SAMPLE: An On-Demand Probabilistic Routing Protocol for Ad-hoc Networks

Eoin Curran and Jim Dowling
Distributed Systems Group, Department of Computer Science,
Trinity College Dublin.

20th January 2004

Abstract

Existing on-demand ad hoc routing protocols assume an idealised wireless network in which all links in the network are either on or off and where all functioning links are equally good. Such a model interprets the fraction of packets that are dropped due to contention or interference as broken links, which can in turn lead to increased routing traffic and radio contention. As an alternative to the traditional hop-count metric, this paper presents a new metric for capturing the cost of a route based on a statistical model of network links. To investigate the impact of using this cost metric, we present a probabilistic routing protocol, SAMPLE, inspired by reinforcement learning techniques. Different scenario-based performance evaluations of the protocol in NS-2 are presented. In comparisons with AODV and DSR, SAMPLE exhibits improved performance in both lossy and congested wireless networks.

1 Introduction

Standard ad hoc routing protocols such as AODV[1] and DSR[2, 3] are based on a discrete model for links in the network. This model is essentially based on an assumption of perfect radio links and ignores the effects of interference and network contention. Discrete models of network links are generally based on the last measurement (i.e. attempt to send a packet, receipt of a packet or link monitoring activity) of a link's status. The first packet dropped on a link is considered as evidence that the link has broken. These models produce protocol implementations where route updates occur whenever the link-level model indicates a link failure, despite the fact that link failure may not be an actual failure but due to interference or congestion in the wireless network.

Real-world wireless networks have properties that are not present in the idealised network. We believe that it is necessary to make statistical observations of the network in order to measure - and hence react to - these properties:

- *Heterogeneous distribution of mobility.* Not all nodes in a network may be mobile at any given time. Hence, some links in the network may be longer-lived than others. The discrete model considers a newly created/discovered link in the network equal to a long-established link.
- *Network congestion and interference.* Under congested network conditions, or when suffering from interference, the radio network interface may not be 100% reliable [4], even for sending packets to nodes that are within transmission range. A discrete model of network links can treat dropped packets due to network congestion as a broken link. This can have the negative side-effect of actually increasing network congestion by generating additional routing traffic. In such congested network conditions, the decision to treat a network link as broken can only be

made on the basis of a number of measurements. A statistical model of network links should be able to operate more efficiently under congested network conditions by adjusting to dropped and delivered packets in a more gradual manner.

- *Shortest Path is not enough.* It has been suggested [5] that the shortest-hop-path criteria is not sufficient to determine optimal routes in an 802.11 based ad hoc network. Multiple shortest-hop routes may be available, with widely varying levels of reliability. Statistical information about the quality of these routes could be used to differentiate between multiple shortest-hop routes.

- *Fluctuating link quality.* Communication at borderline communication ranges in 802.11 is unreliable due to the fluctuating quality of links. Discrete models sometimes update routing entries in a jitter-like manner at communication gray zones, and often replace stable (but sometimes longer) routes with newer, unreliable routes links resulting in poor protocol performance [6].

This paper introduce SAMPLE, a probabilistic on-demand ad hoc routing protocol based on a simple statistical model and inspired by techniques from reinforcement learning. We compare the performance of this protocol with that of AODV and DSR by simulation using NS-2. We find that the use of a statistical model can provide improvements in performance, especially in congested network scenarios. In one network scenario with a subset of stable nodes we show that the SAMPLE routing protocol can deliver throughput approaching the theoretical limit for 802.11 networks.

Section 2 presents background material on probabilistic routing protocols for both wireless and fixed networks. Section 3 discusses the main features and objectives of the SAMPLE routing protocol. Section 4 introduces the statistical models used to represent network link quality in 802.11 networks, the cost metric for routes as well as the probabilistic routing model. Section 5 describes the implementation details of the SAMPLE routing protocol. Section 6 evaluates SAMPLE relative to AODV and DSR in two different network scenarios.

2 Related Work and Background

To date, there has been limited research in the area of probabilistic routing in mobile ad hoc networks. However, statistical models of routing tables and probabilistic flooding have been proposed as more efficient mechanisms for managing routing information and limiting MAC layer flooding traffic respectively.

PERA (Probabilistic Emergent Routing Algorithm [7]) uses swarm intelligence inspired algorithms to build and manage statistical routing information in an ad hoc network. In contrast to the routing protocol presented in this paper, PERA is a proactive routing algorithm and statistical models of routes are maintained and updated by control and signalling packets. These control packets are called ants, and they opportunistically explore multiple paths to a destination. However, due to the overhead of proactive routing PERA actually performs slightly worse than AODV. This is mainly due to the high cost of transmitting extra control packets in a radio network relative to the incremental cost of increasing the packet size. MANSI (Multicast for Ad Hoc Network with Swarm Intelligence [8]) also uses separate control packets to explore multiple routes in a network.

Gossip-Based Ad Hoc Routing [9] applies probabilistic broadcast to route finding within AODV. Probabilistic broadcast is different to probabilistic routing since it operates only for route discovery. It attempts to limit the amount of MAC layer traffic in the network during flooding by exploiting the statistical likelihood of packets arriving at a destination over more than one route in a dense enough network. Gossip has been found to be able to reduce control traffic by up to 35%, although the routes found by gossiping may be 10-15% longer than those found by flooding. A similar approach is outlined in [10].

