Available online at www.sciencedirect.com

ScienceDirect

Ad Hoc Networks 6 (2008) 805–825

Ad Hoc Networks

www.elsevier.com/locate/adhoc

Exploiting path diversity in the link layer in wireless ad hoc networks

Shweta Jain *, Samir R. Das

Computer Science Department, Stony Brook University, New York, NY, United States

Received 19 May 2006; received in revised form 12 April 2007; accepted 18 July 2007
Available online 24 July 2007

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. Further we experiment with anycast in networks that use multiple channels and those that use directional antennas for transmission. In these networks, deafness and hidden terminal problems are the main source of packet loss. We implemented anycast as extension of 802.11 like protocols that were proposed for these special networks. We are able to show that anycast is capable of enhancing the performance of these protocols by simply making use of the path diversity whenever it is available.

Keywords: MAC; Anycast; Ad hoc networks; Multichannel; Directional antenna; Interlayer-interaction

1. Introduction

It is well-known that in wireless ad hoc networks, the “link” between two nodes is a “soft” entity [1]. 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 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 [2] 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

---

[*] Corresponding author.
E-mail address: shweta@cs.sunysb.edu (S. Jain).

1570-8705/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2007.07.002

destructively. Interference is caused by signals from 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 [3,4])[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* [5] 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 [6–9]. 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 Fig. 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.

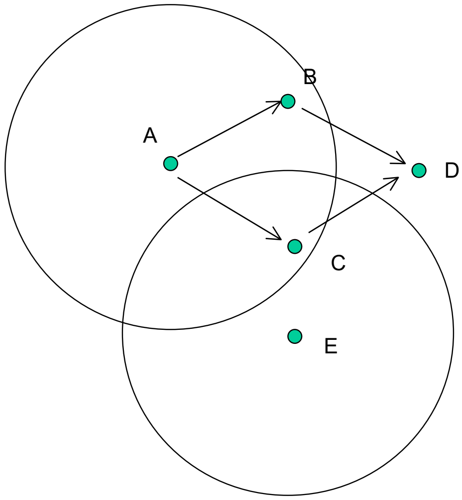


Fig. 1. Example scenario motivating anycast. Node A can forward packets to D either via B or C. But an ongoing transmission at E may interfere at C. If A chooses to forward via C, the transmission will defer until E's transmission is complete. Such instantaneous channel conditions are unknown to the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

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 [10] MAC layer. This makes performance easy to analyze and compare.

The rest of the paper is organized as follows. In Section 2, we provide an overview of the 802.11 MAC protocol operation and describe the properties of a fading channel. In Section 3, 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 multipath routing layer. We then divert attention toward application of anycasting in multichannel and directional antenna networks in Section 4, followed by Section 5 in which we present performance evaluation of anycast. We have analyzed the performance of anycast in a grid network via analytical modeling, and an experimental testbed using Berkeley motes. We have also performed detailed simulation-based evaluations using the popular *ns*-2 simulator. We describe the related work in Section 6 and conclude in Section 7.

---

[1] Note 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.

2. Background and motivation

We start by briefly reviewing the IEEE 802.11 standard distributed coordination function (DCF) [10]. This is the MAC layer functionality that we will later extend in this paper.

2.1. 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) duration 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× 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 $2 \times$ SIFS. This prevents any node in the radio neighborhood of the transmitter or receiver transmit any packet until the ACK is transmitted. Fig. 2 illustrates the entire exchange mechanism.

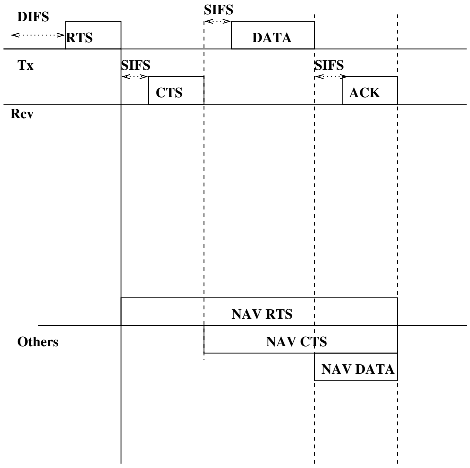


Fig. 2. Time line showing RTS–CTS based data exchange in IEEE 802.11 DCF.

Any node that did not receive the RTS/CTS correctly, because it was received with 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 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.

2.2. 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 [2]. 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 and 5 depending on the environment. The constant factor governing the above proportionality 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 [2] 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, σ^2 is the variance of the multipath, and $I_0(\cdot)$ is the modified Bessel function of the first kind and zero-order. The 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 the 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 $f_m = vf_c/c$, 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 [2]. Knowing N_R, the *average fade duration* (average duration for which the signal level is below a given level R) can be computed as,

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

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

Data presented in [11] for Doppler frequencies that can be encountered in practice[2] 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) and 2 m/s 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.

Common routing protocols in ad hoc networks focus 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]

---

[2] While data for only $f_m = 20$ Hz is presented in [11], the average fade duration for any f_m can be easily computed, given that the relationship between N_R and f_m is linear.

[3] Note 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.

Compare these average fade durations with the fact that it takes approximately 30 ms for the RTS retries to fail seven times causing the MAC to drop the frame. This is computed by using the interframe spacings and slot times from the standard specifications [10], 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.

3. 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. Assuming fading on different links lacks a high degree of correlation, it is unlikely that all links are in deep enough fade at the same time with 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.

3.1. Anycast extension for 802.11

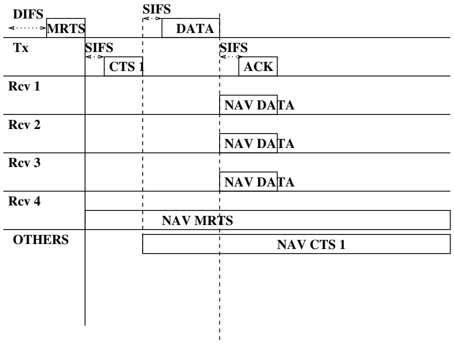
The anycast extension uses a similar handshaking protocol as in 802.11 DCF, but takes advantage of multiple receivers with the goal to transmit the frame to any one of them successfully. It can be thought of an anycasting scheme in the link layer. The routing layer computes multiple routes between the source and destination. We will describe this mechanism in the following 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 the 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 fewer 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.

