

Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks

Shweta Jain and Samir R. Das
State University of New York at Stony Brook

Abstract— We develop an *anycast* mechanism at the link layer for wireless ad hoc networks. The goal is to exploit path diversity in the link layer by choosing the best next hop to forward packets when multiple next hop choices are available. Such choices can come from a multipath routing protocol, for example. This technique can reduce transmission retries and packet drop probabilities in the face of channel fading. We develop an anycast extension of the IEEE 802.11 MAC layer based on this idea. We implement the protocol in an experimental proof-of-concept testbed using the Berkeley motes platform and S-MAC protocol stack. We also implement it in the popular *ns-2* simulator and experiment with the AOMDV multipath routing protocol and Ricean fading channels. We show that anycast performs significantly better than 802.11 in terms of packet delivery, particularly when the path length or effect of fading is large.

I. INTRODUCTION

Wireless ad hoc network provides a very natural platform to study inter-layer protocol interactions as a very rich set of options exist in the design space of every layer, and a richer set of dependencies and interactions exist between such options. Understanding all such interactions is a critical step for better design of ad hoc networks; however, this space is relatively unexplored. In this paper, we address a specific inter-layer interaction between the physical layer and routing/ data forwarding layer and develop a MAC protocol that filters the harmful effect of such an interaction.

It is well-known that in wireless ad hoc networks, the “link” between two nodes is a “soft” entity [8]. From basic communication theory, its existence is governed by whether the signal to interference plus noise power ratio (SINR) at the receiver exceeds a given threshold (called the *receive threshold* γ). γ is determined by the the data rate, the modulation technique, receiver design, and the target bit error rate (BER) the receiver is able to withstand (i.e., able to correct using coding techniques). SINR is again influenced by transient factors such as transmit power, distance between the transmitter and receiver, multipath fading, and interference and noise powers reaching the receiver. Multipath fading [23] is caused by different components of the transmitted signal being reflected by the surrounding objects, and reaching the receiver via paths of different lengths, and combining either constructively or

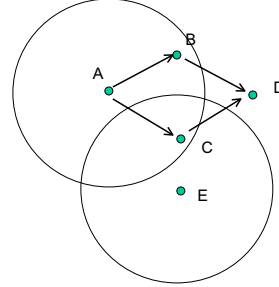


Fig. 1. Example scenario motivating anycast. Node *A* can forward packets to *D* either via *B* or *C*. But an ongoing transmission at *E* may interfere at *C*. If *A* chooses to forward via *C*, the transmission will defer until *E*’s transmission is complete. Such instantaneous channel conditions are unknown to the routing layer that discovers the routes.

destructively. Interference is caused by signals for other, unintended nearby transmitters. Both fading and interference could be time varying. Significant changes in fading and interference levels (beyond that can be masked by changes in sending data rate [5], [11])¹ may lead to transient “loss” of a link. This loss is often sufficient for many common routing and transport protocols to react – either to repair routes or to bring down the offered load. This leads to various operational inefficiencies, given that this loss is transient. Thus, there is a need to incorporate mechanisms that can “withstand” this loss of link at shorter time-scales.

While fundamentally new approaches are necessary to incorporate this soft abstraction for a link in the upper layer protocol design, it is often possible to take an “ad hoc” approach that we pursue in this paper. Here, a “hard” (stable, on or off) abstraction is still used for the link from the viewpoint of the upper layer – something it is designed to handle comfortably. However, now multiple link options are provided to the link layer, and the link layer is given the responsibility to make an instantaneous decision on which link to forward the packet on. We design a MAC-layer *anycasting*[6] scheme to perform this decision making and to forward the packet.

To implement anycasting, the link layer must take advantage of a multipath routing protocol [14], [26], [17],

¹Note that while physical layer techniques can mask effect of fading and interference, this work does not target physical layer techniques. Here, the interest is working on beyond physical layer capabilities, by exploring alternative paths.

[18]. Assume that multiple routing paths have been computed from the source and also from the intermediate nodes to the destination. Typically, the routing layer decides which of the several paths should be used for data forwarding and then the MAC layer is responsible to deliver the packet to the next hop along the chosen path. Now, predominant channel conditions (e.g., because of multipath fading and interference) may cause data transmission to defer or even fail causing the network layer to attempt using an alternate next hop. See a simple example in Figure 1. This leads to multiple transmission retries, wasting bandwidth and increasing delay. A better, alternative approach would be, for the link layer, to choose the next hop by observing the channel conditions on all possible next hop links. This “channel state-based” anycasting should improve performance, requiring very little operational coordination between the routing and MAC layers.

The goal of this paper is to develop an anycast MAC layer protocol to do this “channel state-based” next hop selection. While such a MAC layer protocol can be designed in many ways, a reasonable step is to do this design as an extension/variation of the commonly used IEEE Standard 802.11 [12] MAC layer. This makes performance easy to analyze and compare.

The rest of the paper is organized as follows. In Section II, we provide an overview of the 802.11 MAC protocol operation and describe the properties of a fading channel. In Section III, we describe our extension of 802.11 that implements anycasting to do the channel state based next hop link selection. We also describe the essentials of the the multipath routing layer. Section IV is devoted to performance evaluation. We analyze the performance of the anycast mechanism for a grid network via analytical modeling, and an experimental testbed using Berkeley motes, and proceed to detailed simulation-based evaluations using the ns-2 simulator. We describe the related work in Section V and conclude in Section VI.

II. BACKGROUND AND MOTIVATION

We start by briefly reviewing the IEEE 802.11 standard distributed coordination function (DCF) [12]. This is the MAC layer functionality that we will later extend in this paper.

A. IEEE 802.11 DCF

IEEE 802.11 uses Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA). Carrier sensing is performed by both physical and virtual mechanisms. The virtual carrier sensing is achieved by transmitting control packets to reserve the medium prior to transmission of data packets. The transmitter attempts to sense an idle medium for at least a DIFS (distributed interframe spacing) dura-

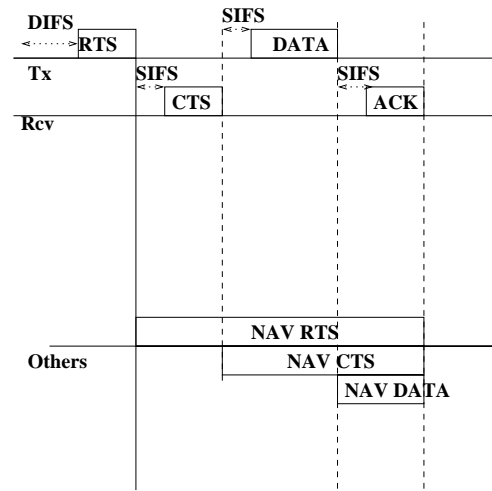


Fig. 2. Timeline showing RTS-CTS based data exchange in IEEE 802.11 DCF.

tion of time. If the medium is sensed busy, the transmitter waits until it becomes idle and then starts a countdown backoff timer set to expire after a number of slot times, chosen randomly between $[0, w]$, w being referred to as the *contention window*. Then it sends an RTS (request-to-send) which contains the address of the receiver and the duration for which the medium is to be reserved. This is the duration of the entire exchange including the control packets. When the intended receiver receives the RTS, and senses the medium to be free, it replies with a CTS (clear-to-send) after waiting for one SIFS (small interframe spacing) period. The CTS also contains the duration of the entire exchange from that point of time. The transmitter upon receiving the CTS transmits the DATA packet after an SIFS period. The receiver responds back with an ACK after an SIFS period following its complete receipt of the DATA packet.

