Understanding Stateful vs Stateless Communication Strategies for Ad hoc Networks

Victoria Manfredi
Dept of Computer Science
Boston University
Boston, MA 02215
vmanfred@cs.bu.edu

Mark Crovella
Dept of Computer Science
Boston University
Boston, MA 02215
crovella@cs.bu.edu

Jim Kurose
Dept of Computer Science
U of Massachusetts Amherst
Amherst, MA 01003
kurose@cs.umass.edu

ABSTRACT

Structural change and uncertainty are fundamental properties of an ad hoc network, making it difficult to develop *communication strategies*, i.e., network-level approaches to transport data from sender to receiver. At a basic level, change and uncertainty affect how long any state maintained by a communication strategy remains useful, and so influence the trade-offs made to collect that state. In this paper, we introduce a framework for organizing the decision space for deciding when a communication strategy should maintain state, and what type of state should be maintained, in an ad hoc network. The framework is based on our observation that three network properties (connectivity, unpredictability, and resource contention) determine when state is useful. Using the framework, we make three contributions. First, we illustrate the framework by showing an instantiation in terms of specific measures that can be used to describe a network setting. Second, we validate the framework by showing it correctly and consistently organizes the decision space for different communication strategies. Finally, we demonstrate the analytic power of the framework by using it to (1) uncover surprising aspects of well-known traces, and (2) identify the need for, and value of, a new strategy for network communication.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; C.2.1 [**Network Architecture and Design**]: Store and forward networks

General Terms

Design, Experimentation, Performance

Keywords

Ad hoc networks, Delay tolerant networks, Communication strategies, State maintenance, Entropy

MobiCom'11 September 19–23, 2011, Las Vegas, Nevada, USA.

1. INTRODUCTION

In an ad hoc network, mobile nodes communicate with each other directly rather than via fixed infrastructure such as cell phone towers or base stations. By not relying on fixed infrastructure, ad hoc networks can be quickly deployed, are more adaptable than typical wireless or wired networks, and are better able to accommodate unforeseen events. However, the full potential of such networks has not yet been realized. To a much greater degree than in a wired network, structural change and uncertainty are fundamental properties of an ad hoc network, making it difficult to develop network-level approaches to transport data from sender to receiver; we call such approaches *communication strategies*. Consequently, a variety of communication strategies have been proposed, with different strategies broadly targeting different classes of ad hoc networks: from routing [1], to epidemic forwarding [22, 11, 26], to delay tolerant network (DTN) forwarding [7, 12, 27]. Unfortunately, there is no precise understanding of how to determine the appropriate communication strategy for an arbitrary ad hoc network.

Fundamentally, change and uncertainty affect how long state maintained by a node remains useful, and so influence the trade-offs made to collect that state (such as whether to expend bandwidth on control packets to find routes or only transmit data packets). In traditional networks the issue is whether to maintain hard state (i.e., state that is explicitly removed) or soft state (i.e., state that is removed via timeouts) [13]. In an ad hoc network the issue is whether to maintain soft state or no state, as well as the needed level of soft state accuracy [2, 3]. One way to organize communication strategies is in terms of the kind of soft state that a strategy might maintain. For instance, information may be stored about the state of the network, such as whether a route exists to a particular destination: we call this *control state*. Alternatively, information may be stored about the state of the data to be transmitted, such as how long data packets should be stored, as well as the data packets themselves: we call this *data state*. At one end of this spectrum, in a routed network nodes primarily store control state. At the other end of the spectrum, in a DTN nodes primarily store data state. Unfortunately, for an arbitrary ad hoc network, it can be difficult to identify when a communication strategy should maintain control state or data state.

In this paper, we introduce a framework for organizing the decision space for deciding when a communication strategy should maintain state, and what type of state should be maintained, in an ad hoc network. The framework is based on our observation that three network properties (connec-

tivity, unpredictability, and resource contention) determine when state is useful. We make three contributions:

- First, we illustrate the framework by showing a concrete instantiation in terms of three empirical measures. The measures we choose to use are: (1) the average number of flows in the network, (2) the probability that an arbitrary route exists, and (3) a metric we define, the *average link-up entropy*. Average link-up entropy measures the value of routing information: it is the average conditional entropy of the current network state (i.e., the set of up/down link states) given the network state at some time in the past.

- Second, we validate the framework by showing that it correctly and consistently organizes the decision space across vastly different networks. The first network is a torus network with time-varying link availability. The second is a network in which nodes move according to random waypoint mobility. Each type of network, depending on the parameter settings, can show a wide range of properties, from mostly-connected to mostly-disconnected. We show that the natural parameterizations of these networks are not comparable, nor are they particularly useful in choosing a communication strategy. However, when placed in the framework, it is easy to identify the situations in which each communication strategy is typically appropriate.

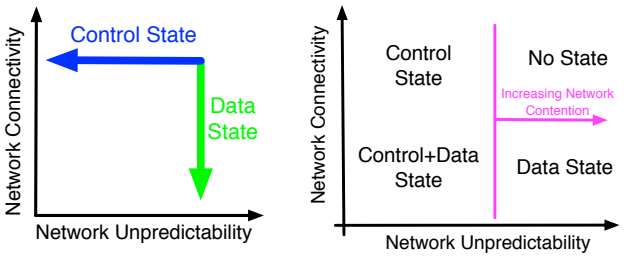
- Finally, we demonstrate the analytic power of the framework by using it to (1) uncover surprising aspects of well-known traces, and (2) identify the need for and value of a new strategy for network communication. We specifically find that many well known contact traces used for DTN studies do *not* fall into the DTN region in the framework. In fact, the region in which they fall has not been extensively studied in prior work. We show that using the communication strategy suggested by the framework for this region achieves higher goodput than routing or DTN forwarding.

Current ad hoc network deployments typically assume *a priori* knowledge of the most appropriate communication strategy to use, whether stateful or stateless. Yet in general the structure and traffic characteristics of an ad hoc network are unknown until deployment. The results in this paper suggest that structural change and uncertainty should not be viewed as an afterthought to protocol behavior, but should instead directly shape that behavior, and that understanding when protocols should maintain state is critical for ensuring that protocols correctly adapt to changes.

The rest of this paper is structured as follows. In Section 2, we introduce the framework. In Section 3, we define the empirical measures we use to instantiate the framework. In Section 4, we validate the framework. In Section 5 we use the framework to classify traces previously studied in the literature. Finally, in Section 6, we summarize our results and describe future work.

2. STATE MAINTENANCE

Our goal is to *organize the decision space for deciding when a communication strategy should maintain state, and what type of state should be maintained, in an ad hoc network*. To achieve communication, nodes can store two kinds



(a) Intuition (b) Framework

Figure 1: Overviewing the framework.

of state: control state and data state. **Control state** is information about the state of the network, such as whether a route exists to a particular destination. **Data state** is information about the state of the data to be transmitted, such as how long data packets should be stored, as well as the data packets themselves. Of course, all communication strategies maintain data state, however briefly, because a data packet must be buffered before being transmitted. For the purposes of our discussion here, we refer to stored data packets as data state only when those packets are stored for an extended period of time (i.e., are not re-transmitted immediately upon receipt). Conceptually, the control plane in a traditional network disseminates control state and the data plane forwards data state. Increasing the amount and quality of control state corresponds to increasing the rate at which updates about the state of the network are received. Increasing the amount of data state corresponds to increasing the amount of time data packets are stored. Like any soft state [13], both control state and data state will eventually be modified as timers expire.

The starting point for our framework is the intuitive observation that *control state is valuable when a network is predictable, while data state is valuable when a network is not well connected*. This idea is shown diagrammatically in Fig. 1(a). The figure immediately suggests four classes of communication strategies.

1. **Control-state strategies.** These strategies maintain primarily control state. Routing [1] is the typical example of a control-state strategy. Routing first disseminates control packets to identify routes (the control state) and then transmits data over those routes.