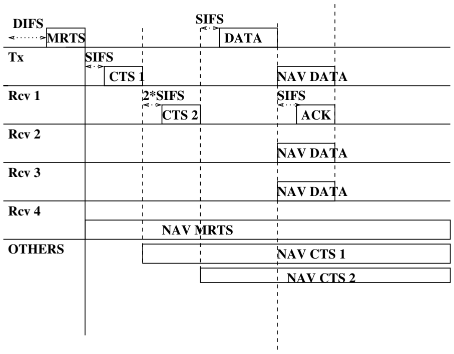
When an intended receiver receives the MRTS packet, it responds by a CTS. These CTS transmissions are staggered in time in order of their priorities. The first receiver in the order transmits the CTS after an SIFS, the second after a period equal to the time to transmit a CTS and 3× SIFS, and so on. See Fig. 3a–c for an illustration. Note that the staggering ensures that the CTSs are separated by at least 2× 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 of the ACK packet. (The DATA packet carries this period in the header just in case these receivers missed the MRTS). See Fig. 3a 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 (Fig. 3b); and when all but the fourth CTS were not received (Fig. 3c).

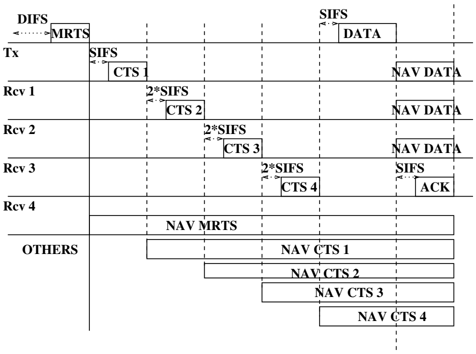
Any other node that hears the MRTS (*exposed* node), sets its NAV for the entire duration mentioned in the MRTS packet. This duration depends upon the number of receivers (which can be a maximum of four) to which MRTS is being sent. For instance, if the number of receivers is k, the NAV is set to $k \times \mathrm{CTS} + (2k+1) \times \mathrm{SIFS} + \mathrm{DATA} + \mathrm{ACK}$ 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 ith CTS, will set its NAV for the period $(2(k-i)+1) \times \mathrm{SIFS} + (k-i) \times \mathrm{CTS} + \mathrm{DATA} + \mathrm{ACK}$ (see Fig. 3a–c).



(a) 1st CTS is received.



(b) 2nd CTS is received.



(c) 4th CTS is received.

Fig. 3. Time line showing the anycast extension of 802.11.

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 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. The anycast mechanism in this manner increases the number of nodes that are exposed terminals and should therefore refrain from any communication. This can potentially reduce the network throughput. One way to cancel this NAV setup if the receiver is not involved in the communication is if the receiver sends explicit NAV cancellation messages. But, while the data is being sent to the last receiver, each of the other receivers would sense a busy channel and therefore they cannot engage in any transmission themselves. Thus, there is no easy way to resolve this problem. However, our simulation studies do show that even with large traffic diversity, anycast performs very well relative to 802.11. Thus, the harmful effect of silencing these nodes 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 [2]. 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.

3.2. 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 [9,12,6]. 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 [13]) to provide more traffic diversity and to reduce load on each individual route [9].

We will use an on-demand multipath routing protocol to provide the MAC layer with multiple next hop links. Specifically, we will use AOMDV [6], a multipath extension of a popular on-demand single path routing protocol AODV [14,15] 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 [6], 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 [6] 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 to 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.

AOMDV uses a sequence number-based method (similar to AODV) to prevent looping and also to eliminate stale routing information. AOMDV is flexible enough to provide disjoint (link- or node-disjoint) or overlapped routes. Naturally, allowing overlapped routes gives a large number of routes providing the protocol as many forwarding choices as possible at each hop. In prior work [6], however, we have explored disjoint path routing as the impact of fading was not analyzed, and links failed primarily because of mobility. This ensures that link failures most often are caused by mobility and thus they are not very transient. Thus, overlapped routes were not useful, as a single link failure may cause failure of many routes at the same time. In the following section simulation results will show that use of disjoint paths (i) bring down the overall performance of either protocol and (ii) the relative advantage of the multiple next hop extension almost vanishes. 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 tradeoff 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 [12] the authors also have noted that limiting the path length difference (l) is a useful optimization in multipath routing.

4. Applications for multichannel and directional antenna networks

It is well known that wireless networks have a limited bandwidth available for communication. This provides a motivation to study network designs which improve the bandwidth utilization. A popular approach is to use multiple channels for communication, known as multichannel networks. Another network model called directional antenna network, uses directional antennas so that the transmission is confined to selected directions with respect to the transmitter, instead of all directions as in regular (omni-directional) networks. Both these network types can potentially improve the bandwidth utilization by increasing the spatial reuse of the available bandwidth.

In multichannel and directional antenna networks just as in regular wireless networks, nodes suffer from deafness and hidden terminal problems. Deafness is said to have occurred when a node makes several futile attempts to communicate with a neighbor who is busy in another transmission and thus is unable to respond to the sender. The hidden terminal problem occurs when a node starts a transmission by incorrectly assuming that the medium is free when in reality there is an ongoing transmission in the neighborhood. The control packet exchange mechanism in 802.11 medium access control protocol (MAC), alleviates the deafness and hidden terminal problems in regular networks. This mechanism assumes a single channel network with omni-directional transmissions. Due to the inability of nodes to listen for transmissions in all directions or in all channels in directional antenna and multichannel networks, deafness and hidden terminal problems may be more rampant in these networks if the 802.11 protocol was used in the MAC layer. Earlier in this paper, anycast was proposed for single channel networks to combat multipath fading, where it was able to alleviate losses due to fading by exploiting path diversity. We will see now that by exploiting the same path diversity, anycast is able to alleviate the deafness problems in both multichannel and directional antenna networks.

4.1. Multichannel networks

While there can be many designs for a multichannel network, we have adapted a “quiescent channel” model that appeared in [16]. In this model, each node in the network is assigned a channel called a quiescent channel. This is the channel to which the node listens to when it is not in transmit mode. This channel assignment is well known to all nodes in the network or can be derived from the node addresses. All channels are used for data transmissions which in a resource constrained network that has a small number of channels, is a more desirable design. Given this network model, we will now describe the receiver directed scheme (RDT) [16], which is a simple adaptation of 802.11 in multichannel networks with the quiescent channel model. We will then use anycast mechanism with RDT to alleviate the deafness problems.

