Computer Communications 48 (2014) 111–120

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Resilience and opportunistic forwarding: Beyond average value analysis

Fredrik Bjurefors [a,*], Merkourios Karaliopoulos [b], Christian Rohner [a], Paul Smith [c], George Theodoropoulos [b], Per Gunningberg [a]

[a] Uppsala University, Department of Information Technology, Box 337, 75105 Uppsala, Sweden
[b] Center for Research and Technology Hellas, 38334 Volos, Greece
[c] AIT Austrian Institute of Technology, 2444 Seibersdorf, Austria

ARTICLE INFO

Article history:
Available online 16 April 2014

Keywords:
Opportunistic networking
Forwarding
Routing
Simulations

ABSTRACT

Opportunistic networks are systems with highly distributed operation, relying on the altruistic cooperation of highly heterogeneous, and not always software and hardware-compatible, user nodes. Moreover, the absence of central coordination and control makes them vulnerable to malicious attacks. In this paper, we study the resilience of popular forwarding protocols to a representative set of challenges to their normal operation. These include *jamming* locally disturbing message transfer between nodes, *hardware/software failures* and incompatibility among nodes rendering contact opportunities useless, and *free-riding* phenomena. We first formulate and promote the *metric envelope* concept as a tool for assessing the resilience of opportunistic forwarding schemes. Metric envelopes depart from the standard practice of average value analysis and explicitly account for the differentiated challenge impact due to node heterogeneity (device capabilities, mobility) and attackers' intelligence. We then propose heuristics to generate worst- and best-case challenge realization scenarios and approximate the lower and upper bounds of the metric envelopes. Finally, we demonstrate the methodology in assessing the resilience of three popular forwarding protocols in the presence of the three challenges, and under a comprehensive range of mobility patterns. The metric envelope approach provides better insights into the level of protection path diversity and message replication provide against different challenges, and enables more informed choices in opportunistic forwarding when network resilience becomes important.

1. Introduction

In opportunistic networks, nodes store, carry, and forward messages when they encounter other nodes using short-range wireless communication. This store-carry-forward (SCF) transport service enables the data flow in the network despite the absence of simultaneous end-to-end connectivity. Yet, the network is a system with highly distributed operation, relying on the good will and cooperation of highly heterogeneous, and not always software and hardware-compatible, user nodes. Moreover, the absence of central coordination and control makes it an easier target for malicious attacks.

Inherent resilience against these challenges to the network operation is provided by data replication. Ideally, data travel in the network over diverse space–time paths, involving disjoint physical spaces and different network nodes. In practice, however, the actual data transfer diversity is highly dependent on the mobility patterns of nodes and the rules of the particular forwarding protocol. In general, forwarding protocols prioritize different performance characteristics such as message delivery ratio or buffer usage, and assign different importance to individual nodes during the data transfer. This, in turn, may render them more vulnerable to a particular type of challenge and more resilient to another.

In general, the performance degradation of opportunistic forwarding in the presence of challenges has been dealt with in literature both analytically [1–3] and with simulations [4–6]. Common to all these works is that the opportunistic forwarding performance in the presence of a challenge is assessed through averages values of the performance metrics, usually computed over several simulation runs.

On the contrary, in this paper, we compute and plot *metric envelopes*, whose upper and lower bounds reflect the best- and worst-case response of a metric, *e.g.*, message delivery ratio, to different realizations of a challenge. The motivating remark is that a simple challenge, such as "*K* selfish nodes" or "*M* jamming devices" can have a widely different impact on the performance of the opportunistic forwarding, depending on *which K* nodes behave selfishly or

* Corresponding author.
E-mail addresses: fredrik.bjurefors@it.uu.se (F. Bjurefors), mkaraliopoulos@iti.gr (M. Karaliopoulos), christian.rohner@it.uu.se (C. Rohner), paul.smith@ait.ac.at (P. Smith), per.gunningberg@it.uu.se (P. Gunningberg).

http://dx.doi.org/10.1016/j.comcom.2014.04.004
0140-3664/

where the M jammers will be physically placed. The metric envelopes implicitly account for the heterogeneity of the opportunistic network nodes in terms of device capabilities and mobility patterns, as well as the varying intelligence of attackers. At the same time, they provide insights that single average values do not. The breadth of the envelope is an indication of how predictably a protocol will perform in the presence of a given challenge; or, equivalently, how much risk is involved in using the protocol in this case. Hence, a protocol with tight metric envelopes may be occasionally preferable to another with better average scores but higher spread of values.

Drawing on earlier work in [7] we use metric envelopes to assess the resilience of three popular forwarding protocols to three representative types of challenges: occasional *software/hardware failures*, *e.g.*, due to incompatibility of the software/hardware the encountered devices may use; intentional *jamming*, a typical example of malicious behavior; and *free-riding*, is a classical instance of non-cooperative behavior emerging in networked settings lacking central coordination and control functionality. The *exact* computation of the metric envelope values for these challenges would require enumerating *all* possible challenge realizations, *e.g.*, combinations of K selfish nodes or placements of the M jammer nodes in the physical space. Clearly such an enumeration becomes computationally intractable already for moderate and even small values of K and M. Therefore, we propose heuristics (cues) for *inferring* "best"- and "worst"-case scenarios for each challenge and *approximating* the respective metric envelopes. The derivation of best- and worst-case partitioning of nodes into software/hardware compatible groups are formulated as instances of the community detection and weighted coloring problems, respectively; jammers are placed in the areas that rank highest (resp. lowest) with respect to the density of encounters; and free riders are let coincide with the most (least) central nodes with respect to message delivery.

We demonstrate the use of envelope metrics and the additional information they can deliver through simulation scenarios with various synthetic and experimental mobility traces. The envelopes can provide arguments in favor of one protocol over the other when they are indistinguishable with respect to average performance values. Their width provides an indication of how much performance differentiation is possible in the presence of a given challenge and given node mobility patterns and how well random simulation runs may fail in predicting the impact of a challenge.

In summary, the contributions of this paper are highly methodological and include: (i) the formulation and promotion of the metric envelope concept as a tool for assessing the resilience of opportunistic forwarding schemes in a way that explicitly accounts for the node heterogeneity (device capabilities, mobility), and when relevant, attacker's intelligence (Section 2); (ii) the proposal of heuristics for approximating the worst- and best-case scenarios for representative challenges (Section 3); and (iii) the demonstration of the methodology in the assessment of three popular forwarding protocols under different challenges and mobility patterns (Section 4). We position this work within the broader literature on opportunistic network resilience in 5 and discuss research directions out of it in Section 6.

2. Assessing resilience: envelopes instead of average values

To assess the performance of forwarding protocols, we consider two standard performance metrics, the message delivery ratio and delay. The message delivery ratio equals the fraction of messages that reach their destinations out of those generated at their sources (ignoring replicas). For every delivered message, message delay equals the time elapsed between the message generation epoch and its arrival at the destination node.

However, and contrary to earlier studies in literature, we are interested in the full range of values a metric can obtain in the presence of a challenge. For example, the impact of K free-rider nodes may vary considerably depending on the importance of the specific K nodes that exhibit this behavior for the forwarding process. Likewise, there are many different ways to place K jammer nodes with jamming radius r_{jam} in the physical space, each placement affecting differently the forwarding operation.

