Resilience and Opportunistic Forwarding: Beyond Average Value Analysis

Fredrik Bjurefors[†], Merkourios Karaliopoulos[‡], Christian Rohner[†],
Paul Smith[§], George Theodoropoulos[‡], Per Gunningberg[†],

[†]Department of Information Technology
Uppsala University, Sweden
name.surname@it.uu.se

[‡]Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Greece
mkaralio@di.uoa.gr

[§]AIT Austrian Institute of Technology
Seibersdorf, Austria
paul.smith@ait.ac.at

ABSTRACT

Opportunistic networks are systems with highly distributed operation, relying on the altruistic cooperation of heterogeneous, and not always software- and hardware-compatible user nodes. Moreover, the absence of central control makes them vulnerable to malicious attacks. In this paper, we take a fresh look at the resilience of opportunistic forwarding to these challenges. In particular, we introduce and promote the use of *metric envelopes* as a resilience assessment tool. Metric envelopes depart from the standard practice of average value analysis and explicitly account for the differentiated impact that a challenge may have on the forwarding performance due to node heterogeneity (device capabilities, mobility) and attackers' intelligence. The use of metric envelopes is demonstrated in the case of three challenges: *jamming*, *hardware/software failures* and incompatibilities, and *free-riding* phenomena. For each challenge, we first devise heuristics to generate worst- and best-case realization scenarios that can approximate the metric envelopes. Then we derive the envelopes of common performance metrics for three popular forwarding protocols under a comprehensive range of mobility patterns. The metric envelope approach enables more informed choices in opportunistic forwarding whenever network resilience considerations become important.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

Keywords

opportunistic networking, forwarding, resilience

CHANTS'13, September 30, 2013, Miami, Florida, USA
Copyright 2013 ACM 978-1-4503-2363-5/13/09 ...$15.00.
http://dx.doi.org/10.1145/2505494.2505500.

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, traversing disjoint physical spaces and involving 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 to their operation has been explored both analytically and with simulations. Most studies have looked into the impact of nodes that defer from message copying/forwarding. In [11] Panagakis *et al.* perform simulations with synthetic random mobility models and let nodes probabilistically defer from copying and forwarding messages. They consider the impact of partial cooperation on the Epidemic, Two-Hop and Binary Spray-and-Wait protocols and find that Epidemic (BSW) is the most sensitive scheme to the message overhead (resp. expected message delivery delay). Keranen *et al.* in [6] perform trace-based simulations for similar misbehavior expressions, including the Prophet protocol, and report that the protocols are overall robust to misbehavior phenomena. Karaliopoulos, on the other hand, formulates analytical Markovian models for the performance of epidemic and two-hop protocols under imperfect cooperation in [3], assuming homogeneous node mobility and exponentially distributed inter-contact times; whereas Li *et al.* [7] 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. A hybrid analytical-simulation study of randomized forwarding protocols is also reported by Resta and Santi in [12]. They derive analytical bounds for the performance of epidemic protocol under probabilistic deferral from forwarding, whereas they use simulations to demonstrate the good robustness properties of BSW to forwarding misbehaviors.

Common to all these works is that they assess the opportunistic forwarding performance in the presence of a challenge through averages values of the performance metrics. 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 misbehaving nodes" can have a widely different impact on the performance of the opportunistic forwarding, depending on *which* K nodes do misbehave. Therefore, the metric envelopes introduced in Section 2 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, metric envelopes 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.

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. The exact computation of the metric envelope values for these challenges would require enumerating *all* possible challenge realizations (*e.g.*, possible ways to select the misbehaving nodes or placements of the jammer nodes in the physical space) and imply a huge computational burden. Therefore, in Section 3 we propose heuristics (cues) for inferring "best"- and "worst"-case scenarios for each challenge and approximating the respective metric envelopes.

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).

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 Figure 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 [14] 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.

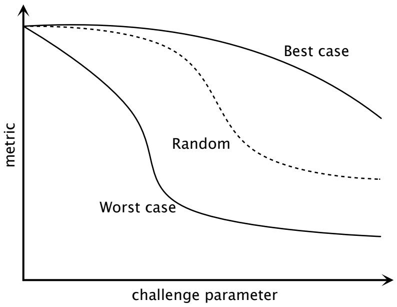


Figure 1: Metric envelope

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.