2. **Data-state strategies.** These strategies maintain primarily data state (i.e., they store data packets for some non-trivial amount of time). DTN forwarding [7, 12, 27] is the typical example of a data-state strategy. In DTN forwarding, nodes store and carry data packets until they encounter other nodes (permitting packets to be delivered even when end-to-end routes are unlikely exist). Essentially, DTN forwarding stores data packets until the network changes in such a way that the data packets can be delivered.

3. **No-state strategies.** These strategies maintain little or no state. Epidemic forwarding [22, 11, 26], specifically flooding, is the typical example of a no-state strategy. In flooding, when a node receives a data packet it is immediately re-transmitted.

4. **Control + data state strategies.** These strategies maintain both control and data state. We observe that for the most part, such strategies have been underexplored in the literature.

2.1 Proposed Framework

Our proposed framework is shown in Fig. 1(b). Illustrated in Fig. 1(b), as **network unpredictability** decreases, we expect that maintaining control state becomes useful since routes remain accurate for long periods of time. As **network connectivity** decreases, we expect that maintaining data state becomes useful, since nodes will likely have to store data packets until the network changes in such a way that the packets can be delivered. Finally, as **network contention** (i.e., the amount of contention for network resources) increases, we expect that the threshold (shown as a vertical line) below which control state should be maintained moves to the right. That is, even less predictable control state becomes useful if network contention is high since control state can be used to mitigate contention.

To evaluate the framework, we need to specify what makes a particular communication strategy **appropriate**. We define the appropriate strategy to be the one that maximizes goodput. Goodput is defined as the number of unique data packets received at the destination and depends on the tradeoffs that the different classes of communication strategies make. For instance, to maintain control state (specifically routes over which data can be efficiently transmitted) control-state strategies expend bandwidth on control packets. In comparison, data-state strategies maintain no control state (so do not expend bandwidth on control packets) but may expend bandwidth on multiple transmissions of a packet.

2.2 Related Work

One way that our framework can be used is to classify different network settings according to the most appropriate class of communication strategies. Other work has also attempted such a classification. The authors in [20] develop a framework specifically for the space of DTNs based on connectivity, while the authors in [4] propose and evaluate a distributed algorithm to classify networks as *connected*, *intermittently connected*, or *disconnected* based on the presence of different types of paths in the network. For each network class, the approach in [4] associates a communication strategy: routing, DTN forwarding, or message ferrying. Our work goes beyond that in [20, 4] by determining the appropriate strategy based on the type of state that should be maintained in the network (rather than the network class) and importantly, takes control signaling costs (i.e., to maintain control state) into consideration.

Other work has also investigated the effect of control signaling on goodput. The authors in [10] determine how much protocol information is needed per packet to allow the start and end times of packets to be recovered at the destination with a given average error, while the authors in [24] determine the minimal rate at which to receive state updates to ensure high probability of packet delivery for link state routing. The authors in [23] develop and evaluate a model to predict the control overhead for different routing protocols. In comparison to the work in [10, 24, 23], our work looks not just at how maintaining control state affects routing goodput, but also at when to use a protocol that maintains less control state than that of a protocol such as routing.

3. INSTANTIATING THE FRAMEWORK

To instantiate the framework we need empirical measures of network connectivity, unpredictability, and contention. By **empirical measure**, we mean any fundamental network or traffic characteristic that a wireless or mobile node could measure. Our goal here is to identify natural empirical measures.

3.1 Measures of network connectivity

These measures indicate the potential benefits of storing data packets for an extended period of time, and therefore of maintaining data state. We use the **probability that an arbitrary route exists** in the network, ϕ, as our measure of network connectivity. The probability that an arbitrary route exists indicates how often an end-to-end route is found between two arbitrary nodes. There exists a critical threshold probability, θ, for the probability that a link exists, p_l, such that when $p_l > \theta$, ϕ is close to one and when $p_l < \theta$, ϕ is close to zero. Thus, θ is the percolation threshold. Knowing θ is not a sufficient empirical measure in itself, however, since different networks will percolate at different values of θ. Hence ϕ is more informative since we can associate the value $\phi = 0.5$ with the percolation threshold, regardless of the value of θ. Intuitively, for $\phi \geq 0.5$ control-state or no-state strategies should be preferred to data-state strategies since an end-to-end route is likely to exist. For $\phi < 0.5$ data-state strategies should become useful since only temporal routes will likely exist for many source-destination pairs.

3.2 Measures of network unpredictability

These measures capture the predictability of route information, such as how long routes remain stable. Thus, these measures indicate whether maintaining control state is useful. Intuitively, the less predictable network changes are, the less preferred control-state strategies will be. We use **entropy** as our measure of network unpredictability. Entropy, a measure of uncertainty, is one natural way to capture the unpredictability of changes in the states (up or down) of links in the network.

To define our entropy-based measure of network unpredictability, let E_t^u be the set of edges up in the network at time t and let $E_{t+i} \subseteq E_t^u$ be the subset of those edges up at time $t+i$. Let g_{t+i} be a random variable representing whether an arbitrary link in E_t^u is up (u) or down (d) at time $t+i$. Then the probability that an arbitrary link in E_t^u is up at time $t+i$ given that all links in E_t^u are up at time t, is computed as follows.

$$P(g_{t+i} = u | g_t = u) \quad = \quad \frac{|E_{t+i}|}{|E_t^u|}$$

We can then define **link-up entropy**, $H(g_{t+i}|g_t)$.

$$H(g_{t+i}|g_t) \quad = \quad -\sum_{y=u,d} P(g_{t+i} = y | g_t = u) \log P(g_{t+i} = y | g_t = u) \qquad (1)$$

In practice, routing protocols typically update routes periodically. We assume routes are updated every T timesteps and refer to T as the **route update interval**. The accuracy of control state decays over the entire interval, T. We also define an **entropy update interval**, T_h, to remove

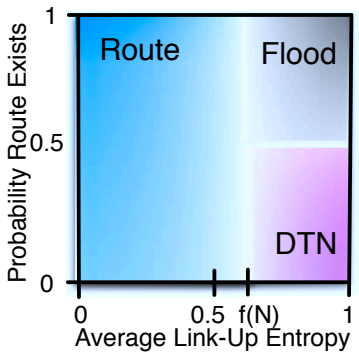


Figure 2: Instantiation of framework using the empirical measures.

the dependence of link-up entropy on a specific time, $t+i$. We can now finally define our proposed measure of network unpredictability, **average link-up entropy**, h:

$$h = \frac{1}{M}\sum_{j=1}^{M}\mathcal{H}_{t(j)+i(j)|t(j)} \quad (2)$$

where M is the total number of timesteps under consideration, $t(j) \leq j$ is the most recent timestep at which an entropy update occurred, $i(j)$ is the number of timesteps since timestep $t(j)$, and

$$\mathcal{H}_{t+i|t} = \begin{cases} H(g_{t+i}|g_t) & \text{if } i \neq 0 \\ 1 & \text{otherwise.} \end{cases} \quad (3)$$

As expressed by $\mathcal{H}_{t+i|t}$, we assume entropy of one (maximum entropy) at entropy update timesteps to capture the fact that during route updates no knowledge of network state is assumed. This mirrors that at timesteps when routes are updated, goodput is zero as no data is transmitted. If routes are updated but no links are found in the network, we assume entropy of one at all timesteps until routes are next updated.

We condition on the links found when routes are updated because it is only those links that will be used in routing. The links that are not up consistently (and are thus unlikely to be found when routes are updated) are not useful for routing and so we do not include their entropy contribution in the calculation of h. As h decreases, the network becomes more predictable and we expect the utility of control-state strategies such as routing to increase. h is similar to the mutual information measure computed in [24] to determine the minimum rate at which to receive link state updates to ensure that the difference between the true and perceived link states is within some error bound (and thereby ensure high probability of packet delivery). We differ from the work in [24] by specifically conditioning on only those links that existed when routes were last updated.