To introduce some terminology that is necessary for the rest of the paper, jamming is a *challenge instance*, which is parameterizable by certain variables such as the number of jamming nodes and their jamming radius. We use the term *challenge realization* to denote a specific implementation of a challenge; for example, a jamming realization describes where exactly the K jamming nodes with jamming radius r_{jam} are placed. On the other hand, *challenge parametrization* denotes the full set of all possible challenge realizations for given values of the challenge parameters. Therefore, "K jammers of radius r_{jam}" is a challenge parameterization, *i.e.*, a shortcut term for all possible challenge realizations involving K jammers of r_{jam} jamming radius.

An example metric envelop diagram is shown in Fig. 1 for a single-parameter challenge. It plots the best- and worst-case values of a metric as the challenge parametrization varies, whereby performance is assumed to be monotonically increasing with the metric value. Each single point at the x-axis corresponds to a certain parametrization and the respective best- and worst-case values enclose (hence, the term envelope) the outcomes of all its realizations. The intermediate curve, between the best- and worst-case, corresponds to the outcome of a random realization or the average of more than one random challenge realizations.

The motivation for promoting envelop diagrams over single average-value curves roots back to longtime practices in engineering different entities, ranging from a single link [8] to a whole system [9]. In all cases, the requirement is to secure an availability of some nines (*e.g.*, three nines corresponds to an availability of 99.9%). Hence, it is much more important for an engineer/designer to know how often performance degrades below some threshold and plan for countermeasures that can make up for this degradation. In the case of opportunistic networks, envelop diagrams explicitly account for the heterogeneity of network nodes with respect to their mobility and hardware/software capabilities and add another dimension to the comparison of the opportunistic forwarding protocols. Since the spread of the envelope is also a measure of the uncertainty/risk related to a certain challenge parametrization, it is possible that one forwarding protocol be preferable to another with higher average performance but broader envelope.

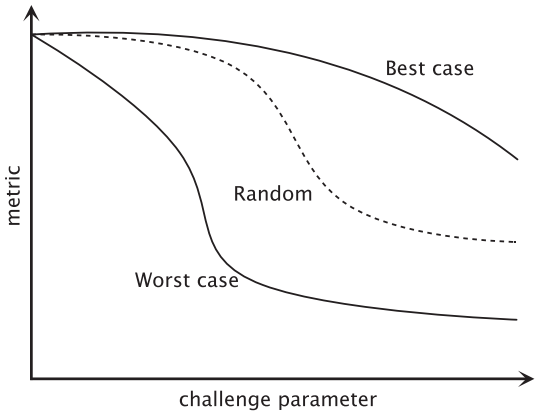


Fig. 1. Metric envelope.

As a final note, envelope diagrams have little to do with confidence intervals, even if they both depict intervals of values. Confidence intervals provide probabilistic guarantees about a point estimate, usually the sample average. They typically assume normal distribution of sample values and get tighter with increasing sample size. In a way, confidence intervals try to suppress the variance of a sample to derive a better point estimate and this may be costly in terms of computations. Envelope diagrams, on the other hand, seek to rather uncover the full variance. Although this can, in principle, bear higher cost, in the following section we propose heuristics that try to minimize it. We comment further on the relation between envelopes and confidence intervals in Section 4.2.

3. Heuristics for performance envelope derivation

We consider three representative challenge instances: software/hardware failures and incompatibilities, jamming, and free-riding phenomena. Note that there is no systematic way to *a priori* find their *actual* extreme realizations and their exhaustive enumeration is computationally expensive. Therefore, we resort to *challenge-specific heuristics* in order to infer best- and worst-case realizations that approximate the envelope bounds for each challenge parametrization.

3.1. Software/hardware failures

This is an instance of unintentional wastage of individual contacts, which may be caused by transient failure of some device component, software or hardware, and may, in turn, hinder the node discovery or data transfer processes. The occurrence of such failures is typically more frequent when there are incompatibilities in the operating systems of the devices or the software/middleware controlling their wireless interfaces.

Modeling-wise, we assign to each encounter a probability of failure, p_f, which is the sum of two factors. The first one, p_{fr} is the baseline common failure probability for all encounters (could be zero, as well). The second probability, p_{fc}, depends on the compatibility of the encountered nodes in terms of software/hardware. The assumption is that such incompatibilities render contact failures more frequent.

To derive then the best- and worst-case scenarios for this challenge, we partition the node set of the opportunistic network into K respectively proper software/hardware-oriented *compatibility groups*. The assumption is that nodes within the same group are perfectly compatible with each other so that their encounters are subject to contact failure probability p_{fr}; whereas encounters between nodes of different compatibility groups fail with increased probability $p_{fr} + p_{fc}$. Ideally, to generate the best-(worst-) case scenario, we would like to maximize the number of encounters between compatible (resp. incompatible) nodes.

Formally, if V is the set of nodes, we seek to derive K-partitions $(V_1, V_2, \ldots, V_K)$ of V with $\bigcup_i V_i = V$ and $V_i \bigcap V_j = \emptyset$, for all $i \neq j$. This partitioning is carried out through a two-step process. First, we derive the weighted contact graph $G_c = (V, E_c; w_c)$ of the network out of the trace of node encounters (ref. Section 4.1). The vertices of G_c correspond to the network nodes and the weights $\{w_c\}$ of the edges to the count of the pairwise encounters between their incident vertices over the trace duration. This step is common for both the best- and worst-case challenge realization. On the contrary, the second step differs.

Best-case scenario: We apply a *modularity-maximizing non-overlapping community detection algorithm* over the weighted contact graph. In general, community detection algorithms [10] identify clusters of nodes (*a.k.a* communities) that are tightly linked with each other as opposed to other nodes in a network. Modularity, on the other hand, is the de-facto criterion for assessing the quality of a given community structure, despite reservations about its sensitivity to smaller communities (*e.g.*, [11]). It also serves as a direct optimization objective for some algorithms such as those in [12,13]. For weighted graphs, it equals [14]

$$Q = \sum_{c \in C} \left[\frac{l_c}{L} - \left(\frac{d_c}{2L} \right)^2 \right] \tag{1}$$

where the summation is carried over the community set C, L is the sum of the weights of all edges in the graph, l_c is the sum of weights over edges lying fully within community c, and d_c the respective sum over the full set of edges incident to nodes in c. Modularity ranges in the interval $[-1/2, 1]$ [15], values above 0.3–0.4 suggesting strong community structure.

Given the context of weights in our weighted graph, output of the community detection algorithm are K node communities, wherein nodes tend to encounter more frequently with each other than with nodes of other communities. By then drawing a one-to-one correspondence between compatibility groups and communities, we minimize the frequency of encounters between incompatible nodes.

Worst-case scenario: We carry out a minimum-cost K-coloring of the weighted graph, whereby the aim is to minimize the overall count of encounters between nodes that are assigned the same color. Formally, letting

$$x_{ik} = \begin{cases} 1 & \text{if node } x \text{ is assigned color } k \\ 0 & \text{otherwise} \end{cases}$$

and $S_C = 1, 2, \ldots K$ be the set of colors, the task can be described by the following nonlinear integer programming problem

$$\begin{aligned} &min \sum_{\substack{i,j \in V \\ i \neq j}} w_{ij} \sum_{k \in S_C} x_{ik} x_{jk} \\ &s.t. \sum_{k \in S_C} x_{ik} = 1 \quad \text{for each } x \in V \\ &x_{ik} \in \{0, 1\} \end{aligned} \tag{2}$$

In general, graph coloring problems are known to be NP-hard; yet, simple approximate algorithms are available (*e.g.*, Ref. [16]). Finally, drawing a one-to-one correspondence between compatibility groups and color classes, we minimize the frequency of encounters between compatible nodes.