Each node maintains a data structure called the *network allocation vector* or NAV to store the aggregate duration of time it knows that the medium would be busy. Any node other than the receiver, who hears the RTS (often called the *exposed nodes*), sets its NAV for the time duration mentioned in the RTS, which is equal to the time required to transmit a CTS, a DATA packet, an ACK and an additional duration equal to $3 \times \text{SIFS}$. This prevents these nodes transmitting any packets during the period the NAV is set. Similarly, any node other than the transmitter, who hears the CTS, but has not heard the RTS before (often called the *hidden nodes*), sets its NAV to the time period mentioned in the CTS, which is equal to the time required to send a DATA packet, an ACK and an additional duration equal to $2 \times \text{SIFS}$. This prevents any node in the radio neighborhood of the transmitter or receiver transmit any packet

until the ACK is transmitted. Figure 2 illustrates the entire exchange mechanism.

Any node that did not receive the RTS/CTS correctly, because it was received with $\text{SINR} < \gamma$, but was able to sense the medium to be busy (a condition that is satisfied when the interference power received is sufficiently higher than the noise floor), would set its NAV to the EIFS duration (extended interframe spacing).

It is possible that the receiver does not receive the RTS correctly because of a collision or fading. Even if it does, it may not always respond with a CTS because, for example, its NAV is set. If the transmitter does not receive CTS within the expected time RTS timeout period, it goes into another random backoff and retransmits RTS when the timer reaches zero. For each backoff, the contention window w is doubled, until it reaches a maximum value. While a node is in backoff, it continues to sense the medium. If the medium is sensed busy or the NAV is set, the backoff counter is frozen during this period. The 802.11 protocol allows a maximum of seven RTS transmission retries. An exception is raised when the packet cannot be transmitted even after the maximum number of retries, causing the frame to be dropped, and possibly sending a feedback to the upper layer (e.g., routing) that may cause route repair activities.

B. Impact of Channel Model

Note that even though RTS retries are allowed in 802.11, it usually takes care of problems due to RTS collision or NAV being set at the receiver. These are indicative of high interference at the receiver. However, the protocol has little option to overcome the effect of time-varying multipath fading – something that cannot be easily removed by simple changes in the protocol. To understand things better, in this subsection we present a well-known radio propagation model, and then analyze how this may influence 802.11 behavior.

Assume that the signal power transmitted by the transmitter is P_T . The signal power P_R received at the receiver at a distance d from the transmitter at time instant t is explained by a combination of large-scale and small-scale propagation models [23]. The large-scale model explains variations in P_R for large changes in d , while the small-scale model explains the same for small changes in d or t . It is well-recognized that in the large-scale, P_R drops with distance following an inverse-power law:

$$P_R \propto \frac{P_T}{d^\alpha},$$

where α is a constant dependent on the exact nature of the model used and is usually between 2–5 depending on the environment. The constant factor governing the above pro-

portionality is a function of parameters not of direct concern to us here, such as antenna parameters, transmit carrier frequency etc. The small-scale model influences this received power with a multiplicative, time-varying factor with known statistical characteristics. When there is a dominant signal component present (say, the line-of-sight or LOS component) among various signal components reflected at various objects and being superimposed at the receiver, this factor follows the *Ricean* probability distribution [23] given by,

$$p(r) = \frac{r}{\sigma^2} e^{-\frac{(r^2+A^2)}{2\sigma^2}} I_0\left(\frac{Ar}{\sigma^2}\right),$$

where A is the peak amplitude of the dominant signal, $2\sigma^2$ is the variance of the multipath, and $I_0(\cdot)$ is the modified Bessel function of the first kind and zero-order. Ricean distribution is typically described in terms of a parameter K , given by

$$K = \frac{A^2}{2\sigma^2}.$$

As A increases (i.e., the dominant path increases in amplitude), K also increases.

When transmitter, receiver or objects in the surrounding environments are moving, there is a *Doppler shift* in the frequency of the received signal. Let us denote the maximum Doppler shift by f_m , where $f_m = v f_c / c$, v being the maximum perceived relative velocity between the transmitter and receiver (which could be caused by the motion of surrounding objects reflecting transmitted signal), f_c is the carrier frequency and c is the speed of light. The Doppler shift causes the signal power to fluctuate in time but with certain temporal correlation property. This fluctuation is usually described by the *level crossing rate* (N_R) which is the rate at which the signal envelop, normalized to the RMS (root mean square) value, crosses a given level R in the positive going direction. N_R depends on the given level R , the parameter K and the maximum Doppler shift f_m [23]. Knowing N_R , the *average fade duration* (average duration for which the signal level is below a given level R) can be computed as,

$$\bar{\tau} = \frac{Pr(r \leq R)}{N_R},$$

where $Pr(r \leq R)$ is the cumulative distribution function of the Ricean distribution.

Data presented in [21] for Doppler frequencies that can be encountered in practice² show that the average fade duration can be in the order of tens of milliseconds. As a specific example, for the 2.4 GHz carrier frequency (f_c)

²While data for only $f_m = 20$ Hz is presented in [21], the average fade duration for any f_m can be easily computed, given that the relationship between N_R and f_m is linear.

and 2 m/sec relative speed (v), the Doppler frequency f_m is 16 Hz. For this Doppler frequency, for 10 dB or more power loss due to fading, the average fade duration is approximately 10 ms; for 5 dB or more it is approximately 20 ms; increasing to approximately 30 ms for 1 dB. Note the increase in fade duration with “deeper” fade.

Common routing protocols in ad hoc networks focuses on optimizing the number of hops between source and destination. This tends to increase the physical distance of each hop, so that the number of hops is minimum. This lowers the received power P_R as modeled by the large-scale propagation model. Thus, even a small reduction in received signal power due to fading may make the SINR fall below the receive threshold γ causing a transient loss of link that may persist for several tens of milliseconds.³

Compare these average fade durations with the fact that it takes approximately 30ms for the RTS retries to fail 7 times causing the MAC to drop the frame. This is computed by using the interframe spacings and slot times from the standard specifications [12], assuming each random backoff lasts for its average duration, and the NAV is never set. Setting of NAV during the time a node is on backoff will extend the backoff time by the NAV period. This analysis shows that it is quite possible that a link is in fade long enough that data transmission will fail in spite of multiple retries. It is also conceivable from the above analysis that it is very likely that 802.11 will need to make a few RTS retries to complete the entire exchange. This fact will later be verified via simulation experiments.

