Congestion Adaptive Routing in Mobile Ad Hoc Networks

Duc A. Tran, *Member*, *IEEE*, and Harish Raghavendra

Abstract—Mobility, channel error, and congestion are the main causes for packet loss in mobile ad hoc networks. Reducing packet loss typically involves congestion control operating on top of a mobility and failure adaptive routing protocol at the network layer. In the current designs, routing is not congestion-adaptive. Routing may let a congestion happen which is detected by congestion control, but dealing with congestion in this reactive manner results in longer delay and unnecessary packet loss and requires significant overhead if a new route is needed. This problem becomes more visible especially in large-scale transmission of heavy traffic such as multimedia data, where congestion is more probable and the negative impact of packet loss on the service quality is of more significance. We argue that routing should not only be aware of, but also be adaptive to, network congestion. Hence, we propose a routing protocol (CRP) with such properties. Our ns-2 simulation results confirm that CRP improves the packet loss rate and end-to-end delay while enjoying significantly smaller protocol overhead and higher energy efficiency as compared to AODV and DSR.

Index Terms—Ad hoc networks, routing protocols, mobile computing, congestion adaptivity.

1 Introduction

According to a recent Gartner Group (www.gartner.com) report in February 2004, the North American mobile data market will grow to 141.1 million connections in 2007, with a compound annual growth rate of 41.7 percent. It is believed that a large portion will be ad hoc connections, which will open many opportunities for MANET applications. To prepare for this promising future, besides other issues, routing is an important problem in need of a solution that not only works well with a small network, but also sustains efficiency and scalability as the network gets expanded and the application data gets transmitted in larger volume. Though essential, routing in MANETs is a nontrivial matter. Since mobile nodes have limited transmission capacity, they mostly intercommunicate by multihop relay. Multihop routing is challenged by limited wireless bandwidth, low device power, dynamically changing network topology, and high vulnerability to failure, to name just a few.

To answer those challenges, many routing algorithms in MANETs were proposed. There are different dimensions to categorize them: *proactive* routing versus *on-demand* routing, or *single-path* routing versus *multipath* routing. In proactive protocols [5], [20], routes between every two nodes are established in advance even though no transmission is in demand. This is realized by a node periodically updating its neighbors with the routing information it has known thus far, hoping that every node eventually has a consistent and up-to-date global routing information for the entire network. This approach is not suitable for large networks because many unused routes still need be maintained and the periodic updating may incur overwhelming processing and communication overhead. The on-demand approach (e.g., [2], [9], [15], [21]) is more efficient in that a route is discovered only when needed for a transmission and released when the transmission no longer takes place. However, when a link is disconnected due to failure or node mobility, which often occurs in MANETs, the delay and overhead due to new route establishment may be significant. To address this problem, multiple paths to the destination may be used as in multipath routing protocols (e.g., [13], [16], [17], [28], [29]). An alternate path can be found quickly in case the existing path is broken. The tradeoff, as compared to single-path routing, is the multiplied overhead due to concurrent maintenance of such paths. Furthermore, the use of multiple paths does not balance routing load better than single-pathing unless we use a very large number of paths (which is costly and therefore infeasible) [30].

There is another dimension for categorizing routing protocols: *congestion-adaptive* routing versus *congestion-unadaptive* routing. The existing routing protocols belong to the second group. In this paper, we propose a new routing protocol that belongs to the first group. We name the proposed protocol CRP (*C*ongestion-adaptive *R*outing *P*rotocol). We note that some of the existing protocols are congestion-aware (e.g., [12], [15]), but they are *not* congestion-adaptive. In congestion-aware routing techniques, congestion is taken into consideration only when establishing a new route which remains the same until mobility or failure results in disconnection. In congestion-adaptive routing, the route is adaptively changeable based on the congestion status of the network.

Our motivation is that congestion is a dominant cause for packet loss in MANETs [15]. Typically, reducing packet loss involves congestion control running on top of a mobility and failure adaptive routing protocol at the network layer.