4.1.1. Receiver directed scheme

In RDT, in order to transmit a packet to the next hop receiver, the transmitting node must switch to the receiver’s channel and perform the CSMA/CA mechanism as in 802.11. If this backoff procedure is completed successfully and the medium is still free, the transmitter performs the RTS/CTS exchange with the receiver in that channel. All overhearing nodes invoke their virtual carrier sensing mechanisms. The virtual carrier sensing mechanism in RDT is achieved by maintaining different network allocation vectors for separate channels. Thus, the overhearing nodes set the NAV corresponding to the channel in which transmission is heard. We distinguish this NAV from the one in regular networks by renaming it as channel NAV or CNAV. Nodes cannot participate in any transmission on a channel as long as the CNAV for that channel is set, but at the same time, nodes are free to switch to and contend for another channel for which the CNAV is not set. This capability of parallel transmissions can potentially increase the network throughput by a large amount.

We note that due to the node's inability to listen to all channels at the same time, it may not have the current state of the channel it intends to transmit in. Thus, when a node switches to a new channel for transmission, it may inadvertently act as a hidden terminal causing collision for an ongoing transmission. Similarly, it can suffer from the deafness problem if the intended receiver happens to be busy in another transmission.

4.1.2. Anycast extension of RDT

The anycast mechanism is capable of alleviating the deafness problems in RDT by exploiting path diversity in the transmission channel. The multipath routing layer may be instrumented to maintain multiple paths on each channel in the network, and provide these node addresses to the MAC layer. Thus, in anycast, the transmitting node switches to the receivers' channel and multicasts a RTS packet to multiple potential next hop receivers in that channel and waits for a CTS. Reception of CTS from any one of the next hop nodes indicates that the channel has been reserved, thus, the transmitter sends data to the receiver from which it received CTS. In case the transmitter did not receive CTS from any next hop receiver, it retries up to six times.

We can see from the protocol description that, anycast would be more successful in alleviating the deafness problems, because it tries to negotiate medium access simultaneously with more than one next hop nodes. This parallel negotiation process greatly increases the probability of success. Note that, the multichannel anycast protocol is similar in principle to its single channel counterpart and thus we can use the same protocol stack without changes in the hardware in both networks.

4.2. Directional antenna networks

In the directional antenna networks, we assume that each node is equipped with a circular antenna array with eight directional elements that divide the entire azimuthal plane into eight 45° beams. (Choice of 45° is not necessary for the protocol to function. This is just representative of commercial directional antennas we have looked at [17].) Directional transmission is achieved by beamforming in the direction of transmission and directional reception is achieved by beamforming toward the angle of arrival of the strongest signal. We have simulated this beamforming for directional transmission by switching on only that antenna element which points toward the direction of transmission. Similarly, directional reception is simulated by turning off all antenna elements except one facing the direction of the strongest signal. Signals from all directions outside the beam width are ignored. We also consider negligible beam switching latency and do not model any back or side lobes. Furthermore, we have assumed that all eight elements can be switched on simultaneously to achieve omni-directional transmission and reception. The transmission radius is assumed to be same in both directional and omni-modes. This can be achieved by decreasing the transmission power by an appropriate factor when the antenna is in the directional mode.

Nodes are able to determine the direction of an incoming transmission by measuring the angle of arrival of the strongest signal. This information provides the relative direction of next hop neighbors and is cached at the routing layer along with the routes to various destinations. Having described the antenna model we will now proceed to discuss the directional virtual carrier sensing (DVCS) [18] protocol followed by the anycast extension.

4.2.1. Directional virtual carrier sensing

In DVCS, if a node is idle, it switches its antenna to omni-directional mode in which it can hear transmissions from all directions. When a node needs to transmit a unicast packet to a receiver, and it is aware of the direction of the receiver, it invokes the CSMA/CA mechanism during which it monitors the medium in the direction where the intended receiver is located. If the medium remains free during the backoff interval, the node beamforms its antenna to the direction of the receiver and transmits a RTS packet. If the receiver perceives a free medium in direction from where the maximum strength signal was received, it beamforms in that direction and transmits the CTS. A successful RTS/CTS exchange is followed by data/ACK exchange. Nodes that overhear RTS/CTS exchange must invoke their virtual carrier sensing mechanism. Nodes maintain separate network allocation vectors for different antenna sectors instead of a single vector. We distinguish this NAV from the NAV in 802.11 by naming it as directional NAV or DNAV. Thus, when making a decision to contend for the medium, nodes check if the DNAV for the direction of transmission is set. The node is free to contend for the medium in all directions for which the DNAV is not set.

When a node switches from directional transmission or reception mode to the omni-directional mode, it is possible that it has missed some control packet exchange that took place while it was in the directional mode. Thus, the node no longer has the current state of the medium. This may lead to the hidden terminal problem. Also while a node is busy in transmission or reception from a direction, a neighbor being unaware of this state might try to communicate with this node from a different direction. This is the well known deafness problem occurring in directional antenna networks. We will see in the following section how *anycast* is able to alleviate these problems.

4.2.2. Anycast extension of DVCS

In the anycast extension to DVCS, the routing layer may be instrumented to maintain different paths for different directions (antenna orientations) and provide multiple next hop options in a particular direction to the MAC layer. Thus, in anycast, the sender multicasts MRTS to multiple next hop neighbors in the same direction and waits for CTS in response. Upon receiving a CTS from any one of the receivers, the sender transmits data to that receiver. All overhearing nodes invoke their directional virtual carrier sensing mechanism just as in DVCS. If the sender does not get any CTS in response to its MRTS, it may retry up to six times.

We observe that, since there may be multiple next hop choices for forwarding the packet, the probability that at least one of the next hop nodes will be able to respond with a CTS is higher. Thus anycast, exploits path diversity to improve packet delivery and alleviate the deafness problem in DVCS. Once again we note that, the directional antenna version of anycast is quite similar to the omni-directional version as well as the multichannel version described earlier.

5. 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 [19] based simulations, that do not have any of these restrictions and can use elaborate scenarios.

5.1. Analysis for a grid network

Consider a two-dimensional grid network as in Fig. 4 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. Two 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 dark in Fig. 4. On these boundary nodes, only one 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 probability 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 probabil-

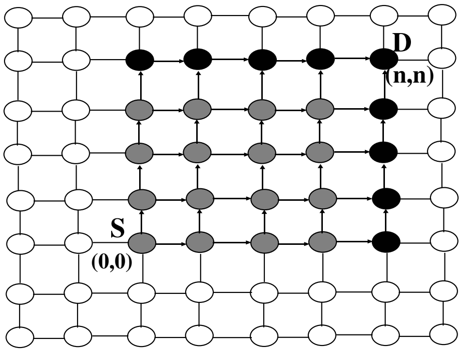


Fig. 4. Grid network for analyzing packet delivery probability.

ity at each hop is either p (if there is only one next hop) or p^2 (if there are two next hops). Note that two next hops are available for each of the first n hops; beyond this, the boundary nodes can provide only one 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 shops.