3.3 Measures of network contention

These measures indicate the potential benefits of reducing the network resources each data packet uses to travel from source to destination, and therefore of maintaining control state. We use the average **number of flows** in the network, N, as our measure of network contention. A flow corresponds to a source-destination node pair, where the source is the origin of some data to be sent to the destination. Ignoring energy concerns, if there is only one flow, no-state or data-state strategies are always preferable to control-state strategies. This is because maintaining control state requires signaling, which diminishes the bandwidth available to carry data traffic.

3.4 Instantiated Framework

In Fig. 2, we instantiate the framework using our proposed empirical measures **(1) the average number of flows,** N, **(2) the probability that a route exists,** ϕ, and **(3) average link-up entropy,** h. The different colored regions correspond to different strategies; the darker the color, the larger the expected goodput gains over other strategies. As shown in Fig. 2, we conjecture that when $h \leq f(N)$, routing is appropriate and is most preferred when h is minimized and ϕ is maximized. As N increases, we expect $f(N)$ also increases. We conjecture that for $h > f(N)$ and $\phi < 0.5$, DTN forwarding is appropriate and is most preferred when h is maximized and ϕ is minimized. Finally, we conjecture that for $h > f(N)$ and $\phi \geq 0.5$, flooding is appropriate and is most preferred when h and ϕ are both maximized. We leave a discussion of the low h, low ϕ region until Section 5.

4. FRAMEWORK VALIDATION

In this section, we validate the framework. Our goal here is to check that the framework correctly and consistently organizes the decision space for selecting a communication strategy across different networks. To enable the full parameter space (given by ϕ, h, and N) to be validated, we use network models rather than traces. We next give our simulation setup in MATLAB.

4.1 Communication Strategies

We use routing, flooding, and DTN forwarding as representative examples of control-state, no-state, and data-state strategies. For generality, we develop idealized abstractions of the strategies that contain only the essential features of each class of strategies. Our abstractions are then as follows.

Routing. We base our routing abstraction on a protocol such as link state routing or AODV [19]. Consider how AODV behaves. When senders have data to transmit but no routes are known, control packets must be sent to identify the shortest routes: during this time period, no data can be sent. Once each sender knows the shortest route for its data, the data packets are sent over that route. Thus, in our routing abstraction we divide time into alternating periods of route updates and data transmission. The relative durations of these periods will be important for identifying the situations in which routing is appropriate. We use a route update interval of T timesteps, with T as defined in Section 3.2. All nodes update routes for all flows at the first timestep of the route update interval T. At each of the remaining $T-1$ timesteps, every source attempts to transmit a data packet over the shortest route to the packet destination. A route must exist end-to-end at a timestep for a packet to be delivered using the route. The amount of control overhead incurred then depends on how frequently routes are updated, impacting both the amount and accuracy of the control state maintained.

Notation	Meaning
C_i^t	Capacity of the route used by flow i at time t when routing
L	Average link capacity
τ	Number of timesteps over which DTN forwarding broadcasts each data packet
T	Route update interval
T^*	Optimal route update interval
T_h	Entropy update interval
T_h^*	Optimal entropy update interval
M	Number of timesteps under consideration
$\mathcal{I}_R(t, T^*)$	Indicator function: 1 if routes are not updated at timestep t, 0 otherwise
$\mathcal{I}_F(i, t)$	Indicator function: 1 if destination of flow i is reachable at timestep t, 0 otherwise
$\mathcal{I}_D(i, t, \tau)$	Indicator function: 1 if destination of flow i is reachable over interval from t to $t+\tau$, 0 otherwise

Table 1: Notation used in paper.

Strategy	Goodput for One Flow, i	Goodput for N Flows
Routing	$G_R^i(T^*) = \frac{1}{M}\sum_{t=1}^{M} C_i^t \ \mathcal{I}_R(t, T^*)$	$G_R(T^*, N) = \sum_{i=1}^{N} G_R^i(T^*)$
Flooding	$G_F^i(1) = \frac{1}{M}\sum_{t=1}^{M} \frac{L}{N} \ \mathcal{I}_F(i, t)$	$G_F(1, N) = \sum_{i=1}^{N} G_F^i(1)$
DTN Forwarding	$G_F^i(\tau) = \frac{1}{M\tau}\sum_{t=1}^{M} \frac{L}{N} \ \mathcal{I}_D(i, t, \tau)$	$G_F(\tau, N) = \sum_{i=1}^{N} G_F^i(\tau)$

Table 2: Per-timestep goodput computations used in Section 4. If multiple flows share a link, the available link capacity is shared equally among all flows. Notation given in Table 1.

Flooding. In our flooding abstraction, sources transmit data at every timestep, broadcasting each packet to all nodes reachable at the timestep. With N flows, each source floods data at each timestep at rate L/N, where L is the average link capacity.

DTN forwarding. In our DTN forwarding abstraction, sources broadcast each packet to all nodes reachable in at most τ timesteps. Thus, nodes maintain data state for τ timesteps. With N flows, each source transmits data at rate $\frac{L}{\tau N}$. For $\tau = 1$, DTN forwarding corresponds to flooding. While DTN forwarding for $\tau > 1$ may be more likely than flooding to deliver a particular data packet, it takes longer to do so, and goodput is correspondingly less. We use $\tau = 2, 3$ in the simulations in Section 4.3. We intentionally abstract the spectrum of DTN forwarding strategies using one simple strategy which is to hold the data (maintain data state).

To generate data packets, our simulations assume that each flow generates one new data packet at each timestep. We additionally assume that when a node transmits a data or control packet, the packet is received by all nodes reachable at the timestep (including by nodes multiple hops away). Using these assumptions and our abstractions for the different strategies just described, we give per-timestep goodput computations for the strategies in Table 2.

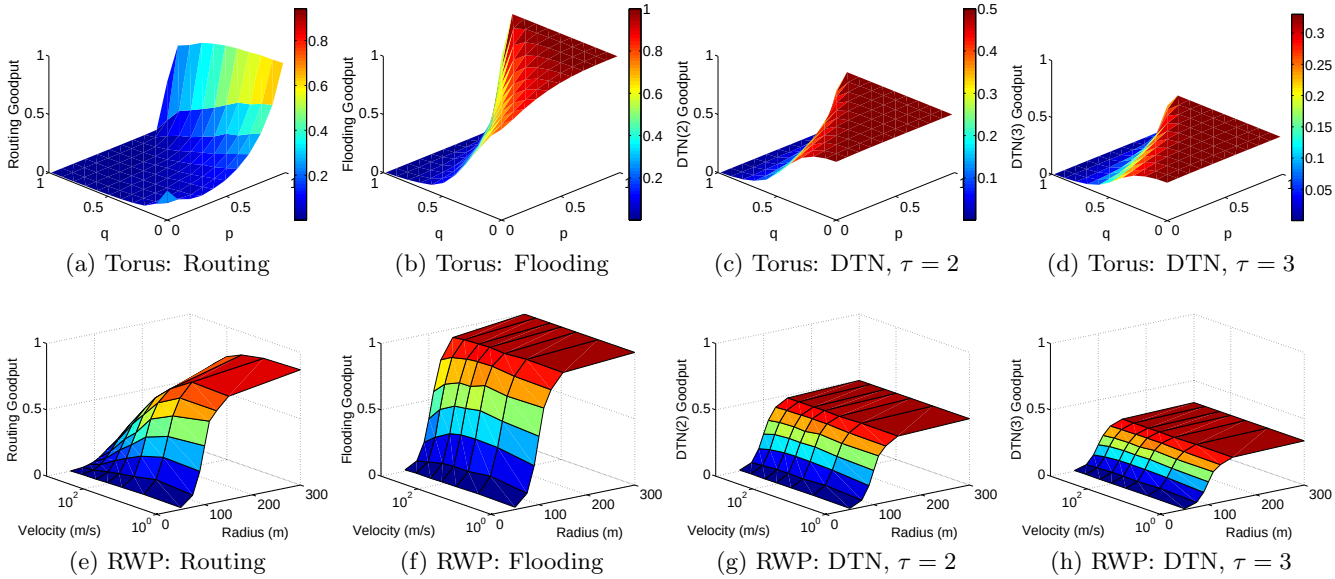
4.2 Network Models

We consider two network models: (1) a torus network with time-varying link availability, and (2) a network in which nodes move according to the random waypoint mobility model [6]. Neither model is expected to be realistic in practice. However, the torus model allows us to consider the simplest possible network case where links are independent and nodes are stationary. The random waypoint model then allows us to build in the complexity of the network and consider correlated links and mobile nodes. We next describe our simulation setups for these models.