3. EXPERIMENTATION METHODOLOGY

To demonstrate the use of the metric envelope approach, we carry out experiments with two performance analysis tools: the ONE simulator [5] and the trace-driven Space-Time-Graph methodology in [4]. ONE provides implementations for various mobility models and routing protocols. The Space-Time-Graph scripts can be used to evaluate the performance of opportunistic networking protocols based on contact-traces.

3.1 Challenge realization

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 the *actual* extreme realizations, while 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.

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 the wireless interface.

Implementation-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. The second probability, p_{fc}, depends on the compatibility of the encountered nodes in terms of software/hardware. The assumption is that such incompatibilities make occasional contact failures more frequent.

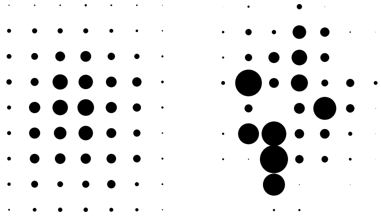


Figure 2: Contact map for the RWP and SPMBM scenario. The size of the circles indicates the number of contacts within the area.

To derive then the best-case scenario for this challenge, we properly partition the node set into software/hardware-based *compatibility groups.* We do this by running a modularity-maximizing community detection algorithm over the weighted contact graph of the opportunistic network. The vertices of the contact graph correspond to the network nodes and the weights of the edges between two vertices to the count of their pairwise encounters. Output of the community detection algorithm is a (predefined) number of node clusters (communities), wherein nodes tend to encounter more frequently with each other than when compared to nodes of other communities. By mapping these communities to compatibility groups, we minimize the encounters that are subject to the higher failure probability p_{fc}.

We follow the same process to derive the worst-case scenario, only now the weights of the contact graph edges are inverted[1]. Therefore, nodes that tend to meet frequently with each other are assigned to different compatibility groups and "high-risk" pairwise node encounters increase.

Jamming: Jamming is an example of an 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. These places could include train and subway stations. The jammer is assumed to be blocking any communication between node pairs if one of the nodes is located within a certain radius around it. Compared to software/hardware failures, jamming introduces strong *spatial correlation* in the pattern of wasted contact opportunities.

The jamming challenge is implemented by extending the ONE simulator's logging functionality and adding the physical coordinates of the encountered nodes to each encounter record. The simulated area is organized into a grid of 100 × $100m^2$. Multiple jammers with jamming radius $r_{jam} = 50$, in line with current technology[2], are placed in the middle of the squares to jam an area up to 50% of the simulation area. To create the worst (and best)-case scenarios, we selectively place the jammers at the middle of the squares with the most (and fewest) contacts over the trace duration. To this end, we generate a contact map, as shown in Figure 2. Node contacts involving even a single node within distance r_{jam} of a jammer are then removed from the contact trace and the residual modified trace is used for performance analysis.

[1] When an edge weight is zero, we substitute it with a small enough value so that its inverse yields a finite value.

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

Table 1: Characteristics of experimentation datasets

Configuration	Cambridge	Infocom06	SPMBM	RWP
Collection	iMote	iMote	ONE	ONE
Duration(days)	6	4	0.5	0.28
Scan time(sec)	5-10	5-10		
Granularity(sec)	120	120	1	1
Mobile Devices	12	78	126	126
Stationary Dev.	0	20		
# of Contacts	6732	227657	30959	30000

Free-riding: Free-riders are a common phenomenon in all networks necessitating some form of collaboration and mutual contributions (*e.g.*, [2]). 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 social nodes, *i.e.*, nodes that are highly mobile and frequently encounter other nodes.

We extend the routing protocols in the ONE simulator to allow for free-rider node behavior. Free-riders are selected based on their centrality values. We calculate the centrality index of each node in a given trace by counting how many times it participates in the shortest space-time paths under the Epidemic protocol. The number of randomly generated message-samples is chosen such that the ranking of nodes remains stable and for most traces lies close to 5000 messages.

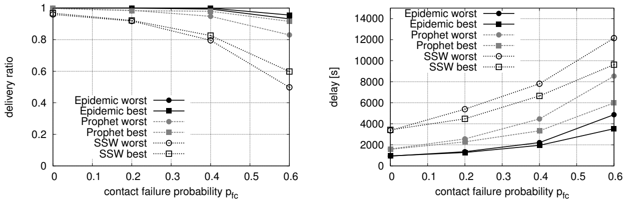
Then, as a best-case realization, we choose as free-riders those nodes with the lowest centrality values, *i.e.*, nodes that would be expected to contribute little in the forwarding process under normal operation. In the worst-case scenario, the free-riders are the nodes with the top centrality values.