If a node (i,j) is at a distance l from S (i.e., the node is at the lth 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 $l \geqslant n$. 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 $j = (n-1), (n-2), \ldots, (l-n+1)$. This determines that the number of such paths going through *all* boundary nodes at hop $n \leqslant l < 2n$ 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 $n \leqslant l < 2n$ 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.

Fig. 5 plots the packet delivery probability P versus the path length ($2n$) for different link loss probabilities (p) for both single (unicast) and multiple next hop forwardings (anycast). Note that even though only a maximum of two 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.

5.2. Evaluation on experimental testbed

We implemented the anycast protocol on Berkeley motes platform, manufactured by Crossbow Technology [20,21]. While our original intention is

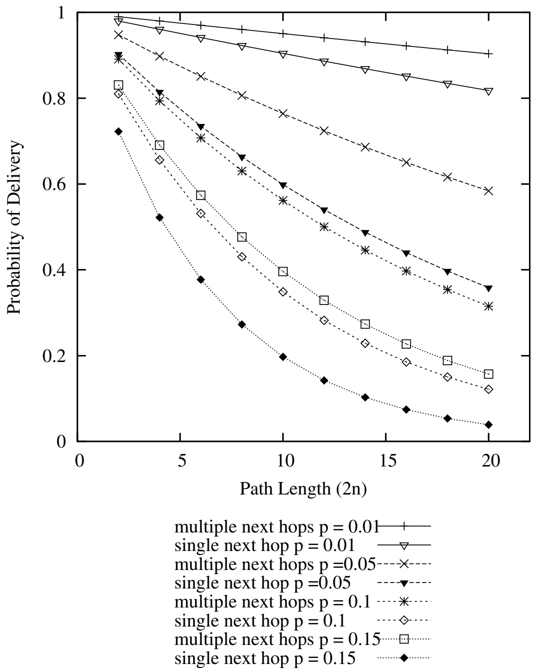


Fig. 5. Packet delivery probabilities for the grid network of Fig. 4 with single (unicast) and multiple next hop forwarding (anycast).

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 motes platform, where link layer protocols are implemented in software as a part of the protocol stack in the TinyOS operating system [22,21]. We did a proof of concept implementation in software using the TinyOS [22,21] platform on Mica motes. We used the Mica platform for our experiments that uses an Atmel ATMEGA series microcontroller (4 MHz, 8-bit) as the processor and an RFM TR1000 transceiver operating at 916 MHz 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 SMAC protocol stack [23,24] 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 [23,24] 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, back-offs 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 albiet 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 Fig. 4. 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 5.1). Fig. 6 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 experiments.[4] An average of a large number of experiments is reported in Fig. 6. 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.

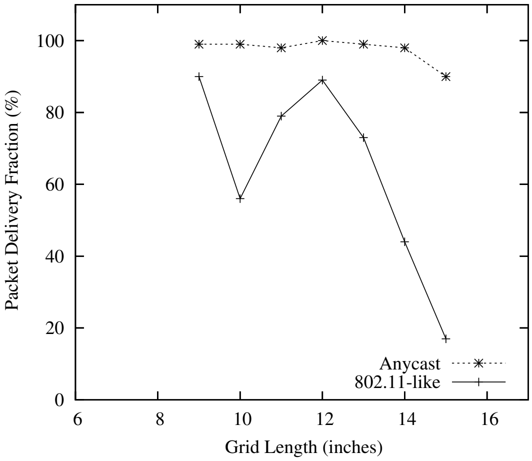


Fig. 6. Packet delivery fraction in the 4 × 4 Berkeley motes testbed with S-MAC protocol stack.

5.3. Simulation model

We used the *ns*-2 [19] simulator with the AOMDV protocol [6] in the routing layer and the anycast protocol in the MAC layer. As mentioned before, the AOMDV model used here allows over-

---

[4] We 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.

[5] We 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 in.) the performance was relatively poor, possibly due to some multipath effects created at these lengths.

lapped 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 pkt/s (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 [25] 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 [2] for the large-scale propagation model (as in the *ns*-2 distribution), augmented by a small-scale model modeling Ricean fading as presented in Section 2.2. The *ns*-2 extension provided by the authors of [3] 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 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 pregenerated dataset containing the components of the time-sequenced fading envelop.

The original implementation in [11] 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 [4] 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/s has been used to compute the Doppler shift f_{m}.

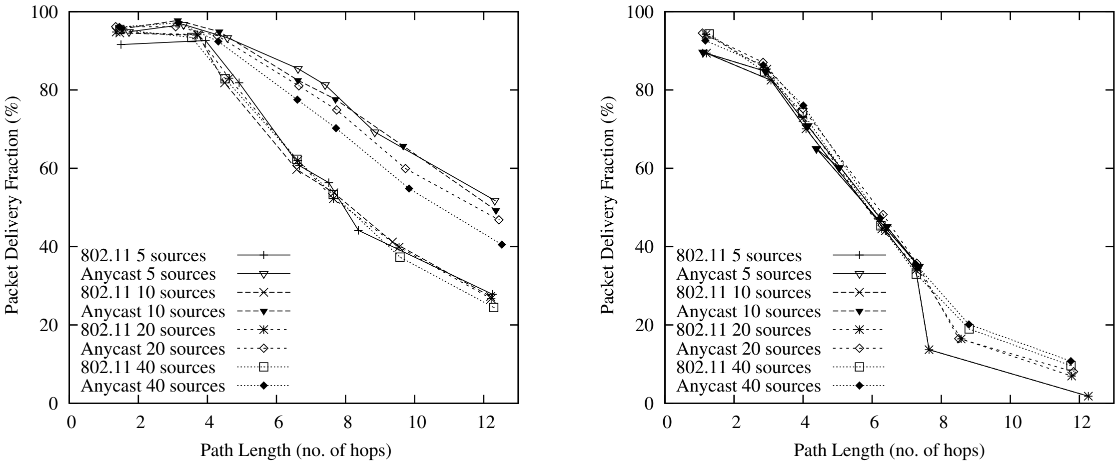
Three different network models have been used for evaluation each with 200 nodes and various number of traffic flows: The first model is a stationary grid network similar to Fig. 4. Here, the grid is, however, rectangular 40×5 with the distance between adjacent nodes in the grid being 100 m. Note that the nominal radio range (without fading) being about 250 m, 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 m × 500 m). Similar experiments were run by controlling the random choices of source destination pairs so that their shortest path lengths fall close to preselected specific values. The third model uses the same number of nodes in the same area; but now the 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 five runs.