3.2. Jamming

Jamming is an instance of intentional challenge to (*a.k.a* attack against) the network with the malicious intention to degrade its performance. The jammer's intelligence may vary. To maximize network damage, the jammers could be located in the most densely populated areas, where most contacts take place, such as train and subway stations. Compared to software/hardware failures, jamming introduces strong *spatial correlation* in the pattern of wasted contact opportunities.

Our jammer's model assumes blocking of any communication between node pairs if even one of the encountered nodes lies within the jammer's range, r_{jam}. Hence, the existence of jammers eliminates contact opportunities from the original contact trace to generate a residual contact trace that can, thereafter, be analyzed as usual. The jammers are homogeneous with a jamming radius $r_{jam} = 50$, in line with current technology.[1] We generate a square tiling of the simulated area and place a variable number of

[1] http://jammerfun.com/latest-high-power-12w-4g-lte-cell-phone-wifi-signal-jammer-blocker-with-remote-control.html.

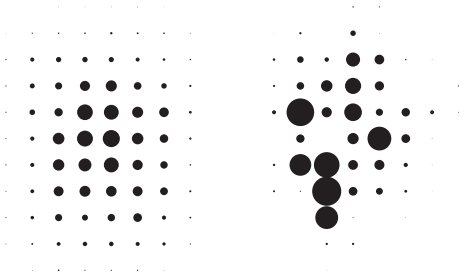


Fig. 2. Contact map for the RWP and SPMBM scenario. The size of the circles indicates the number of contacts within the area.

jammers in the centers of the squares so that the jammed area accounts for up to 50% of the simulation area.

To determine worst/best-case realization, we first generate a contact map that reflects the count of encounters in each tile of the simulated area, as shown in Fig. 2. Then the worst (best)-case scenarios consist in selectively placing the jammers at the centers of the squares with the most (resp. fewest) contacts over the trace duration.

3.3. Free-riding

Free-riders are a common phenomenon in all networks necessitating some form of voluntary cooperation and mutual contributions (*e.g.*, [17]). In opportunistic networks such nodes generate and receive messages, participate in the operation of the protocol, but are not willing to forward the messages of others. The impact of free-riders becomes more profound when they coincide with the most *central* nodes, *i.e.*, nodes that are highly mobile and most frequently contribute to space–time paths traversed by the produced messages.

To guide the selection of free-rider nodes, we first assess their centrality. The centrality index we use is reminiscent of the way Hui et al. compute nodes' centralities in [18]. We generate M messages with randomly chosen source and destination nodes and time epochs and forward them optimally over the traces, assuming *a priori* knowledge of future contact opportunities. We then count how many times each node participates in the shortest space–time paths of these messages and set the centrality indices to the normalized count values. The parameter M is chosen such that the ranking of nodes are stabilized, which for most traces is achieved when 5000 messages are sent.

Then, as best-case realization, free-riders are mapped to the nodes with the lowest centrality values, *i.e.*, nodes that would be expected to contribute little to the forwarding process under optimal operation; whereas, for the worst-case realization, the free-riders are chosen to be the nodes with the top-centrality values.

In all three cases, an encounter that could otherwise result in data exchange may not actually do so. Each challenge essentially thins out the original density of contact opportunities, as this emerges from the pure mobility of the network nodes. In general, the resilience of each forwarding protocol to these challenges depends on the way it manages its message replication budget and its capacity to find uncorrelated space–time paths, both with respect to the physical space these paths traverse and the user nodes that realize them. However, additional variance in the forwarding performance is induced by the randomness (software/hardware failures, free-riders) or intelligence (jamming) these challenges are actually realized. The evaluation in Section 4 seeks to illustrate and alert against this variance.

4. Experimentation methodology and results

4.1. Methodology: tools, traces, protocols

To demonstrate the use of the metric envelope approach, we carry out experiments with two performance analysis tools: the ONE simulator [19] and the trace-parsing Space–Time-Graph tool described in [20]. ONE provides a broad variety of mobility model and forwarding protocol implementations and is the de-facto simulator for opportunistic networks. The Space–Time-Graph tool, on the other hand, parses traces of encounters and yields significantly faster run times for the narrower selection of protocols it supports.

We have chosen to analyze three most popular forwarding schemes, the Epidemic, Spray and Wait, and Prophet protocols. *Spray and Wait* [21] is selected as a representative instance of randomized (controlled-flooding) forwarding and simulations are carried out with both its source (SSW) and binary (BSW) versions. Prophet [22], on the other hand, is one of the very first and extensively studied instances of utility-based forwarding. The Prophet variants used in our experiments implement the modifications to the original protocol described in [23] and carry out both constrained (as in the Spray and Wait protocol) and unbounded (as in the Epidemic protocol) message replication.

Our evaluation uses several traces of pairwise node encounters with different properties, as summarized in Table 1. Two of these traces have been synthesized from mobility models built in the ONE simulator. These traces also log the physical coordinates of the nodes' movement and support the experimentation with the jamming challenge (ref. Section 3.2). The remaining traces directly report Bluetooth sightings within groups of users carrying iMotes. In more detail:

Shortest Path Map Based Movement (SPMBM) traces: Nodes move between two randomly chosen locations by following the shortest path along connected edges representing roads on a map [19]. In our case, a street map of Helsinki is used. The simulation area is 4500×3400 meters and includes 126 nodes. The SPMBM model has a strong behavioral modeling element resulting in structured node mobility patterns. A second SPMBM trace with 86 nodes and the same settings is used to represent a sparse network with otherwise the same properties.

Random Waypoint (RWP) traces: Nodes move at random speed and in random direction within a specified area, making pauses every time they complete a fixed-direction trip. Well known theoretical results suggest that when the mobility of nodes is homogeneous, the distribution of pairwise inter-contact times can be well approximated by an exponential distribution [24] and the node density within a square area experiences a peak at the center of the area and gradually declines away from it with fixed density contours resembling concentric rectangles [25].

Haggle traces: These five well-known experimental traces have been gathered in the context of the Haggle Project [26]. They include Bluetooth sightings between users carrying iMotes during the experiments ('internal' contacts) but also encounters with other Bluetooth-enabled devices ('external' contacts) in the vicinity of iMote carriers. Herein we analyze the internal contacts that represent data transfer opportunities among the experiment participants, assuming that they are all equipped with always-on devices; that is, each internal Bluetooth sighting is assumed to correspond to a contact whereby nodes can exchange information. The experimental settings for the generation of the Haggle traces are detailed in [26].

For the SPMBM and RWP traces, we use the same simulation area size and number of nodes. The message source and destination nodes are chosen at random for all traces.

Table 1
Characteristics of experimentation datasets.

Configuration	Cambridge	Infocom05	Infocom06	SPMBM regular/sparse	RWP
Collection	iMote	iMote	iMote	ONE	ONE
Duration (days)	6	4	4	0.5/0.5	0.28
Scan time (sec)	5–10	5–10	5–10		
Granularity (sec)	120	120	120	1/1	1
Mobile devices	12	41	78	126/86	126
Stationary dev.	0	0	20		
# of Contacts	6732	28,216	227,657	30,959/7277	30,000

4.2. Envelopes vs. average-values

Before proceeding with the derivation of envelopes for the traces and protocols of Section 4.1, we present an example of how the average-value analysis compares with the envelope approach and to what extent random challenge realizations can approach the best- and worst-case scenarios. The example we consider is the operation of the source Spray and Wait protocol with $L = 5$ message replicas over the SPMBM trace in the presence of free rider nodes. Note that there are $\binom{N}{K}$ different random challenge realizations, *i.e.*, ways to choose K free rider nodes out of N network nodes. Fig. 3 plots the standard deviation (bold red lines) as well as the min- and max-values (faded grey lines) of the measured average message delay distribution over 100 random challenge realizations. It also depicts the envelope curves, as these have been computed using the heuristic in Section 3.3.