• *The authors are with the Department of Computer Science, University of Dayton, 300 College Park, Dayton, OH 45469.*
E-mail: {duc.tran, raghavhz}@notes.udayton.edu.

Manuscript received 24 Nov. 2004; revised 20 May 2005; accepted 6 Aug. 2005; published online 26 Sept. 2006.
Recommended for acceptance by K. Nakano.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0278-1104.

Routing may let a congestion happen which is later detected and handled by congestion control. Congestion nonadaptiveness in routing in MANETs may lead to the following problems:

- **Long delay**: It takes time for a congestion to be detected by the congestion control mechanism. In severe congestion situations, it may be better to use a new route. The problem with an on-demand routing protocol is the delay it takes to search for the new route.
- **High overhead**: In case a new route is needed, it takes processing and communication effort to discover it. If multipath routing is used, though an alternate route is readily found, it takes effort to maintain multiple paths.
- **Many packet losses**: Many packets may have already been lost by the time a congestion is detected. A typical congestion control solution will try to reduce the traffic load, either by decreasing the sending rate at the sender or dropping packets at the intermediate nodes or doing both. The consequence is a high packet loss rate or a small throughput at the receiver.

The above problems become more visible in large-scale transmission of traffic intensive data such as multimedia data, where congestion is more probable and the negative impact of packet loss on the service quality is of more significance. Unlike well-established networks such as the Internet, in a dynamic network like a MANET, it is expensive, in terms of time and overhead, to recover from a congestion. Our proposed CRP protocol tries to prevent congestion from occurring in the first place. CRP uses additional paths (called "bypass") to reduce packet delay, but tries to minimize bypass use to reduce the protocol overhead. Traffic is split over the bypass and the primary route probabilistically and adaptively to network congestion. Hence, 1) power consumption is efficient because traffic load is fairly distributed and 2) congestion is resolved beforehand and, consequently, CRP enjoys a small packet loss rate. These advantages of CRP are verified in our ns-2 [19] based performance study.

The remainder of this paper is organized as follows: Next, we present the protocol details of CRP. We provide the results of our simulation study in Section 3. In Section 4, we discuss works related to routing in MANETs and how CRP is different from them. The paper is concluded in Section 5.

2 Congestion Adaptive Routing

Congestion Adaptive Routing (CRP) is a congestion-adaptive unicast routing protocol for MANETs. We introduced its preliminary concepts and evaluation in [22] and [27], respectively. In this paper, we present a complete design with more insight and an in-depth evaluation for this routing protocol. In CRP, every node appearing on a route warns its previous node when prone to be congested. The previous node uses a "bypass" route for bypassing the potential congestion area to the first noncongested node on the primary route. Traffic is split probabilistically over these two routes, primary and bypass, thus effectively lessening the chance of congestion occurrence. CRP is on-demand and consists of the following components:

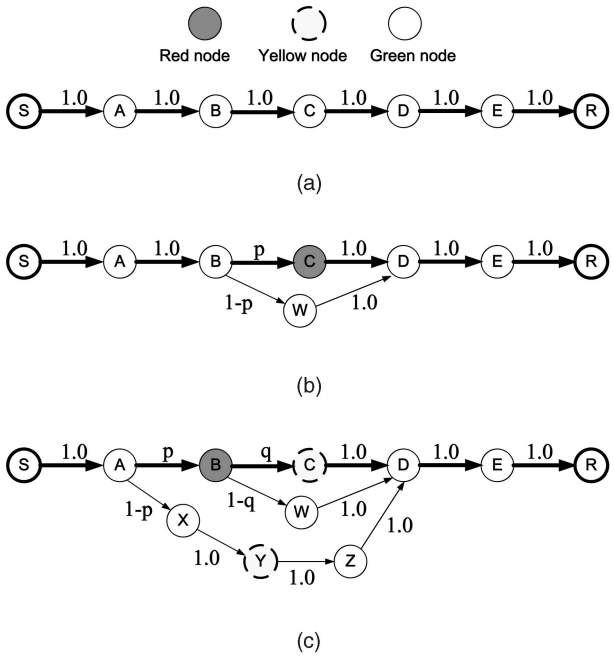


Fig. 1. Use of bypass to reduce congestion.