In fixed network environments such as WANs, both AntNet [11] and Ant-based Control [12] have shown that statistical routing protocols can perform as well and often better than traditional routing protocols. AntNet is an adaptive, mobile-agents-based algorithm inspired by work on the ant colony metaphor. Ant-based Control uses a very similar approach to

AntNet, but has been designed specifically for telephone networks. Both protocols periodically launch network exploration agents, called *forward ants* to every destination to find lower cost routes through the network. For AntNet, at each node the ants will choose their next hop probabilistically using that node's routing table. As the ants visit a node, they record their arrival time and the node identity. An ant reaching its destination is converted to a *backward ant* that calculates a round-trip time to the destination over the route chosen by the forward ant. This round-trip time is compared to the average round-trip time to that destination. If the new round-trip time is smaller, the probability of choosing that route is increased. If the new time is larger, that route's probability is decreased.

3 Objectives and Summary of the Approach

This project had the initial aim of building a reactive routing protocol for ad hoc networks based on a statistical model of network links. Existing statistical routing algorithms such as AntNet and PERA are, however, proactive routing protocols. They use separate control and maintenance packets (ants) to discover new routes in the network. Each node keeps a routing table, which for each destination gives the probability of choosing each neighbouring node as the next hop. However, only the exploratory ant packets are routed probabilistically to the next hop, as normal network traffic (i.e. non-ant packets) is routed to the next hop with the highest probability in the routing table for a given destination. In order to construct a reactive routing protocol that is suitable for mobile ad hoc networks, we must be able to exploit new routes and changing routes as nodes move in the network. In effect, we required on-demand exploration of routes but without the overhead of techniques such as flooding.

SAMPLE is a reactive probabilistic routing protocol based on a statistical model of network links. Routing information is distributed in the network in an on-demand manner by attaching it to data packets, and not as separate control traffic. The routing of packets is done probabilistically at each node, so packets may travel on new routes or suboptimal routes. The potential increased cost of this probabilistic routing is a trade-off against the benefit of discovering lower cost routes and distributing routing information in the network.

SAMPLE is strongly inspired by the field of reinforcement learning. Reinforcement-learning [13, 14] describes a class of problems within machine learning in which agents attempt to optimize its interaction with a dynamic environment through trial and error. In SAMPLE, the routing agent interacts with the environment by deciding which neighbour to forward a packet to, or to take the option of broadcasting the packet. In response to these decisions, the routing agents obtain information about the quality of network links with their neighbours.

The statistical model that SAMPLE uses attempts to estimate the probability of a transmission over a given link being successful. This model is used to calculate the cost of using a given link in the network. SAMPLE uses a model for the *cost* of links and routes in the network that is designed to approximate the number of radio transmissions[1] needed to deliver a packet along that link or route. The cost of a route is represented as the sum of the costs of each of its links.

Each data packet that a routing agent transmits in the network allows it to advertise the cost of its routes for both the source and destination of the data packet. The routing information advertised by routing agents is sampled by other routing agents when receiving or promiscuously receiving packets. These routing agents then calculate the cost of their route to the given source or destination via the node which it received the packet from. The cost advertised by routing agents is the optimal cost from its neighbouring nodes. When routing, the estimated route cost via each available neighbouring node is used to decide which node to forward a packet to. The next hop is chosen probabilistically, with the lowest cost route being chosen with the greatest probability.

[1] transmissions made by the 802.11 protocol, i.e. including the number of retransmissions that are made until the packet is acknowledged successfully

SAMPLE can send packets along routes which are currently considered to be suboptimal. This is done in the hope of discovering better routes by improving the routing information available to the routing agents. There is a trade-off between the quality of the information available for routing and the amount of 'sub-optimal' actions required to gather that information which is closely related to the idea of *Exploration vs. Exploitation* within machine learning. We use a greedy heuristic and action selection mechanism inspired by reinforcement learning techniques in order to try and limit the amount of exploration we perform and to restrict that exploration to useful areas of the network.

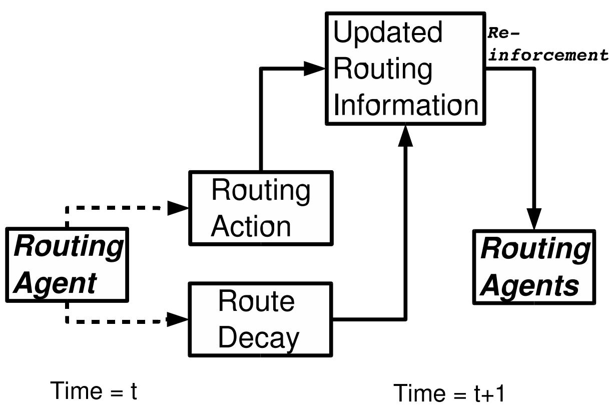


Figure 1: Routing Information Updating using Reinforcement Learning

We have introduced the idea of *route decay* in SAMPLE as a mechanism to remove stale routing information from the tables of the routing agents. This is similar to pheromone trail evaporation used in [15]. Each agent stores the last-advertised route cost to a given destination from each of its neighbours. However, the agent considers this value to decay (i.e. grow steadily larger) from the time it is advertised. In this way, routes that are not advertised are gradually eliminated from consideration for routing decisions. This ties in closely with the idea of *reinforcement* in SAMPLE: the use of a route increases the probability of that route being used, and the more successful a route is, the more it is advertised throughout the network. Figure 1 illustrates how the routing decision taken by a routing agent and route decay combine to update the routing information used by other routing agents in the network.