In all three cases, an encounter that could otherwise result in data exchange, does not actually do so. The question we seek to answer is how resilient the different forwarding protocols are to these challenges, which essentially thin out the original density of contact opportunities, as this emerges from the pure mobility of the network nodes. Intuitively, the protocol resilience is closely related to the way they manage the message replication budget and their capacity to find uncorrelated paths, both with respect to the physical space they traverse and the user nodes that realize them.

3.2 Forwarding protocols and mobility traces

We analyze the performance of three of the most popular forwarding schemes, the Epidemic, Spray and Wait and Prophet protocols. *Spray and Wait* [13] is a representative instance of randomized forwarding, whereas Prophet [8] does utility-based forwarding. Our evaluation uses several traces of pairwise node encounters with different characteristics. Two of these traces have been synthesized from mobility models provided by the ONE simulator. These traces enable logging of the physical coordinates of the nodes' movement and are more appropriate for the experimentation with the jamming challenge (ref. Section 3.1). The remaining traces directly report Bluetooth sightings by groups of users carrying iMotes. Table 1 shows the characteristics of the traces used in this work.

Shortest Path Map Based Movement (SPMBM) traces: Nodes move between two randomly chosen locations by following the shortest path along connected segments representing roads on a map [5]. In our case, a street map of Helsinki is used. The simulation area is 4500×3400 meters and includes 126 nodes. The message source and destination nodes are chosen at random.



(a) SPMBM trace (b) SPMBM trace

Figure 3: Effect of *software/hardware failure* on delivery ratio and delay.

Random Waypoint (RWP) traces: Nodes move with random speed and random direction within a specified area, generating a trace with exponentially distributed contact times. We use the same simulation area size, number of nodes, and message load as with the SPMBM.

Haggle traces: These are five well-known experimental traces, gathered in the context of the Haggle Project [1]. They include Bluetooth sightings by users carrying iMotes during the experiments. Each Bluetooth sighting is assumed to be a contact whereby nodes can exchange information. The experimental settings are detailed in [1].

4. EXPERIMENTATION RESULTS

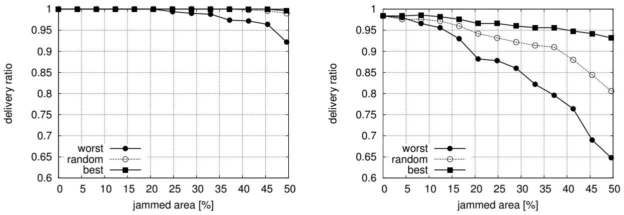
In this section, we derive and plot example metric envelopes corresponding to the challenges in Section 3.1, for the forwarding protocols and contact traces of section 3.2. The aim is to demonstrate the additional hints they provide when assessing the resilience of opportunistic forwarding.

4.1 Software/hardware failures

The experiments in this section are carried out over the SPMBM trace. We have applied the greedy community detection algorithm of Newman in [10], which produces the number of communities that maximizes the modularity of the partition. When the algorithm is applied to the original weighted contact graph of the network (best-case scenario), we derive three "communities" of 51, 24, and 51 nodes, respectively. When it is applied to the contact graph with the edge weights inverted, we obtain six "communities" of sizes ranging in 10-30 nodes. The baseline contact failure probability for nodes within the same community is set to $p_{fr} = 0.1$; whereas, for encounters between nodes lying in different communities, the baseline contact failure probability is incremented by the variable p_{fc}.

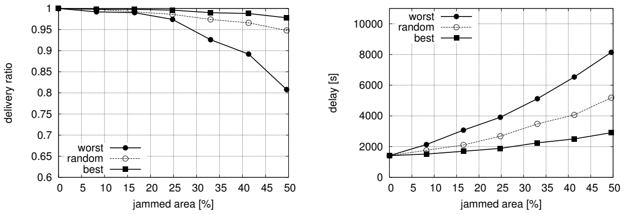
Overall, and compared with the other two challenges (Sections 4.2 and 4.3) the performance envelopes of the two metrics are tight. Source Spray and Wait (SSW) ($L = 5$), in particular, presents the largest sensitivity to the way the network is partitioned to compatible and incompatible nodes, as can be seen in Figure 3.