III. CHANNEL STATE-BASED LINK SELECTION

Assume now that multiple possible next hop options are presented to the transmitter, and its responsibility is to transmit to *any one* of these receivers successfully. Since fading on different links is expected to be uncorrelated, it is unlikely that all links are in deep enough fade at the same time with $\text{SINR} < \gamma$. Thus, it is likely that transmission on at least one link is possible without any significant number of retries in the average case. In the next sub-section, we describe an extension of 802.11 that uses this idea.

A. Anycast Extension for 802.11

The anycast extension uses a similar handshaking protocol as in 802.11 DCF, but takes advantage of multiple receivers with the goal to transmit the frame to any one of them successfully. It can be thought of an anycasting

³Note that physical layer techniques such as transmit power control and rate control can be used to tackle such link loss to some extent. In general, the design of an anycast MAC should subsume the transmit power and rate control approaches in the physical layer. However, with a given physical layer design, loss of link will still be a reality, and anycasting can always play an important role in the design space.

scheme in the link layer. The routing layer computes multiple routes between the source and destination. We will describe this mechanism in the next subsection. At each hop, the routing layer passes on the multiple next hop information to the MAC layer. The transmitter now “multicasts” the RTS to these multiple next hops (it is actually a broadcast control packet as before). We will refer to the multicast RTS as MRTS; it contains all the next hop receiver addresses. Because of practical considerations (such as RTS packet size), we limit the number of next hops to use to a maximum of four.

The four next hops are assigned a priority order, which can be determined by their respective positions of their addresses in the MRTS packet. The priority can come from the routing or any lower layer. As an example for routing layer, the next hop leading to a shorter path to the destination gets higher priority, or the next hop that has a less number of packets waiting in the interface queue gets higher priority. As an example for the MAC/physical layer, relevant statistics related to the amount of error correction can be used as an indicator for the quality of the link and hence to determine its priority. A combination of the above can also be used.

When an intended receiver receives the MRTS packet, it responds by a CTS. These CTS transmissions are staggered in time in order of their priorities. The first receiver in the order transmits the CTS after an SIFS, the second after a period equal to the time to transmit a CTS and $3 \times \text{SIFS}$, and so on. See Figures 3, 4, 5 for an illustration. Note that the staggering ensures that the CTSs are separated by at least $2 \times \text{SIFS}$ period; thus they do not collide.

When the transmitter receives a CTS (which may or may not be the first CTS transmitted), it transmits the DATA frame to the sender of this CTS (which would be the highest priority receiver that responded) after an SIFS interval. This ensures that other, lower priority receivers hear the DATA *before* they send CTS — as the next one in priority will not send a CTS until another SIFS interval — and suppress any further CTS transmission. All such receivers then set their NAV until the end the ACK period. (The DATA packet carries this period in the header just in case these receivers missed the MRTS). See Figure 3 for an illustration when the very first CTS transmitted has been successfully received. We provide two other illustrations demonstrating the scenarios when the first CTS was not received, but the second was received (Figure 4); and when all but the fourth CTS were not received (Figure 5).

Any other node that hears the MRTS (*exposed node*), sets its NAV for the entire duration mentioned in the MRTS packet. This duration depends upon the number of receivers (which can be a maximum of four) to which MRTS is being sent. For instance, if the number of receivers is k ,

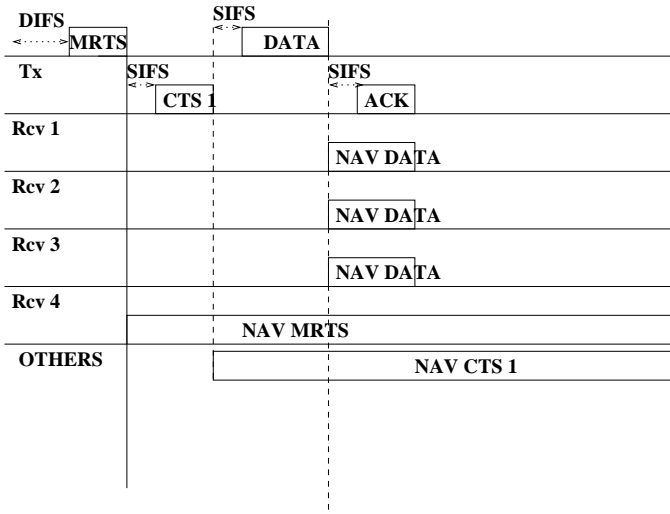


Fig. 3. Timeline showing the anycast extension of 802.11 when the first CTS is successfully received.

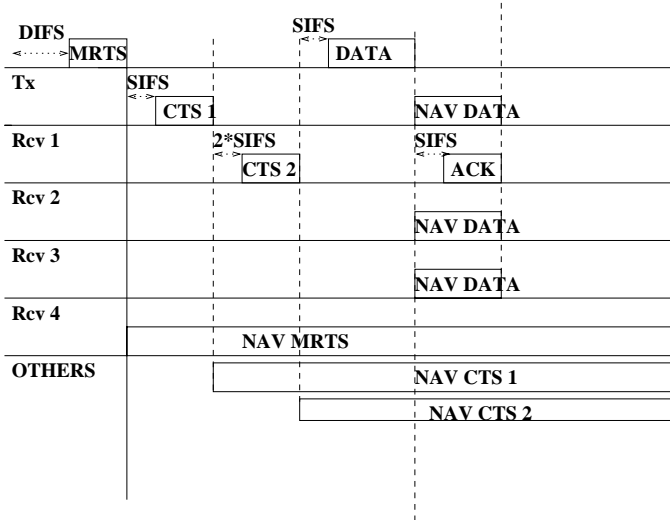


Fig. 4. Timeline showing the anycast extension of 802.11 when the first CTS is lost; but the second CTS is received successfully.

the NAV is set to $k \times \text{CTS} + (2k + 1) \times \text{SIFS} + \text{DATA} + \text{ACK}$ time. This time is the maximum time needed for the data transfer to complete. Similarly, any node that hears any of the CTSs (*hidden node*) sets its NAV until the ACK period. For example, such a node upon receiving the i -th CTS, will set its NAV for the period $(2(k - i) + 1) \times \text{SIFS} + (k - i) \times \text{CTS} + \text{DATA} + \text{ACK}$. See Figures 3, 4, 5.

If none of the CTSs are received successfully, the transmitter goes into a random backoff and then retries again with the same receivers. The random backoff procedure is exactly as in 802.11 except that in the experiments we have allowed a lower number of maximum retries – six instead of seven. This is because the possibility of failure is much less with multiple choices of the next hop.