Torus. We use a 10×10 torus network with IID links. To model link behavior, we use a 2-state Markov model: links transition from up to up with probability p and down to down with probability q at each timestep. All links have unit capacity. We perform 1500 runs to obtain each point in a plot; each run lasts for 100 timesteps. We use the steady-state probability that a link is up ($\pi = \frac{1-q}{2-p-q}$) to initialize each run of the simulation in steady-state. At each timestep, we update the states of the links and check which routes exist at the timestep, for routing strategies using route update intervals, T, of 1, 2, 5, 10, 25, 50, and 100. We randomly select without replacement N new source-destination pairs (i.e., N flows), every 100 timesteps. At route re-computation timesteps, all capacity is used for control.

Random waypoint (RWP). We use code from [18] to generate traces of random waypoint mobility with constant velocity. For the random waypoint model, instead of p and q, our tunable parameters are now the node *velocity*, v, and *transmission radius*, r. Each simulation is 4000 seconds long. We let the length of one timestep be the transmission time of a packet times the logarithm of the network size. To obtain a realistic timestep, we use a transmission rate of 11 Mbps and a packet size of 1500 bytes. This gives a timestep of 0.012 $s \times \log_{10}(n)$ where n is the number of nodes in the network. Since we use 100 nodes, the timestep is 0.024 s. Every timestep, we sample the network, checking whether any route or a specific route exists for each flow. If a route exists both at the start and end of a timestep, we assume it exists for the duration of the timestep. When the node velocity, v, is greater than 100 m/s we additionally subsample each timestep every $0.024\frac{100}{v}$ seconds and check that the route exists at each of the subsampled times. We use T values of 1, 2, 3, 6, and 10, and randomly select without replacement N new source-destination pairs every 300 timesteps.

For both the torus and random waypoint results, we show average routing goodput results for the T values that maxi-



(a) Torus: Routing (b) Torus: Flooding (c) Torus: DTN, $\tau = 2$ (d) Torus: DTN, $\tau = 3$

(e) RWP: Routing (f) RWP: Flooding (g) RWP: DTN, $\tau = 2$ (h) RWP: DTN, $\tau = 3$

Figure 3: Average per-timestep goodput, one flow.

mize goodput, (T^*), and average link-up entropy results for the T values that minimize h, (T_h^*).

4.3 Results

We first examine our results in the (p, q) and (r, v) spaces that naturally parameterize the torus and random waypoint models, and then examine our results in the (ϕ, h, N) space of the instantiated framework.

4.3.1 *Impact of* (p, q) *and* (r, v) *on Goodput* [1]

Fig. 3(a) to (d) examine how p and q affect goodput in the torus when there is one flow. Fig. 3(a) plots routing goodput: the transition from low to high(er) goodput occurs where $p > 0.5$ and q is small. Next, Fig. 3(b) plots flooding goodput: the transition from low to high goodput now occurs uniformly along the $p = q$ axis. Fig. 3(c) and (d) then plot DTN goodput for $\tau = 2$ and $\tau = 3$ respectively: again, the transition from low to high goodput occurs near the $p = q$ axis, but now the maximum relative goodput is 1/2 and 1/3 that of flooding. Thus, routing transitions from low to high goodput differently than flooding or DTN forwarding.

Fig. 3(e) to (h) examine how v and r affect goodput in random waypoint when there is one flow. Fig. 3(e) plots routing goodput: goodput is maximized for large r and low v. Next, Fig. 3(f) plots flooding goodput: as long as r is greater than $\sim$ 100 m (the percolation threshold), flooding goodput is close to one. Finally, Fig. 3(g) and (h) plot DTN forwarding goodput for $\tau = 2$ and $\tau = 3$ respectively: as long as r is above the 100 m percolation threshold, goodput is maximized (again about 1/2 and 1/3 that of flooding).

4.3.2 *Validating the Framework*

Fig. 4(a) to (d) investigate where each communication strategy maximizes goodput for the torus and random waypoint models as a function of their natural (p, q) and (r, v) parameterizations. As the natural parameterizations are not comparable we have no common decision space from which to select communication strategies for the two models.

Fig. 4(e) to (h) then investigate where each strategy maximizes goodput for the torus and random waypoint models as a function of the empirical measures. ***Fig. 4(e) to (h) show that by transforming to the*** (ϕ, h, N) ***space, the regions where the different strategies maximize goodput are similar for both models. The empirical measures thus consistently organize the decision space and validate the framework in Fig. 2.*** Note that for the random waypoint model, as the probability of a route decreases (i.e., when $\phi < 0.5$), the probability that the route used will be longer than one hop also decreases, and so the average route length also decreases. In comparison, the route length distribution for the torus is fixed since nodes are stationary. DTN forwarding with $\tau = 2, 3$, however, only has an advantage over flooding (which is equivalent to DTN forwarding with $\tau = 1$) when routes exist temporally over exactly two or three timesteps (but not one timestep). Consequently, for random waypoint, DTN forwarding with $\tau = 2, 3$ (as was used in the simulations) is unlikely to be preferred to flooding, and we observe in Fig. 4(h), that DTN forwarding never maximizes goodput.

5. USING THE FRAMEWORK

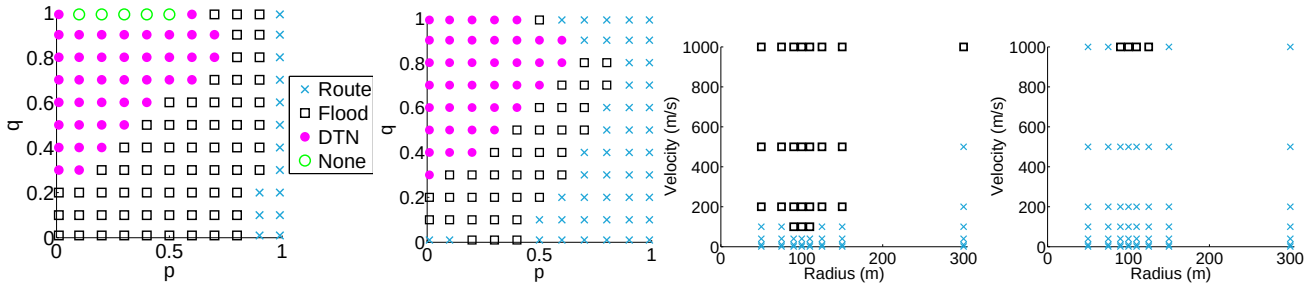
Having validated the framework in the previous section, in this section we use the framework to examine a number of well-known traces. We specifically investigate what type of state should be maintained by communication strategies for those traces.