1. congestion monitoring,
2. primary route discovery,
3. bypass discovery,
4. traffic splitting and congestion adaptivity,
5. multipath minimization, and
6. failure recovery.

We start with an example to get used to the concept of "bypass" and then discuss those constituent components in detail.

2.1 Example

A simplified example is illustrated in Fig. 1. A route $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow R$ is initially found for the sender S to the receiver R. This route is called the "primary route" from S to R. Each link is labeled with a probability that the incoming packet is forwarded along this link. In Fig. 1a, every packet follows the primary route. Some time later, node C detects that a congestion is likely to occur and sends a warning to its neighborhood. Its previous node (node B, regarding destination R) is aware of this situation. In response, node B finds a route bypassing C. This route is destined for D, which is the first noncongested node after C, as shown in Fig. 1b. All the primary nodes other than B and D are not included. Traffic coming to B will be spread over the primary link $B \rightarrow C$ and the bypass route $B \rightarrow W \rightarrow D$ with probabilities p and $1 - p$, respectively. Effectively, since the traffic coming to C is lessened, C will less likely become a congestion spot. Similarly, shown in Fig. 1c is the case where B is nearly congested. In response to B's warning, A finds a bypass to D which is the first noncongested node after B. Traffic coming to A will be split over the primary link $A \rightarrow B$ and the bypass route $A \rightarrow X \rightarrow Y \rightarrow Z \rightarrow D$. Node B thus improves its congestion status.

TABLE 1
Primary Routing Table

Attribute	Description
dst	destination
hop	next node on the primary route
hop_status	congestion status of the next primary hop
prob	probability to forward a packet to the primary link
bypass_dst	destination of the bypass route
bypass_hop	next node on the bypass route
bypass_status	congestion status of the bypass route
green_hop	next node on the primary route that is green
green_metric	distance to green_hop in hops

Intuitively, bypass can be used in routing to prevent congestion. Technically, there are important issues not mentioned in this example that need to be addressed. The next subsection discusses congestion monitoring at a node.

2.2 Congestion Monitoring

When the number of packets coming to a node exceeds its carrying capacity, the node becomes congested and starts losing packets. We can use a variety of metrics at a node to monitor congestion status. For instance, we can be based on the percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets timed out and retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion. The design of CRP allows it to work on top of any of these methods. For ease of presentation and as a proof of concept, we adopt the following simple method as an example in the paper. A node periodically checks the occupancy of its link-layer buffer. The congestion status is determined based on the ratio r between the the number of packets currently buffered to the buffer size. A node is said to be "green" (i.e., far from congested), "yellow" (i.e., likely congested), or "red" (very likely or already congested) if $r \leq 1/2$ (buffer is less than half full), $r \in (1/2, 3/4]$ (buffer is more than half full but far from being full), or $r \in (3/4, 1]$ (buffer is close to or already full), respectively. As later discussed, a bypass is a path from a node to its *next green node*. The next green node is the first "green" node at least two hops away downstream on the primary route.

2.3 Primary Route Discovery

The sender discovers the route to the receiver in a simple way. It broadcasts an REQ packet toward the receiver. The receiver responds to the first copy of REQ by sending back an REP packet. The REP will traverse back the path that the REQ previously followed. This path becomes the *primary route* between the sender and the receiver. Nodes along this route are called *primary nodes*.