5.4. Simulation results in grid, random and mobile networks

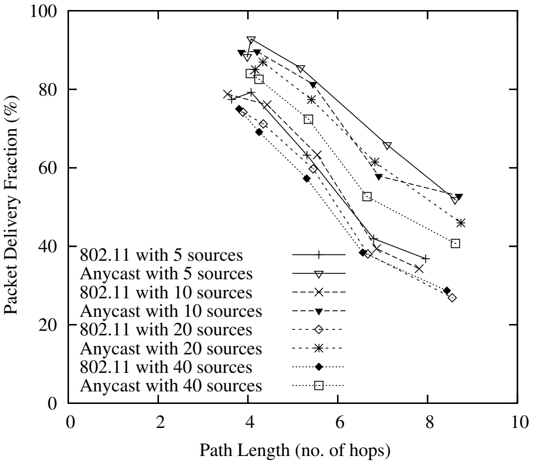
Fig. 7a plots the average packet delivery fraction for the stationary grid network model for the two link layer models. As expected, the delivery fraction goes down with increase in path lengths with anycast performing better – with the performance differential increasing with the path length. A performance 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



(a) Stationary grid network with overlapping path routing.

(b) Stationary grid network with disjoint path routing.



(c) Stationary random network with overlapping path routing.

Fig. 7. Packet delivery fraction with 802.11 and anycast in static networks.

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. Fig. 10 demonstrates this EFFECT, where the percentage of MRTSs that have one, two, three or four 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 three or four next hops. When routing is modified to restrict the routing to discover only link-disjoint paths, the performance improvement with anycast is almost non-existent. Fig. 7b demonstrates this. This figure uses the same simulation runs as before, only with a change in routing. We investigated the reason for the lack of performance gain with disjoint path routing. As

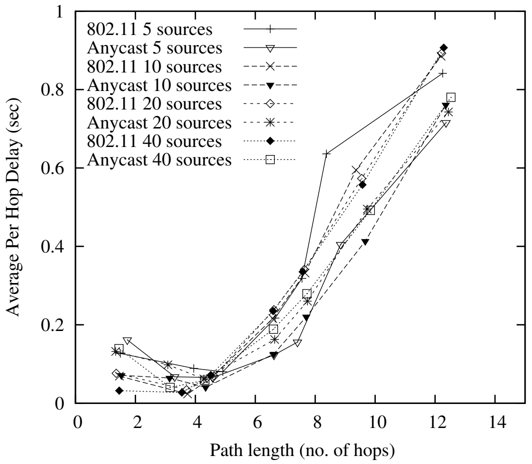


Fig. 8. Average per hop delay with 802.11 and anycast in static grid network with overlapping path routing.

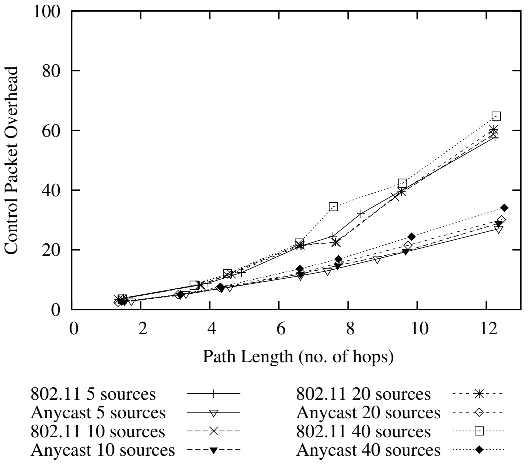


Fig. 9. Control packet overhead in 802.11 and anycast in static grid network with overlapping path routing.

alluded to before in Section 3.2, the major cause is lack of sufficient number of next hops. Fig. 11 confirms this hypothesis by comparing the fraction of unicast MRTSs (MRTSs with only one next hop) for these two variations. Note the large number of unicast MRTS for disjoint path routing relative to the overlapped paths case, showing that multiple next hops are not often available for disjoint path routing.[6] From this point onward, only overlapped path was used for routing.

Fig. 7c shows the packet delivery performance in the stationary random network. Note again that performance improvement varies from about 20% to up to about a factor of 2 for large path lengths. Because of the randomness involved the hop-wise 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. Table 1 shows packet delivery fraction for a specific set of scenarios with 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, we will look at mobile scenarios with different mobility. Fig. 12a presents the packet delivery performance in a mobile scenario with average speed of 20 m/s, respectively. Note that anycast is performing about about 25–40% relative to the unicast performance. In these set 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. Fig. 12b–d shows a scenarios in which average speed of each node is 15, 10 and 5 m/s, respectively. 802.11 delivers less that 60% of the packets at high mobility while anycast is able to deliver up to 75% of the packets.

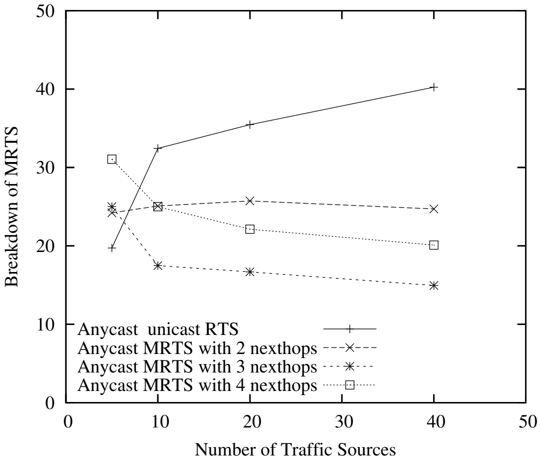


Fig. 10. Percentage of MRTS packets with different numbers of next hops in stationary grid network (average path length is approximately 6).