SAMPLE has been designed with a certain type of ad-hoc network in mind, that used to provide internet access to a metropolitan area. In this network scenario, the majority of traffic either originates or terminates at a small number of nodes in the network (i.e. those providing internet access). We also assume that traffic flows will have a certain level of bi-directionality (e.g. acknowledgement packets in TCP). In this scenario, attaching routing information to every data packet has some interesting properties:

- The amount of effort used for routing to a given destination is relative to the popularity of that destination. Hence the quality of routing information is higher for more popular traffic destinations, such as nodes providing internet gateway services.
- Good routes in one direction will often also be good routes in the reverse direction (since 802.11 requires bidirectional communication). For this reason, routes to either end of a traffic flow are closely related. By attaching routing information to data packets we exploit routes in one direction of flow in order to efficiently transfer routing information about the reverse flow to where it is needed.

4 Statistical Model of Network Links in SAMPLE

This section introduces the statistical model of network link quality and the probabilistic routing model used in the design of the SAMPLE protocol.

4.1 Statistical Model for Network Links

A statistical model for network links should provide a statistical measure of the link's performance over a period of time. The quantity that we attempt to measure is the probability of successfully transmitting a unicast packet to a given neighbour node. In order to do this, we sample the rate of a number of different events within a small time window τ into the past[2] . The events that we monitor are:

- Attempted Unicast Transmissions, r_A
- Successful Unicast Transmissions, r_S
- Received Unicast Transmissions, r_U
- Received Broadcast Transmissions, r_B
- Promiscuously Received (overheard) Unicast Transmissions, r_P

The rate of these events is used to estimate the probability of an attempted unicast transmission being successful, i.e. we attempt to estimate the future value of $\frac{r_S}{r_A}$. Since a successfully received packet is indicative of a functioning network link, we allow receive events to influence our estimation to a configurable extent:

$$E\left(\frac{r_S}{r_A}\right) = \frac{r_S + \alpha\beta\left(r_U + r_B + r_P\right)}{r_A + \beta\left(r_U + r_B + r_P\right)} \qquad (1)$$

The parameter α represents our *belief* about the probability of successfully transmitting a packet in the case that we have received but not attempted to transmit. The parameter β controls how much received packets are weighted compared to transmitted packets. For the experiments carried out in this paper we used values of 0.5 and 0.2 for α and β respectively.

Note that this is quite a simple estimate of delivery ratio. More complicated measures could be devised and other variables considered. Our goal here is not to design an accurate predictor for delivery ratio, but rather to show how such a predictor be incorporated into a routing protocol. For comparison, the $E\left(\frac{r_S}{r_A}\right)$ function used by AODV is 1 if the latest transmission was successful and there has been communication within a timeout period, and 0 otherwise.

[2]For the experiments in this paper τ=10s

4.2 Model of route cost

We use our measure of link quality to define a measure of cost for routes in the network. The cost metric is then used to compare routes in the network, where routes with a smaller cost will probabilistically be more likely to be chosen. The cost for a route, D, is defined as the sum of each cost d over $r-1$ network links in the route:

$$D\left(\{n_{i_1}, n_{i_2}, \ldots, n_{i_r}\}\right) = \sum_{s=1}^{r-1} d\left(n_{i_s}, n_{i_{s+1}}\right) \qquad (2)$$

The d function is based on the estimated delivery ratio introduced in Section 4.1. The d function should reflect the number of radio transmissions (including retransmissions) needed by the 802.11 MAC protocol to deliver one packet over a network link. Labelling the estimated delivery ratio as λ, then, the following are desirable properties of our d function:

- If $\lambda = 1$, then $d = 1$. i.e. with a perfect link, exactly one transmission is required to deliver each packet
- If $\lambda = 0$, then $d = \infty$. i.e. if a link is completely broken, then no number of attempted transmissions will deliver a packet over it

We define further restrictions on the d function by reference to the 802.11 MAC protocol. Note that the event counts that we use to calculate λ in Equation 1 are the event counts reported by the actual MAC protocol. In 802.11 a radio transmission will be retried up to 7 times[3] before it is considered failed and causes a unicast failure notification to be sent to the routing protocol [16, 17].

Let us consider an example where we attempt n 802.11 unicasts over a given link and only one of them succeeds. In this case, $r_A = n$ and $r_S = 1$, and $\lambda = E\left(\frac{r_S}{r_A}\right) = \frac{1}{n}$. We know that this required at least $1 + 7(n-1) = 1 + 7\frac{(1-\lambda)}{\lambda}$ radio transmissions. So to deliver the one successful packet required $1 + 7\frac{(1-\lambda)}{\lambda}$

[3]For simplicity, we ignore the case when only 4 retransmissions are used.

radio transmissions. We note that this expression meets the criteria that we set our d function at the boundary values of λ. We therefore use this as our d function:

$$d(\lambda) = 1 + 7\frac{(1-\lambda)}{\lambda} \quad (3)$$

This cost function d is our expected minimum number of radio transmissions to successfully deliver one packet over the given link.

Using this cost function, the goal of our routing protocol becomes to compute the optimal cost from the source of packets to a given destination. Each node computes its cost to the destination from that of its neighbours and from the link costs to the neighbours. We label the optimal cost from node N to a given destination, O, as $D_P(N)$. Thus, we get:

$$D_P(N) = \min_M \left(D_P(M) + d(N, M)\right), \qquad D_P(P) = 0 \quad (4)$$

A similar cost function is found in AODV, where the corresponding D function is the hop-count.

4.3 Probabilistic Routing using Estimated Route Cost

The previous section introduced a model for the cost of routes in the network based on statistical measurement of the links in those routes. Theoretically, if the true values of the delivery ratios, d were available, we could calculate the values of the D function for every source and destination node in the network. In the case that the true D values were known and available at all nodes, the decision of where to forward a given packet can be made quite easily. The routing protocol can choose to forward to the next-hop with the lowest D value, or in the case where a number of (nearly) optimal choices are available we can choose the next-hop on a per-packet basis according to some policy (e.g. sticky, round-robin, etc.).