Note that the protocol reduces to 802.11 when there is only one next hop receiver. This gives us an opportunity

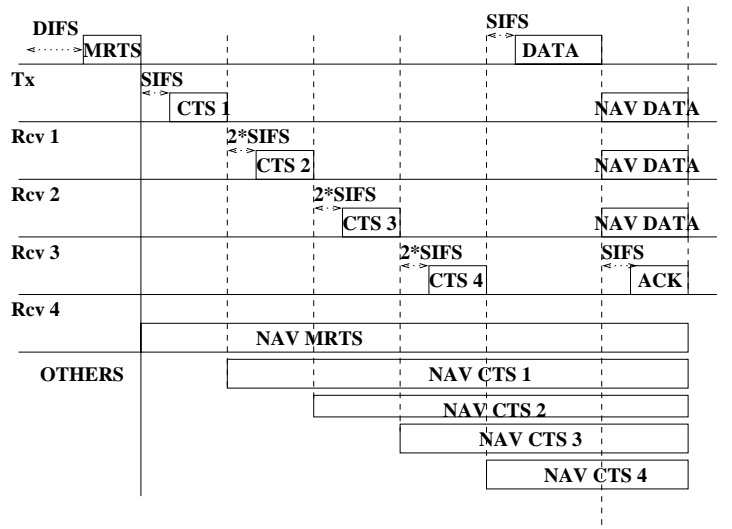


Fig. 5. Timeline showing the anycast extension of 802.11 when all but the fourth CTS is lost.

for a fair performance comparison. Also, note that when multiple next hops are indeed available and the CTS from the highest priority receiver is received successfully, this would be the same receiver sending CTS in an equivalent 802.11-based scenario. In this case again, the protocol behaves similar to 802.11, but it sets a longer NAV period for the hidden and exposed terminals. In this context, also note that in situations when multiple CTS's come back, all nodes in the vicinity of the receivers sending CTS's set up their NAV, while only the last one is involved in communication. There is no easy way to resolve this problem. Our simulation studies do show that even with large traffic diversity, anycast performs very well relative to 802.11. Thus, the harmful effect of this additional exposed terminal problem is not high enough to mask the benefit of the protocol.

It is possible that the fade state of the channel can change from the point when CTS is transmitted to when DATA or ACK is transmitted, causing the exchange to fail. But we claim that it is unlikely. The *coherence* period (T_c) of a fading channel defines the approximate interval the channel state remains very correlated or, in other words, does not change significantly [23]. T_c is approximately equal to the inverse of the Doppler frequency (f_m). From the values we have used in the previous section, it is easy to see that the coherence period is expected to be large enough for the DATA transmission to succeed if a CTS indeed has succeeded. As an example, for $f_m = 16$ Hz, $T_c = 62.5$ ms. Compare this with the time to transmit a 1000 byte DATA frame. At 2 Mbps the transmission time would be 4 ms; at 11 Mbps it would be 0.73 ms.

It is obvious that the protocol benefits the most when a fair number of choices for the next hop is available. This increases the probability that the data transmission takes

place successfully. Thus the effective operation of the protocol is dependent on a routing layer being able to compute enough redundant routing paths. The next subsection discusses the design choices we make in the routing layer that plays a significant role in the performance.

B. Design of Multipath Routing Layer

Multipath routing protocols have been explored in mobile ad hoc networks to maintain multiple redundant routes to provide fault tolerance and also for load balancing [18], [16], [14]. Availability of multiple routes reduces route maintenance overhead as routes need to be recomputed only when all available routes fail. Also, it is possible to forward data packets over multiple routes simultaneously (dispersity routing [15]) to provide more traffic diversity and to reduce load on each individual route [18].

We will use an on-demand multipath routing protocol to provide the MAC layer with multiple next hop links. Specifically, we will use AOMDV [14], a multipath extension of a popular on-demand single path routing protocol AODV [19], [20] that is based on the distance vector concept. In AODV, when a traffic source needs a route to the destination, it initiates a route discovery by flooding a route request (RREQ) for the destination in the network, and then waits for the route reply (RREP). When an intermediate node receives the first copy of a RREQ packet, it sets up a reverse path to the source using the previous hop of the RREQ as the next hop on the reverse path. In addition, if there is a valid route available to the destination, it unicasts a RREP back to the source via the reverse path; otherwise it rebroadcasts the RREQ packet. Duplicate copies of the RREQ are discarded. The destination, on receiving the first copy of a RREQ packet, behaves the same way. As a RREP proceeds to the source it builds a forward path to the destination at each hop.

In AOMDV, a node can form multiple reverse routes to the source using the duplicates of the RREQ packet; but it still rebroadcasts only one RREQ. Additionally, the destination or any node having a path to the destination may choose to respond to multiple RREQs it receives via multiple reverse paths already formed. As presented in [14], AOMDV uses mechanisms to ensure link disjointness of the multiple paths; however, in this work we have turned off these mechanisms to allow overlapped routes. The benefit is that removal of the disjointness constraint automatically provides many more paths. We will see later that more paths are beneficial for performance.

Note that this is a significant departure from multipath routing techniques that try to guarantee some form of disjointness [14] to ensure independence of path failures. However, this is important only when link failures are viewed as a more “stable” event, i.e., links change

state (from off to on, for example) in the time scale of route changes in the routing protocol. In the model we are interested in, link failures are transient, and links are expected to change state within a much shorter time scale. This may not be true, however, when link failures may be caused by mobility. In the simulation experiments we report later, we still see significant improvement with overlapped paths even in mobile scenarios, making it a sensible design choice.

Note that in our model, the routing packets also face the same fading channel as the data packets. Thus, transient link failures impact the route discovery process, which is unavoidable. Routing may also form next hop links that could be too weak normally, but just had been strong enough during route discovery. We have made simple optimizations AOMDV to make routing more efficient. As an example, the RREPs are broadcast instead of unicast. This gives an opportunity to at least some of the next-hop neighbors on the reverse path to receive the packet successfully, and form the forward paths. Here again, we rely on the assumption of lack of correlation between the channel state of different links on the same node. The traditional timer-based route expiry in AODV or AOMDV is not used, because this may delete unused, but possibly valid routes. Other key techniques in AOMDV, such as use of sequence numbers for loop prevention and determining freshness of routes, and the route error-based route erasure process are not altered.

One other design choice we need to make, is whether to allow paths that are too long relative to the shortest paths. This issue presents a trade-off that must be carefully orchestrated. To understand this, take an example where 802.11 fails to transmit on a next hop link because of fading, causing it to retry. Assume that we are using the shortest path routing and the data packet is still k hops away from the destination needing at least k more transmission attempts for the packet to reach the destination. If we use anycast instead, under an identical scenario, the protocol will choose an alternate next hop. Assume that the current node is $k+l$ hops away from the destination via this alternate next hop. This means that even though this transmission is successful, the packet still needs at least $k+l$ transmission attempts to reach the destination. Thus, the 802.11 transmission must fail at least l times for the multipath extension to be of any value. Of course, $l = 0$ is an ideal possibility; but this may reduce the number of alternate paths drastically. We empirically evaluated various possibilities for l , and found that $l = 1$ to be a reasonable choice. Thus, we allow only those paths to be formed in the routing table that are at most one hop larger than the shortest path. The value of l can be a parameter of the protocol. It is worthwhile to mention here that in [16] the

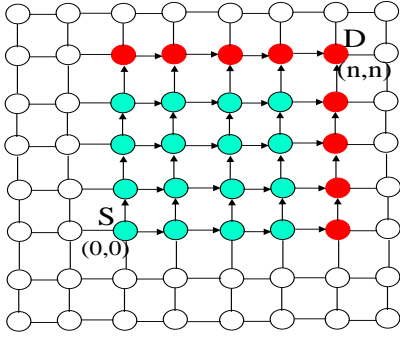


Fig. 6. Grid network for analyzing packet delivery probability.

authors also have noted that limiting the path length difference (l) is a useful optimization in multipath routing.

