Message Lost or Message Taken
- On Message Ferry Selection in DTNs -

Morten Lindeberg and Jan Erik Haavet
Department of Informatics
University of Oslo, Norway
(janehaa,mglindeb)@ifi.uio.no

Sergio Cabrero Barros
Department of Informatics
University of Oviedo, Spain
cabrerosergio@uniovi.es

Vera Goebel and Thomas Plagemann
Department of Informatics
University of Oslo, Norway
(goebel,plageman)@ifi.uio.no

Abstract—**Delay tolerant networking (DTN) exploits the mobility of nodes to serve as message ferries for content delivery. It is the goal of message ferry selection to choose a node that soon moves close to the destination. Proper ferry selection can strongly affect system performance. We analyse this impact for data intensive applications in emergency and rescue (ER) operations. Through extensive simulation studies capturing the entire protocol stack, we compare approaches that have a priori static knowledge about the typical mobility patterns in such operations with evolving knowledge based approaches. Our results show that evolving strategies achieve almost the performance of approaches with static a priory knowledge in ER scenarios at the cost of control overhead, especially in non-DTN like dense networks. The static approaches are prone to malfunction if real mobility strongly deviates from the assumed mobility pattern. We demonstrate that it is important to simulate the entire protocol stack for such applications in DTNs. Delay tolerant forwarding is strongly affected by the underlying layers. Simulation results would be of low relevance for real-world implementations without modelling them.**

I. INTRODUCTION

Delay tolerant networking (DTN) is an active research area developing protocols for (partially) disconnected networks. Typical application domains are sensor networks, vehicle area networks, and emergency and rescue operations (ER). DTN utilizes store-carry-forward solutions, i.e., nodes do not only forward packets, but exploit also the mobility of so-called message ferries in times when the destination is in a separate network partition. Message ferries are used to transport the packets physically closer to the destination.

There are two core approaches for selecting message ferries: First, random-driven approaches that assume no knowledge of mobility. These approaches often use replication to assure that packets are eventually received at the destination. The higher the degree of replication, the higher the degree of epidemic spread, and the higher the delivery probability, evidently at the cost of overhead. Second, knowledge-driven approaches exploit knowledge of mobility patterns to select ferries that are likely to come in contact with the destination. In most cases, such approaches do not exploit message replication. It is essential that the sender knows which nodes can be used as a ferry, and that these nodes are aware of their role and responsibility. Message ferry selection strongly affects the performance of the entire system, both positive and negative, depending on its quality of ferry choice. In ER operations we can assume some knowledge regarding node mobility. In addition, to preserve the lifetime of the involved handheld devices, we regard epidemic spread as too costly in terms of communication overhead. Thus, it is our aim to investigate knowledge-driven approaches for the application domain of ER.

In the literature, we find two main approaches for knowledge-driven message ferry selection: The first approaches utilize static a priori knowledge, where nodes are selected based on knowledge of their future movement [1]. The second approach is referred to as evolving selection. It assumes that there exist message ferries, but instead of using a priori knowledge the approaches probabilistically classify nodes as ferries from topology statistics such as link contacts [2] and route states [3]. Strategies following the evolving approach differ in the way they calculate the probability of nodes to act as ferries. Furthermore, nodes need to exchange the results of their probability calculation. This data exchange has two dimensions of design options: when to exchange (time-based or event-based), and which data to exchange. Time-based data exchange, like in DSMC [4] is performed periodically at a constant rate, while event-based exchange, like in Prophet [2], is triggered by network events like route changes. A priori knowledge determines whether the data exchange must include probability values for all nodes in the network, or only for the destination(s). The latter requires that the destination(s) is known a priori and fixed.

This work aims to better understand the impact of message ferry selection on system performance for knowledge-driven approaches. Therefore, we evaluate two evolving-based approaches (our DSMC strategy and a related approach called Prophet [2]), with one approach with static a priory knowledge that we have developed for ER scenarios with designated ferry nodes. Additionally, we include one hybrid approach that combines static a priory knowledge, with evolving-based knowledge (EOR). As workload in our simulation studies we use transmission of video data envisioned from head mounted cameras to the command and control center in an ER operation. We simulate the entire protocol stack in ns-3 to assure the relevance of our results for real-world deployment. Several experiments are used to study the following questions: (1) How much does a priori knowledge of message ferries help in delivery? (2) What if the mobility patterns do not

978-1-4673-4404-3/12/$31.00 ©2012 IEEE

correspond to the a priori knowledge? (3) What is the effect of simulating the whole protocol stack, as opposed to only node encounters and upper layers? (4) What is the difference of the approaches in terms of performance?

