Dynamic Selection of Persistence and Transport Layer Protocols in Challenged Networks

Aaron M. Rosenfeld, Robert N. Lass, William C. Regli
Department of Computer Science
Drexel University, Philadelphia, PA
Email: ar374, urlass, regli@cs.drexel.edu

Joseph P. Macker
Naval Research Laboratory
Washington, D.C.
Email: joseph.macker@nrl.navy.mil

***Abstract*—This work applies a distributed algorithm utilizing Markov Random Fields (MRFs) to the problem of dynamically selecting Session and Transport layer protocols in challenged networks such as mobile ad-hoc networks. It motivates the problem by identifying the primary network properties which affect Message Delivery Ratio (MDR) in networks with varying degrees of connectivity and traffic load. Using this information, local and remote observations are used to select a set of protocols which should perform the best.**

Analysis shows that dynamically selecting a set of protocols can deliver up to 50% more messages in challenged environments, and never under-performs statically choosing protocols.

I. Introduction

Maintaining a consistent set of state information across group-oriented applications in tactical edge, mobile ad-hoc networks (MANETs) is a challenging, yet often mission-critical task. In enterprise systems, nodes may exchange information over a network using a protocol such as TCP. On typical enterprise networks, TCP provides reliable ordered data that can be assumed to be free of errors. On MANETs, in contrast, applications requiring long-term information consistency among a group of hosts cannot always use classical reliable transport or rely heavily on a centralized server due to network partitioning, high rates of packet loss, and inconsistent network views.

Some systems are deployed in environments where the network cannot support a protocol such as TCP due to frequent link changes, packet loss, extended disconnection, and high latency [1]. In these situations, protocols which can function in these environments must be employed. However, these protocols have varying levels of fidelity, assorted delivery models, and perform inconsistently in different situations making *a priori* selection of the "optimal" protocol difficult, and scenario-specific.

Further complicating protocol selection is the fact that multiple layers of the network stack can be composed to improve performance. For example, using a caching strategy on top of multicast UDP may act similarly to a reliable multicast protocol in some situations. This demonstrates the need for an intelligent method of choosing protocol stacks based on various network and data parameters.

II. Problem Statement

The problem addressed in this paper is two-fold: the first is determining the state of the network (or any other measurable environmental factor) distributedly, in some neighborhood of each node. The second is dynamically selecting the best algorithm to operate in that environment.

The implicit assumption in the above problem is that there exist environmental factors that have an influence on the performance of the algorithm being selected. In previous work [2], Rosenfeld, et al. analyzed the performance of various persistence protocols in different networking environments, showing that this is the case at least for persistence algorithms.

In general, the problem can be formalized as a set of n environmental factors, each of which can be described by a label:

$$\mathcal{L}_i = \{l_1, l_2, \ldots l_{|\mathcal{L}_i|}\}$$

Each l_i is a label describing a possible state of that environmental factor. There could be different sets of labels to be simultaneously applied (e.g., describing bandwidth and connectivity), represented as:

$$\mathcal{L}^* = \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n$$

There are also a number of algorithms to choose from:

$$\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$$

Finally there is a performance function p for each time t:

$$p : \mathcal{L}_1 \times \mathcal{L}_2 \times \cdots \times \mathcal{L}_n \times \mathcal{A} \to \mathbb{R}$$

The goal is to choose an $a \in \mathcal{A}$ such that p is maximized; that is $a = \arg\max_{a \in \mathcal{A}} p$.

Every scenario would have to define its own algorithms, labels, and performance function.

In summary: we want to select, on each node, a persistence algorithm that is optimal for that node to use, given the network state in its neighborhood. This is different from many other approaching that try to distributedly choose an algorithm that will be used globally. This approach takes into account the fact that different algorithms may be more suited for different subsets of the network.