IV. PERFORMANCE EVALUATION

We present three sets of performance results. The first set builds a simple model to analytically evaluate packet delivery probability in a grid network when single or multiple next hop links are available. The second set presents experimental evaluation on the Berkeley motes platform in a similar grid network. Both these networks provides valuable insights, even though they are restricted in some form — because of tractability reasons for the analytical model and logistical reasons in the experimental motes testbed. The third set of results use *ns-2* [9] based simulations, that do not have any of these restrictions and can use elaborate scenarios.

A. Analysis for a Grid Network

Consider a two dimensional grid network as in Figure 6 with 4-nearest neighbor connectivity. This model is representative of networks with a rich set multipaths such that many forwarding options are available. This network model is simple enough to study closed form expressions for packet loss probabilities for multihop routing with unicast or anycast forwarding. Suppose, nodes S and D are the source and destination nodes respectively. Without loss of generality assume that the coordinate of S is $(0, 0)$ and that of D is (n, n) . The shortest path length between S and D is $2n$. The nodes falling on the shortest paths are shaded. 2 next hops are possible on all hops on all shortest paths *except* on the boundary nodes on the $n \times n$ rectangle beyond n hops from S . These nodes are shaded in red/dark in Figure 6. On these boundary nodes, only 1 next hop is possible.

Now, assume that the probability of a link loss is p and the probabilities are independent. If only a single next hop is used for packet forwarding and their is no retry, the packet drop probability at each hop is p . Thus, the proba-

bility P that a packet from S will reach D is given by,

$$P = (1 - p)^{2n}.$$

If multiple next hops are available (in this case the maximum is a modest 2), the packet drop probability at each hop is either p (if there is only one next hop) or p^2 (if there are 2 next hops). Note that 2 next hops are available for each of the first n hops; beyond this, the boundary nodes can provide only 1 next hop, but the rest of the nodes can still provide 2. Thus, in the last n hops, each hop can undergo a packet loss with probability p or p^2 . To determine the combined probability, we need to evaluate the proportion of paths that go through boundary and non-boundary nodes for each hop beyond the first n hops.

If a node (i, j) is at a distance l from S (i.e., the node is at the l -th hop), $i + j = l$. Simple combinatorics can determine that the number of (shortest) paths of length l from S to node (i, j) is

$$\frac{(i + j)!}{i!j!}.$$

A node could be a boundary node only if $l \geq n$. A boundary node on a shortest path must satisfy i or $j = n$, and a non-boundary node on a shortest path must satisfy i or $j = (n - 1), (n - 2), \dots, (l - n + 1)$. This determines that the number of such paths going through *all* boundary nodes at hop $n \leq l < 2n$ is given by

$$B(l) = \frac{2(l!)}{n!(l - n)!},$$

the factor 2 coming from the fact there are two boundary nodes at each hop. Also, the number of paths going through all non-boundary nodes at hop $n \leq l < 2n$ is given by,

$$NB(l) = \sum_{k=1}^{2n-l-1} \frac{l!}{(n - k)!(l - n + k)!}.$$

Since all paths are equally likely in our model, at hop l a boundary or a non-boundary node will be used simply in proportion to the number of paths going through them. Accordingly the packet drop probability at hop l will be either p or p^2 , respectively. Combining all these, the probability P that a packet from S will reach D is given by

$$P = (1 - p^2)^n \times \prod_{l=n}^{2n-1} \left\{ 1 - \frac{B(l)p + NB(l)p^2}{B(l) + NB(l)} \right\}.$$

The first term is for the first n hops and the second term is for the following n hops.

Figure 7 plots the packet delivery probability P versus the path length ($2n$) for different link loss probabilities (p)

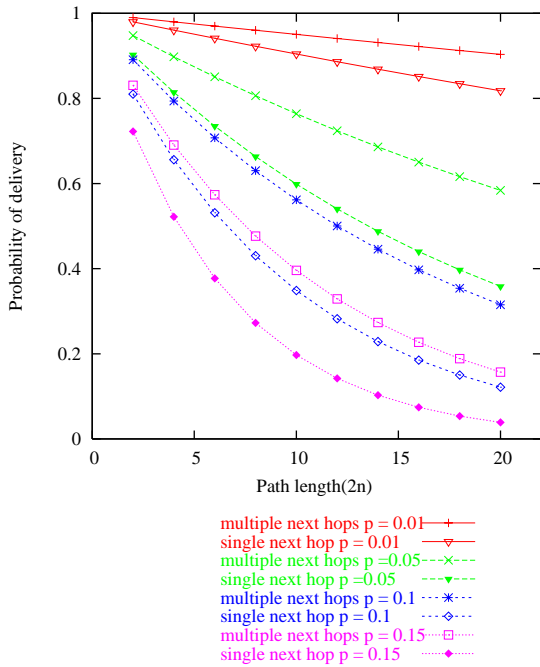


Fig. 7. Packet delivery probabilities for the grid network of Figure 6 with single (unicast) and multiple next hop forwarding (anycast).

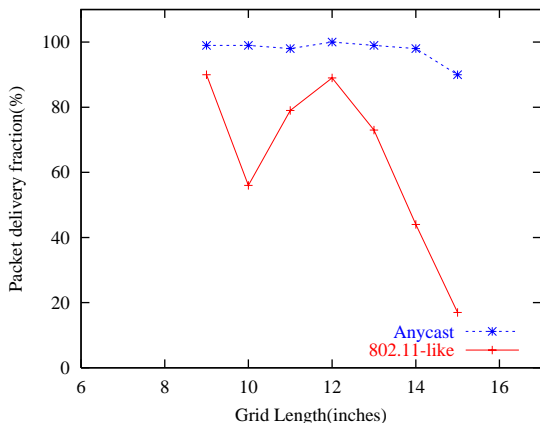


Fig. 8. Packet delivery fraction in the 4×4 Berkeley notes testbed with S-MAC protocol stack.

for both single (unicast) and multiple next hop forwardings (anycast). Note that even though only a maximum of 2 next hops are used, there is a significant relative improvement in delivery probability with multiple next hops, particularly as the path length increases. Larger number of next hop possibilities should improve the probability further.

B. Evaluation on Experimental Testbed

We implemented the anycast protocol on Berkeley notes platform, manufactured by Crossbow Technology [7], [1]. While our original intention is to use anycast

as a replacement for 802.11-based MAC layer protocol, implementing anycast on 802.11-based hardware requires modification of the firmware in the network interface card. This requires working with the chipset and/or card manufacturers. However, a proof-of-concept implementation is possible on the Berkeley notes platform, where link layer protocols are implemented in software as a part of the protocol stack in the TinyOS operating system [10], [1]. We used the Mica platform for our experiments that uses an Atmel ATMEGA series microcontroller (4MHz, 8-bit) as the processor and an RFM TR1000 transceiver operating at 916MHz as the radio interface. In the Mica platform the radio bit rate limited to about 50 Kbps. This speed is CPU limited, as the protocol processing happens at the sole processor on the mote.