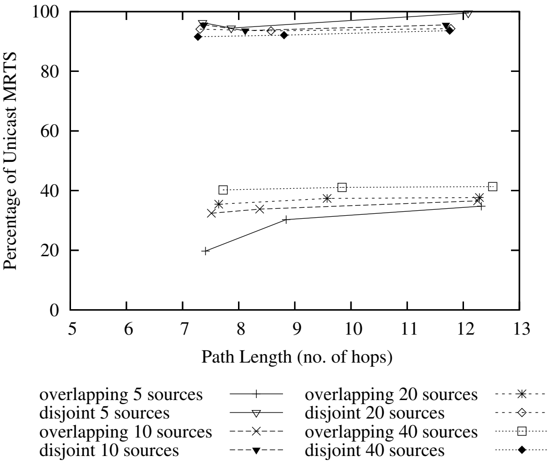


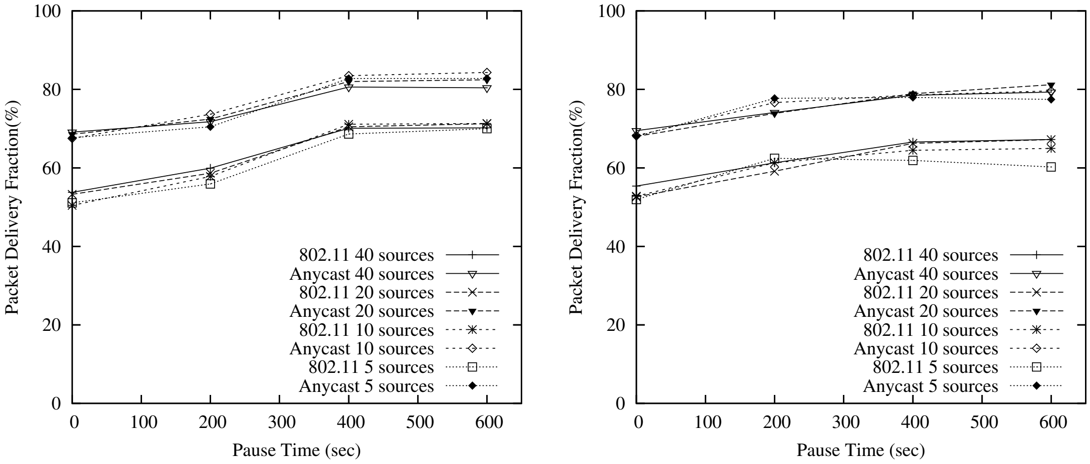
Fig. 11. Percentage of unicast MRTS packets in the stationary grid network for disjoint path and overlapping path routing.

---

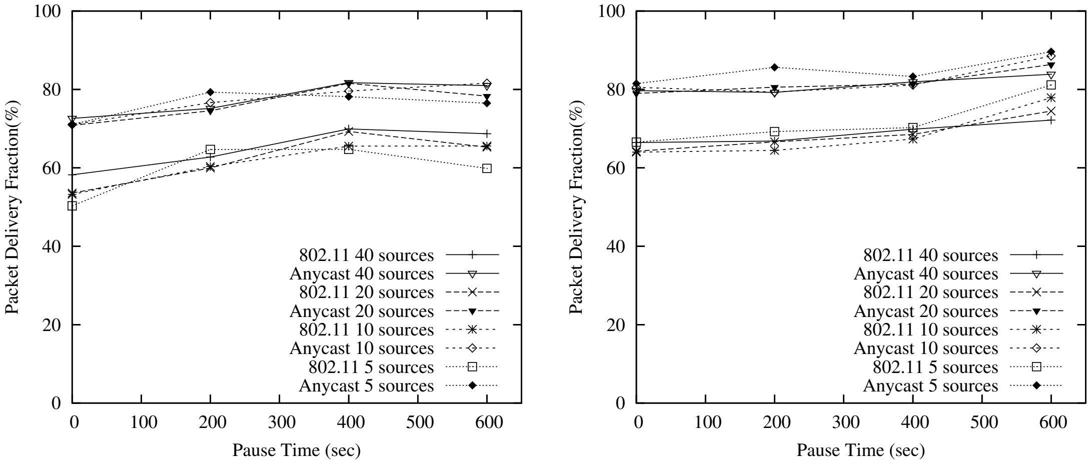
[6] It may appear that disjoint path routing means that only the source has more than one next hop and not any of the intermediate nodes. However, the protocol used here follows the disjoint path definition in [6] where a node I on the path P_1 from S and D is allowed to form an independent path P_2 to D which is link-disjoint from P_1.

Table 1
Effect of Ricean *K* factor on packet delivery fraction

Number of sources	Ricean *K* factor					
	5 db		10 db		15 db	
	802.11	Anycast	802.11	Anycast	802.11	Anycast
20	47%	64%	72%	82%	92%	95%
40	50%	66%	74%	86%	93%	96%



(a) average speed=20m/s (b) average speed=15m/s



(c) average speed=10m/s (d) average speed=5m/s

Fig. 12. Packet delivery fraction for 802.11 and anycast in mobile scenarios.

At 5 m/s, anycast delivers 80% of the packets while 802.11 is barely able to cross the 60% mark.

5.5. Comparison of overheads in anycast and 802.11

In this section we have presented results that compare overheads in the anycast and 802.11 protocols. We have observed from the analysis in Section 5.1 as well as the packet delivery fraction graphs in the previous section, that the benefits of anycasting is more prominent when hop-wise distance between the source and destination is longer than four hops. We can obtain a larger range of path lengths in the grid networks than in random or mobile networks where path lengths are difficult to control due to randomness and mobility. In order to compare the overheads of the two protocols over a large range of path lengths as well as for the sake of brevity, we will present the overhead results for static grid networks only. We have seen that the other scenarios also follow similar trends.

We have compared average per hop delays incurred by packets that were successfully received at the destination. This is computed as the ratio of the average delay incurred by the packets and the average number of hops traversed from the source to the destination. We observe in Fig. 8 that this delay in the anycast scheme is higher than in 802.11 when the paths are on an average less than four hops long. Thus for shorter path lengths simultaneous transmission to reach any next hop incurs more delay than the retry mechanism in 802.11. However, as path lengths increase, packets in the anycast mechanism have lower delay than in 802.11. At path length of approximately 12 hops, anycast has up to 12% lower delay than 802.11. This observation is generally true for other experiments, where we do not always show delay plots.