However, in reality the d values that are available in the network are estimates. Hence the D values calculated from the d values are estimates. Also, since the effect of changing d values may take a number of messages passed in the network to propagate to all the D values based on that d value, the D values in the network may also be out of date. This inaccuracy in the D values available to the routing protocol has two implications:

- The accuracy of the d values improves the more frequently the links are sampled - we should attempt to measure link quality in the network regularly. In order to compare available routes, we require accurate information about the route cost for each of the options. For these reasons, it is desirable to sample the quality of the links along possible routes continuously or at least periodically.
- The ideal routing policy may not be to forward to the next-hop with the lowest D value - since these values are estimates, the choice of next-hop should be made statistically, i.e. we should examine and use routes which are currently estimated to be sub-optimal since they may in fact be optimal, or may become optimal as the network changes.

It is important for the routing protocol to attempt actions which are currently considered sub-optimal in order to gather accurate information about the network. In artificial intelligence research, this trade-off between using the optimal choices that have been already calculated and choosing sub-optimally in order to improve the knowledge of the system is described as *Exploration vs. Exploitation*. One standard technique for exploration is to use *Boltzmann-distributed exploration* [18]. In SAMPLE, for each of the possible next hops M the decision of which link to route a packet from node N is chosen probabilistically between the set of possible next hops:

$$P(M) = \frac{e^{-(D(M)+d(N,M))/T}}{\sum_{M'} e^{-(D(M')+d(N,M'))/T}} \quad (5)$$

The parameter T is called the *temperature*, and determines the likelihood of choosing sub-optimal actions. The higher the temperature, the more likely a sub-optimal action is likely to be chosen. Varying the temperature controls the amount of exploration that will be taken. For the experiments conducted in Section 6, a value of $T = 3$ was used. The effects of varying T for different network scenarios are examined in [...].

We will also use a simple *greedy heuristic* in order to restrict exploration to useful areas of the network: we will only allow node N to forward to those neighbouring nodes with a D value that is less than that of N. The greedy heuristic also helps prevent packets from entering routing loops after which they would be dropped.

The sampling of available routes and links in the network by SAMPLE is done in a continuous, on-demand manner by using each data packet transmitted to sample a possible link. We use each data packet to advertise information about routes to both its source and destination. In this way, we relate the amount of routing traffic for a given destination to the frequency of its use as a traffic source or destination.

5 The SAMPLE Routing Protocol

This section presents the implementation of SAMPLE based on the statistical models of routes and probabilistic routing presented in Section 4.

In the SAMPLE routing protocol we transfer routing information on-demand and opportunistically as much as possible, including the use of promiscuous receiving by nodes in the network. Separate routing packets are used as little as possible.

Each routing agent stores route cost information for its neighbouring nodes, as well as the last advertised D value from those neighbours for each destination or source node in use. Each packet sent by the routing protocol will contain the information shown in Table 1. Any node receiving or promiscuously receiving a packet will store the D values for its neighbouring node, and also update the event counts identified in Section 4.1 for that neighbouring node. Using this information, the routing agent can calculate its D values to attach to any packet that it transmits.

5.1 Probabilistic Packet Forwarding

In normal operation, a routing agent receiving a unicast packet updates its routing tables and then chooses a next hop according to the greedy heuristic and Boltzmann-distributed selection identified in Section 4.3. The agent then unicasts the packet towards the next hop that it has selected. If this unicast fails, the routing agent updates its routing tables and repeats the procedure (decrementing the TTL of the packet).

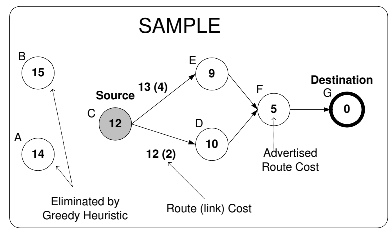


Figure 2: Routing decision

Figure 2 illustrates the choices available to the routing agent at node C and the probabilistic routing decision made. The figure shows the cost that each node has advertised for routing towards G.

The figure shows how node C has calculated its route cost. D has advertised a route cost of 9, and C estimates the link cost between C and D as 4. Hence C calculates its route cost via D as 13. Similarly, C calculates its route cost via E as 12. C advertises its lowest available route cost, 12.

The two nodes A and B at the left of figure 2 are not considered as next hops by C since their advertised route cost is higher than that of C. When making the decision about which node to forward a packet to, C considers only D and E as next hops. The decision as to whether E is chosen as the next hop is made according to the Boltzmann selection formula:

$$P(E) = \frac{e^{-12/T}}{e^{-12/T} + e^{-13/T}} = \frac{1}{1 + e^{-1/T}} \qquad (6)$$

The higher the value of T, the less likely that E will be chosen as the next hop. i.e. if the temperature parameter T is increased, the estimated sub-optimal next-hop D will be chosen more often. D is

Field Name	Field Type	Field Contents
ORIGIN	IP Address	The original source of the packet
DESTINATION	IP Address	The final destination of the packet
SEQUENCENUMBER	Integer	Generate at source node
SOURCEDIST	Floating-Point	D_{src}(N)
DESTINATIONDIST	Floating-Point	D_{dest}(N)
HADERROR	Boolean	if the packet's previous transmission failed
IPPACKET	Data Packet	IP Packet, or empty

Table 1: SAMPLE Packet Format (N is the node transmitting the packet)

sub-optimal as its total route cost is 13 compared to a value of 12 for E.