Two important remarks can be made out of Fig. 3. First, for a wide set of challenge parameterizations (here: K value), the distribution of the metric (here: message delay) values over all possible challenge realizations exhibits long tails so that typical sample sizes (*i.e.*, number of random simulation runs) cannot capture its extreme, worst- and best-case, values. In other words, an inefficiently high number of random simulation runs would be required in order to have meaningful chances to sample the extreme metric values. Secondly, the metric envelope curves carry additional information that is not visible through an average-value analysis. In this particular example, we see that there are 45 nodes with a highly central role in data dissemination. The message delay increases superlinearly as these nodes gradually adopt free riding behavior, whereas its increase is much less dramatic as more nodes do so.

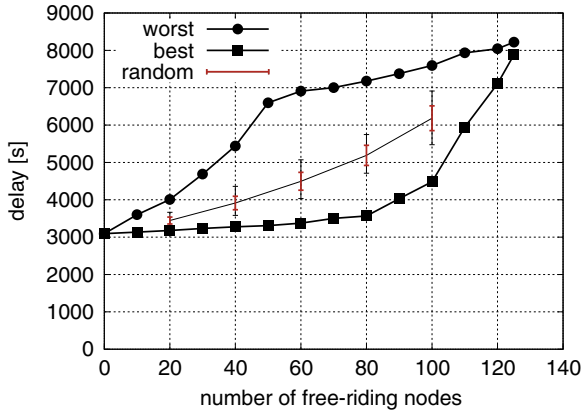


Fig. 3. Metric envelope/vs average values and standard deviation over random challenge realizations: SPMBM trace, message delay under source Spray and Wait ($L = 5$) in the presence of free riders.

4.3. Results

4.3.1. Software/hardware incompatibilities

The experiments in this section are carried out over the two variants of the SPMBM trace, to compare the impact of contact diversity on resilience against software/hardware incompatibilities. For the best-case scenario derivation, we have used the greedy community detection algorithm of Newman in [12][2], which produces the number of communities that maximizes the modularity of the partition. For the dense and sparse SPMBM trace, the algorithm yields 4 communities of {62, 21, 30, 13} nodes per community, resp. 3 communities with {55, 17, 14} nodes per community. For the worst-case scenario, we use an adaptation of the largest-degree-first algorithm for weighted graphs.

Algorithm 1. Min-cost vertex coloring for weighted graph

Input : $G = (V, E_c; w_v)$
Output : *a* K *− vertex − coloring* f *of graph* G

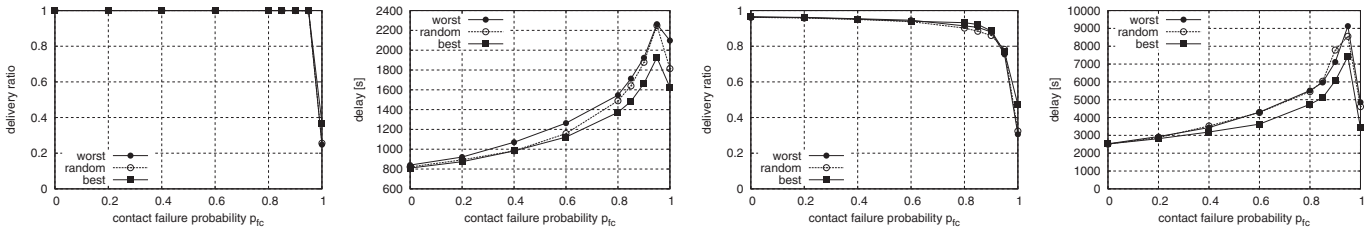
While there are uncolored vertices of G
  choose the vertex v *with the maximum weighted degree*
  assign it the color that minimizes the sum of weights of its edges
  to already colored vertices with the same color, $f(v) = c$
Return vertex coloring f

The minimum weight coloring of the weighted graph yields color classes of {34, 27, 35, 30}, and {33, 29, 24} nodes per community for the dense and sparse SPMBM trace, respectively. The baseline contact failure probability for pairwise encounters among compatible nodes (*i.e.*, nodes grouped in the same community) is set to $p_{fr} = 0.0$; whereas, for encounters between nodes having different software/hardware releases (*i.e.*, belonging to different communities), the baseline contact failure probability is incremented by the variable p_{fc}.

Overall, and compared with the other two challenges (Sections 4.3.2 and 4.3.3) the performance envelopes of the two metrics are tight. Therefore, the uncertainty due to the mobility patterns of the nodes is fairly limited, particularly with respect to the message delivery ratio.

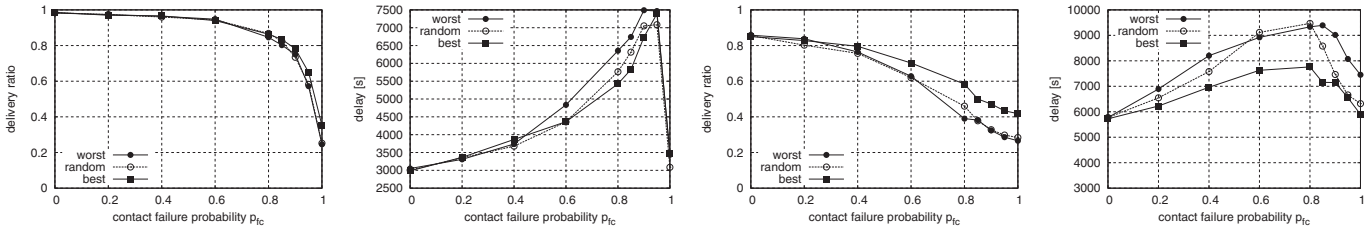
The Epidemic protocol is almost insensitive to contact failures due to incompatibilities (see Fig. 4). Even though 95% of the contacts between incompatibility groups fail, the few contacts that are left is enough to deliver messages between communities. However, the delay is three times higher then with no challenge, if the network is dense, and almost four times as big, if the network is sparse. Messages with high delivery delay under challenged

[2] We have also experimented with the Extremal Optimization algorithm in [27], which is more persistent in searching to maximize the modularity of the structure and is reported to outperform greedy algorithms. The difference was negligible and does not change the remarks made in this subsection.



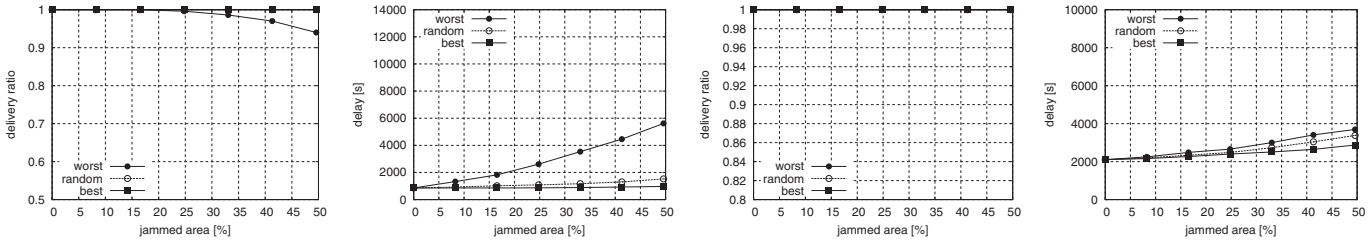
(a) SPMBM trace (b) SPMBM trace (c) Sparse SPMBM trace (d) Sparse SPMBM trace

Fig. 4. Effect of *software/hardware incompatibilities* on delivery ratio and delay: *Epidemic*.