978-0-7695-5124-1/13 $31.00 © 2013 IEEE
DOI 10.1109/MILCOM.2013.248

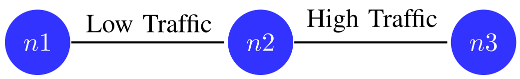

Fig. 1: Illustration of the necessity to incorporate remote observations for protocol selection decisions.

III. Technical Approach

We propose a technique which uses sensed network information to intelligently select both a Transport and Session layer protocol which is suited for the estimated network state. This is novel as it alleviates the need for applications' users to specify a protocol to use, and instead allows them to simply describe the level of fidelity needed.

In this work two Session and two Transport layer protocols were applied to various networks in different combinations. The Session layer protocols were:

1) *Rumor Mongering [3]:* Broadcasts messages with some frequency. Naïve, high-bandwidth, but very fast.
2) *Trickle [4]:* Broadcasts summary data and only retransmits messages when necessary. Low-bandwidth, but slower.

For the Transport layer, multicast UDP was used either with or without Classical Flooding (CF). When CF is on, every message is forwarded exactly once by every node.

Figures 2 and 3 show that both the degree of connectedness and bandwidth utilization can drastically affect the effectiveness of different protocol combinations. Specifically, the more verbose protocols (Rumor Mongering and CF) perform extremely well in low-traffic networks, but delivery drastically fewer messages in high-traffic networks.

It is therefore clear that no static set of protocols can adequately handle an extremely dynamic environment, and there is a need for dynamic selection.

Further, it is insufficient to take into account only local observations about the state of the network: they must be balanced with observations from other hosts as network attributes may be different even a few hops away. For example, consider a simple topology as depicted in Figure 1. If node $n1$ bases its decision about the best protocol solely on local observations, it may choose a protocol that would send too much traffic over the link to node $n2$, exacerbating the problem of high traffic at node $n2$. This is analogous to the hidden terminal problem. One node ($n1$) is unable to detect that its actions conflict with another node's ($n3$) actions from the point of view of a third node ($n2$).

The approach taken in this paper is an application of the work of Doyle, et al. on distributedly estimating network conditions [5] to our previous work on the performance of different data dissemination / persistence protocols in varying network conditions [2]. The network's connectivity and data rate is estimated, and then a protocol selected based on the previously seen performance characteristics of that protocol in the environment the node estimates the network to be in.

Further, each host will not estimate the network state based solely on their own experience, but also that of their neighbors.

		$\tau_c(c')$		
		High	**Medium**	**Low**
$\tau_t(t')$	**Low**	Rumor	Rumor	Rumor + CF
	High	Trickle + CF	Trickle	Trickle

TABLE I: Empirically determined best performing algorithms.

A. Markov Random Fields

The overall goal of using MRFs is to assign labels to the network based upon observations. Labels could be anything, such as link change rate, loss in the network, etc.

Given a set of network observations θ and labels $\mathcal{L}$, the goal is to find an $l \in \mathcal{L}$ with maximal posterior probability. Each posterior can be determined with Equation 1.

$$\underbrace{p(\mathcal{L}|\theta)}_{Posterior} \propto \underbrace{p(\theta|\mathcal{L})}_{Likelihood} \times \underbrace{p(\mathcal{L})}_{Prior} \tag{1}$$

B. Merging Local and Remote Observations

Each node maintains an estimate of the probability that each label is correct. The set of label estimates, $l_e \in (L)_e$, is a one-to-one mapping from the set of labels $estimate : \mathcal{L} \to \mathcal{L}_e$. Every time a new θ_i is received, this data is incorporated into the local value of l_e. Specifically, the current value of l_e is used as input to a Normal Distribution which will indicate the probability of the label being consistent with local observations. Then, a weighted sum of remote observations is added to this, as shown in Equation 2, and the local observation value, θ_i, is updated.