The core insights from our extensive experiments and contributions in this paper are: (1) The evolving approaches achieve almost the performance in choosing message ferries, compared to the static approaches. (2) In cases where the assumed knowledge is wrong in a realistic ER scenario, the evolving approaches perform better than the static approaches. (3) By simulating also the lower layers of the protocol stack, we find a severe drop in performance by not adapting to the lower layer states. Thus, simulating the whole stack, as opposed to only the upper layers is important, but this is neglected by most performance evaluations in DTNs. (4) For the particular domain of ER networking, we find performance gains in avoiding carrier-to-carrier loops and mechanisms of ageing. However, the static parameter that is often used in evolving approaches to achieve ageing should be adapted to the particularities of the scenario.

The outline is as follows: Section II briefly surveys the strategies we analyse. Section III explains our evaluation setup, the results and the analysis. We conclude our findings in Section IV and present future work.

II. MESSAGE FERRY SELECTION STRATEGIES

The application domain of ER is well suited for the development of knowledge-based ferry selection approaches, because the organization of ER follows strict rules that are very similar in many countries. This paper focuses on the fact that a command and control center (CCC) must be established outside the incident area at a fixed location. This is backed up by rescue personnel mobility observations from large catastrophe manoeuvres [5], [6]. As such, the incident cite and the CCC often consist of two network partitions. Personnel will travel between the two partitions, e.g., to enter the incident area, to rest at the CCC, or to carry injured people away from the incident cite.

We now present the four ferry selection strategies that we compare: (1) *Static* that leverage exact knowledge about existing ferries. (2) *EOR* [6], which is a hybrid solution that combines evolving and static knowledge. (3) Our own evolving approach *DSMC* [4]. (4) The well-known approach *Prophet* [2].

A. Static

The strategy assumes that all message ferries are known in advance; they are assigned a certain range of IP addresses. Routing is performed in three stages. (1) Nodes make use of routes established by the local routing protocol, which is OLSR. Packets are only sent one hop at a time, with the intention to forward packets as close as possible to the destination. (2) Local route decisions are cached on each node. If a route breaks, these cached decisions enable that nodes forward packets towards the last working link on the broken route. (3) If there exist no route and no former route is cached, a message ferry is used. Nodes search through their routing tables for entries within the pre-assigned IP address range, i.e., for message ferries. The closest ferry (based on hop-count) with an IP address within this address range is selected. To avoid looping of messages between two or more ferries, we prohibit that nodes assigned as ferries transmit messages to other ferries. This is accomplished by that nodes always check their local IP address, before carrying out routing in stage (3), i.e., message ferry selection.

B. EOR

Emergency Overlay Routing (EOR) [6] is a special purpose overlay routing protocol for ER scenarios. The goal is to identify message ferries that deliver packets with minimum delay. For that purpose, EOR combines a priory static knowledge, with parameters that are evolving during runtime. The statically assigned parameter is the type of unit carrying the network device (N_t), where different nodes are assigned different values. We currently consider two types of units: vehicles and firemen on foot. A device that is carried by a vehicle is considered to provide more stable links than devices carried by personnel and is therefore assigned a higher score. It is expected that such devices have less battery constraints, and lower probability of malfunctioning. Turning to the dynamic parameters, T_c is a measure of the time passed since the nodes' last encounter with the CCC. This requires that all nodes are aware of the CCC IP address. T_c is stored every time a route is discovered to the CCC. The second dynamic parameter is the distance (hop-count) to the potential ferry node, entitled H_c. A ferry value F_c is calculated for each node. The intuition for this approach is: the longer a ferry node is away from the CCC, the higher the probability that it will return soon to the CCC. We calculate the ferry value as follows:

$$F_c = \frac{T_c * N_t}{H_c} \tag{1}$$

We set N_t for vehicle nodes to 5, and for firemen to 1. These values ensure the selection of vehicles over firemen, unless T_c or H_c for the firemen are clearly superior. The values (5 and 1) have been identified as close to optimal in previous experiments over the targeted scenario (ER 2) [6]. EOR chooses the node with the highest ferry value in the network partition to forward the stored packets. Ferry values towards a destination are calculated by exchanging T_c when the members of the network partition change (event-based).