For a meaningful implementation, we used the S-MAC protocol stack [25], [24] developed in USC/ISI. S-MAC replaces the MAC and PHY layer implementations in the original TinyOS network protocol stack and provides a flexible architecture to develop new MAC protocols by providing a flexible packet format and clear separation between the MAC and PHY layers. The original S-MAC implementation [25], [24] uses a protocol very similar to the IEEE 802.11 DCF for channel access operating in the ad hoc mode, including implementations of inter-frame spacings, physical and virtual carrier sensing, backoffs and retries, RTS/CTS/DATA/ACK based handshake, and network allocation vectors. It also uses several innovations for energy management, which we turned off to make the protocol very similar to regular 802.11. Since the entire implementation is in software, this provides an excellent platform to experiment with new MAC protocols albeit with low data rate radios.

We modified the S-MAC protocol stack to implement anycast by modifying the base 802.11-like implementation. In the test scenario we placed 16 motes in a square 4×4 grid configuration as in Figure 6. Back-to-back data packets are transmitted from one corner of the 4×4 grid to the opposite corner. Routes are manually set up exploring all possible paths (similar to the analysis in Section IV-A). Figure 8 shows the relative packet delivery performance of the 802.11-like protocol and our anycast implementation in the S-MAC protocol stack. The length of a side of the unit grid is varied to provide an independent means to control the radio performance. Increasing the length beyond a threshold makes the signal strength fairly weak and radio performance very much prone to multipath fading and other noise. The experiments were performed in a small laboratory room in a computer science department in its natural state, i.e., with usual furniture, people moving around and possible sources of radio noise; but no noise was intentionally created to influence the exper-

iments.⁴ An average of a large number of experiments is reported in Figure 8. The positions (including pose) of the motes were kept unaltered across experiments with the same grid size. Note the poor packet delivery performance for the 802.11-like protocol as the grid size is increased.⁵ Anycast provides an excellent performance over the entire range.

C. Simulation Model

We used the *ns-2* [9] simulator with the AOMDV protocol [14] in the routing layer and the anycast protocol in the MAC layer. As mentioned before, the AOMDV model used here allows overlapped paths; and only those paths are used that are at most one hop larger than the shortest path the protocol is able to find. With 802.11, the traditional forwarding model is followed. The next hop link on the shortest path is attempted first. Upon failure (i.e., when maximum retry count is exceeded), this link is marked down and the next shortest alternative is used. A route error is generated only when all alternatives are exhausted. In the anycast protocol, the next hop priorities are generated based on path lengths alone.

The traffic model uses CBR (constant bit rate) traffic with randomly chosen source-destination pairs. A traffic rate of 1 packet/sec (512 byte packet) per flow was used in the experiments. Load is varied by varying the number of flows (number of sources). For each packet delivered to the destination the number of hops it traveled is logged, and its average statistics is used as a parameter in the performance plots. For mobile experiments, the popular *random waypoint* mobility model [4] is used. Here, a mobile node alternately pauses and moves to a randomly chosen location with a constant but randomly chosen speed. The pause times and the average speed are parameters of this model.

The radio propagation model uses the two-ray ground reflection path loss model [23] for the large-scale propagation model (as in the *ns-2* distribution), augmented by a small-scale model modeling Ricean fading as presented in Subsection II-B. The *ns-2* extension provided by the authors of [5] has been used for the fading model. Here, the Ricean fading is modeled using an efficient simulation technique that also captures the time correlation of the signal envelop depending on the Doppler spread created by

⁴We indeed have seen significant improvements in performance of the 802.11-like implementation in remote, quiet and open outdoor environments, where not much link diversity could be obtained to make anycast significantly meaningful. Such environment also provided a much larger radio range.

⁵We also noticed some amount of unstable performance for the 802.11-like protocol for lack of diversity. For example, at certain grid lengths (10 and 11 inches) the performance was relatively poor, possibly due to some multipath effects created at these lengths.

the relative motion of the transmitter and/or receiver (could also be caused by the motion of reflecting objects). The technique employs a lookup operation in a pre-generated dataset containing the components of the time-sequenced fading envelop.

The original implementation in [21] uses the simulation time instant to index into a channel table that causes all next hop links from a node to undergo exactly similar fading which is unrealistic. In order to make them uncorrelated, the index uses both simulation time (to provide time correlation) and the next hop node id (to prevent correlation between channel conditions on all next hops links). Similar “corrections” for the same the code base has also been reported in [11] in the context of multi-rate MAC implementations. A value of 5 dB for the Ricean K factor has been used unless otherwise stated. For stationary networks, a max relative velocity v of 1 m/sec has been used to compute the Doppler shift f_m .

Three different network models have been used for evaluation each with 200 nodes and various number of traffic flows: The first model is a stationary grid network similar to Figure 6. Here, the grid is, however, rectangular 40×5 with the distance between adjacent nodes in the grid being 100m. Note that the nominal radio range (without fading) being about 250m, it gives a fair number of routing paths between random pairs of source and destination. We ran several simulations with various numbers of sources. Since the distance between the source-destination pairs is a sensitive parameter (as we have seen in the model developed in the previous subsection), we have controlled the random selection of source and destinations in a way to give us specific values for the “shortest” path lengths (in hops).

The second model uses a network of 200 randomly positioned stationary nodes in the same area ($4000\text{m} \times 500\text{m}$). Similar experiments were run by controlling the random choices of source destination pairs so that their shortest path lengths fall close to pre-selected specific values. The third model uses the same number of nodes in the same area; but now they are mobile and follow the random waypoint mobility model. The pause times and speed are varied to control the mobility. Because of mobility, it was not possible to control the hop-wise distance. All simulations are run for 900 simulated seconds. Each data point represents the average of 5 runs.

D. Simulation Results

Figure 9 plots the average packet delivery fraction for the stationary grid network model for the two link layer models. As expected, the delivery fraction goes down with increase in path lengths with anycast performing better – with the performance differential increasing with the path

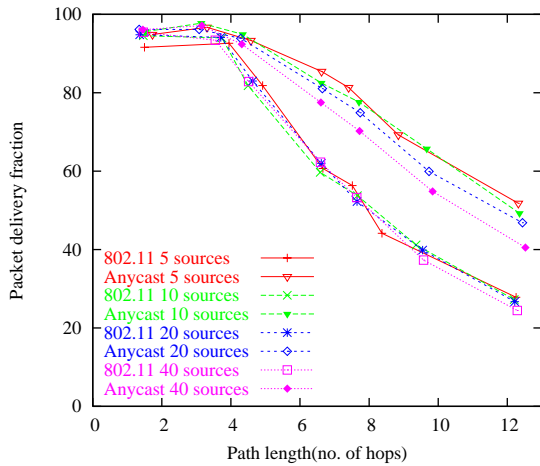


Fig. 9. Packet delivery fraction with 802.11 and anycast in a stationary grid network with 200 nodes.