(a) SPMBM trace (b) SPMBM trace (c) Sparse SPMBM trace (d) Sparse SPMBM trace

Fig. 5. Effect of *software/hardware incompatibilities* on delivery ratio and delay: *Spray and Wait*.



(a) SPMBM trace (b) SPMBM trace (c) RWP trace (d) RWP trace

Fig. 6. Impact of *jamming* on message delivery ratio and delay: *Epidemic*.

conditions typically have sender and destination nodes in different incompatibility groups. When 100% of the contacts between incompatibility groups fail, those messages are not delivered anymore and therefore are no longer accounted for in the average delay.

The SSW protocol is affected by the challenge much earlier, already at $p_{fc} = 0.4$ and $p_{fc} = 0.6$, with 40 resp. 60 percent contact loss, for sparse and dense networks, respectively. Compared to the Epidemic protocol, the delay does not have the same relative increase when SSW is used (see Fig. 5(b) and (d)). The main reason is due to that the delay is larger to begin with and that messages that are delayed for a long time do not have the chance to be delivered due the lost contacts.

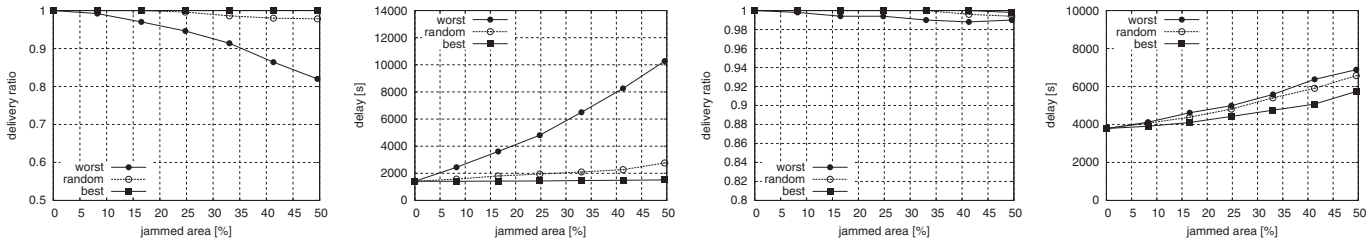
In general, the degradation in performance of the protocols is worse, with respect to delivery ratio, when the contact trace is sparse, as the challenge increase. Under the same conditions, the performance envelope of the delay metric increases in width, particularly when the replication budget of the routing protocol is limited. Then, as the challenge increases, a low frequency of encounters between compatible nodes in the worst case scenario can no longer be compensated by the help of incompatible nodes.

4.3.2. Jamming

The experiments in this section are carried out over the SPMBM and RWP traces. Prophet is run without a hard bound on the number of message replicas and Spray and Wait in its source-spraying mode with replication budgets $L = 8$ (SSW_8) and $L = 16$ (SSW_{16}).

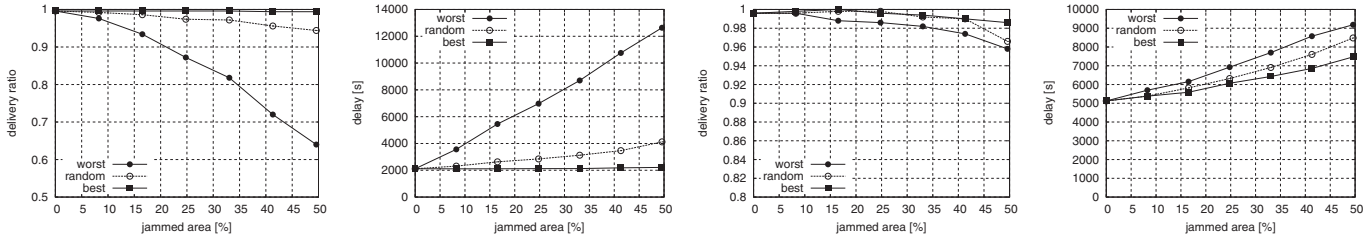
Several aspects of the relative performance of the three protocols are evident already through the outcomes of the random realizations. Hence, as shown in Fig. 6 and intuitively expected, the Epidemic protocol exhibits the highest resilience to the jamming attacks. Over the theoretical RWP traces, in particular, the protocol manages to deliver all messages irrespective of how intelligently the jamming nodes are placed in the physical space. Yet the jammers' placement affects the delay experienced by the delivered messages. Since the RWP mobility model induces a higher concentration of nodes in the middle of the area, as shown in Fig. 2, visibly higher delays are obtained when the jammers are placed in cells around the center of the simulation area rather than at the edge cells (best-case scenario), where fewer encounters occur. The performance envelopes of the Epidemic protocol broaden over the SPMBM traces, the ratio of worst- over best-case delay approaching a factor of six when half the area is jammed.

Prophet, on the other hand, experiences substantial performance degradation in terms of both message delivery ratio and delay. The jammed encounters do not only represent wasted opportunities for message forwarding but also prevent the protocol from correctly updating its state, *i.e.*, the delivery predictability indices. Hence, the protocol cannot deliver all its messages to their destinations (Fig. 7(a) and (c)) and when it does so, the delivery delay is significantly higher than with the Epidemic protocol, as



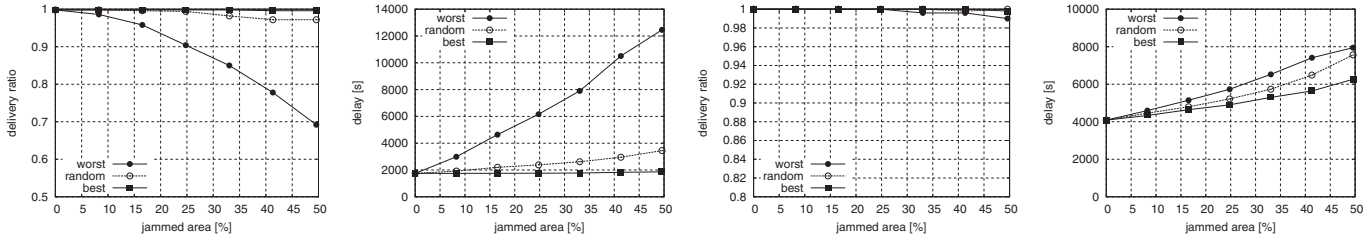
(a) SPMBM trace (b) SPMBM trace (c) RWP trace (d) RWP trace

Fig. 7. Impact of *jamming* on message delivery ratio and delay: *Prophet*.



(a) SPMBM trace (b) SPMBM trace (c) RWP trace (d) RWP trace

Fig. 8. Impact of *jamming* on message delivery ratio and delay: *source Spray and Wait*, $L = 8$.



(a) SPMBM trace (b) SPMBM trace (c) RWP trace (d) RWP trace

Fig. 9. Impact of *jamming* on message delivery ratio and delay: *source Spray and Wait*, $L = 16$.

can be seen in Fig. 7(b) and (d). On the other hand, Prophet achieves similar message delivery rates to SSW_{16} (see Fig. 9) and slightly better ones than SSW_8 (see Fig. 8), whereas it maintains a small advantage with respect to message delivery delays.