5.1 Traces

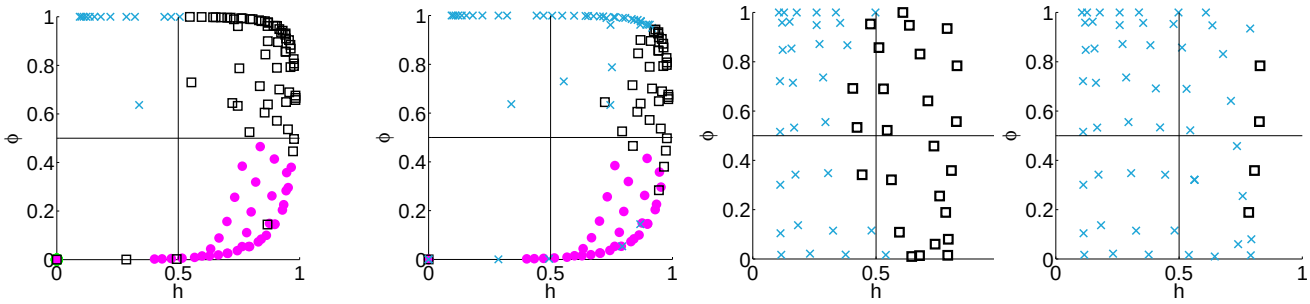
We use five traces from the CRAWDAD repository [14].

- **Infocom06 [8, 9, 21].** This dataset consists of contacts over time among 78 participants and 20 access points at the Infocom 2006 conference.

[1] To understand how (p, q) and (r, v) impact average link-up entropy, h, we refer the reader to Appendix A.



(a) Torus, 2 Flows (b) Torus, 7 Flows (c) RWP, 2 Flows (d) RWP, 7 Flows



(e) Torus, 2 Flows (f) Torus, 7 Flows (g) RWP, 2 Flows (h) RWP, 7 Flows

Figure 4: Validating the framework. None refers to no single strategy maximizing goodput.

- **Infocom05 [8, 9, 21].** This dataset consists of contacts over time among 41 participants at the Infocom 2005 conference.
- **DieselNet [5].** This dataset consists of contacts over time among buses. Bus routes are repeated daily, so we focus only on a trace for one day, October 22, 2007. To the end of the trace (which corresponds to the termination of the bus schedule for the day) we add two hours where we assume buses are parked adjacent to each other in the garage, and that adjacent buses can communicate. This garage interval is used to specifically explore whether the framework correctly classifies the interval as in the control-state regime.
- **Intel [8, 21].** This dataset consists of contacts over time among eight employees at the Intel Research Cambridge Corporate Laboratory.
- **Cambridge [8, 21].** This dataset consists of contacts over time among 12 graduate students at the University of Cambridge.

We implement our simulations in MATLAB. For all traces we use discrete-time simulation, with one timestep equal to one second. In practice the appropriate timestep (such as to compute T_h^*) is the expected time needed for a useful packet transmission to occur, which may depend on the network. By useful we mean the time needed for a data packet to be received by its destination, or for routes to be found.

We compute the empirical measures at each timestep, since we are interested in time-varying behavior. This is different from the simulations in Section 4, where we were interested in steady-state behaviour. We plot an exponentially weighted moving average of the measures using a weight of $\alpha = 0.001$ to average in new values. For instance, instead of computing h as in Equation (2), we now compute h at each timestep j as follows.

$$h_j \quad = \quad (1-\alpha)h_{j-1} + \alpha\mathcal{H}_{t(j)+i(j)|t(j)} \tag{4}$$

We assume $\mathcal{H}_{t(j)+i(j)|t(j)} = 1$ if no links existed in the network at timestep $t(j)$ when routes were updated. We similarly compute a time-averaged ϕ at each timestep j, (ϕ_j), using only the mobile nodes (i.e., we do not include connectivity to access points and external contacts). The T^* and T_h^* values used in the plots are given in Table 3.

5.2 Trace Classification

We compute time-averaged h_j and ϕ_j values for the traces and plot the values computed at each timestep j in Fig. 5, coloring each point according to the corresponding trace. Fig. 5 shows that only DieselNet has behavior distinctly different from the other traces, first, spending time in the *(high h, low ϕ)* region (the data state regime) and then transitioning to the *(low h, high ϕ)* region when in the garage (the control state regime). Because we are plotting time-

	Info06	Info05	DieselNet	Intel	Cambridge
T^*	10	10	6 / 300	30	30
T_h^*	10	30	30	30	30

Table 3: Update intervals used in Fig. 5 to 9. The DieselNet results use $T^* = 6$ until the buses enter the garage, and then $T^* = 300$. For the DieselNet trace we use T and T_h values of 1, 2, 3, 6, 10, 30, 60, 300; the other traces use 1, 2, 3, 6, 10, 30, 60, 3000.

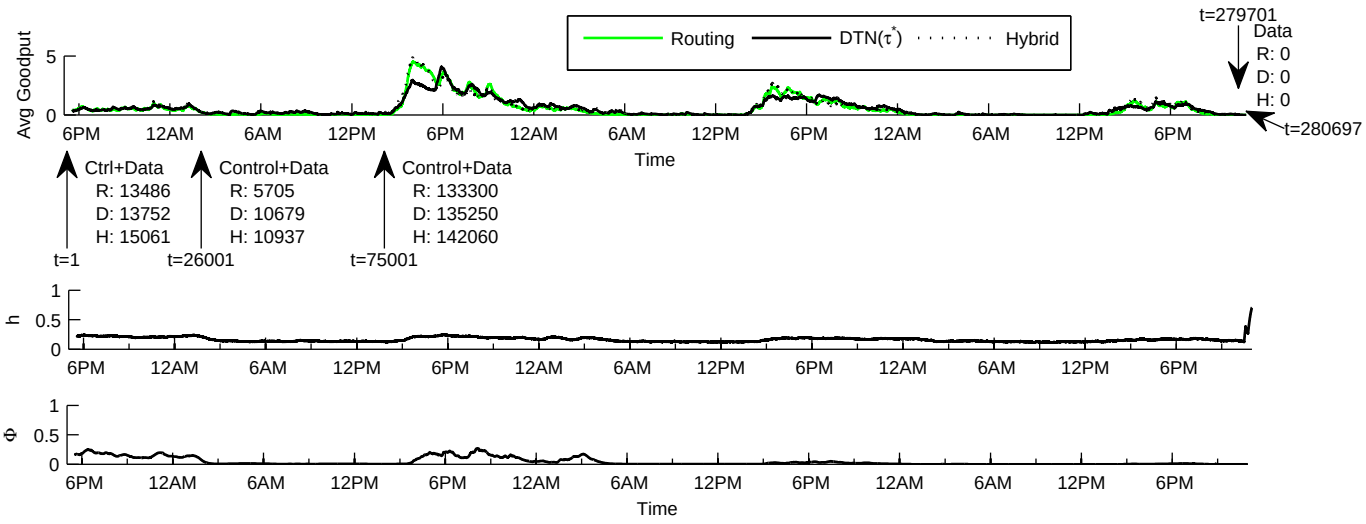


Figure 6: Infocom06 trace results. We annotate each interval with the regime identified by the framework below the goodput plot, and show the number of packets delivered by routing (R), DTN forwarding (D), and hybrid forwarding (H) within the interval.

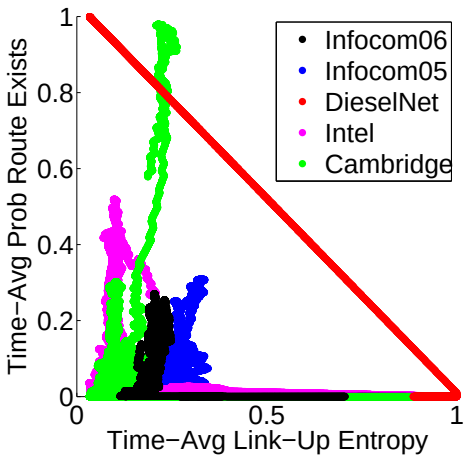


Figure 5: Classifying the traces. Each point represents a time period in a dataset.

averaged values, Fig. 5 shows DieselNet transitioning gradually, rather than jumping directly, between the two regions; however the raw data transitions directly. In comparison, the other traces spend most or all of their time in the *(low h, low ϕ)* region (the control+data state regime). We next examine this region further.