C. DSMC

Dynamic Selection of Message Carriers (DSMC) [4] is inspired by CAR [7], and designed for message ferry selection in ER scenarios. DSMC requires no static a priory knowledge about carriers and instead detects message ferries dynamically based on their probability of delivering packets. Each node estimates its contact probability with all other nodes periodically, and stores this probability in a delivery probability (DP) table. This DP table is exchanged with 1-hop neighbour nodes periodically. When a node receives a DP table from a different node, it updates its own DP table. It only maintains the entry of the node with the best probability of delivery for the destinations. The more time a node spends connected to another node, the higher is its delivery probability. As in [8], we use an exponentially weighted moving average (EWMA) to calculate the contact probability. Node A maintains a table of contact probabilities E_{ab} for every other node B. E_{ab} is updated each second. The calculation is shown in Equation 2, where α is a constant parameter between 0 and 1 that represents the ageing factor of the contact probability.

$$E_{ab} = \begin{cases} (1-\alpha)[E_{ab}]_{old} + \alpha & \text{if } A \text{ in contact with } B \\ (1-\alpha)[E_{ab}]_{old} & \text{if not in contact} \end{cases} \quad (2)$$

Through observations, we discovered performance degradation due to a looping behaviour between ferries. This is remedied by introduced a mechanism that prevents ferries from forwarding packets if their DP is above a threshold entitled cf_limit. In this paper, we set the ageing factor $\alpha = 0.001$ and $cf\_limit = 0.01$. These values are based on systematic studies of well-suited α and cf_limit values for ER scenarios.

D. Prophet

Prophet [2] is a well-known protocol that dynamically calculates the probability of encounters for every node in the network. It is inspired by the movement patterns of people and assumes that nodes often in contact with each other have higher probability of encountering each other again in the future. Each time a node A is in contact with node B, A updates its encounter probability $P_{(a,b)}$ to B:

$$P_{(a,b)} = P_{old} + (1 - P_{old}) * P_{init} \quad (3)$$

P_{init} is a constant representing the initial probability. P_{old} is the value of the last calculated encounter probability. Nodes exchange periodically probability values with each other. For all exchanges, each node first updates all of its probabilities according to: $P_{(a,b)} = P_{old} * \gamma^k$. This causes the value to decrease over time, i.e., ageing. The value γ is a statically set constant. By varying this constant, one can achieve stronger or weaker ageing. k represents the distance in time since the last update. In this paper, we use the values $\gamma = 0.98$ and $P_{init} = 0.75$, as recommended in [2]. Our implementation does not include transitivity properties. In the original research paper, the authors made use of replication by not deleting packets in the storage queue when transmitting to a newly assigned ferry, except when there is no more space in the storage queue. However, to make the strategies more comparable, we do not include the use of replication for Prophet.

III. EVALUATION

For a fair and meaningful comparison of the four strategies, we evaluate them in a common system called Dts-Overlay [9], running on top of a set of standard MANET protocols. Otherwise, strategies would be strongly affected by lower layer protocols, mechanisms of store-carry-forwarding (like the rate of emptying buffers) and the way that ferry values are exchanged. This section introduces Dts-Overlay, our test-setup, workload, network scenarios and metrics.

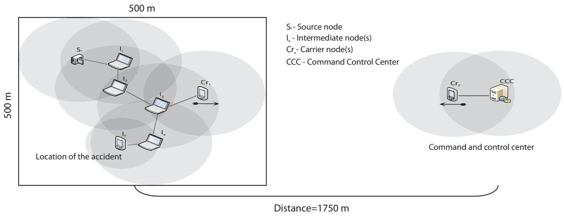

Fig. 1. Scenario ER 1.

A. Dts-Overlay System

Dts-Overlay is a delay tolerant network overlay implemented in C++, incorporated into the ns-3.10[1] network simulator. It runs on all nodes in the network on top of UDP, IP / OLSR routing, IEEE 802.11 MAC and PHY. The overlay facilitates four essential functions: (1) Send and receive primitives for networked applications; (2) methods for storing packets temporarily in a buffer during times when nodes appear disconnected; (3) provisioning of cross-layer information; and (4) CarrierManager to implement arbitrary message ferry selection strategies. The CarrierManager is a new component to enable experiments with multiple strategies. It supports the exchange of probability values as part of the ferry selection routine. Currently, these exchanges are performed periodically every four seconds for all neighbour nodes, i.e., following the time-based approach. For fair comparison, we eliminate effects caused by the different ways the evolving strategies are carrying out the exchanges. This means that we utilize time-based exchanges for each of the evolving strategies, although EOR and Prophet in their original versions utilize event-based exchanges. Future work should study the effect of time-based versus event-based exchanges.