The added value of the metric envelopes is more clear for the SPMBM trace, where the stronger structure in the mobility patterns yields considerable variance in the performance under different realizations of the attack. In the message delivery plots of Figs. 6(a) and 7(a), the worst-case curves of the Epidemic and Prophet protocols start deviating already when 5% of the area is jammed, well before the random realization curves start doing so (25% of the area is jammed). In this sense, they give a much clearer signal for the preferability of Epidemic over Prophet at low jammer densities than what can be inferred by only looking at the average values. Likewise, the envelope curves resolve the tie (in terms of average message delivery probabilities) between Prophet and SSW_{16} in favor of the former: with SSW_{16} there is much higher risk to deliver significantly fewer messages than what the average values predict for both protocols.

4.3.3. Free-riders

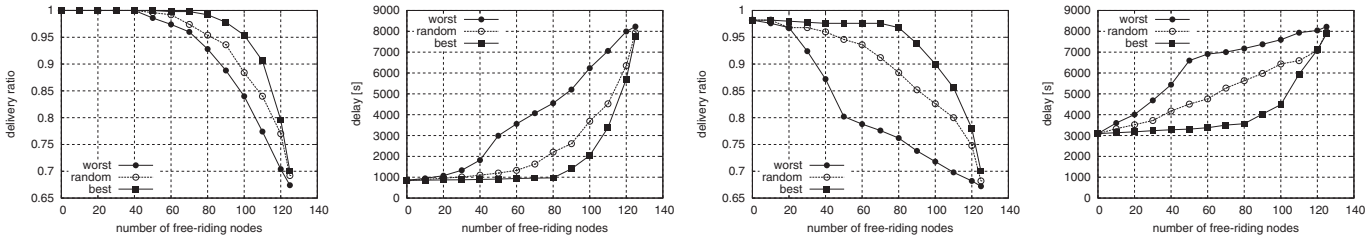
Fig. 10 compares the Epidemic and SSW protocols under the SPMBM traces, when SSW has replication budget $L = 5$ (SSW_5). The performance envelopes of the epidemic protocol are consistently tighter for both performance metrics thanks to its inherent diversity. As the free-rider nodes are allowed to grow to the full network population, the performance of both protocols under all three ways to assign free-rider nodes converges to the performance of Direct Encounter forwarding.[3] However, the performance degradation of the epidemic protocol is more graceful and predictable than that of the SSW_5, whose worst-case performance widens the metric envelopes already in the presence of few free-riders.

On the other hand, the envelopes for both protocols and both metrics degenerate into almost a single curve for the RWP trace. In this case, the envelopes in Fig. 11 "fail" to show any differentiation between worst- and best-case scenarios only because this does not exist. Nodes move homogeneously and the dissemination load of randomly generated messages is spread almost uniformly across all nodes. It makes, thus, little difference which specific nodes misbehave; the challenge impact is purely volume-based.

More interesting is the comparison of the binary versions of Spray and Wait and Prophet protocols under the Haggle traces. Although binary spraying has shown to be optimal under nominal cooperation conditions [21], it is much more vulnerable to free-rider nodes that can waste its replication budget faster than they do for source spraying [28]. The two protocols are compared under identical replication budgets ($L = 4$ for the Cambridge and Infocom05 traces, $L = 8$ for the Infocom06 trace).

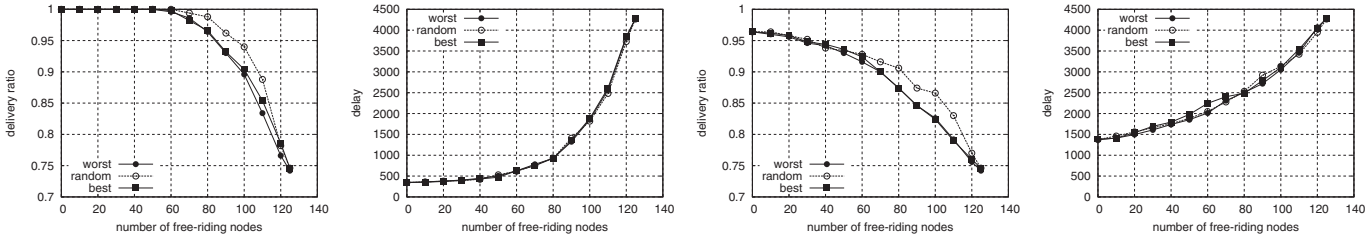
As shown in Figs. 12–14, BSW features consistently smaller delivery ratio envelopes than Prophet. In all traces, Prophet starts

[3] Under Direct Encounter forwarding messages are only transferred directly, over a single hop, from their source to their destination nodes. No message relaying is involved.



(a) Epidemic (b) Epidemic (c) SSW (d) SSW

Fig. 10. Impact of *free-riders* on the message delivery ratio and delay: SPMBM trace, *Epidemic vs. SSW* ($L = 5$).



(a) Epidemic (b) Epidemic (c) SSW (d) SSW

Fig. 11. Impact of *free-riders* on the message delivery ratio and delay: RWP trace, *Epidemic vs. SSW* ($L = 5$).

from higher message delivery probabilities under perfect node cooperation but sees its delivery capacity degrade faster as more nodes practice free-riding. More notable, however, is that the message delivery delay envelopes of Prophet are not aligned with its message delivery ratio envelopes. In fact, there are cases, as shown in Figs. 13(d) and 14(d), where for a broad range of free-rider node values, the correlation between the best- and worst-case values of the two metrics is negative.

This can be traced to the specific way that free-riders affect the operation of the Prophet protocol. Namely, free-rider nodes do not hurt Prophet's ability to update its delivery predictability indices, as software/hardware failure and jamming do; however, they thin out more aggressively the sequence of useful encounters than they do for BSW. With Prophet, the message tends to travel along a gradient of increasing delivery predictability values. Namely, they travel towards nodes with increasingly higher delivery predictability values. Since the worst-case scenario turns these nodes into free-riders, messages essentially have a window of opportunity to get to the destination before they hit one of these free-rider nodes. This is why many more messages get lost in the worst-case scenario, when compared to BSW. This is also reflected in the plots of message delivery delay. As more nodes succumb to free-riding, the delay of the monotonically fewer messages decreases as well. The messages that make it to the destination are those that find a space–time path to it fast enough. In the best-case scenario, that turns the least central nodes to free-riders, the message delays are less dramatically affected with the increase of free-rider nodes so that there is some turning point, where the expected message delay under the worst-case scenario is lower than that for the best-case scenario. The disadvantage of Prophet is that the replication budget is not at risk only during the initial spraying phase (as with BSW) but may further diminish after the spraying phase is over, upon encounters with free-rider nodes that feature higher delivery predictability indices to the message destination nodes.

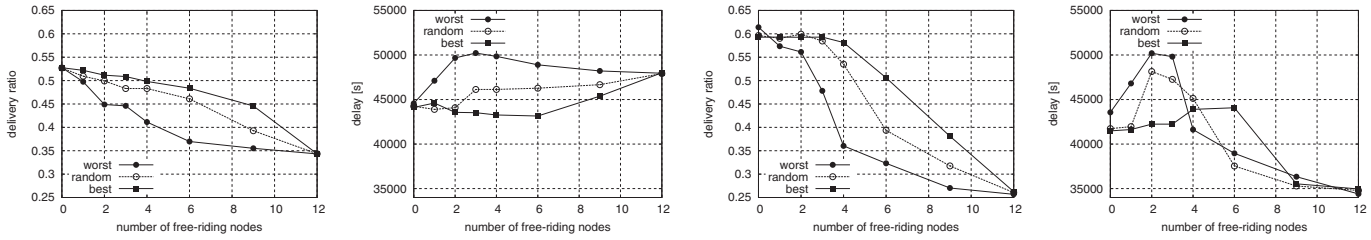
4.3.4. Discussion

The performance envelopes we have derived in Sections 4.3.1–4.3.3 are of different breadth and shape. As a general rule, the envelopes tend to be tighter for traces that have higher degrees of message path diversity due to the way nodes move and encounter with each other. For those traces, the difference between best- and worst-case scenarios are small and the additional information the envelopes provide when compared to average value analysis is minimal. Random challenge realizations suffice to predict the protocol performance degradation.