λ is a constant for weighting remote observations. A high value gives more weight to remote observations than local, and a low value gives more weight to local observations. In all experiments this value was set to 80 as described in [5].

$$\begin{aligned} c' &= N(c) + \frac{\lambda}{|\mathcal{R}|}\sum_{i\in\mathcal{R}} c_i \\ t' &= N(t) + \frac{\lambda}{|\mathcal{R}|}\sum_{i\in\mathcal{R}} t_i \end{aligned} \tag{2}$$

Further we establish two functions which will classify observations. $\tau_c(x) \in \mathcal{C}$ (Equation 3) classifies the network into high, medium, or low where $x = c'$. $\tau_t(x) \in \mathcal{T}$ (Equation 4) classifies the network as high or low for traffic where $x = t'$.

$$\tau_c(x) = \begin{cases} h, & 4 \le x \le 9 \\ m, & 2 \le x < 4 \\ l, & 0 \le x < 2 \end{cases} \tag{3}$$

$$\tau_t(x) = \begin{cases} h, & .5 \le x \le 1 \\ l, & 0 \le x < .5 \end{cases} \tag{4}$$

IV. Empirical Validation

To test this approach, the same scenarios from [2] were used. These scenarios involve ten nodes moving around a bounded area using the Reference Point Group Mobility (RPGM) model [6]. The parameters and details are omitted in this paper, but can be found in the original publication.

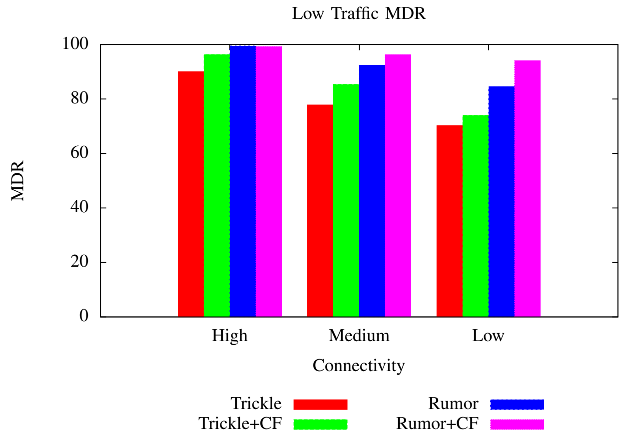

Fig. 2: Message delivery ratio for low-traffic scenarios.

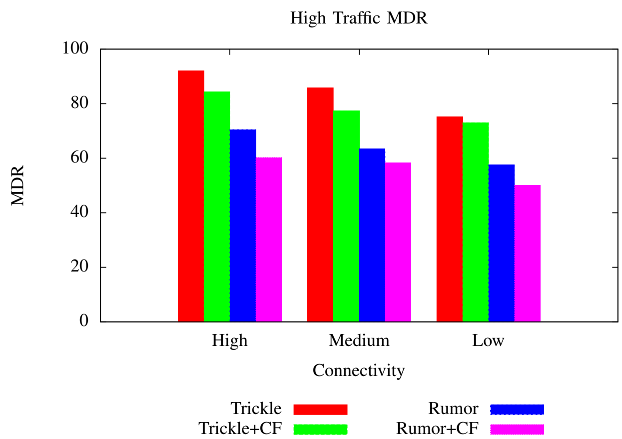

Fig. 3: Message delivery ratio for high-traffic scenarios.

A. Algorithm

For this analysis, there are two sets of labels as shown below.

1) $\mathcal{L}_C = \{h, m, l\}$ indicating if the network has $h$igh, $m$edium, or $l$ow connectivity.
2) $\mathcal{L}_T = \{h, l\}$ indicating if there is $h$igh amounts of traffic or $l$ow amounts of traffic.

Each node i maintains a set of observations $\theta_i = \langle c, t \rangle$ where c is the estimated number of neighbors and t is the estimated amount of bandwidth utilization. The number of neighbor is estimated using `HELO` messages. The bandwidth usage t is estimated by monitoring the number of packets sent over the network.