Each node has two routing tables: primary table (denoted as $prTab$) and bypass table (denoted as $brTab$). $prTab$ is used to direct packets on the primary route, while $brTab$ directs packets on bypass routes. Thus, $brTab = \emptyset$ for a node that does not appear on a bypass route of any connection. We will revisit $brTab$ later in Section 2.4. For $prTab$, the main attributes are described in Table 1. An entry in $prTab$ is unique to a destination node {*dst*}. We denote by $prTab[N, R]$ the entry for destination R in the routing table of node N and by $prTab[N, R].attr$ the value for attribute $attr$. For instance, the entry for node A regarding destination R in Fig. 1c is $prTab[A, R]$ = (R, B, red, p, D, X, yellow, D, 3).

To reduce traffic due to primary route discovery and better deal with congestion in the network, we employ two strategies. First, an REQ is dropped if arriving at a node with a "red" congestion status. Second, if an REQ for receiver R arrives at a node N, this node may already have an entry for R as a result of a previous connection establishment. In this case, N just needs to forward REQ to $prTab[N, R].hop$—the next primary node of N toward destination R. Broadcasting is avoided and, therefore, we do not send too much traffic over the network.

On receipt of an REP initiated by destination R, a node N adds a new entry for R, or replaces the old entry with the new one, into $prTab$. The entry for R is removed if no data packet destined for R arrives or $prTab[N, R].hop$ is not heard of after a certain timeout period.

2.4 Bypass Discovery

A primary node periodically broadcasts a UDT (update) packet with TTL = 1. The UDT packet contains the node's congestion status and a set of tuples [*destination R, next green node G, distance to green node m*], each for a destination appearing in the primary routing table. This packet is created by *Procedure createUDT*, shown below:

1. Procedure *createUDT* at node N
2. output: packet p = (s, set of [R, G, m])
3. s = current congestion status of N
4. for (each destination R in $prTab[N]$)
5. N_{next}: next primary node
6. if (N_{next} is green OR $N_{next} = R$)
7. $G = N_{next}$
8. m = 2
9. else
10. $G = prTab[N, R].green\_hop$
11. $m = prTab[N, R].green\_metric$ + 1
12. add [R, G, m] to packet p

The purpose is that, when a node N receives a UDT packet from its next primary node N_{next} regarding destination R, N will be aware of the congestion status of N_{next} and learn that the next green node of N is G, which is m hops away on the primary route. This information is crucial in case a bypass is needed. The primary table is updated accordingly, as shown in *Procedure recvUDT* below:

1. Procedure *recvUDT* at node N
2. input: packet p = (s, set of [R, G, m])
3. from: node N'
4. for (each destination R in p)
5. if (N' is the next primary node of N regarding R)
6. $prTab[N, R].hop\_status = s$
7. $prTab[N, R].green\_hop = G$
8. $prTab[N, R].green\_metric = m$

An exception applies to the node immediately before destination R, where $prTab[., R].green\_hop = R$, $prTab[N, R].hop\_status =$ "green," and $prTab[., R].green\_metric = 1$. For

TABLE 2
Bypass Routing Table

Attribute	Description
dst	final destination
bypass_src	start node of bypass route
hop	next node on bypass route
status	congestion status of bypass route

every other node, $prTab[., R].green\_hop$ is set to -2 initially, meaning that this information is not yet available.

Let us suppose that a node N receives a UDT packet from its next primary node N_{next} (regarding a destination R). If N_{next} is yellow or red, a congestion is likely ahead if data packets continue to be forwarded on link $N \rightarrow N_{next}$. Since CRP tries to keep congestion from occurring in the first place, N starts to discover a bypass route toward node G—the next green node of N known from the UDT packet. For this purpose, N broadcasts a BPS_REQ packet destined for G. The next green hop G may receive multiple instances of BPS_REQ, but responds only to the earliest by sending a BPS_REP to N. This BPS_REP follows backward the path that the BPS_REQ traveled earlier. This path is the bypass route that N will use.