5.3 The Hybrid Region

Our analysis of the traces reveals that most of the traces spend most or all of their time in the low connectivity and low unpredictability region in Fig. 1(b); we will refer to this region as the **hybrid region**. Intuitively, in the hybrid region there are few routes but those routes that do exist are relatively stable. Such a situation could arise, for example, when a subset of nodes are stationary and connected, while the remaining nodes are highly mobile.

The fact that so many traces fall into the hybrid region is surprising. These traces have all been used to evaluate DTN forwarding strategies, which are data state strategies. But our analysis suggests that a control+data state strategy is appropriate for the majority of these traces. Conceptually, a simple control+data state strategy for networks in the hybrid region should perform routing over any predictable routes and DTN forwarding over the remaining unpredictable links. In fact, strategies for networks that fall in the hybrid region have not been extensively explored in the literature, although a few studies have looked at this sort of communication strategy [17, 25, 15]. We next compare the performance of a simple hybrid strategy on the traces with that of routing and DTN forwarding.

5.4 Evaluating Strategies on the Traces

5.4.1 Communication Strategies

We use routing, DTN forwarding, and a hybrid forwarding strategy (defined below) as representative examples of control-state, data-state, and control+data state strategies. In comparison to the simplified strategies used in Section 4 to validate the framework, we use more realistic strategies to evaluate the traces.

To model data traffic, we assume flows exist between all node pairs (and that queues are maintained at nodes for all flows). At each timestep, the source of a randomly selected flow generates one packet to be delivered. Our goal with this traffic model is to isolate the effects of the network traffic from those of the network topology and to ensure that no strategy maximizes goodput simply because of the traffic characteristics. Like with the empirical measures, we plot an exponentially weighted moving average of goodput using a weight of $\alpha = 0.001$ to average in new values. We next describe the strategies.

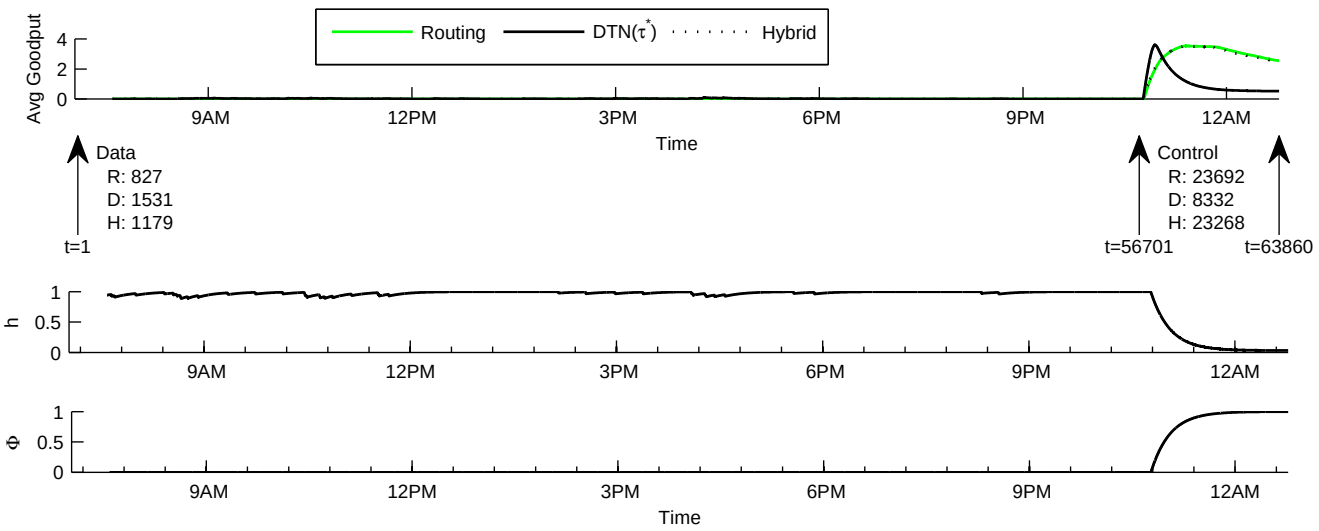


Figure 7: DieselNet trace results.

Routing. We again assume that all nodes update routes during a single timestep. To determine the routing goodput at the other timesteps, we first find all reachable destinations using the routes available from the last route recomputation. We then randomly choose a reachable destination for which there are packets to deliver, and deliver a packet to that destination. We then again check whether any more destinations are reachable, but now only consider those routes from the last route recomputation that exclude all nodes on the route just used: this reflects link contention due to interference. We repeat this process until we find no reachable destinations. We simulate a set of T values and use results for the T value that maximizes goodput over the entire simulation.

DTN forwarding. Rather than using a fixed τ as in Section 4, nodes now store a data packet until it is delivered. At each timestep, DTN forwarding randomly selects two nodes, i and j, that are in contact: if i is holding a packet destined for j, that packet is transmitted and removed from the queues of all nodes in addition to being removed from node i's queue. If i does not have a packet destined for j, i transmits a copy of a randomly selected data packet that j does not yet have. This process is repeated for all node pairs in contact that have not yet participated in a packet exchange at the timestep. Again, we intentionally abstract the spectrum of DTN forwarding strategies using the simple strategy of holding the data.

Hybrid forwarding. This strategy is a purposefully simple hybrid of our routing and DTN forwarding strategies: at each timestep, we first use routing to deliver packets to all reachable destinations using the routes available from the last route recomputation. Once all reachable destinations have been exhausted, we use DTN forwarding to transmit data packets over any remaining links that exist that are not incident to nodes previously used in routing. For comparison purposes, we plot the hybrid forwarding results for the T value that was optimal for routing; we leave optimizing the T value for hybrid forwarding for future work.

5.4.2 *Results*

Infocom06. Fig. 6 shows the empirical measures and goodput for the Infocom06 trace as a function of the time of day, averaged over 10 runs. Below the goodput plot, each interval is annotated with the regime identified by the framework and the number of packets delivered by routing (R), DTN forwarding (D), and hybrid forwarding (H) within the interval. These annotations are used to show cumulative goodput since the plots themselves only show time-averaged instantaneous goodput. The figure shows distinct time-of-day behaviour for h_j; however, we note that h_j is always below 0.5 and ϕ_j is always low. The figure also illustrates a number of features of the framework. First, during periods when ϕ is relatively large, goodput is also greatest. Second, while no strategy can achieve high goodput when both h and ϕ are small, the strategy that consistently achieves the highest goodput throughout the trace is the hybrid strategy. This can be seen from the annotations at times $t = 1$, $t = 26001$, and $t = 75001$, which show the goodput achieved during the three intervals. In each of the intervals, the hybrid strategy has the highest goodput. This is as predicted by the framework, given that each interval lies entirely within the *(low h, low ϕ)* region. Our analysis of the Infocom05 trace shows that hybrid forwarding again has highest goodput and is not shown due to space constraints.