5.2 Routing table decay

In order to remove stale values from our routing tables, we let the D values in our routing tables 'decay' unless they are re-advertised. We do this by growing these values exponentially. $E(D) = D.\alpha^T$, where T is the time elapsed since the D value was advertised. In our experiments, we set this value to 1.1.

5.3 Packet flooding and duplicate suppression

There are situations where the broadcast of packets is both required and desirable. In the case that no routing information is available, such as before the network has been bootstrapped with traffic or the routing information has gone stale, the routing protocol must still be able to function correctly. Also, in continuous operation, we would like to be able to discover new network links and routes that become available. For these reasons, we allow the routing agent to broadcast a packet. In the case that no routes are available, the broadcast action will always be taken. However, we also allow this action to be chosen during normal operation, with a certain (albeit low) probability.

The probability of choosing the broadcast action is a configurable parameter of the SAMPLE routing protocol. It is effected by assigning a virtual cost to the action of broadcasting, and making the broadcast action one of the options considered when using the Boltzmann Equation. The broadcast action is assigned a cost by adding a fixed amount[4] to the estimated cost of the current node. In this way, the more options that are available to the routing agent, and the better the quality of those options, the less likely the broadcast action is to be chosen. When there are very few next-hops to choose between, or if the links to those next-hops are very poor, the broadcast action is more likely to be chosen.

Any routing agent receiving the broadcasted packet will forward it if its D value for the packet destination is lower than that of the previous node, and if it has not received the packet before. This serves to reduce but not eliminate the amount of duplicate packets in the network.

Each routing agent keeps a record of the last packet it forwarded from a given destination by its sequence number. If this packet is received again, it will be discarded. In the case that a routing agent retries transmitting a packet after a failed transmission, the packet will be marked as 'HADERROR'. This field allows a receiving routing agent to disable its duplicate suppression mechanisms for that packet.

5.4 Pro-active replies

In order for routing information to spread effectively in the network, there should be traffic in both directions along a flow. In many real-world networks, this will be the case (e.g. TCP acknowledgements). However, in the case of CBR traffic (used in our evaluation) we allow the routing protocol to generate rout-

[4]For the experiments described in Section 6, this value was 7.

ing packets to received packets. This is controlled by a configurable parameter, MAXRECEIVEDWITHOUTSEND, which specifies how many packets a node can receive without sending a packet in response. In the case of TCP traffic this mechanism will not be invoked except for the first packet received. This response packet is the only packet that SAMPLE sends which contains only routing information. For the experiments in 6, the routing protocol was configured to send 1 response packet for every 10 unacknowledged packets delivered.

6 Experimental Results and Discussion

We have implemented the SAMPLE routing protocol described in Section 4 in the NS-2 network simulator [19]. We compare the performance of the SAMPLE routing protocol to that of AODV and DSR in two different network scenarios.

Since the SAMPLE routing protocol combines routing information with data packets, the metric of *number of routing packets* is not a valid one for comparison with AODV and DSR. For this reason, we use the number of transmissions (unicast or broadcast) that each protocol makes per application packet sent during the simulation run as a metric to compare the protocols. This metric represents the cost to the network of routing each data packet. For AODV and DSR, this metric should indicate the amount of routing traffic generated in the simulation. For SAMPLE, this will represent both the pro-active response packets discussed in section 5.4 and the data packets broadcast by the exploration mechanism discussed in section 5.3.

6.1 Effect of Packet Loss in a Random Network

This scenario is approximately that used in [20]. A simulation arena of 1500m x 300m is used, with the transmission power of the radio interfaces set to 250m. Random way-point mobility model is used, with a maximum speed of 20 m/s and varying pause times. Constant bit rate traffic of 64 byte packets, 4 packets a second, with 10 flows between random pairs of nodes is used. We introduce packet loss to the simulation in order to measure how well the different protocols operate in networks with lossy links.

Firstly, we compare the SAMPLE routing protocol to AODV and DSR with no packet loss added to the simulation. Figure 3 shows the packet delivery ratio and transmissions-per-packet metrics as they vary with the level of mobility in the network. In this scenario, SAMPLE actually has a marginally worse packet delivery ratio than AODV and DSR, with a cost (in terms of network transmissions) similar to that of DSR.

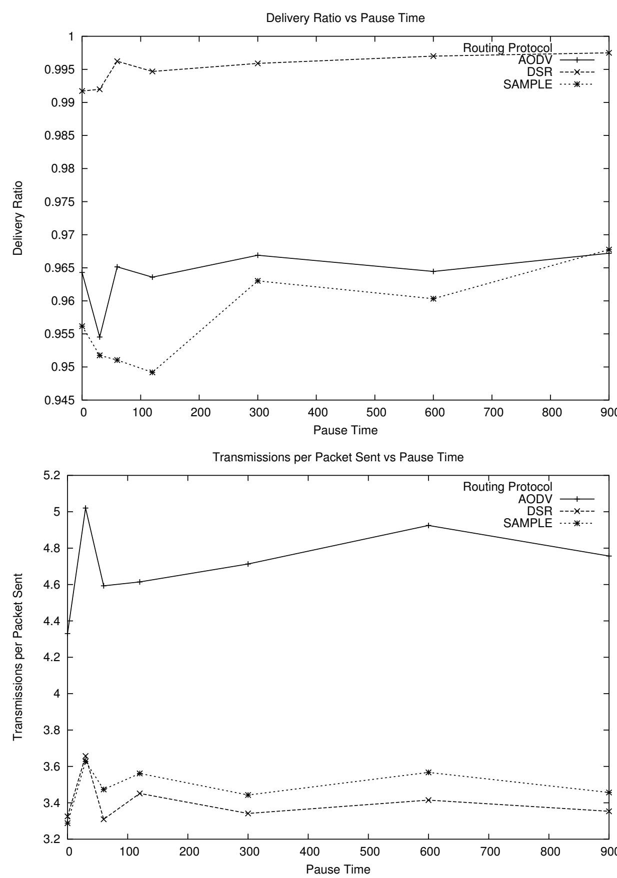


Figure 3: Comparison of SAMPLE with AODV and DSR with no packet loss

Radio interference is simulated by introducing random packet loss into the simulation. An NS-2 ErrorModel is used to drop packets both at the transmitter and receiver (each with half the rate shown in the results). This is a simplistic simulation of interference in the wireless network as packet loss is not introduced as a function of signal strength. However, it is indicative of the effect of lossy network links on the routing protocols.