At some fixed interval, each node broadcasts its θ_i to its neighbors. This serves two purposes. First, the receiving nodes aggregate these into $\mathcal{R}$, the magnitude of which is used to update their own c value. Second, $\mathcal{R}$ is used to take into account remote observations.

B. Emulation Environment

Experiments were run in the CORE [7] emulator with a basic on/off radio model. That is, a pair of nodes can either communicate with each other or not based on distance. The bandwidth for all links was 11 Mbps.

In all scenarios, a constant stream of UDP packets is broadcast by all nodes with MGEN [8] to simulate traffic from other applications. For low-traffic scenarios, each nodes' traffic uses approximately 1% of the total available bandwidth. In high-traffic scenarios, this value is increased to 8%. The value of 8% was chosen so that in a fully-connected network, approximately 20% of the bandwidth would be available. We must emphasize that we are using a very abstract link layer model, where utilization metrics perhaps make more sense than they would on a "real" wireless network. However, our approach of empirically determining what "high" and "low" mean would be the same on an actual wireless network.

There are two categories of metrics that were collected. The first measures the performance of this approach at judging the state of the network and the speed at which nodes come to agreement. The metrics used for this are:

1) **Average Delay to Agreement:** The average time it takes for all nodes in the network to agree on a protocol set to use.
2) **Average Degree of the Network vs. Belief Percentage:** A simple metric comparing the actual average degree of the network nodes versus what percentage of the nodes believe the network is dense or sparse.

The second category of metrics (which is less important for this class, but interesting nonetheless) measures the effectiveness of switching protocols with this method. The sole metric for this is Message Delivery Ratio (MDR) which is the percentage of messages delivered to all nodes. The MDR of this approach is compared to the best protocol combination shown in Table I.

C. Approach Performance

Figure 4 shows that the average network degree properly influences how strongly nodes believe the network is sparse or dense. Note that as the average degree of the network increases to fully connected (9 neighbors) the percentage of the nodes that believes the network is highly-connected increases linearly.

This is desirable since the transition between labels should be smooth and not have sudden spikes.

The delay to agreement across all experiments is shown in Figure 5. As the network becomes more connected, the latency is lower because nodes have more neighbors and therefore θ_i exchanges occur more quickly.

It is possible that the network will never completely convert to a single value. This can happen when there are pockets of the network that have different network conditions than the rest of the network. We view this as a good feature of this approach, because it will cause the protocol selection algorithm to select a protocol that is suited for the actual condition in these pockets.

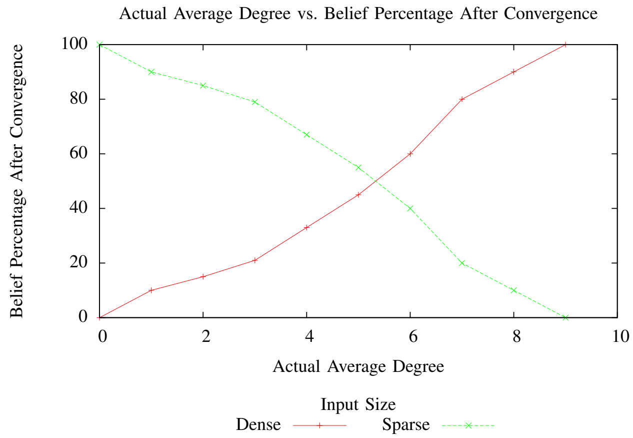

Fig. 4: Agreement amongst nodes about the density of the network.