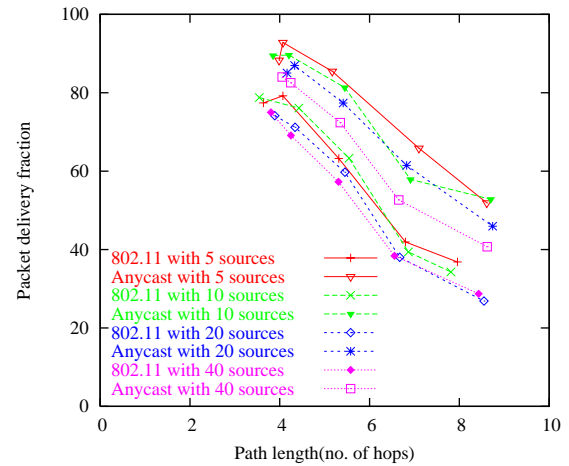


Fig. 11. Packet delivery fraction for 802.11 and anycast in the stationary random network.

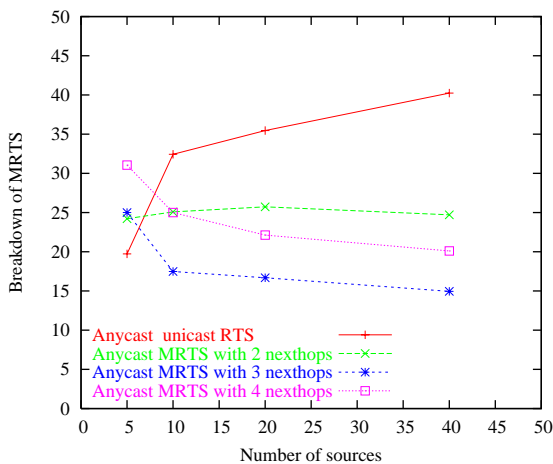


Fig. 10. Percentage of MRTS packets with different numbers of next hops in stationary grid network (average path length is approx 6).

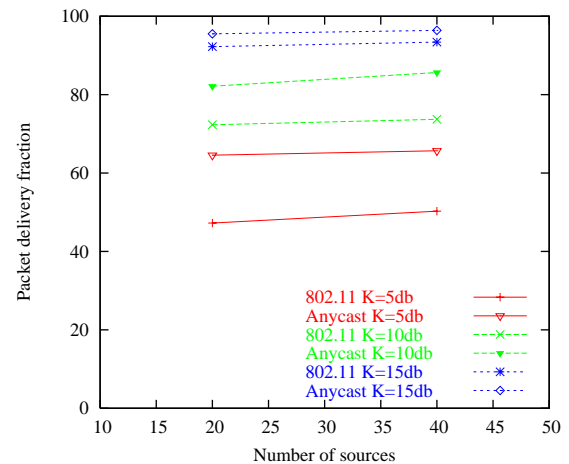


Fig. 12. Demonstrating the effect of Ricean K factor (stationary random network, hop-wise distance = 4).

length. A performance gain of up to a factor of 2 is observed for large path lengths.

Note also that the anycast performance is going down with increase in number of traffic sources, while for 802.11, the performance is almost independent of this parameter. It turns out that with more traffic diversity the route discovery is unable to provide a large number of routes because of loss of route request packets due to increased interference. Note that route request packets are broadcast packets and thus they are more susceptible to fading and interference as they cannot be retransmitted. Figure 10 demonstrates this effect, where the percentage of MRTSs that have 1,2,3 or 4 next hops are plotted against number of sources. Note the increase in unicast MRTS (i.e., MRTS with only one next hop receiver) with traffic, and corresponding decrease in MRTSs with 3 or 4 next hops.

Figure 11 shows the packet delivery performance in the

stationary random network. Note again that performance improvement varies from about 20% to upto about a factor of 2 for large path lengths. Because of the randomness involved the hopwise distances could not be varied over as wide a value as in the grid network. We also analyzed the impact of the changes in fading in this set up. Figure 12 shows packet delivery fraction for a specific set of scenarios with 20 and 40 sources when the hop-wise distance is about 4. Here, the Ricean K parameter is varied which influences the relative amplitude of the dominant signal component. Note that the dominant component is relatively stronger (larger K value) the impact of fading is less. Thus, with smaller K , the absolute performance degrades, but the performance differential between multiple and single next hops increases.

Finally, Figure 13 presents the packet delivery performance in a mobile scenario with average speed of 20 m/s respectively. Note that anycast is performing about 25–40% relative to the unicast performance. In these set

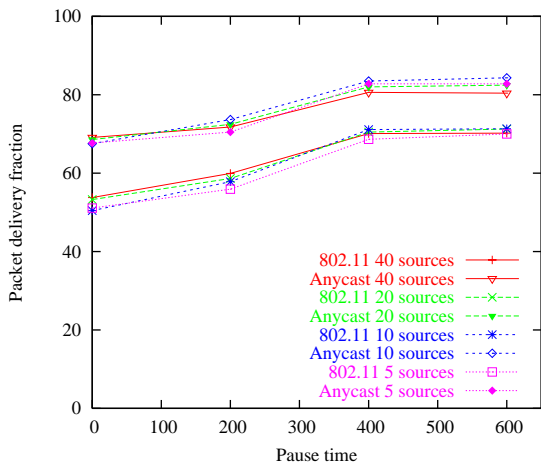


Fig. 13. Packet delivery fraction for 802.11 and anycast in the mobile scenario (average speed =20m/s)

of experiments the impact of increasing load (number of sources) is minimal. This is because of relatively small average path lengths (about 3.5) realized in these experiments.

V. RELATED WORK

In [13], a combination of forwarding and MAC layer protocol called *selection diversity forwarding* has been proposed. Here, the data frame is multicast to a set of candidate nodes, each of which send back ack control packets. Then only one node is chosen from this set by the forwarding node and issued a *forwarding order* control packet, which is again acknowledged. This is the node that will forward the data packet further; and others will discard the packet. Note that there is no channel reservation such as 802.11 or our anycast extension. Data packets can easily collide, and the overall exchange takes longer as the forwarding order has to wait to for all acks. The criterion to choose the forwarding node depends on the upper layer protocol. For example, the forwarding node could be the one that provides the maximum forward progress in geographic forwarding. Selection diversity forwarding has been shown to perform better than fixed forwarding mechanisms, such as NFP (nearest with forward progress) or MFR (most forward with fixed radius) for Rayleigh fading channels.

Several recent articles build on the 802.11 standard to estimate the channel condition and automatically adapt the sending bit rate to match the channel conditions. However, they still use single next hop, and use the unicast forwarding model in 802.11. In the RBAR protocol [11], the receiver estimates the channel condition by the physical layer analysis of the RTS packet and determines the best rate to send the data frame. The control packets are sent using the base (lowest) rate so that they are always success-

fully delivered. The OAR protocol [5] extends this idea to send multiple back-to-back packets when the channel condition is determined to be good. OAR also takes care to ensure fairness, as there is a chance in this protocol that links with better channel conditions can get more share of the channel bandwidth.