Figure 4 shows the performance of the routing protocols as the level of packet loss in the network increases. Data points shown are the average of at least 30 simulation runs with varying traffic and mobility scenarios.

The SAMPLE routing protocol manages to maintain good performance for packet loss levels at which AODV and DSR show significantly reduced performance. For packet loss rates of up to 20%, SAMPLE has packet delivery ratio above 85%, with only slight increase in the number of transmissions made per packet. At a 20% packet loss rate, however, AODV and DSR have packet delivery ratios of 60% and 10% respectively, and produce more than double the amount of radio transmissions for each packet sent compared to SAMPLE.

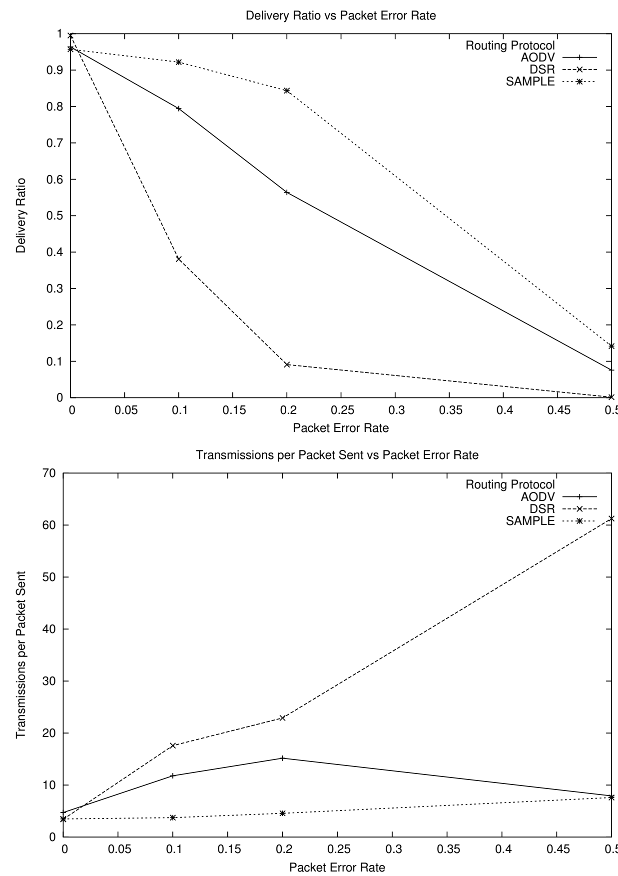


Figure 4: Effect of Packet Loss Rate

6.2 Metropolitan Area Ad Hoc Network

We have also evaluated the performance of SAMPLE against that of AODV and DSR in a network scenario based on a metropolitan ad-hoc network. In this scenario, there are a subset of nodes in the network that are not mobile. The network scenario is motivated by the recent appearance of ad-hoc networks designed to supply internet access to mobile nodes. We believe this scenario is representative of the type of metropolitan area ad hoc networks that are found in Mesh Networks[21].

In this scenario, we anticipate that certain nodes in the network will be immobile for extended periods of time, and that the traffic patterns in the network will be concentrated on those subset of nodes which have internet connectivity. In the experiments presented here we use 3 server nodes. Each client sends constant-bit-rate traffic to one of the servers at a rate of 4 packets per second. The number of client nodes in the network is varied in order to create congestion in the network.

Figure 5 shows the layout of the simulation arena. There are 33 fixed nodes in our simulations, and 50 mobile nodes. The 3 server nodes are the fixed nodes at the centre of the simulation arena. The fixed nodes in the simulation provide stable links in the network which the routing protocols could exploit.

Figure 7 shows the variation in performance of the three routing protocols as the number of clients in the network is increased. For these figures the packet size sent by clients was kept fixed at 64 bytes, sent 3 times a second. Figure 8 shows the same experiment,

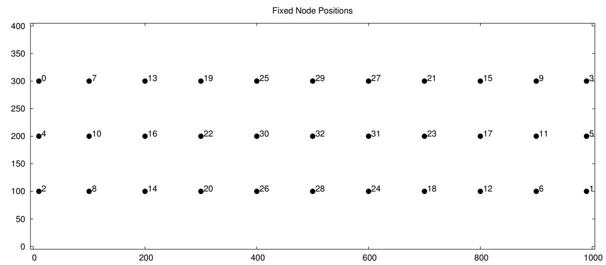


Figure 5: Simulation Arena, showing fixed node positions (transmission range is 100m)

this time with 512 byte packets. *Offered Throughput* is used in both figures to enable comparison of the results.

As the number of clients in the network is increased, the offered throughput to the routing protocols is increased. This in turn increases the level of network congestion and the amount of contention that the MAC protocol must deal with. This increased congestion increases the number of failed MAC unicasts in the network.