DieselNet. Fig. 7 shows the empirical measures and goodput for the DieselNet trace as a function of the time of day, averaged over 30 runs. The figure shows that during the daytime interval (until $t = 56701$), the network is in the data-state regime, which suggests that DTN forwarding is appropriate. Indeed, as seen in the annotation, DTN forwarding has the highest goodput during this interval. The nighttime interval (after $t = 56701$, when buses are in the garage) is in the control-state regime which suggests that routing is appropriate. The annotations show that routing has the highest goodput during this interval. Unlike the other traces, the DieselNet trace shows properties that cor-

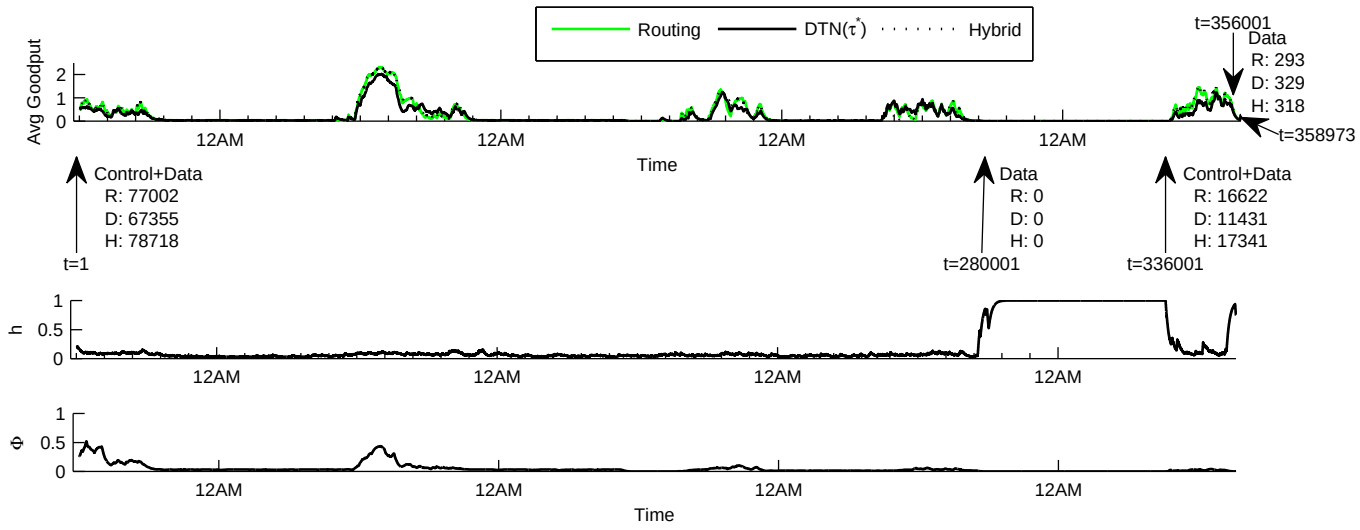


Figure 8: Intel trace results.

respond only to data-state or control-state; the DieselNet traces are never in the hybrid region.

Taken together, the Infocom06 and DieselNet traces show behavior that spans all three regimes of interest (control-state, data-state, and control+data state) and show that in each case, the framework gives the correct guidance for the appropriate strategy to use. Interestingly, the DieselNet trace is the only trace we analyzed that primarily falls in the data-state region (corresponding to DTN forwarding), despite the fact that all the traces we analyzed have been used in studies evaluating DTN forwarding.

Intel. Fig. 8 shows the empirical measures and goodput for the Intel trace as a function of the time of day, averaged over 10 runs. Like the other traces, for the intervals falling in the *(low h, low ϕ)* region, hybrid forwarding has the highest goodput. At the end of the Intel trace, after $t = 35601$, there is a brief interval falling in the *(high h, low ϕ)* region, for which DTN forwarding has the highest goodput, again as predicted by the framework.

Cambridge. Fig. 8 shows the empirical measures and goodput for the Cambridge trace as a function of the time of day, averaged over 10 runs. Unlike the other traces, the Cambridge trace has an initial interval, until $t = 5001$, falling in the *(low h, high ϕ)* region at the start of the trace. As predicted by the framework, routing has highest goodput during this interval.

5.5 Discussion

The framework gives insight into what type of state (control or data) should be stored as a function of the given network characteristics (h and ϕ). Our analysis also shows the importance of the hybrid region in practice, and that as predicted by the framework, control+data state strategies (like hybrid forwarding) are indeed the most appropriate strategy for this region.

The framework also provides insight into why flooding (a no-state strategy) is not often seen in practice. The *(high h, high ϕ)* region where we expect no-state strategies to be appropriate occurs when routes are very likely to exist but are constantly breaking. Such a scenario can be envisioned by the motion of heated gas molecules trapped in a container. People or vehicles moving in the same way as molecules seems unlikely to arise in practice, and none of the traces we analyzed spent time in the *(high h, high ϕ)* region.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a framework for understanding when a communication strategy should maintain state, and what state should be maintained, in an ad hoc network. We made the following three contributions. First, we illustrated the framework by showing an instantiation in terms of specific measures (ϕ, h, N) that can be used to describe a network setting. Second, we validated the framework by showing it correctly and consistently organizes the decision space for different communication strategies (using the torus and random waypoint models). Finally, we demonstrated the analytic power of the framework by using it to (1) uncover surprising aspects of well-known traces, and (2) identify the need for and value of a new strategy for network communication, hybrid forwarding.

In future work, we will examine how to practically estimate the empirical measures, and will investigate how quickly the estimated measures converge. Gossip algorithms are one way to estimate (ϕ, h, N) in a distributed way, with information piggy-backed on data and control packets whenever possible. In this way, ϕ could be approximated using the average fraction of nodes in the network with which a node shares a connected component. Conversely, each node can locally estimate h using passive observation of its links; more accurate estimation can be achieved by having nodes additionally gossip their estimated h values (or the conditional probabilities needed to compute h and the fraction of timesteps at which no links are found). As potentially significant overhead is needed to compute measures, it may be useful is to locally compute the measures for different regions of a network. Since different nodes may have different views of the ad hoc network, different nodes may also estimate different values for the empirical measures. We will thus also investigate the consequences of different nodes in the network

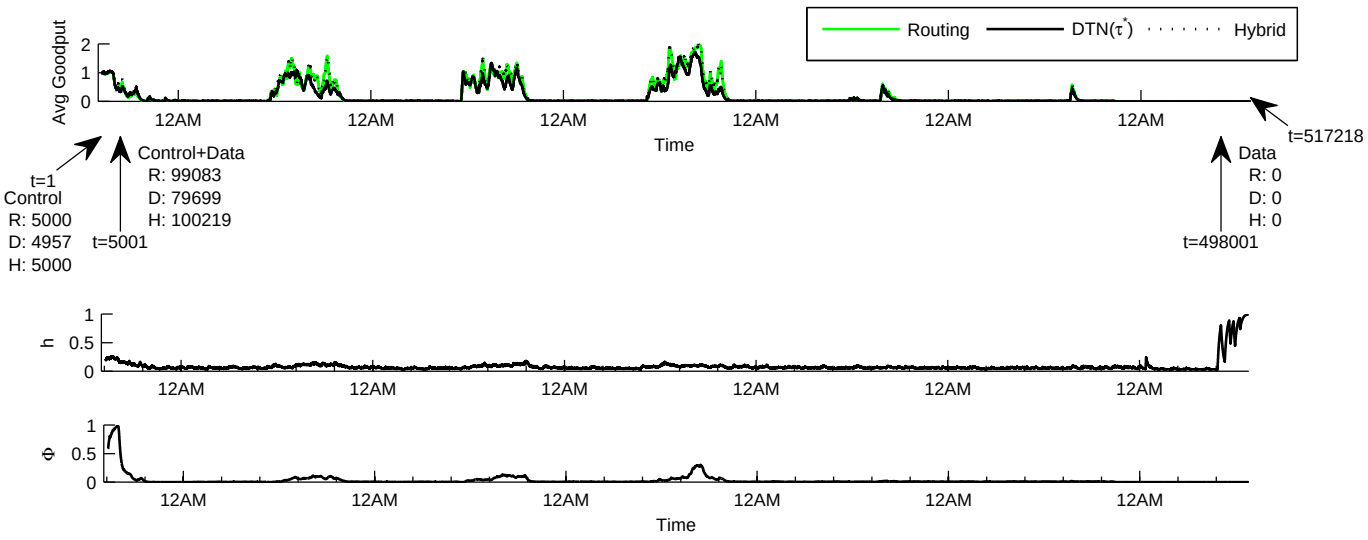


Figure 9: Cambridge trace results.

maintaining different amounts and types of state, as well as further explore strategies that simultaneously maintain both control state and data state. Finally, we have not considered energy issues in this work: further work is needed since control can be used to manage energy (such as in a duty-cycling network) as well as increase goodput.