For both metrics, there is almost no overlapping between the envelopes of the three protocols (at least for p_{fc}<0.6) so that they can be ordered deterministically in terms of performance: Epidemic outperforms Prophet, which in turn scores better than SSW. Interestingly, the Epidemic and Prophet



(a) Epidemic (b) SSW $L = 5$

Figure 4: Impact of *jamming* on message delivery ratio in the SPMBM trace.



(a) SPMBM trace (b) SPMBM trace

Figure 5: Impact of *jamming* on message delivery ratio and delay: *Prophet.*

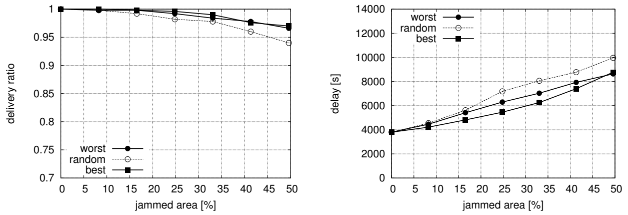
protocols are almost insensitive to contact failure phenomena, even when as many as 40% of the contacts between incompatible nodes actually fail; whereas the SSW replication budget and two-hop limitation end up being particularly restrictive when run over a trace like SPMBM, with strong structure in the nodes' mobility patterns and relatively long shortest space-time paths.

4.2 Jamming

The experiments in this section are carried out over the SPMBM and RWP traces. Spray and Wait is evaluated in its source-spraying mode with a replication budget $L = 5$.

As shown in Figure 4(a) and intuitively expected, the Epidemic protocol exhibits by far the highest resilience to the jamming challenge. Even when jammers are placed in the areas hosting the most encounters, one would need to jam more than 35% of the area before the protocol experiences some notable degradation in the order of 2%. 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 (Figures 5(a) and 6(a)) and when it does so, the delivery delay is significantly higher than with the Epidemic protocol, as can be seen in Figures 5(b) and 6(b). The SSW protocol is the most vulnerable to the jamming challenge, especially when jammers are placed in central locations the degradation in terms of message delivery ratio and delay is more severe (Figure 4(b)).

Over the serving-as-reference RWP trace, our heuristic fails to yield consistent envelopes for delivery ratio; the random placement of jammers yields worse message delivery probabilities and delays than their "worst-case" placement



(a) RWP trace (b) RWP trace

Figure 6: Impact of *jamming* on message delivery ratio and delay: *Prophet.*

in the hotspot areas, even though it is very similar to the SPMBM trace. An average value analysis would thus not show this different behavior. In RWP, everyone meets everyone else many times so even if contacts are jammed at one occasion messages can be delivered at a later point in time, see effect on delay in Figure 6(b). That makes it difficult to heuristically choose a best- and a worst-case.

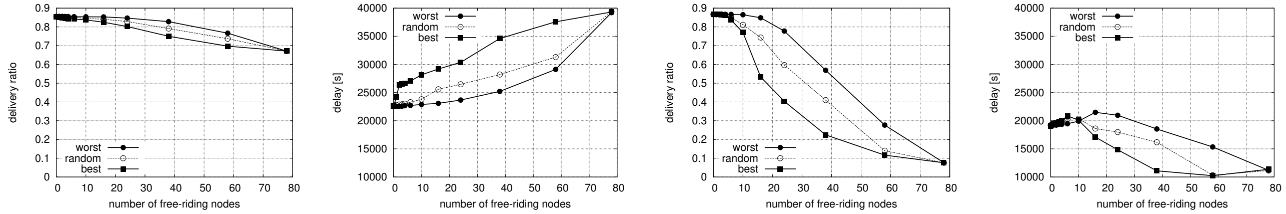
4.3 Free-riders

Figure 8 compares the Epidemic and SSW protocols under the SPMBM traces, when SSW has a replication budget of $L = 5$. The performance envelopes of the Epidemic protocol are consistently tighter for both performance metrics, in line with its advantage in terms of replication budget. Note that 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. The SSW protocol does this more smoothly under the best- and random-case while the Epidemic protocol eventually degrades faster when a strong majority of the nodes adopt such behavior.

More interesting is the comparison of Spray and Wait and Prophet under the Haggle set of traces. In these comparisons, we have experimented with the protocol versions that implement binary message spraying. The two protocols are compared under similar replication budgets.

