	49.3	33.9	25.6	66.5	44.1	37.1	46.9	48.8	32.2	53.2
		Replications	67.6	71.8	59.8	52.7	79.7	50.9	59.4	71.0	76.7	52.0	77.1
		Relayed-nodes	74.8	83.6	60.5	62.4	83.1	59.4	61.6	79.6	70.6	56.9	82.1
		Time-to-live	43.7	49.3	37.6	28.1	47.2	38.2	38.3	48.0	41.7	39.3	50.1
	Descending	Message-size	58.4	64.3	59.6	59.6	66.1	44.8	54.1	60.4	60.8	53.8	69.5
		Arrival-time	68.8	73.5	60.5	51.4	76.6	53.5	59.3	69.6	74.3	51.6	78.1
		Replications	38.8	45.3	29.9	26.0	55.5	42.7	34.5	45.7	47.5	31.4	50.9
		Relayed-nodes	42.5	48.8	35.8	31.6	52.1	27.9	37.5	48.5	46.0	23.9	55.4
		Time-to-live	80.8	86.6	73.7	62.4	85.6	62.5	70.0	81.4	82.7	62.5	89.2
Message-size	57.4	62.3	53.4	53.0	68.8	56.4	51.1	61.4	61.0	50.2	60.2		

Fig. 3. Scenario 3

			Drop Policy										
			X	Ascending					Descending				
			Random	Arrival-time	Replications	Relayed-nodes	Time-to-live	Message-size	Arrival-time	Replications	Relayed-nodes	Time-to-live	Message-size
Metric			Delivery Ratio (percentage, more is better)										
Queue Policy	X	Random	46.2	52.7	35.4	23.9	66.8	20.8	38.2	53.1	64.9	27.7	56.7
		Arrival-time	36.2	46.5	30.0	18.0	64.0	17.5	32.6	44.6	59.0	21.8	54.3
		Replications	56.3	59.8	45.2	31.5	70.1	24.8	44.7	60.5	70.6	31.4	60.6
		Relayed-nodes	58.1	66.9	50.0	35.6	69.5	24.1	42.9	60.9	65.2	30.2	61.4
		Time-to-live	34.9	38.5	25.2	18.1	57.3	17.0	36.1	50.0	55.6	23.3	52.8
	Descending	Message-size	54.4	58.3	49.0	43.4	64.3	30.5	47.4	59.0	65.1	37.3	56.9
		Arrival-time	51.9	54.5	43.0	29.5	62.6	23.8	44.2	56.6	67.1	28.5	58.1
		Replications	35.7	39.8	28.9	17.8	52.5	17.2	31.9	44.9	55.0	21.4	50.2
		Relayed-nodes	33.4	37.7	24.3	20.7	52.3	15.9	32.8	49.4	56.7	19.6	53.2
		Time-to-live	60.0	64.9	50.0	32.2	66.0	27.6	45.5	60.6	69.4	35.6	61.9
Message-size	26.7	30.0	21.4	16.0	41.6	17.5	34.7	39.9	49.6	18.6	51.9		
Metric			Overhead Factor (transmissions per message, less is better)										
Queue Policy	X	Random	98.4	80.9	123.3	180.3	139.4	103.4	118.6	105.0	122.4	131.6	98.2
		Arrival-time	103.9	81.0	118.1	178.2	114.0	91.8	105.5	92.1	96.0	130.1	93.5
		Replications	93.1	79.0	114.3	167.7	154.0	123.8	126.7	97.5	127.4	127.8	94.8
		Relayed-nodes	75.2	55.0	94.9	131.0	114.2	117.6	99.8	114.0	134.2	113.5	87.8
		Time-to-live	88.7	80.7	121.3	156.8	118.0	93.5	92.5	71.0	97.1	110.0	86.8
	Descending	Message-size	116.3	101.4	144.8	176.1	192.0	249.3	129.9	107.9	154.1	135.0	103.2
		Arrival-time	101.8	82.3	118.4	173.7	165.7	102.7	116.3	110.4	141.1	135.0	92.3
		Replications	113.6	99.3	127.5	181.4	141.1	90.4	111.1	109.1	97.5	137.8	97.3
		Relayed-nodes	104.5	92.9	141.4	166.7	147.5	117.0	113.7	77.7	108.5	157.0	88.8
		Time-to-live	80.1	59.8	101.5	150.4	157.6	120.7	123.9	139.1	144.8	119.2	86.5
Message-size	38.3	33.4	52.9	68.2	100.7	52.2	44.4	35.5	59.6	72.8	108.0		
Metric			Average Delay (minutes, less is better)										
Queue Policy	X	Random	71.3	68.2	75.4	57.9	58.4	59.2	79.6	72.6	85.7	67.6	72.9
		Arrival-time	79.6	78.6	80.5	58.3	72.6	67.2	85.6	81.7	107.2	70.1	86.6
		Replications	54.5	52.5	64.7	47.6	44.3	51.1	65.2	54.8	66.8	54.0	51.3
		Relayed-nodes	50.9	43.4	74.5	50.2	43.6	51.8	65.1	49.6	81.7	58.7	48.7
		Time-to-live	79.6	76.1	72.0	62.3	93.3	66.4	93.5	93.9	110.0	70.1	91.4
	Descending	Message-size	63.9	61.3	64.1	53.2	57.8	57.8	70.6	67.6	79.6	57.5	58.0
		Arrival-time	48.4	41.2	60.9	47.1	37.5	49.4	64.9	51.9	64.6	51.4	48.8
		Replications	77.3	68.4	80.3	60.0	68.2	62.5	86.7	81.9	100.8	68.0	83.3
		Relayed-nodes	71.4	70.9	72.0	61.3	73.6	65.7	84.6	88.4	99.7	62.5	84.2
		Time-to-live	36.0	33.5	52.6	39.8	30.7	44.9	55.0	43.0	58.5	46.8	40.8
Message-size	68.0	61.0	73.9	59.5	55.3	60.2	87.3	75.3	93.1	65.2	82.2		
Metric			Composite (percentage, more is better)										
Queue Policy	X	Random	55.4	62.6	43.6	36.3	72.8	46.2	44.6	59.4	62.1	40.1	59.3
		Arrival-time	45.6	55.2	39.8	32.3	68.1	43.9	41.5	51.4	52.2	36.2	53.3
		Replications	69.0	73.1	55.2	47.4	79.0	49.2	53.8	71.0	72.6	48.4	70.5
		Relayed-nodes	74.9	85.8	57.4	53.9	83.1	50.8	56.6	73.7	63.2	48.3	73.4
		Time-to-live	47.5	52.0	41.0	34.1	54.6	41.8	42.3	52.8	48.5	40.7	51.7
	Descending	Message-size	60.1	65.0	54.0	52.1	65.1	37.6	52.0	62.9	60.4	49.2	64.6
		Arrival-time	68.1	74.6	55.1	45.8	76.3	50.9	54.4	69.2	70.5	46.7	71.0
		Replications	44.5	52.2	37.2	31.4	59.5	44.6	39.3	49.9	51.8	35.4	51.9
		Relayed-nodes	47.3	51.7	37.1	34.7	56.8	36.5	40.8	53.4	51.9	32.3	54.7
		Time-to-live	81.5	87.8	65.5	52.9	82.6	54.1	59.7	73.8	74.4	55.2	77.5
Message-size	57.6	62.9	50.6	51.1	67.0	53.2	52.8	62.8	60.0	47.6	55.8		

Fig. 4. Composite Scenario Evaluation