We attributed in [9] high packet loss to lower layers. This loss was eliminated through techniques of cross-layer adaptation. Dts-Overlay avoids sending packets over links where the MAC retransmission queue is filling, indicating a non-working link. Furthermore, links are avoided where ARP fails in resolving the link IP destination address. Finally, the MAC Return function in the MAC layer implementation passes packets to the overlay for temporary storage, instead of dropping them.

B. Test-Setup and System Configuration

The application workload consists of a single unicast stream of the video entitled Foreman[2], resolution CIF (352x288) and 25 fps frame rate as in [9]. The video is pre-encoded with H.264 baseline profile and target bitrate 256 kb/s. The 12-second video stream is continuously repeated for the entire simulation. In all network scenarios we model IEEE 802.11b in ad-hoc mode. The devices simulate direct-sequence spread

---

[1] Available at http//www.nsnam.org.

[2] Obtained from Video Traces Research Group - http://trace.kom.aau.dk/

spectrum (DSSS) modulation, and a constant data transmission mode of 11 Mbps as in [9]. The wireless channel is modelled using the constant speed propagation delay model, and the Friis propagation loss model. Non-unicast transmissions, used by OLSR, are assured the same range as unicast transmissions to avoid so-called greyzones [10].

We include two scenarios that model realistic node movement in ER. To explore the question what is happening if the movements are quite different than assumed in the a priori knowledge, we analyse two core aspects. First, the overall movement pattern is as assumed, however, the ferries do not work as assumed. This is modelled as a variation of ER 1. Second, the overall movement pattern is substantially different than assumed. This is address with a traditional random waypoint scenario.

1) ER 1: The scenario models two network partitions and a set of designated message ferries moving between them (see Figure 1). Ten incident area nodes move according to the Random Walk mobility model with a speed of 2 m/s, 10 s pause time in an area of 500 m x 500 m. Three message ferry nodes pause at the CCC and in the incident area for 60 seconds, allowing data exchange. During the data carrier phase, they move following a straight line at 10 m/s, the distance is 1750 m. Total duration is 3600 seconds, i.e., 1 hour.

A variation of this scenario, entitled ER 1B investigates what happens when the assumed knowledge is wrong. At 1800 seconds into the simulation, one ferry stops in the incident area until the simulation ends.

2) ER 2: This scenario models the mobility of another ER operation based on fire fighter mobility observations and GPS traces [6] from the Asturian Fire Service (Bomberos de Asturias/112). A CCC is located at a fixed position 400 meters from an incident area of size 1000 m x 1000 m. At time $t = 0$, the source node (camera) moves from the CCC to the incident area, where it moves according to the random waypoint mobility for the remaining time. There are four teams consisting of four firemen and a vehicle. All teams stay at the CCC at $t = 0$, and later alternate between rest and intervention phases for 6000 seconds. During a rest phase, teams stay in the CCC for a random time between 300 and 600 seconds. An intervention phase starts with the car and firemen travelling together with 25 m/s speed to a randomly selected point in the incident area. When they arrive at this point, the car stops and firemen move according to random waypoint for a time period randomly selected between 500 and 1500 seconds. Node speed for the firemen, and the camera node as well, is 3 m/s and pause times are randomly selected between 0 and 600 seconds. After the intervention phase, the firemen move towards the car. When all firemen are gathered, they travel back to the CCC with the speed of 25 m/s. The communication range of nodes is limited to $\sim$100 m (ns-3 RxGain parameter set to -16 dB).

3) Random Waypoint: This scenario is similar to that of the random waypoint scenario used in [11]. The area size is 300 m x 1500 m, node count is 50, and node speed is uniformly distributed between 0 m/s and 20 m/s. Communication range is fixed to 100 m by using the range propagation loss model.

4) Metrics: To compare the performance of the strategies, we use four metrics: (1) Rx_v: the percentage of correctly received video packets at the destination; (2) P_l: the percentage of lost packets; (3) Tx_t: the total number of bytes from the packets transmitted at the physical layer; (4) D_e: the average delay of successfully received packets. The delay of buffered and lost packets are not accounted for. Many of the packets that are not delivered to the destination might be buffered in the overlay and do not count as lost. All numbers are averages from five experiment runs, unless otherwise stated, and we present the standard deviation (σ).