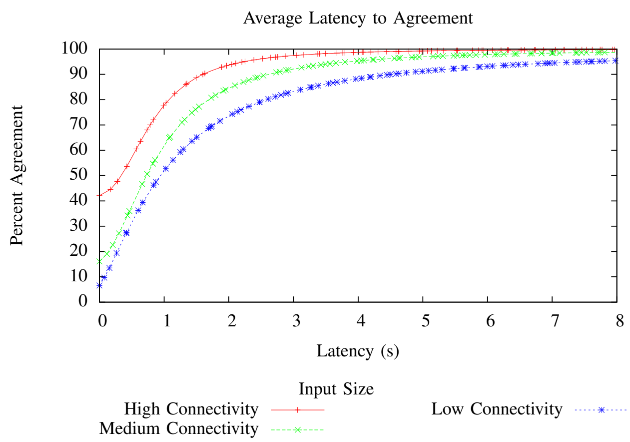

Fig. 5: Percentage of nodes agreeing on a protocol after a certain amount of time.

D. Effectiveness of Dynamic Selection

This section analyzes how useful it is to apply the described approach to improve MDR in challenged networks. Figure 6 shows this approach results in slightly higher MDR in low-traffic scenarios. In the highly connected, low-traffic scenario, there was only an improvement of around 4%.

The largest gains in MDR occur in high-traffic (and therefore bandwidth constrained) scenarios as shown in Figure 7. Even when highly-connected, dynamically selecting a protocol increases MDR by 5%. In sparser networks, the gain is as much as 50%.

V. Related Work

A. Persistence

Network transport protocols for large-scale group dissemination have been developed since the early 1990s. The Scalable Reliable Multicast (SRM) protocol [9] was developed to provided localized, many-to-many content repair for receivers with missing data. The Multicast Dissemination Protocol

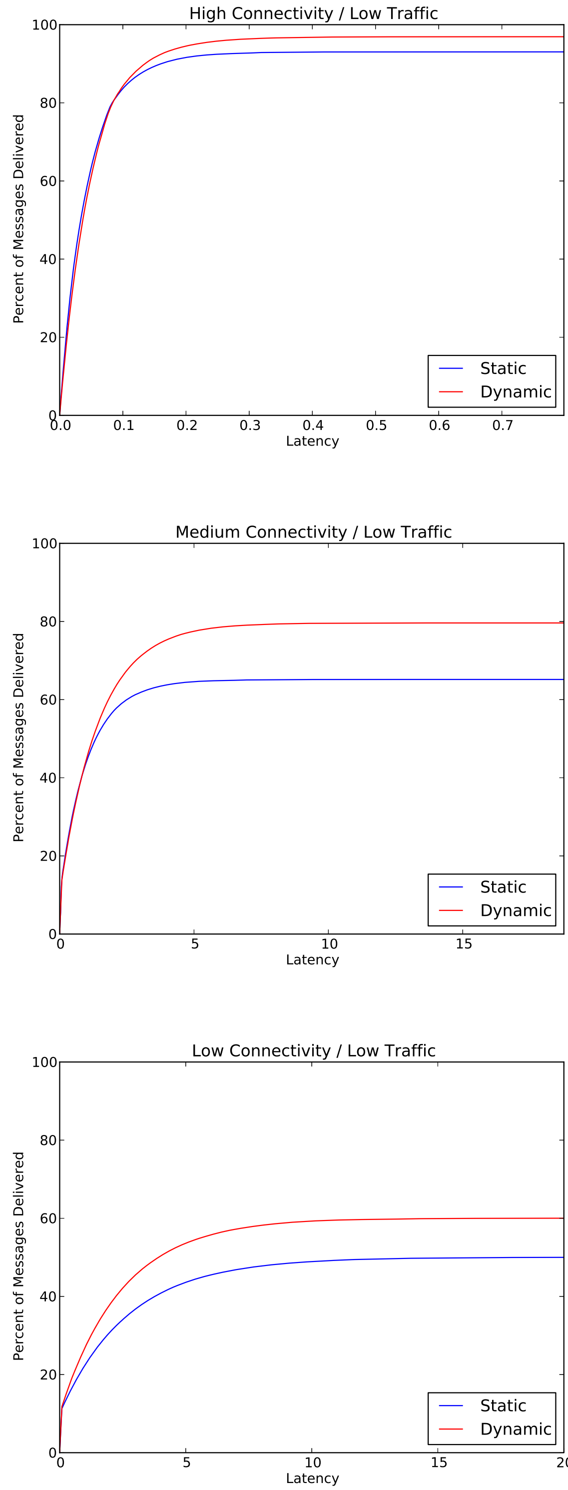

Fig. 6: Comparison of MDR using dynamic protocol selection vs static assignment in low-traffic scenarios with varying connectivity.