Figures 7 and 8 show that this increased network congestion affects the AODV and DSR protocols quite heavily, but that the SAMPLE protocol is able to continue to operate effectively with high levels of network congestion.

In [22] it was demonstrated that for multi-hop 802.11 networks, the achievable throughput is significantly less than the transmission rate of the radio interfaces. For simple chain topologies, they demonstrated that there is a theoretical maximum throughput of $\frac{1}{4}$ of the maximum single-hop throughput, but that 802.11 only achieves about $\frac{1}{7}$ in practice. The maximum throughput is further decreased when operating with network topologies other than the chain. For a 2Mbps transmission rate, the single-hop throughput for data when packet headers and interframe timing is taken into account is around 1.7Mbps. Thus, the maximum achievable data throughput in an 802.11 ad-hoc network is 1.7Mbps$\times\frac{1}{7}$, or approximately 0.25Mbps (which [22] achieved using 1500 byte packets). Figure 6 shows that SAMPLE manages to approach this limit in this scenario.

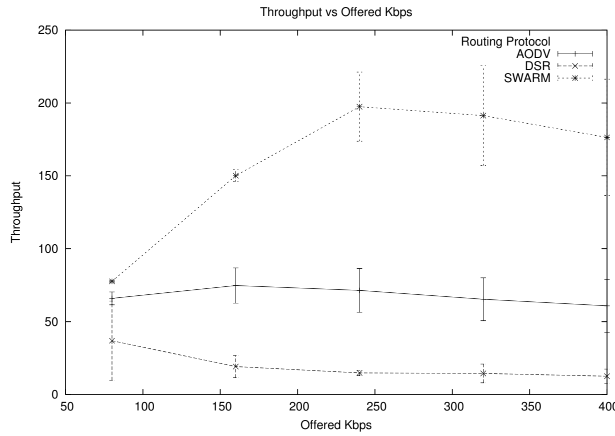
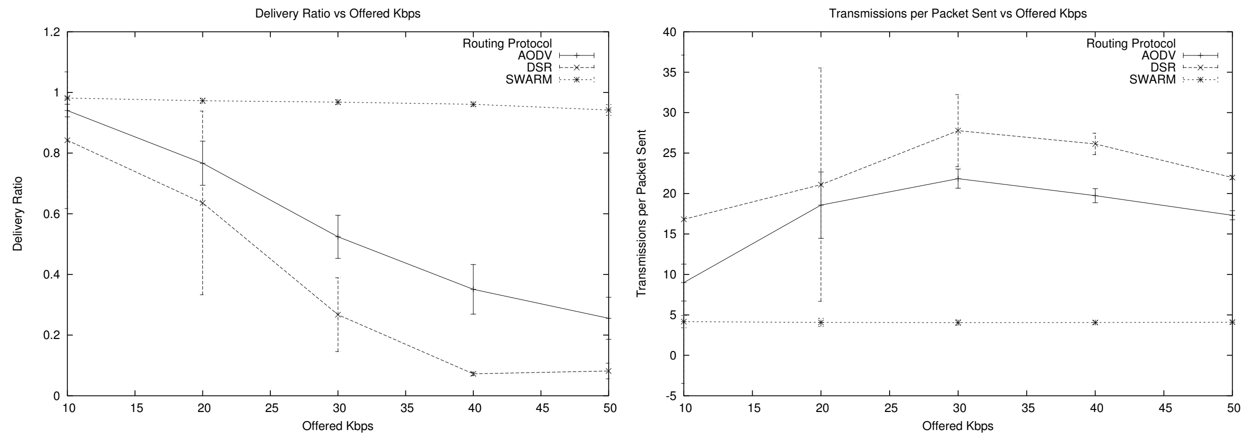


Figure 6: Delivered Throughput with varying load, 512 byte packets.

6.3 Discussion of Results

The results presented in the previous section show that the use of a statistical model and probabilistic routing allow for better performance in the face of adverse network conditions. We believe that the SAMPLE protocol performs better than AODV and DSR in these conditions for a number of reasons:

- In congested networks, or when suffering from interference, all links will have less than 100% reliability. In this situation, AODV and DSR generate increased routing traffic in response to dropped packets (Figure 7 for example shows a clear increase in traffic as congestion increases). In congested networks particularly this increased traffic can add to radio contention and further worsen the problem. By not treating every dropped packet as a broken link, SAMPLE avoids generating a large number of routing packets.
- In the scenario of the metropolitan ad-hoc network, there are a subset of the links in the network which are stable. In congested network scenarios, a statistical model can distinguish these



(a) Delivery Ratio

(b) Throughput

Figure 7: Performance with Varying Load. 64 byte packets

stable links in the network from those links which are changing. A discrete model, however, does not allow differentiation between multiple available links in this manner.