C. Results

The strategies are compared in two different configurations of Dts-Overlay. Configuration *C1* with cross-layer adaptations to lower layers as identified in [9]. These adaptations are (1) handing dropped packets at the MAC layer back to Dts-Overlay, (2) reducing the MAC layer retransmission limit to three, (3) avoiding links with a high amount packets queued for retransmission at the MAC layer, and finally (4) avoiding links where ARP cannot resolve the IP address on the link receiver side. Configuration *C2* comprises only adaptation to the route table of the OLSR routing protocol. The obtained performance measures are listed in Table I, and we now summarize the most important results.

ER 1: Packet delivery is generally high in configuration C1. Strategies Static, DSMC and EOR achieve Rx_v at 96%, Prophet 90%. Packet loss is 0%. The biggest difference between the strategies is their bandwidth consumption (Tx_t) and average delay (A_d). Tx_t is 60% higher for Prophet than for Static. A_d is generally high, and heavily affected by the pause time and carrying phase of carrier nodes. Prophet achieves substantially higher A_d at 332.6 s, than the other strategies (at around 200 s).

It is apparent that performance decreases by not adapting to lower layers. Rx_v strongly decreases from C1 to C2. Rx_v is 75% for Static and DSMC, 69% for EOR and 64% for Prophet. Packet loss is high: 20% for DSMC, 21% for Static, 25% for Prophet, and 29% for EOR. It can be observed that A_d is lower than in C1, especially for EOR at 135.3 s. This is probably caused by the fact that several packets in C2 are lost, rather than later retransmitted to another ferry which is the case in C1.

ER 1B: The scenario facilitates mobility where the assumed static knowledge is wrong. This negatively affects the performance of the static approaches. In C1, Rx_v is 61% for Static and EOR. The other strategies perform better: Rx_v is for DSMC at 84%, and for Prophet 90%. Packet loss is still at 0% for all strategies.

In C2, Rx_v is higher for DSMC and Prophet (equally at 65%) than for Static and EOR (48% and 46% respectively). Loss for DSMC and Prophet is substantially higher (at 21% and 27%) than for Static (Rx_v = 13%) and EOR (Rx_v=16%). This is probably because they actively make use of ferries

TABLE I
RESULTS FOR ALL SCENARIOS

Scenario: ER 1					
Config	Metric	Static	DSMC	EOR	Prophet
C1	Rx_v	96 % (σ:0)	96 % (σ:0)	96 % (σ:1)	90 % (σ:3)
	P_l	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)
	Tx_t	317 MB (σ:21)	358 MB (σ:20)	462 MB (σ:41)	517 MB (σ:42)
	A_d	171.6 s (σ:1.1)	202.9 s (σ:15.7)	191.8 s (σ:32.0)	383.7 s (σ:46.8)
C2	Rx_v	75 % (σ:4)	75 % (σ:6)	69 % (σ:2)	64 % (σ:8)
	P_l	21 % (σ:4)	20 % (σ:6)	29 % (σ:2)	25 % (σ:8)
	Tx_t	309 MB (σ:23)	363 MB (σ:23)	440 MB (σ:35)	554 MB (σ:33)
	A_d	159.5 s (σ:2.3)	173.4 s (σ:6.7)	135.3 s (σ:9.6)	332.8 s (σ:44.5)
Scenario: ER 1 B					
Config	Metric	Static	DSMC	EOR	Prophet
C1	Rx_v	61 % (σ:0)	84 % (σ:1)	61 % (σ:2)	90 % (σ:2)
	P_l	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)
	Tx_t	247 MB (σ:17)	345 MB (σ:25)	340 MB (σ:29)	522 MB (σ:23)
	A_d	170.2 s (σ:2.1)	208.1 s (σ:17.4)	205.9 s (σ:41.5)	379.8 s (σ:35.4)
C2	Rx_v	48 % (σ:2)	65 % (σ:5)	46 % (σ:1)	65 % (σ:13)
	P_l	13 % (σ:2)	21 % (σ:5)	16 % (σ:1)	27 % (σ:11)
	Tx_t	246 MB (σ:15)	340 MB (σ:26)	329 MB (σ:26)	524 MB (σ:38)
	A_d	158.6 s (σ:2.1)	174.8 s (σ:7.1)	143.0 s (σ:14.5)	351.0 s (σ:43.3)
Scenario: ER 2					
Config	Metric	Static	DSMC	EOR	Prophet
C1	Rx_v	9 % (σ:5)	21 % (σ:7)	13 % (σ:7)	6 % (σ:2)
	P_l	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)
	Tx_t	58 MB (σ:26)	143 MB (σ:41)	90 MB (σ:35)	38 MB (σ:13)
	A_d	2309.1 s (σ:492.1)	2541.7 s (σ:354.5)	2099.9 s (σ:272.7)	2777.9 s (σ:406.9)
C2	Rx_v	15 % (σ:9)	25 % (σ:7)	19 % (σ:14)	4 % (σ:1)
	P_l	28 % (σ:19)	31 % (σ:19)	29 % (σ:36)	12 % (σ:17)
	Tx_t	126 MB (σ:28)	237 MB (σ:51)	153 MB (σ:93)	72 MB (σ:88)
	A_d	1794.4 s (σ:725.8)	1663.1 s (σ:396.6)	1727.6 s (σ:448.2)	3018.1 s (σ:698.4)
Scenario: Random Waypoint					
Config	Metric	Static	DSMC	EOR	Prophet
C1	Rx_v	80 % (σ:9)	72 % (σ:4)	88 % (σ:10)	73 % (σ:4)
	P_l	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)	0 % (σ:0)
	Tx_t	1237 MB (σ:222)	732 MB (σ:110)	1192 MB (σ:136)	900 MB (σ:236)
	A_d	261.5 s (σ:49.3)	398.4 s (σ:68.8)	235.8 s (σ:65.8)	403.3 s (σ:97.9)
C2	Rx_v	32 % (σ:4)	36 % (σ:5)	34 % (σ:3)	34 % (σ:6)
	P_l	66 % (σ:5)	61 % (σ:5)	63 % (σ:2)	64 % (σ:6)
	Tx_t	522 MB (σ:36)	343 MB (σ:34)	453 MB (σ:28)	398 MB (σ:20)
	A_d	34.0 s (σ:33.3)	54.7 s (σ:20.6)	34.4 s (σ:16.4)	36.8 s (σ:14.6)