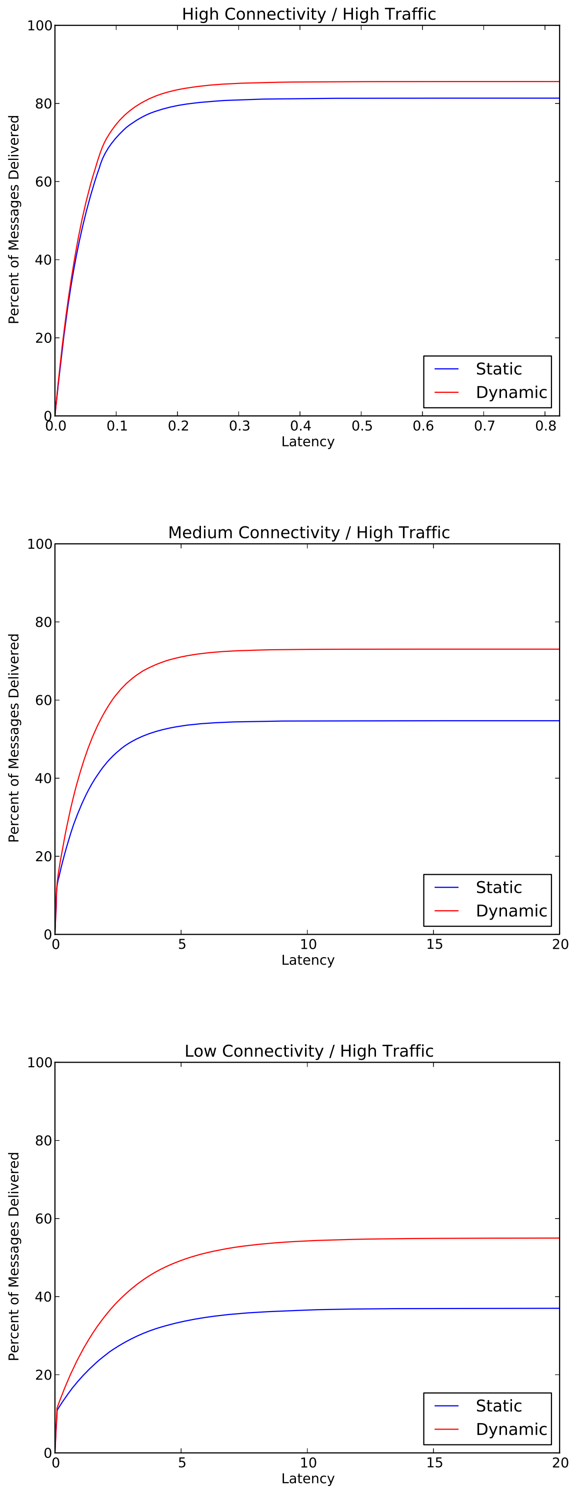

Fig. 7: Comparison of MDR using dynamic protocol selection vs static assignment in high-traffic scenarios with varying connectivity.

(MDP) [10] extended classical UDP multicast using a one-to-many model to provide a scalable, negative-acknowledgment (NACK) protocol with missing message discovery, repair, and retransmission. MDP was one of the first protocols to use Reed-Solomon (RS) codes, potentially improving scalability and delay with respect to the repair process [11].