However, the actual envelope shape varies also with the challenge instance. For example, in the RWP traces with homogeneous node mobility, nodes tend to have similar centrality values since all of them tend to meet with each other. As a result, the best- and worst-case scenarios for the free-riding challenge almost coincide (Fig. 11 in Section 4.3.3). On the other hand, in the same RWP traces, nodes tend to be more densely distributed and encounter more frequently in the center of the physical area. As a result, placing the jammers there (worst-case scenario) makes a visible different than placing them at the edges of the area, at least for forwarding protocols with restricted replication budget, as shown in Fig. 9(c) and (d).

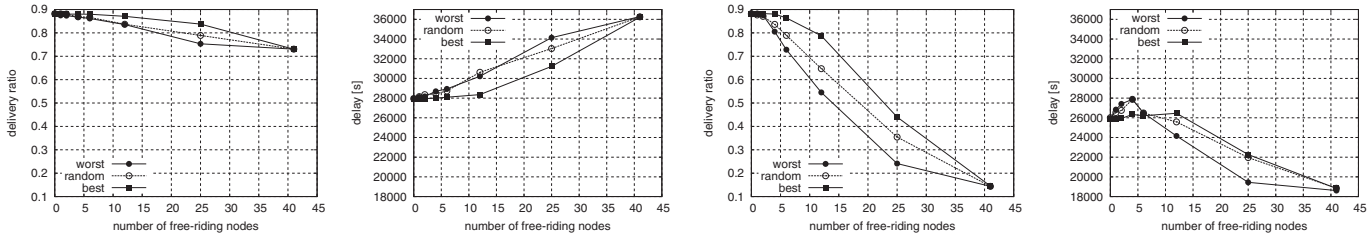
Worst-case scenarios can be relevant to various network planning and engineering tasks. Consider a wireless network operator that would like to complement their infrastructure with opportunistic communication. The operator will probably assume that some percentage of nodes will act as free-riders, but they cannot know who. Trying to compute how many nodes are required in an average scenario would significantly underestimate what is actually needed in a worst-case scenario, *i.e.*, free-riders coinciding with the nodes that are highly mobile. Note that the concept of the envelope could draw on any percentile of the performance metrics; however, it would probably be even more difficult to make guesses about what would be a scenario yielding, say, the 95% percentile of some metric(s) instead of the worst/best-case scenario for it.

Less apparent may seem the relevance of best-case scenarios. However, consider the challenge of software/hardware incompatibilities. One would expect that nodes that tend to meet often, at least when these encounters are driven by kinship or common interests rather than simply physical proximity (*e.g.*, during daily commuting to workplace), are software/hardware compatible to be able to communicate without disruptions. In this case, the best-case realization would be a more realistic predictor of the expected protocol performance rather than a random or the worse-case one.



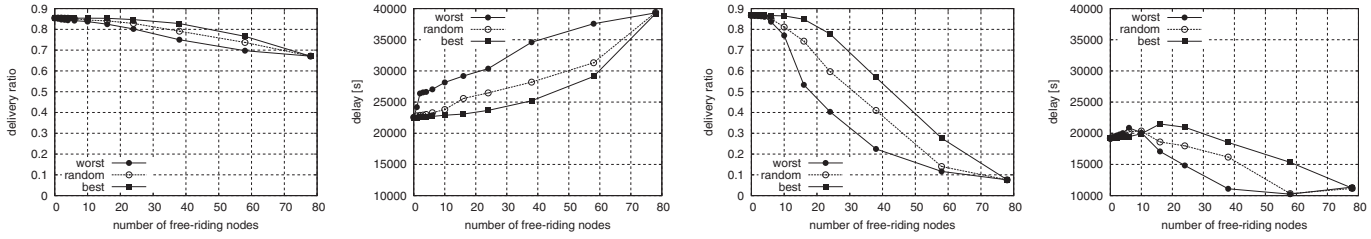
(a) BSW (b) BSW (c) Prophet (d) Prophet

Fig. 12. Impact of *free-riders* on the message delivery ratio and delay: Cambridge trace, *BSW* ($L = 4$) *vs. binary-spraying Prophet* ($L = 4$).



(a) BSW (b) BSW (c) Prophet (d) Prophet

Fig. 13. Impact of *free-riders* on the message delivery ratio and delay: Infocom05 trace, *BSW* ($L = 4$) *vs. binary-spraying Prophet* ($L = 4$).



(a) BSW (b) BSW (c) Prophet (d) Prophet

Fig. 14. Impact of *free-riders* on the message delivery ratio and delay: Infocom06 trace, *BSW* ($L = 8$) *vs. binary-spraying Prophet* ($L = 8$).

5. Related work

In general, the performance degradation of opportunistic forwarding schemes in the presence of challenges to their operation has been explored both analytically and with simulations. Most studies have looked into the impact of the free-riding phenomenon, whereby nodes do not (occasionally) contribute to the relaying of other nodes' messages. In [5] Panagakis et al. perform simulations with synthetic random mobility models and let nodes probabilistically abstain from copying and forwarding messages. They compare the impact of partial cooperation on the Epidemic, Two-Hop and BSW protocols and find that Epidemic (BSW) is the most sensitive with respect to the message overhead (resp. expected message delivery delay). Keranen et al. in [4] consider similar misbehavior expressions. They perform trace-based simulations that include Prophet and report that the protocols are overall robust to misbehavior phenomena. Karaliopoulos, on the other hand, has formulated analytical Markovian models for the performance of epidemic and two-hop protocols under imperfect cooperation in [1], assuming homogeneous node mobility and exponentially distributed ICTs; whereas Li et al. [2] draw on the same assumptions to study the vulnerability of epidemic protocol to "social" selfishness, whereby nodes organize themselves in groups/communities and exhibit different levels of cooperation depending on group memberships. The resilience of forwarding protocols of the controlled-flooding type is also the subject of a hybrid analytical-simulation study by Resta and Santi in [3]. Analysis is used to derive bounds for the performance of the epidemic protocol in the presence of free riders, whereas simulations are carried out to demonstrate the good robustness properties of BSW under similar conditions.

A richer set of attacks is studied through simulations in [6] for different variants of the MaxProp protocol. The authors also point to the difference between random and worst-case scenarios but overall find their protocol adequately robust to attacks. This conclusion has been later disputed by the simulation study in [29], which reports more severe performance degradation of the protocol under a new set of devised attacks and argues that some form of authentication is necessary to cope with attacks in general. A slightly different research thread relates to the vulnerability of particular protocol primitives met in a range of opportunistic protocols. Hence, in [28] the focus is on the management of the message replication budget in multi-copy protocols. The work compares the source and binary message spraying techniques as implemented in the Spray and Wait and Prophet protocols. Through analysis and simulations, it shows that the original advantage of binary over source spraying under perfect node cooperation turns to a disadvantage as more free rider nodes are present in the network. It also proposes an alternative spraying function that achieves a better tradeoff between the performance under fully cooperative nodes and the robustness to free riders.