for transmission of packets, which increases Rx_v but induces higher packet loss.

ER 2: The scenario is sparse, therefore packet reception is low. In configuration C1, DSMC (Rx_v at 21%) has the best performance. EOR and Static achieve Rx_v at 13% and 9%, Prophet 6%. Packet loss is 0%, and Tx_t is lower compared to the other scenarios. Average delay is substantially larger in ER 2, than in the ER 1 scenario. This can be attributed to mobility; ferry nodes spend a substantial amount of time away from the CCC.

Packet reception is higher in C2 (than in C1) for all strategies, except Prophet (down from 6% to 4%). Rx_v is 15% for Static, 25% for DSMC, and 19% for EOR. The higher reception comes at the cost of overhead and loss. P_l is close to 30% for Static, DSMC and EOR, and 12% for Prophet. Concerning overhead, we notice for Static a drastic increase in Tx_t (more than a factor of two).

Random Waypoint: In C1, Rx_v is now in the range between 72% and 88%, and loss is 0% for all strategies. EOR (at 88%) performs better than DSMC (at 72%), but standard deviation is high due to the random mobility. Tx_t varies between 732 MB and 1237 MB, and delay between 235.8 s and 403.3 s. Standard deviation is high. Turning to C2, we see that performance is significantly lower. Rx_v is down to less than 40% for all protocols, and loss varies between 61% and 66%. Tx_t is strongly reduced for all protocols, this also applies to the average delay. As an example, A_d for Prophet is reduced from 403.3 s to 36.8 s.

D. Analysis

This analysis focuses on five topics: (1) The performance of static versus evolving approaches; (2) The effect of ageing when the assumed knowledge is wrong; (3) Occurrences of ferry-to-ferry looping; (4) The importance of modelling and adapting to lower layers; (5) The overhead induced by the evolving approaches.