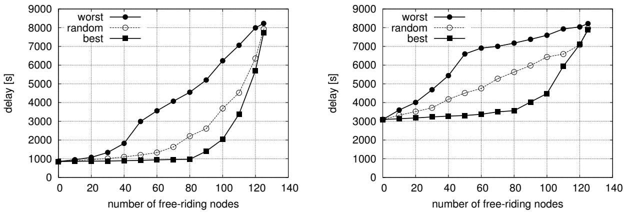
As shown in Figure 7, Binary Spray and Wait (BSW) has a consistently tighter envelope than Prophet, when it comes to delivery ratio. In all traces, Prophet starts from higher message delivery probabilities under perfect node cooperation but sees its delivery capacity degrade faster as more nodes practice free-riding. The free-riders do not hurt Prophet's ability to update its delivery predictability indices, as software/hardware failure and jamming do; however, free-riders appear to thin out more aggressively the sequence of useful encounters than they do for BSW. This is also reflected in the plots of message delivery delay. As more nodes are recruited as free-riders, the delay of the monotonically fewer messages decreases as well. Hence, there seems to exist a window of opportunity for messages to get to the destination before the protocol replication budget is exhausted. The disadvantage of Prophet is that this 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.

The performance envelopes we have derived in Sections 4.1 to 4.3 are of different breadth and shape. As a general



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

Figure 7: Impact of *free-riders* on the message delivery ratio and delay: Infocom06 trace, *BSW*$(L = 8)$ *vs. binary-spraying Prophet* $(L = 8)$.



(a) Epidemic (b) SSW $L = 5$

Figure 8: Impact of *free-riders* on the message delay in the SPMBM trace.

rule, the envelopes tends to be narrower for traces that have an inherently high degree of diversity due to the way nodes move and encounter each other. In those cases, the differences between best- and worst-case scenarios are smaller.

However, the actual envelope shape has also to do with the challenge itself. For example, under RWP traces, 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 do not yield a big difference (not shown in Section 4.3). On the other hand, under 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) is different than placing them at the edges of the area, at least for forwarding protocols with restricted replication budget, shown in Figures 6(a) and 6(b).

5. CONCLUSIONS

In this paper, we have proposed the use of metric envelopes as a tool for assessing the resilience of opportunistic forwarding. The use of envelopes explicitly accounts for the differentiated impact of a given challenge due to node heterogeneity and, when relevant, attacker's intelligence. Therefore, it enables more informed conclusions about the resilience of opportunistic forwarding protocols when compared to the typically used average value analysis.

Since the enumeration of all possible challenge realizations is a computationally expensive task, we have postulated heuristic ways to determine worst- and best-case scenarios realizing the envelope boundaries for representative challenges. Finally, we have demonstrated the use of the envelopes in the case of three popular forwarding protocols using a variety of contact traces.

6. REFERENCES

[1] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *Proc. IEEE INFOCOM '06*, pages 1–13, April 2006.

[2] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *Selected Areas in Communications, IEEE Journal on*, 24(5):1010 – 1019, May 2006.

[3] M. Karaliopoulos. Assessing the vulnerability of DTN data relaying schemes to node selfishness. *Communications Letters, IEEE*, 13(12), Dec 2009.

[4] M. Karaliopoulos and C. Rohner. Trace-based performance analysis of opportunistic forwarding under imperfect cooperation conditions. In *Proceedings of the INFOCOM 2012 mini-conference*, 2012.

[5] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proc. Simutools '09*. ICST, 2009.

[6] A. Keranen, M. Pitkanen, M. Vuori, and J. Ott. Effect of non-cooperative nodes in mobile dtns. *WoWMoM*, 2011.

[7] Q. Li, S. Zhu, and G. Cao. Routing in socially selfish delay tolerant networks. In *Proc. INFOCOM 2010*, pages 1 –9, march 2010.

[8] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE MCCR*, 2003.

[9] J. Meyer. On evaluating the performability of degradable computing systems. *Computers,IEEE Transactions on*, C-29(8):720 –731, Aug. 1980.

[10] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, Jun 2004.

[11] A. Panagakis, A. Vaios, and I. Stavrakakis. On the effects of cooperation in DTNs. In *Proc. COMSWARE*, pages 1–6, Jan. 2007.

[12] G. Resta and P. Santi. A framework for routing performance analysis in delay tolerant networks with application to noncooperative networks. *IEEE Trans. Parallel Distrib. Syst.*, 23(1):2–10, Jan. 2012.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Trans. Netw.*, 16(1):77–90, Feb. 2008.

[14] L. W. C. Y. *Mobile Cellular Telecommunications Systems.* New York: McGraw-Hill, 1989.