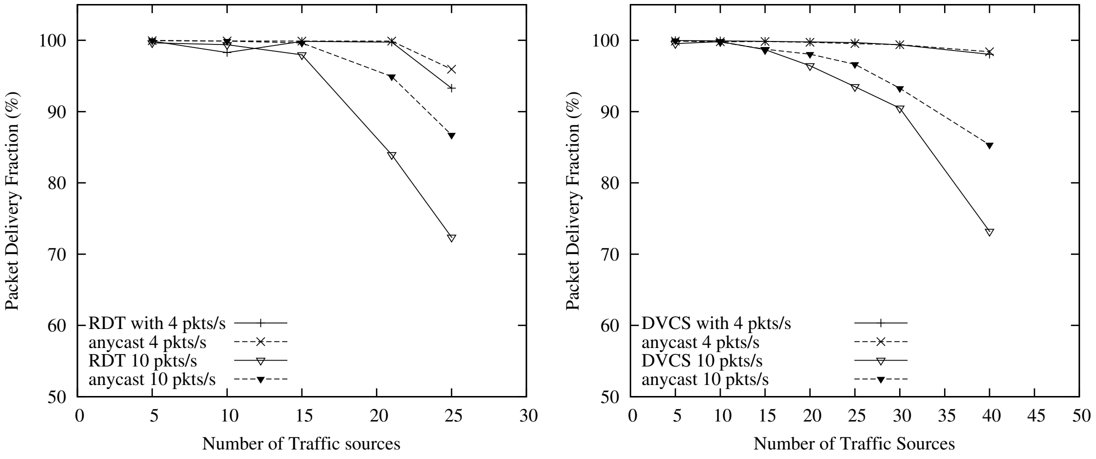
In anycast, the traffic due to additional CTS packets might cause additional overhead. In order to understand the effect of additional CTS in anycast, we will now compare the control overheads in the two protocols. We compute the control overhead as the ratio of the total number of RTS and CTS packets sent along the entire path from the source to the destination and the total number of data packets that are successfully received at the destination. We present the result in Fig. 9. As expected, the control overhead is low when the path length is small but it increases as the data packets have to be routed through more nodes to reach the destination. It is interesting to note that the control overhead in anycast is actually lower than that in 802.11 and as the path length increases, the difference becomes wider. In 12 hop paths, 802.11 sends more than 60 control packets for every data packet that reaches the destination, while anycast sends only around 30 control packets per data packet. Note that in an ideal scenario, for 12 hop paths, the number of control packets per data packet would be 24, two packets for each hop in the path. This result clearly shows that the multiple CTS transmissions in the anycast protocol presents a much lower overhead than the multiple RTS/CTS sent in 802.11 as it retries several times before succeeding in sending packets to the next hop node.

Our results establish the benefits of anycast in practical wireless networks that have far from ideal channel conditions. In wireless networks where the path lengths are larger than four hops, the anycast mechanism not only provides a higher packet delivery fraction but does so with lower packet delays and exchanges less number of control packets as compared to the 802.11 protocol.

5.6. Experiments with multi-channel and directional antenna networks

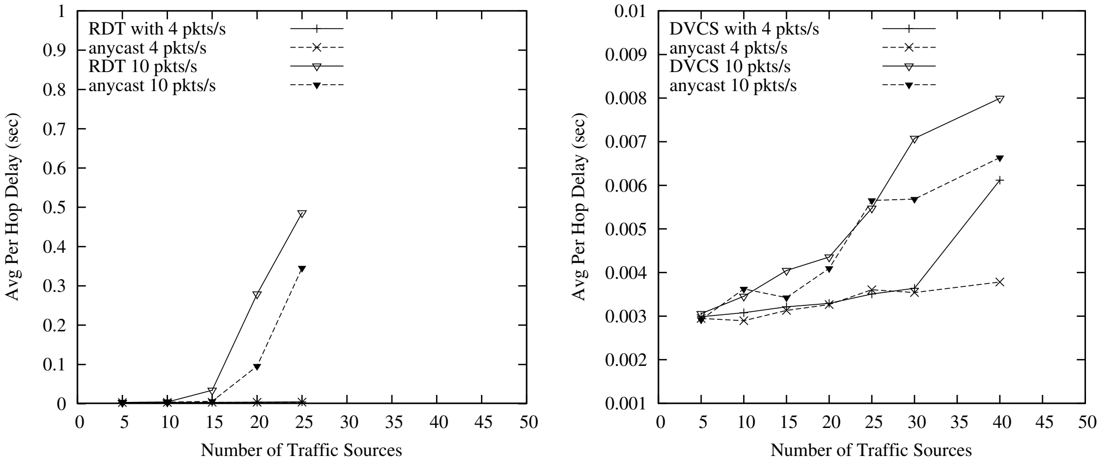
We implemented the multichannel and directional antenna protocols in the popular *ns*-2 simulator. We used multipath AODV in the routing layer with appropriate modification so that the routing layer can maintain separate paths for separate channels or directions. We performed experiments in a static scenario with 100 nodes placed randomly in a 1000×1000 m area. We ran experiments for different scenarios with 5, 10, 20, 25, 30 and 40 traffic connections and with data rates of 4 and 10 pkts/s and the packet size was 512 bytes. In the multichannel network experiments, there are three channels available for communication. Fig. 13a shows the graph of packet delivery fraction achieved by RDT and multichannel anycast protocols when the number of traffic connections is varied at two different rates (4 and 10 pkts/s). Similarly Fig. 14a shows the average per hop delay for the same scenario. The results clearly show how anycast outperforms RDT both in terms of delay and packet delivery fraction. As the number of traffic sources increases the difference between the two protocols constantly increases and at high load scenarios with 25 sources and 10 pkts/s, anycast delivers 88% packets while 802.11 delivers only 73%. This result clearly shows the advantage of anycast when the network load is high and the deafness problem is more prominent in multichannel networks.

Figs. 13b and Fig. 14b compare packet delivery fraction and average per hop delay for anycast and DVCS in directional antenna networks. We see that, anycast provides higher packet delivery fraction than DVCS and the performance difference increases with the network load. For example, anycast delivers 12% more packets to the destination and incurs 16% lower delay in the scenario with 40 sources each transmitting 10 pkts/s. These results confirm that anycast is more robust in high load scenarios where there are more possibilities of packet



(a) Multichannel network (b) Directional Antenna network.

Fig. 13. Packet delivery fraction vs number of traffic sources for anycast and 802.11 like protocols.



(a) Multichannel network. (b) Directional Antenna network.

Fig. 14. Average per hop delay vs number of traffic sources for anycast and 802.11 like protocol.

loss due to deafness problems in the directional antenna network.

6. Related work

In [26], 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 [4], 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 successfully delivered. The OAR protocol [3] 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 [27] 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 [5] 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 [28], which is very similar in spirit to the selection diversity forwarding work described earlier. Another protocol called GeRaF [29] also contains similar ideas, but has been specifically applied for geographic forwarding. Here, the interest is more on modeling, rather than a practical implementation.

Two recent studies [30,31] used a protocol similar to ours in spirit, however, for a different goal.[7] These protocols exploit multiuser diversity in the context of an access point-based system. Similar exploitation of multiuser diversity was also explored earlier in channel state based scheduling [32] protocols. In contrast, we exploit path diversity.

7. 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 3.2). Some mechanisms to do this on a per-packet basis has been discussed in [5].

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).

By anycasting the deafness problem in a multichannel or directional antenna network may be alleviated if not solved without the use of additional hardware or a separate control channel and even without synchronization requirement. Anycast can alleviate these problems by exploiting the availability of different routes to the destination. Thus, if one of the next hop nodes is “deaf”, another node may be able to route the data packet. Similarly, if a transmission is interrupted by a hidden terminal, the transmitter may be able to re-negotiate the channel with a different neighbor thereby, reducing the

---

[7] Note that both these papers refer to an earlier, tech report version of our work.

possibility of another collision. We have presented anycast in single channel, multiple channels and directional antenna networks. It is also possible to use the same protocol in hybrid networks containing all three features. Thus, unlike other protocols that were designed either for multichannel or for directional antenna networks, anycast is suitable for both types as well as single channel and omnidirectional networks.