MDP's successor protocol, NORM (NACK-Oriented Reliable Multicast) [12], provides similar features but also streaming capabilities, and enables reliable unicast transport as a wireless alternative to TCP. NORM has been well-documented and is transitioning into a standards track Internet RFC [12].

Other related transport work has focused on modifying traditional transport protocols for MANETs. For example, ATCP (TCP for Ad-hoc networks) [13] provides additional features to TCP, improving its performance in networks with high bit-error rate.

Due to their origins in high-fidelity enterprise networks, however, these protocols tend to have limitations when applied to networks with high mobility and topological churn, such as tactical-edge MANETs. Additionally, most of these protocols focus on point-to-point communications, and do not sufficiently address group communications.

Long-term communication challenges cannot always be solved with reliable transport protocols alone, because they typically focus on short-term network transport sessions. For example, a situational awareness application would need all position information to be repopulated after significant session disruption, system failure, or network failure. In the case of chat clients, a message may need to be delivered even if the intended client was unreachable for a long period of time after the original transmission.

This necessitates a mechanism by which two or more nodes can exchange and synchronize application data, long after the data is originally sent. To do so, messages must be maintained and synchronized across the network.

Early work in this area can be traced to Demers' seminal paper [3] which applied epidemic protocols to distributed database synchronization. Multicast-based consistency techniques where also examined in Van Hook's work on consistency objects in large-scale distributed simulation projects [14].

Since then, many protocols have been developed for the purpose of synchronizing application data. Trickle [4], for example, uses small metadata advertisements to exchange necessary data for synchronization. DIP [15] improved the advertisement process, using a binary-search to reduce overhead. GoSyP [16] employs a more stateful approach, sending unicast messages between pairs synchronizing nodes. Scuttlebutt [17] uses sequence numbers and unicast exchanges to transmit only what is necessary to synchronize nodes.

Although each protocol uses different mechanisms, the overall goal is the same: exchange small amounts of information, locate missing or outdated messages, and then reconcile those discrepancies through message repairing.

B. Dynamic Protocol Selection

Selecting individual layers of the network stack dynamically is not a new topic. Most of this work involves only a single layer of the stack, however. Quenum describes a method of dynamically selecting an application-level protocol for use in Multi-Agent Systems [18] but makes assumptions about the transport and lower layer functionality. Nakajima [19] introduces a cross-layer technique, but it is exclusively applied to CORBA networks.

A slight departure from this is an adaptive routing protocol selection algorithm [20], which is aimed at primarily reducing resource utilization on host machines.

Selecting an optimal combination of protocols can prove critical. An empirical analysis of different protocols in various environments has shown that the combination of protocol and environment can have a dramatic effect on the measures of effectiveness (latency, message delivery rate) [2].

Finally, the most obviously related work is that of Doyle, et al [5]. The authors describe a very similar to framework to the one used in this paper, for dynamically selecting routing protocols in a MANET.

VI. Conclusions and Future Work

In all experiments, the described approach resulted in an increase in MDR across the network. Only in the highly-connected scenarios was this increase negligible, and in the most challenged networks substantial improvements in MDR of up to 50% were achieved.

For the mobility patterns used, the network came to agreement reasonably quickly. This agreement was long-tailed: most nodes agreed very quickly, but the remaining took considerably longer.

In future work it would be beneficial to analyze the bandwidth usage of static and dynamic approaches. Further, it would be interesting to investigate using continuous values for labels rather than a discrete set. That is, the network would have a degree of connectivity and bandwidth usage, rather than just being high, medium, or low. Using a continuous value may not be useful for dynamically selecting from a discrete set of protocols [21], but it may be useful for tuning parameters of protocols that are sensitive to network conditions.

Another area worth looking at is statistical techniques such as a Kalman filter to smooth out the oscillations in network state. The measurements taken by each node are inherently noisy, and cutting through the noise could yield a more stable approach.