In this paper, we consider a broader and more representative set of challenges that involve node behaviors, malicious attacks, but also software/hardware inconsistency issues. More importantly,

we differentiate between challenge instances, parameterizations and realizations, and propose systematic ways to select the latter in order to approximate the performance envelope of the opportunistic protocols. This way, we generate richer representations of the vulnerability of opportunistic forwarding protocols to challenges, which bear additional information to anyone interested in predicting the protocols' performance degradation in their presence. Methodologically, our paper bears strong similarities to the work by Doerr and Hernandez in [30]. They study the robustness of network topologies (graphs) by deriving envelopes for the values different graph attributes take when they are subject to k node/edge failures. Contrary to our approach, they get the envelope bounds by enumerating *all* possible realizations of a failure scenario. This way their method is exact but gets computationally intractable already for small values of k.

6. Conclusions

We assess the resilience of opportunistic forwarding schemes through the metric envelope concept, which explicitly accounts for the differentiated challenge impact due to node heterogeneity in terms of device capabilities, mobility and, when relevant, attacker's intelligence. The envelope approach gives a more complete assessment of a protocol's resilience than average value analysis. As a general rule, the envelope tends to be narrower for traces that have inherently high degree of diversity due to the way nodes move and encounter each other. The actual envelope shape has also to do with the challenge itself.

Since the enumeration of all possible challenge realizations to get relevant statistics is a computationally expensive task, we have proposed heuristic ways to infer their best- and worst-case scenarios that could enclose the full variation of the forwarding protocol performance. Hence, the derivation of these scenarios has been formulated as an instance of the community detection (resp. weighted coloring) problem in the case of software/hardware incompatibilities; accounted for the spatial density of encounters in the case of jamming; and relied on the node centrality indices in the case of free-riders.

Finally, we have carried out a systematic experimental evaluation of three popular representative forwarding protocols using a variety of contact traces. We have argued that the metric envelope approach provides better insights into the level of protection path diversity and message replication provide against different challenges, and enables more informed conclusions about the resilience of opportunistic forwarding protocols.

A fundamental property of the proposed heuristics is that they are protocol-agnostic, *i.e.*, the worst- and best-case scenario derivation do not explicitly account for the particular protocol forwarding rules. Doing so, they set a common reference for a fair comparison of the resilience properties of different forwarding protocols. Further performance variation may be achieved by attacks carefully instrumented for a specific protocol. The design of intelligent attacks for specific opportunistic networking protocols or applications constitutes a distinct research thread implying worst-case scenarios that differ from protocol to protocol. However, the envelope schema could still be relevant for assessing the predictability of the attack's impact on the protocol performance.

Acknowledgements

This research was funded by the ResumeNet project under the EU grant FP7-224619. The work of the second author has been supported by the European Commission under the Network of Excellence in Internet Science project EINS (FP7-ICT-288021). Our paper benefited from the community detection functionality of the radatools v3.2 software, which is provided freely to the research community by Prof. Sergio Gomez and his colleagues.

References

[1] M. Karaliopoulos, Assessing the vulnerability of DTN data relaying schemes to node selfishness, Commun. Lett. IEEE 13 (2009).
[2] Q. Li, S. Zhu, G. Cao, Routing in socially selfish delay tolerant networks, Proc. INFOCOM 2010 (2010) 1–9.
[3] G. Resta, P. Santi, A framework for routing performance analysis in delay tolerant networks with application to noncooperative networks, IEEE Trans. Parallel Distrib. Syst. 23 (2012) 2–10.
[4] A. Keranen, M. Pitkanen, M. Vuori, J. Ott, Effect of non-cooperative nodes in mobile dtns, WoWMoM, 2011.
[5] A. Panagakis, A. Vaios, I. Stavrakakis, On the effects of cooperation in DTNs, in: Proc. COMSWARE, pp. 1–6, 2007.
[6] J. Burgess, G.D. Bissias, M.D. Corner, B.N. Levine, Surviving attacks on disruption-tolerant networks without authentication, in: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '07, 2007, pp. 61–70.
[7] F. Bjurefors, M. Karaliopoulos, C. Rohner, P. Smith, G. Theodoropoulos, P. Gunningberg, Resilience and opportunistic forwarding: beyond average value analysis, in: Proceedings of the 8th ACM MobiCom workshop on Challenged networks, CHANTS '13, 2013, pp. 19–24.
[8] W. Lee, Mobile Cellular Telecommunications Systems, McGraw-Hill, New York, 1989.
[9] J. Meyer, On evaluating the performability of degradable computing systems, IEEE Trans. Comput. C-29 (1980) 720–731.
[10] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113+.
[11] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. (USA) 104 (2007) 36–41.
[12] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (2004) 066133.
[13] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (2008) P10008.
[14] M.E.J. Newman, Analysis of weighted networks, Phys. Rev. E 70 (2004) 056131.
[15] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, D. Wagner, On modularity clustering, IEEE Trans. Knowl. Data Eng. 20 (2008) 172–188.
[16] L. Kiaer, J. Yellen, Weighted graphs and university course timetabling, Comput. Oper. Res. 19 (1992) 59–67.
[17] M. Feldman, C. Papadimitriou, J. Chuang, I. Stoica, Free-riding and whitewashing in peer-to-peer systems, IEEE J. Sel. Areas Commun. 24 (2006) 1010–1019.
[18] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: social-based forwarding in delay-tolerant networks, IEEE Trans. Mob. Comput. 10 (2011) 1576–1589.
[19] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: Proc. Simutools '09, 2009, ICST, 2009.
[20] M. Karaliopoulos, C. Rohner, Trace-based performance analysis of opportunistic forwarding under imperfect cooperation conditions, in: Proceedings of the INFOCOM 2012 mini-conference, 2012.
[21] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Efficient routing in intermittently connected mobile networks: the multiple-copy case, IEEE/ACM Trans. Netw. 16 (2008) 77–90.
[22] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, SIGMOBILE MCCR (2003).
[23] S. Grasic, E. Davies, L.A., D.A., The evolution of a dtn routing protocol - prophetv2, in: Proc. ACM MOBICOM CHANTS, San Francisco, CA, USA, 2011.
[24] P. Nain, D. Towsley, B. Liu, Z. Liu, Properties of random direction models, in: Proceedings IEEE of 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005, vol. 3, IEEE, 2005, pp. 1897–1907.
[25] C. Bettstetter, G. Resta, P. Santi, The node distribution of the random waypoint mobility model for wireless ad hoc networks, IEEE Trans. Mob. Comput. 2 (2003) 257–269.
[26] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Impact of human mobility on the design of opportunistic forwarding algorithms, in: Proc. IEEE INFOCOM '06, 2006, pp. 1–13.
[27] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2005).
[28] M. Karaliopoulos, I. Stavrakakis, Involve relays or leave it to the source? On message replication in DTNs under imperfect cooperation, Technical Report, online: <http://cgi.di.uoa.gr/mkaralio/wp-content/uploads/2012/07/MsgReplicationOppnets.pdf>, 2012
[29] F.C. Choo, M.C. Chan, E.-C. Chang, Robustness of dtn against routing attacks, in: 2010 Second International Conference on Communication Systems and Networks (COMSNETS), 2010, pp. 1–10.
[30] C. Doerr, J.M. Hernandez, A computational approach to multi-level analysis of network resilience, in: Third International Conference on Dependability (DEPEND), 2010.