References

[1] A. Ephermides, Ad hoc networks: not an ad hoc field anymore, wireless communications and mobile computing, Mobile Ad Hoc Networking: Research, Trends and Applications 2 (5) (2002) 441–448 (special issue).

[2] T. Rappaport, Wireless Communication: Principles and Practice, Prentice-Hall, 2002.

[3] B. Sadeghi, V. Kanodia, A. Sabharwal, 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), 2002, pp. 24–35.

[4] G. Holland, N. Vaidya, P. 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), 2001, pp. 236–251.

[5] R.R. Choudhury, N.H. Vaidya, Mac-layer any-casting in ad hoc networks, SIGCOMM Computing and Communications Review 34 (1) (2004) 75–80.

[6] M. Marina, S.R. Das, On demand multipath distance vector routing in ad hoc networks, in: Proceedings of the International Conference on Network Protocols (ICNP), 2001, pp. 14–23.

[7] W. Zaumen, J.J. Garcia-Luna-Aceves, Shortest multipath routing using generalized diffusing computations, in: Proceedings IEEE INFOCOM 98, 1998, pp. 1408–1417.

[8] A. Nasipuri, 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), Boston, 1999, pp. 64–70.

[9] M. Pearlman, Z. Haas, P. Scholander, S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, in: Proceedings of ACM MobiHoc 2000, pp. 3–10.

[10] I.E.E.E., Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11-1997.

[11] R. Punnoose, P. Nikitin, D. Stancil, Efficient simulation of ricean fading within a packet simulator, in: Proceedings of the IEEE VTC-Fall 2000, vol. 2, 2000, pp. 764–767.

[12] A. Nasipuri, R. Castaneda, S.R. Das, Performance of multipath routing for on-demand protocols in ad hoc networks, ACM/Kluwer Mobile Networks (MONET) Journal 6 (4) (2001) 339–349.

[13] N.F. Maxemchuk, Dispersity Routing, in: Proceedings of the IEEE ICC, 1975, pp. 41:10–41:13.

[14] C. Perkins, E. Royer, Ad hoc on-demand distance vector routing, in: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[15] R.R. Choudhury, N.H. Vaidya, Mac-layer any-casting in ad hoc networks, SIGCOMM Computing and Communications Review 34 (1) (2004) 75–80.

[16] N. Shacham, P. King, Architectures and performance of multichannel multihop packet radio networks, IEEE Journal on Selected Areas of Communication SAC-5 (6) (1987) 1013–1025.

[17] 802.11 Phocus Array Antenna System by Fidelity Comtech, http://www.fidelity-comtech.com/.

[18] M. Takai, J. Martin, R. Bagrodia, A. Ren, Directional Virtual Carrier Sensing for directional antennas in mobile ad hoc networks, in: ACM Mobihoc, 2002, pp. 39–46.

[19] K. Fall, K.V. (Eds.), *ns* Notes and Documentation, 1999. Available from http://wwwmash.cs.berkeley.edu/ns/.

[20] Crossbow technologies, inc. http://www.xbow.com.

[21] Tinyos community forum http://www.tinyos.net.

[22] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler, K.S.J. Pister, System architecture directions for networked sensors, in: Architectural Support for Programming Languages and Operating Systems, 2000, pp. 93–104.

[23] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: Proceeding of INFOCOMM, 2002, pp. 1567–1576.

[24] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated, adaptive sleeping for wireless sensor networks, ACM/IEEE Transactions on Networking 12 (3) (2004) 493–506.

[25] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multihop wireless ad hoc network routing protocols, in: Proceedings of the 4th International Conference on Mobile Computing and Networking (ACM MOBICOM'98), 1998, pp. 85–97.

[26] P. Larsson, Selection diversity forwarding in a multihop packet radio network with fading channel and capture, ACM SIGMOBILE Mobile Computing and Communications Review 5 (2001) 79–282.

[27] M. Pursley, H. Russell, J. Wysocarski, 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, vol. 2, 2001, pp. 967–971.

[28] S. Biswas, R. Morris, Opportunistic routing in multi-hop wireless networks, SIGCOMM Computing and Communications Review 34 (1) (2004) 69–74.

[29] M. Zorzi, R.R. Rao, Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance, IEEE Transactions on Mobile Computing, 2003, pp. 337–348.

[30] Z. Ji, Y. Yang, J. Zhou, M. Takai, R. Bagrodia, Exploiting medium access diversity in rate adaptive wireless LANs, in: MobiCom'04: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, ACM Press, New York, NY, USA, 2004, pp. 345–359.

[31] J. Wang, H. Zhai, Y. Fang, Opportunistic packet scheduling and media access control for wireless LANs and multi-hop ad hoc networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), vol. 2, 2004, pp. 1234–1239.

[32] P. Bhagwat, P. Bhattacharya, A. Krishna, S. Tripathi, Using channel state dependent packet scheduling to improve TCP

throughput over wireless LANs, ACM/BaltzerWireless Networks Journal (1997) 91–102.

Shweta Jain is currently a Senior Engineer at Staccato Communications Inc. San Diego CA. She received her MS and PhD in Computer Science from Stony Brook University, Stony Brook, NY, in 2005 and 2007. She received her B.E. degree in Electronics and Telecommunication engineering from Bengal Engineering and Science University in 2001. Her research interests are in wireless medium access protocols, focusing on system design and development for embedded wireless devices.

Samir R. Das is currently an associate professor in the Computer Science Department in the State University of New York at Stony Brook. He received his Ph.D. in Computer Science from Georgia Institute of Technology, Atlanta, in 1994. His research interests are in wireless networking and mobile computing, focusing on protocols, systems and performance evaluation. He received the U.S. National Science Foundation's CAREER award in 1998 and the best paper award in *ACM MobiSys* conference in 2007. He has been a speaker in the Distinguished Visitor program of the IEEE Computer Society during 2001–03. He co-chaired the technical program committee for the *ACM MobiHoc Symposium* in 2001 and *ACM MobiCom Conference* in 2004. He currently serves or has served on the editorial board of the *IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, ACM/Kluwer Wireless Networks Journal and Ad Hoc Networks* journal. More information about him and his research can be found at http://www.cs.sunysb.edu/~samir.