Since the distance to G should be short, the BPS_REQ is set with TTL = $2 \times m$ to limit broadcast traffic. (Here, m is the distance from N to G on the primary route and α is a small constant.) The broadcast traffic is reduced even more because BPS_REQ is dropped if arriving at a node (neither N nor G) already present on the primary route. As a result, the bypass path is disjoint with the primary route, except that they join at the end nodes N and G. It is possible that no bypass is found due to the way that BPS_REQ approaches G. In this case, we continue using the primary route. However, the work in [6] shows that the chance to find a "short-cut"[1] from a node to another on a route is high. Our bypass is more flexible than and not necessarily a short-cut and, therefore, the probability for finding a bypass is even higher.

As mentioned before in Section 2.3, a node maintains $brTab$—a bypass routing table to direct packets along bypass paths. This table, whose main attributes are described in Table 2, is updated upon receipt of a BPS_REP. Each entry in $brTab$ is identified by a tuple {destination *dst*, bypass source *bypass_src*}. Referring to Fig. 1, the entry for node X in $brTab$ regarding destination R is (R, A, Y, yellow). Similarly to $prTab$, an entry of $brTab$ is removed if, after a certain timeout period, no data packet corresponding to this entry arrives or the next bypass node is not heard of.

Bypass routing tables are only used by bypass nodes (e.g., nodes X, Y, Z, W in Fig. 1c). For the begin node of a bypass, say node A in Fig. 1c, when the BPS_REP comes back, A will assign $prTab[A, R].bypass\_dst = D$—the node that initiates BPS_REP and $prTab[A, R].bypass\_hop = X$—the first bypass node.

1. A short-cut from a node to another on a route is a shorter path between these two nodes not crossing the main route.

2.5 Traffic Splitting and Congestion Adaptability

Now that the bypass at a node has been found, data packets coming to this node are not necessarily spread over the bypass and the primary link. Indeed, as long as the next primary node is not red, no packet is forwarded on the bypass. This is because the primary route is still far from congested and we do not want to impose any unnecessary burden on the bypass nodes. We find the bypass proactively for we can use it immediately if the next primary node becomes red (indicating severe congestion).

Let us consider a node N on the primary route from sender S to receiver R and assume that a bypass from N to the bypass destination $prTab[N, R].bypass\_dst$ is currently maintained but unused. When the next primary node of N (i.e., node $prTab[N, R].hop$) first becomes red, incoming packets will follow primary link $N \rightarrow prTab[N, R].hop$ with a probability $p = prTab[N, R].prob = 0.5$ and follow bypass link $N \rightarrow prTab[N, R].bypass\_hop$ with an equal chance $(1 - p = 0.5)$. Hence, this traffic splitting effectively reduces the congestion status at the next primary node.

To adapt with congestion due to network dynamics, the probability p is modified periodically based on the congestion status of the next primary node and the bypass route. The congestion status of a bypass is the accumulative status of every bypass nodes. For instance, in Fig. 1c, the status of the bypass route $A \rightarrow X \rightarrow Y \rightarrow Z \rightarrow D$ is yellow. The basic idea is that we should increase the amount of traffic on the primary link if the primary link leads to a less congested node and reduce otherwise. The probability adjustment policy is described in Table 3 and explained below:

1. Next primary link is **green**: The primary link is not congested and we can increase p to better utilize this link (we can remove the bypass when no data is forwarded there and, so, save maintenance cost). However, we have to be conservative in increasing p because if we increase it too much, the link may become congested soon. For this reason, we increase p by $(1 - p)/4$ (25 percent of the gap between p and 1) when the bypass congestion status is **green**. When the bypass status is **yellow**, we should increase p by more to help relax the traffic on the bypass; in this case, we increase p by 33 percent of the gap between p and 1. If the bypass congestion status is **red**, we increase p by 50 percent of the gap between p and 1 to significantly reduce bypass traffic.
2. Primary link is **yellow**: We should not change p for the primary link if the bypass status is either **green** or **yellow**. If we increase p, the primary link may become red. If we decrease p, the bypass may become yellow or red, which does not improve the overall situation much. However, when the bypass status is **red**, p will be increased to avoid more losses on the bypass. This increase is conservative (25 percent of the gap between p and 1) to prevent the primary link from becoming red.
3. Primary link is **red**: The primary link is congested and, therefore, p is decreased to make it less congested. The decrease is more significant (50 percent of the gap between p and 1) if the bypass