Acknowledgments

This work was supported by NSF grants CIF-235, CIF-A-235, CNS-0905565, CNS-1018266, and CNS-1012910, by the NSF under grant #0937060 to the Computing Research Association for the CIFellows Project, and by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

7. REFERENCES

[1] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2:1, 2004.

[2] U. G. Acer, A. A. Abouzeid, and S. Kalyanaraman:. An evaluation of weak state mechanism design for indirection in dynamic networks. In *INFOCOM*, 2009.

[3] U. G. Acer, S. Kalyanaraman, and A. A. Abouzeid. Weak state routing for large scale dynamic networks. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, MobiCom '07, pages 290–301, New York, NY, USA, 2007. ACM.

[4] D. Antonellis, A. Mansy, K. Psounis, and M. Ammar. Towards distributed classification for mobile ad hoc networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet*, 2008.

[5] A. Balasubramanian, B. N. Levine, and A. Venkataramani. Enabling interactive applications for hybrid networks. In *ACM Mobicom*, 2008.

[6] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:5:483–502, 2002.

[7] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant networking architecture. *Request for Comments: 4838*, 2007.

[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6:606–620, June 2007.

[9] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot. The diameter of opportunistic mobile networks. In *CoNEXT '07*, 2007.

[10] R. Gallager. Basic limits on protocol information in data communication networks. *IEEE Transactions on Information Theory*, 1976.

[11] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical Report IRB-TR-02-003, Intel Research Berkeley, 2002.

[12] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM*, 2004.

[13] P. Ji, Z. Ge, J. Kurose, and D. Towsley. A comparison of hard-state and soft-state signaling protocols. *IEEE/ACM Transactions on Networking*, 15:2:281–294, 2007.

[14] D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD data set dartmouth/campus (v. 2004-12-18). Downloaded from http://www.crawdad.org/dartmouth/campus, Dec. 2004.

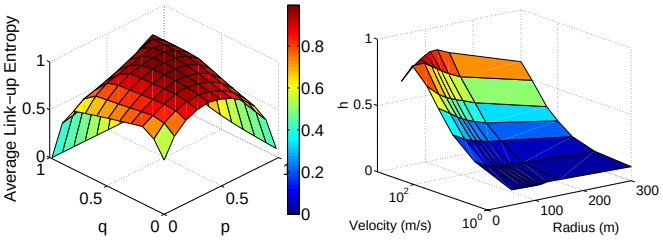
[15] J. Lakkakorpi, M. Pitkanen, and J. Ott. Adaptive routing in mobile opportunistic networks. In *MSWiM*, 2010.
[16] J. Norris. *Markov Chains*. Cambridge University Press, 1998.
[17] J. Ott, D. Kutscher, and C. Dwertmann. Integrating DTN and MANET routing. In *CHANTS*, 2006.
[18] S. PalChaudhuri, J.-Y. L. Boudec, and M. Vojnovic. Perfect simulations of random trip models. In *Proceedings of the 38th Annual Simulation Symposium*, 2005.
[19] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
[20] R. Ramanathan, P. Basu, and R. Krishnan. Towards a formalism for routing in challenged networks. In *CHANTS*, 2007.
[21] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29), May 2009.
[22] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-200006, Duke University, 2000.
[23] L. Viennot, P. Jacquet, and T. Clausen. Analyzing control traffic overhead versus mobility and data traffic activity in mobile ad-hoc network protocols. *Wireless Networks*, 10:4, 2004.
[24] D. Wang and A. Abouzeid. Link state routing overhead in mobile ad hoc networks: a rate-distortion formulation. In *Proceedings of IEEE INFOCOM*, 2008.
[25] J. Whitbeck and V. Conan. HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks. *Computer Communications*, 2010.
[26] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Computer Networks*, 51:10:2867–2891, 2007.
[27] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys & Tutorials*, 8:1, 2006.

APPENDIX

A. AVERAGE LINK-UP ENTROPY PLOTS

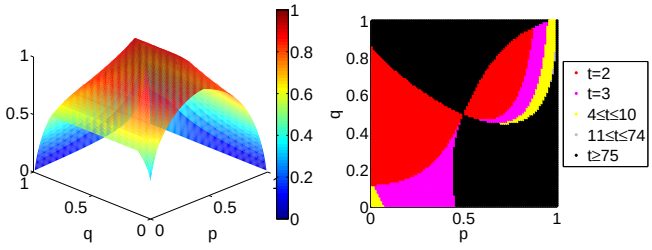
Fig. 10(a) plots average link-up entropy, h, for the torus as a function of p and q. Fig. 10(a) shows that h is maximized in the region where $p = q$. Conversely, Fig. 10(a) shows that h is minimized when q is very large and p is very small, or q is very small and p is very large: this is because links are very likely to stay down (or up), respectively, and so links are very predictable. Similarly, when p and q are both small, h is again low: this is because while links may be frequently breaking and reappearing, links are predictably breaking. Thus, the frequency of network change is important, but so is the predictability of that change. Note that h is slightly lower in the $q < p$ region versus the $p > q$ region, because h focuses only on the links that existed when routes were last updated. ***Finally, observe that the region where h is maximized in Fig. 10(a) corresponds to the region where flooding is most preferred in Fig. 4(a).***

Since we use a two-state Markov model for link behavior



(a) Torus (b) RWP

Figure 10: Average link-up entropy.



(a) h (b) T_h^*

Figure 11: Average link-up entropy, h, and optimal update interval, $t = T_h^*$, for the torus, computed analytically.

in the torus, we can also analytically compute h. We first compute the t-step transition probabilities for a link (see, e.g., Ch. 1 of [16]) as follows.

$$p_{uu}^{(t)} = \begin{cases} \frac{1-q}{2-p-q} + \frac{1-p}{2-p-q}(p+q-1)^t & \text{if } p+q<2 \\ 1 & \text{if } p+q=2 \end{cases}$$

$$p_{dd}^{(t)} = \begin{cases} \frac{1-p}{2-p-q} + \frac{1-q}{2-p-q}(p+q-1)^t & \text{if } p+q<2 \\ 1 & \text{if } p+q=2 \end{cases}$$

The notation $p_{ud}^{(t)}$ indicates the probability that a link is in state $d = down$ after t transitions and having started in state $u = up$. We assume $p_{uu}^{(0)} = p_{dd}^{(0)} = 1$ and $p_{ud}^{(0)} = p_{du}^{(0)} = 0$. From the $(p+q-1)^t$ term, when $p+q-1 < 0$ links tend to oscillate. The network is thus stable when $p+q-1 = 1$ and bistable when $p+q-1 = -1$.

The link-up entropy of Equation (1) computed analytically for the torus is then as follows.

$$H^u(g_{t+i}|g_t) = -\sum_{y=u,d} p_{uy}^{(t+i)} \log p_{uy}^{(t+i)} \qquad (5)$$

Average link-up entropy, h, can then be computed analytically and is shown in Fig. 11(a). Fig. 11(b) plots the T_h^* values used to compute the h values in Fig. 11(a).

Finally, Fig. 10(b) plots h for the random waypoint model as a function of r and v: h is maximized where r is around the percolation threshold of 100 m and v is high. Although many fewer links may actually exist than could potentially exist (unlike in the case of the torus), h focuses only on the predictability of those links found when routes were last updated. ***As with the torus, again observe for the random waypoint model that the region where h is maximized in Fig. 10(b) corresponds to the region where flooding is most preferred in Fig. 4(c).***