1) Static vs. Evolving Approaches: In scenario ER 1, the static approaches show only limited gains over the evolving approaches. In ER 1B, the evolving approaches perform better. In ER 2, DSMC actually outperforms both EOR (the hybrid approach) and Static in terms of packet reception. It can be expected that DSMC achieves the highest reception, because all nodes that have been in contact with the destination are considered ferries. Static, which only "knows" vehicles as ferries achieves significant lower packet reception. EOR chooses vehicles over firemen, and therefore misses some opportunities of message ferrying. It is important to note that EOR chooses vehicles over firemen since these nodes are

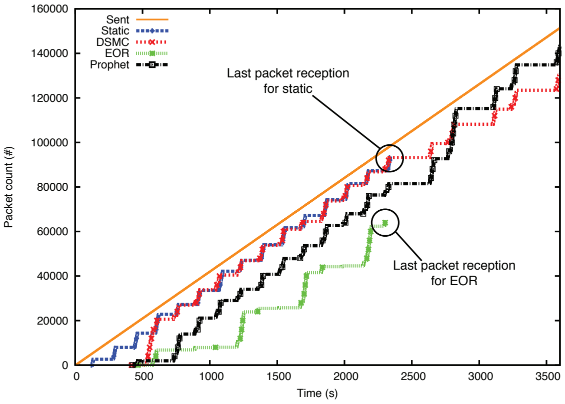

Fig. 2. ER 1B - Packet reception in a single experiment run

expected to be more reliable. Those nodes have higher storage capacity, and are not constrained by battery lifetime. This effect is however not captured in our simulation studies.

The evolving approaches are expected to achieve higher delay in the initial phase due to the learning phase. Delay in ER 1 is slightly higher for DSMC than for Static, and this can be attributed to the initial learning phase. Even though this initial learning phase also affects EOR, it still achieves low average delay (in C2 ER 1 actually the lowest). The reason is that EOR chooses ferries (when multiple exist) that are the most likely to leave the earliest, i.e., the ferry that has been the longest away from the CCC. This decreases delay, but at the cost of higher packet loss in the configurations where we do not adapt to lower layers. This loss can be attributed to the fact that it takes quite a long time for the routing protocol to identify that the link is no longer existing. Thus, when a ferry leaves the partition, the link is broken, but DtsOverlay in our setup still tries to send packets over this link.

2) *Effect of Ageing:* It is important to consider scenarios where the assumed a priory knowledge is wrong. We investigate this in two scenarios: (1) Scenario ER 1B, where one ferry node stops moving after 1800 seconds, and remains stationary in the incident area for the rest of the scenario. (2) Random waypoint scenario, where all nodes move randomly. The effect of wrong assumed knowledge in case (2) had lesser effect than for case (1), where DSMC and Prophet clearly outperform the static strategies. The reason is that these approaches incorporate ageing. Ferries that have not seen the destination for a long time are disregarded as ferries. To illustrate this effect, we include in Figure 2 the amount of packets received over time for a single run for all strategies in Scenario 1B. It can be observed that all strategies perform relatively well until some point in time after the ferry stops (at $t = 1800$ s). Static and EOR elect for this particular run the stopped node as ferry and are from thereon not delivering any packets. The strategies elect the stopped ferry for two different reasons. For EOR, since it has been the longest away from the CCC. For Static, most probably since it appeared first in the routing table among ferries within 1-hop reach. DSMC and Prophet, disregard the stopped ferry after some time, thus packets are after some time again delivered to the destination through other working ferries.

The importance of ageing (ageing factor) has also been in identified in [12]. This work compares five evolving approaches and finds that ageing in Prophet negatively affects performance, because carrier nodes “forget” that they have been in contact with the destination. It should be noted that the original Prophet factor is used for these experiments, however the mobility scenario is very different from the one used in the original research paper [2]. In the comparison study [12], node speed is 0.5-1.5 m/s, and scenario duration is in the order of days. In the original Prophet paper [2], node speed is 0-20 m/s, and the scenario duration in the order of hours. We found in ER 1B positive effects of Prophet’s ageing mechanism. This means that ageing can negatively impact performance if the factor does not match well with the rate of contact between ferries and the destination, and can improve performance if the factor matches well.

3) *Ferry-to-Ferry Looping:* Prophet in scenario ER 1 achieves lower packet reception than other strategies, but bandwidth consumption is still higher. We attribute this to ferry-to-ferry looping. Investigation of packet traces revealed several occasions of packet exchanges between ferry nodes. This comes at the cost of higher Tx_t, higher average delay, and lower packet reception. For the particular scenario of ER 1, ferry nodes moving towards the CCC will meet one or more ferry nodes moving from the CCC (i.e., in the opposite direction). The ferry moving towards the CCC should avoid packet transmission to the other ferry node, to avoid transmission in the opposite direction to the destination. Even worse, the packet might end up looping back and fourth among the ferries. In EOR, the returning nodes will be attributed a lower probability score than the ferries moving towards the CCC. This prevents such loops. Static and DSMC both operate with mechanisms that prevent this behaviour.