Finally, we plan to look at other algorithms and protocols that can be dynamically selected to improve performance, as well as other variables that affect the performance of these algorithms and protocols. This will require more empirical analysis of leading algorithms and protocols.

References

[1] M. S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," Internet Requests for Comments, RFC Editor, RFC 2501, January 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2501.txt

[2] A. M. Rosenfeld, R. N. Lass, D. S. Ingram, W. C. Regli, and J. P. Macker, "A comparison of group-based data persistence techniques in manets," in *Military Communications Conference*, 2012.

[3] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, ser. PODC '87. New York, NY, USA: ACM, 1987, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/41840.41841

[4] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *In Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2004, pp. 15–28.

[5] L. E. Doyle, A. C. Kokaram, S. J. Doyle, and T. K. Forde, "Ad hoc networking, markov random fields, and decision making," *IEEE Signal Processing Magazine*, vol. 23, no. 5, pp. 63–73, 2006.

[6] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[7] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "CORE: A real-time network emulator," in *IEEE Military Communications Conference*, 2008, pp. 1–7.

[8] Naval Research Laboratory — Networks and Communication Systems Branch, "Multi-generator (mgen)," 2005. [Online]. Available: cs.itd.nrl.navy.mil/work/mgen

[9] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transactions on Networking (TON)*, vol. 5, no. 6, pp. 784–803, 1997.

[10] J. Macker and R. Adamson, "The multicast dissemination protocol (mdp)," Internet Engineering Task Force, 1999. [Online]. Available: http://cs.itd.nrl.navy.mil/pubs/docs/draft-macker-rmt-mdp-00.txt

[11] J. Macker, "Reliable multicast transport and integrated erasure-based forward error correction," in *MILCOM 97 Proceedings*, vol. 2. IEEE, 1997, pp. 973–977.

[12] B. Adamson, C. Bormann, M. Handley, and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol," RFC 5740 (Proposed Standard), Internet Engineering Task Force, Nov. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5740.txt

[13] J. Liu and S. Singh, "Atcp: Tcp for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 7, pp. 1300 –1315, jul 2001.

[14] D. Van Hook, J. Calvin, and J. Smith, "Data consistency mechanisms to support distributed simulation," in *Proceedings of the 12th Workshop on Standards for the Interoperability of Distributed simulations (Orlando, FL, Mar.)*. Citeseer, 1995.

[15] K. Lin and P. Levis, "Data discovery and dissemination with dip," in *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, april 2008, pp. 433 –444.

[16] A. Rosenfeld, D. Kusic, and W. Regli, "A gossip-based synchronization protocol for state consistency in distributed applications," *MobiHoc Tactical MANET Workshop*, 2011.

[17] R. van Renesse, D. Dumitriu, V. Gough, and C. Thomas, "Efficient reconciliation and flow control for anti-entropy protocols," in *Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware*, ser. LADIS '08. New York, NY, USA: ACM, 2008, pp. 6:1–6:7. [Online]. Available: http://doi.acm.org/10.1145/1529974.1529983

[18] J. Quenum, S. Aknine, O. Shehory, and S. Honiden, "Dynamic protocol selection in open and heterogeneous systems," in *Intelligent Agent Technology, 2006. IAT '06. IEEE/WIC/ACM International Conference on*, dec. 2006, pp. 333 –341.

[19] T. Nakajima, "Dynamic transport protocol selection in a corba system," in *Object-Oriented Real-Time Distributed Computing, 2000. (ISORC 2000) Proceedings. Third IEEE International Symposium on*, 2000, pp. 42 –51.

[20] M.-S. Jian, T.-Y. Chou, and C. Lee, "An adaptive routing protocol selection algorithm corresponding to dynamic wireless network environment," in *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, vol. 2, feb. 2006, pp. 6 pp. –944.

[21] L. Lamport, "Buridan's principle," *Foundations of Physics*, vol. 42, no. 2, pp. 1056–1066, August 2012.