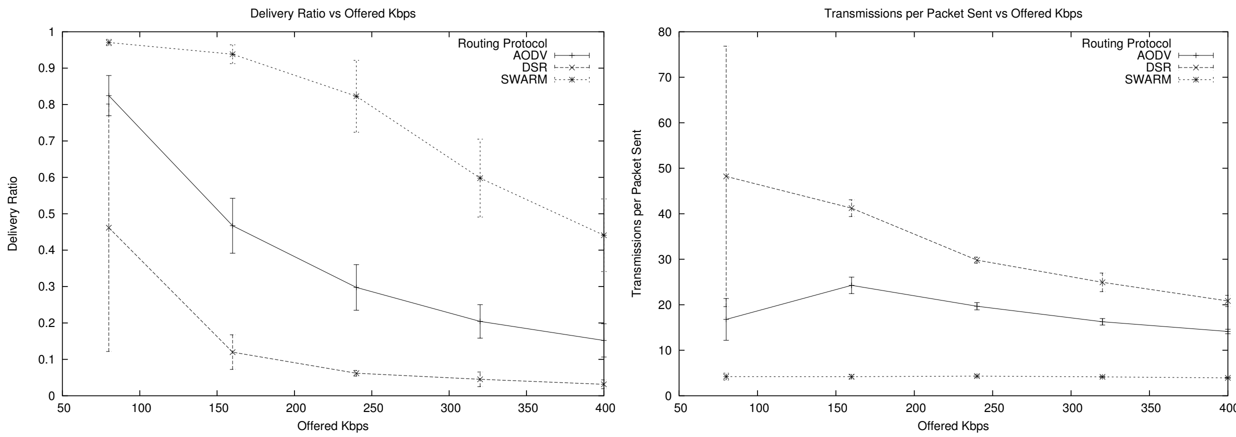
6.4 Future Work

However, the SAMPLE protocol as presented has a number of issues which require further research:

- All nodes in the network use promiscuous listening as part of normal operation. This results in increased battery usage and processing by nodes taking part in the network. Research is ongoing to reduce the amount of promiscuous listening required by the protocol.
- The use of multiple routes in a probabilistic manner means that packets will be delivered out-of-order more often using SAMPLE than using AODV or DSR. The interaction of this out-of-order delivery with higher level network protocols such as TCP is a possible drawback to this approach.
- Configuration parameters used by the protocol. The routing protocol uses a number of configurable parameters to control its behaviour and the level of exploration carried out. We are investigating how these parameters affect the performance of the protocol in different network scenarios, in order to determine reasonable default values. We are also investigating whether these configuration parameters could be automatically adjusted by the routing protocol in response to observed network traffic.
- The statistical model used. In SAMPLE, we use quite a simple model of links in the network. Research is ongoing to investigate the accuracy of this model, and to investigate the use of other statistical models.

7 Conclusions

We have presented a new metric to represent the cost of routes in an ad-hoc network that is based on a statistical model of network links. Based on this cost metric, we have presented SAMPLE, a new probabilistic on-demand ad hoc routing protocol inspired by techniques from reinforcement learning. We have



(a) Delivery Ratio

(b) Throughput

Figure 8: Performance with Varying Load. 512 byte packets

carried out a series of scenario-based experiments that demonstrate higher packet delivery ratios with less transmissions than AODV and DSR in lossy and congested ad-hoc networks. In the particular scenario of a metropolitan area ad-hoc network SAMPLE displays significantly improved performance and throughput over AODV and DSR. Also, we have identified a number of areas for future research on the use of statistical models, probabilistic routing and reinforcement learning in ad-hoc routing protocols.

References

[1] C. Perkins, "Ad Hoc On Demand Distance Vector (AODV) Routing," 1997. .

[2] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (Imielinski and Korth, eds.), vol. 353, Kluwer Academic Publishers, 1996. .

[3] D. Johnson, D. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multihop wireless ad hoc networks," 2001. .

[4] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Effects of Loss Rate on Ad Hoc Wireless Routing," Tech. Rep. MIT-LCS-TR-836, MIT Laboratory for Computer Science, March 2002. .

[5] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of Multihop Wireless Networks: Shortest Path is Not Enough," in *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I)*, (Princeton, New Jersey), ACM SIGCOMM, October 2002. .

[6] H. Lundgren, E. Nordstr"om, and C. Tschudin, "The gray zone problem in ieee 802.11b based ad hoc networks," in *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, pp. 104–105, July 2002.

[7] J. S. Baras and H. Mehta, "A Probabilistic Emergent Routing Algorithm (PERA) for Mobile Ad Hoc Networks," in *WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[8] C.-C. Shen and C. Jaikaeo, "Ad hoc Multicast Routing Algorithm with Swarm Intelligence,"

Tech. Rep. 2003-08, DEGAS Group, Dept. of CIS, University of Delaware, 2003.

[9] L. Li, J. Halpern, and Z. Haas, "Gossip-based Ad Hoc Routing," 2002. .

[10] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, 2002. .

[11] G. D. Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998. .

[12] R. Schoonderwoerd, O. E. Holland, J. L. Bruten, and L. J. M. Rothkrantz, "Ant-Based Load Balancing in Telecommunications Networks," *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1996. .

[13] L. P. Kaelbling, M. L. Littman, and A. P. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996. .

[14] S. ten Hagen and B. Kröse, "A short introduction to reinforcement learning," in *Proc. of the 7th Belgian-Dutch Conf. on Machine Learning* (W. Daelemans, P. Flach, and A. van den Bosch, eds.), (Tilburg), pp. 7–12, October 1997. .

[15] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in *New Ideas in Optimization* (D. Corne, M. Dorigo, and F. Glover, eds.), pp. 11–32, London: McGraw-Hill, 1999. .

[16] B. Crow, I. Widjaja, J. Kim, and P. T. Sakai, "IEEE 802.11: Wireless Local Area Networks," September 1997. .

[17] IEEE, "IEEE Std. 802.11: Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications," 1999. .

[18] S. B. Thrun, "The role of exploration in learning control with neural networks," in *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches* (D. A. White and D. A. Sofge, eds.), (Florence, Kentucky), Van Nostrand Reinhold, 1992. .

[19] Information Sciences Institute, "NS-2 network simulator." Software Package, 2003. .

[20] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *Mobile Computing and Networking*, pp. 85–97, 1998. .

[21] LocustWorld, "MeshAP," 2003. .

[22] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, (Rome, Italy), pp. 61–69, July 2001. .