In [22] an adaptive transmission protocol is used that adjusts the power and code rate of the transmitted signal to adapt to the channel conditions. But this scheme does not work when a poor quality link has not been used by the routing protocol for some time. The work suggests an alternate forwarding technique dependent on multipath routing that alters routing paths to discover links that may have improved recently.

Three recent papers also motivate use of anycasting in the MAC layer. In [6] authors motivate anycast as a general-purpose MAC layer method to take decisions on packet forwarding in short time scales. They describe potential use of anycast from the point of view of improving spatial reuse and reducing interference. They describe applications with power-controlled multiple access and directional antenna. However, since this is a position paper, no performance evaluation is reported. In the same forum, an “opportunistic” routing mechanism is presented [2], which is very similar in spirit to the selection diversity forwarding work described earlier. Another protocol called GeRaF [27] also contains similar ideas, but has been specifically applied for geographic forwarding. Here, the interest is more on modeling, rather than a practical implementation.

VI. CONCLUSIONS

We have proposed an anycast mechanism at the link layer that forwards packets to the best suitable next hop link to enable efficient packet forwarding on a multihop route. This mechanism is dependent on the availability of multiple next hops, which could be computed by a multipath routing protocol. We have designed the link layer protocol as an extension of the popular IEEE Standard 802.11 and carried out an extensive performance evaluation using both an experimental testbed and detailed simulation modeling. The anycast protocol provides a significantly better packet delivery relative to 802.11 in a variety of ad hoc network models, both regular and random, stationary and mobile. The performance differential was observed to increase when path lengths increase.

Note that when multipath routing is combined with anycast, the forwarding decisions taken at each hop is a local decision. This can easily increase the overall path length unless the forwarding is orchestrated carefully (see the discussion on the value of l at the end of section III-B). Some mechanisms to do this on a per-packet basis has been dis-

cussed in [6].

Another point of concern is the operation of the routing protocol. The routing protocol itself suffers from the transient weak channel conditions, and may fail to discover links that (transiently) fail to deliver routing messages. This does not seem to be a significant problem in the our simulations. However, we anticipate a different method of delivery for routing messages can improve performance (such as using higher transmit power to counteract fading).

Packet reordering can be a problem too in the anycast forwarding model that we used. Frequent reordering can impact TCP performance, as a series of duplicate acknowledgments is perceived as an indicator of congestion and packet loss. However, we contend that this is a more general problem, and will be present in any form of multipath forwarding technique. The internetworking community is paying attention to this problem, and some solutions are indeed available (see, for example, [3]).

In summary, exploiting the diversity in the channel conditions on various next hop links (path diversity) provides better immunity from channel errors. Other methods of recovering from error would be to increase transmit power, or decrease effective bit rate to mask the impact of the poor channel conditions. Our future work will focus on combining these methods to path diversity to provide a more “holistic” improvement in performance.

REFERENCES

- [1] TinyOS Community Forum. <http://www.tinyos.net>.
- [2] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. In *Proc. ACM Hotnets-II Workshop*, 2003.
- [3] Stephan Bohacek, Joao Hespanha, Junsoo Lee, Chansook Lim, and Katia Obraczka. TCP-PR: TCP for persistent packet reordering. In *Proc. of the IEEE 23rd Int. Conf. on Distributed Computing System*, pages 222–231, May 2003.
- [4] 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 *Proceedings of the 4th International Conference on Mobile Computing and Networking (ACM MOBICOM'98)*, pages 85–97, October 1998.
- [5] B.Sadeghi, V.Kanodia, A.Sabharwal, and E.Knightly. Opportunistic media access for multirate ad hoc networks. In *Proceedings of the 8th International Conference on Mobile Computing and Networking (ACM MOBICOM'02)*, pages 24–35, September 2002.
- [6] Romit Roy Choudhury and Nitin H. Vaidya. MAC-layer anycasting in wireless ad hoc networks. In *Proc. ACM Hotnets-II Workshop*, 2003.
- [7] Crossbow Technology, Inc. <http://www.xbow.com>.
- [8] Anthony Ephermides. Ad hoc networks: not an ad hoc field anymore. *Wireless Communications and Mobile Computing, Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):441–448, 2002.
- [9] Kevin Fall and Kannan Varadhan (Eds.). *ns notes and documentation*, 1999. available from <http://www-mash.cs.berkeley.edu/ns/>.
- [10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler, and K.S.J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, November 2000.
- [11] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proceedings of the 7th International Conference on Mobile Computing and Networking (ACM MOBICOM'01)*, pages 236–251, July 2001.
- [12] I.E.E.E. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11–1997, 1997.
- [13] Peter Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5:79–282, October 2001.
- [14] M. Marina and S. R. Das. On demand multipath distance vector routing in ad hoc networks. In *Proceedings of the International Conference on Network Protocols (ICNP)*, pages 14–23, Dec. 2001.
- [15] N. F. Maxemchuk. Dispersity Routing. In *Proceedings of the IEEE ICC*, pages 41:10–41:13, 1975.
- [16] A. Nasipuri, R. Castaneda, and S. R. Das. Performance of multipath routing for on-demand protocols in ad hoc networks. *ACM/Kluwer Mobile Networks (MONET) Journal*, 6(4):339–349, 2001.
- [17] A. Nasipuri and S. R. Das. On-demand multipath routing for mobile ad hoc networks. In *Proceedings of the 8th. IEEE International Conference on Computer Communications and Networks (IC3N)*, pages 64–70, Boston, October 1999.
- [18] M.R. Pearlman, Z.J. Haas, P. Scholander, and S.S. Tabrizi. On the impact of alternate path routing for load balancing in mobile ad hoc networks. In *Proceedings of ACM MobiHoc 2000 Workshop*, August 2000.
- [19] Charles Perkins and Elizabeth Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.
- [20] Charles Perkins, Elizabeth Royer, and Samir R. Das. Ad hoc on demand distance vector (AODV) routing. RFC 3561, July 2003.
- [21] R.J. Punnoose, P.V. Nikitin, and D.D. Stancil. Efficient simulation of ricean fading within a packet simulator. In *Proc. IEEE VTC-Fall 2000*, volume 2, pages 764–767, September 2000.
- [22] M.B. Pursley, H.B. Russell, and J.S. Wycosarski. An improved forwarding protocol for updating channel state information in mobile FH wireless networks. In *IEEE Communications for Network-Centric Operations: Creating the Information Force*, volume 2, pages 967 – 971, October 2001.
- [23] T.S Rappaport. *Wireless Communication: Principles and Practice*. Prentice-Hall, 2002.
- [24] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, June 1994. To appear.
- [25] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, June 2002.
- [26] W.T. Zaumen and J. J. Garcia-Luna-Aceves. Shortest multipath routing using generalized diffusing computations. In *Proceedings IEEE INFOCOM 98*, March 1998.
- [27] M. Zorzi and R.R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. *IEEE Trans. on Mobile Computing*, pages 349–364, Oct-Dec 2003.