4) *Importance of Modelling Lower Layers:* In general, we see that modelling lower layers and considering their properties is very important for wireless networks. This is confirmed by substantial performance gains from adapting to lower layers (C1), over not adapting to lower layers (C2). As an example, all strategies experience more than 60% packet loss in the random waypoint scenario in C2. In C1, packet loss is 0%, and reception is higher at least with a factor of two. A substantial amount of research efforts that target DTNs and ferry selection does not simulate the entire protocol stack during performance studies. We argue that this lowers the usefulness of the results. As an example, [2] does not describe any loss in lower layer protocols when evaluating Prophet in a scenario similar to our random waypoint scenario. To ensure accuracy and realism, we stress the importance of incorporating more accurate lower layer models in DTN research.

5) *Overhead Discussion:* The evolving approaches introduce control packet overhead (the exchange of ferry values). In dense network partitions, this does not scale well with

an increasing number of nodes. The overhead is strongly affected by the way these exchanges take place. In time-based approaches, overhead can be controlled by lowering the frequency of exchanges at the cost of lower accuracy. For event-based approaches, especially if the network has high mobility, we can expect an exponential increase in topological events, e.g., route changes. Thus, event-based approaches in dense networks face severe problems of scalability.

IV. CONCLUSION

This paper studied how a priori static knowledge can help in message ferry selection. This has to the best of our knowledge not been studied before. Our analysis of four different strategies revealed that a priory knowledge gives very limited gains in performance; the evolving approaches adapt in many cases even better to network dynamics. They performed superior in cases where the assumed a priory knowledge was incomplete. Our own strategy DSMC achieved the best performance in the targeted ER scenarios: It avoids ferry-to-ferry-looping, and incorporates ageing. Importantly, our evaluation showed that lower layers protocols plays an important role in message ferry selection and DTNs. The performance gains of a highly tuned message ferry selection mechanism, are typically low compared to the gains that can be obtained from adapting to lower layer protocols.

In the future, we plan to investigate mechanisms to achieve "dynamic" ageing for evolving approaches. In addition, evaluate means to lower overhead. The exchange of probability values causes control traffic that does not scale well with high node densities. To manage this, we aim to further investigate the use of non-intrusive clustering for ferry selection. Such an approach would not rely on probability exchanges, but on topological information found locally on each node when applying MANET routing. Proactive routing protocols such as OLSR, store network topology in an adjacency matrix that can be used to identify network partitions through clustering algorithms. Nodes that tend to move in, out or preferably between such clusters can be seen as good candidates.

Acknowledgements: This work is funded by the VERDIKT program of the Norwegian Research Council, through the DT-Stream project (no. 183312/S10).

REFERENCES

[1] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *International symposium on Mobile ad hoc networking and computing (MobiHoc)*. ACM, 2004, pp. 187–198.

[2] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, pp. 19–20, July 2003.

[3] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," in *International Conference on Computer Communications (INFOCOM)*. IEEE, 2006, pp. 1–11.

[4] J. E. Haavet, "Dynamic selection of message carriers in a sparse and disruptive MANET," Master's thesis, University of Oslo, Department of Informatics, 2011.

[5] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, "Modeling mobility in disaster area scenarios," *Performance Evaluation*, vol. 66, no. 12, pp. 773 – 790, 2009.

[6] S. Cabrero, X. García Pañeda, T. Plagemann, D. Melendi, and R. García, "Towards reliable video transmission over sparse MANETs in emergencies," *Informatica na educacao: teoria & pratica*, vol. 14, no. 1, 2011.

[7] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *World of Wireless Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2005, pp. 183–189.

[8] H. Dang and H. Wu, "Clustering and cluster-based routing protocol for delay-tolerant mobile networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 1874–1881, June 2010.

[9] M. Lindeberg, S. Kristiansen, V. Goebel, and T. Plagemann, "MAC Layer Support for Delay Tolerant Video Transport in Disruptive MANETs," in *IFIP/TC6 NETWORKING*. Springer, 2011, pp. 106–119.

[10] H. Lundgren, E. Nordströ, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *International workshop on Wireless mobile multimedia (WOWMOM)*. ACM, 2002, pp. 49–55.

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

[12] E. Pagani and G. P. Rossi, "Utility-based forwarding: A comparison in different mobility scenarios," in *Proceedings of the third ACM international workshop on Mobile Opportunistic Networks (MobiOpp)*. ACM, 2012, pp. 29–36.