TABLE 3
Splitting Probability Adjustment for Congestion Adaption

Congestion	bypass status = **green**	bypass status = **yellow**	bypass status = **red**
next primary node is **green**	$p := p + (1-p)/4$	$p := p + (1-p)/3$	$p := p + (1-p)/2$
next primary node is **yellow**	p unchanged	p unchanged	$p := p + (1-p)/4$
next primary node is **red**	$p := p - (1-p)/2$	$p := p - (1-p)/4$	find another bypass

status is **green** because the bypass has been in good condition. If the bypass status is much, the decrease is more conservative (25 percent) to prevent the bypass from becoming red. If the bypass is **red** (congested), we need to find another uncongested bypass to shoulder some traffic for the primary link.

We now work on an example demonstrated by Fig. 2. We start from the top diagram, where, currently, the bypass from A is $A \rightarrow X \rightarrow Y \rightarrow C$, from B is $B \rightarrow Y \rightarrow Z \rightarrow E$, and from D is $D \rightarrow W \rightarrow F$. Note that the bypass from B is destined for E because, earlier, both C and D were not green. D now is green and C becomes red. Since B is aware of C becoming red, B adjusts the probability to forward on the primary link $B \rightarrow C$ to $0.5 - 0.5/2 = 0.25$ because the bypass congestion status is green. Similarly, if node D learns that its bypass becomes yellow, it will change the probability to forward on link $D \rightarrow E$ to $0.4 + 0.6/3 = 0.6$. The new probability values are shown in the middle diagram. Now, supposing that node Y becomes red and nodes C and W remain green, the new probability values on link $A \rightarrow B$, $B \rightarrow C$, and $D \rightarrow E$, shown on the bottom diagram, are changed to $0.8 + 0.2/4 = 0.85$ (because we assume node B is currently yellow), $0.25 + 0.75/2 = 0.625$ (because we assume node C is currently green), and $0.6 + 0.4/2 = 0.7$, respectively.

Although a bypass and the primary route cannot include more than two common nodes, we allow different bypass paths to share nodes. For instance, bypass node Y in Fig. 2 belongs to two bypass paths. The rationale is that enabling bypass paths to share nodes increases the chance to discover a bypass. A bypass node may become too congested if it has to carry large loads of bypass traffic. This case is handled by our congestion adaptation explained above: This node will eventually be removed. We will discuss this kind of problem in the following section.

2.6 Multipath Minimization

To keep the protocol overhead small, CRP tries to minimize the use of multiple paths. If $prTab[N, R].prob$ approaches 1.0 (e.g., within a predefined threshold ϵ), this means the next primary node is far from congested or the bypass route is very congested. In this case, N removes the bypass. If $prTab[N, R].prob$ approaches zero, this means that the next primary node is very congested. In this case, the primary link is disconnected and the bypass becomes primary. These two extreme cases are illustrated in Fig. 3. In either case, all the bypass nodes are informed of the decision and their routing tables are modified accordingly.

To further reduce the use of multipathing and keep the protocol simple, CRP does not allow a node to use more than one bypass. Therefore, the bypass route discovery is only initiated by a node if no bypass currently exists at this node. The protocol overhead for using bypass is also reduced because of short bypass lengths. A bypass connects to the first noncongested node after the congestion spot, which should be just a few hops downstream.

2.7 Failure Recovery

A desirable routing protocol should gracefully and quickly resume connectivity after a link breakage. CRP is able to do so by taking advantage of the bypass routes currently available. There are three main cases of failures and we address them below. For ease of presentation, we consider

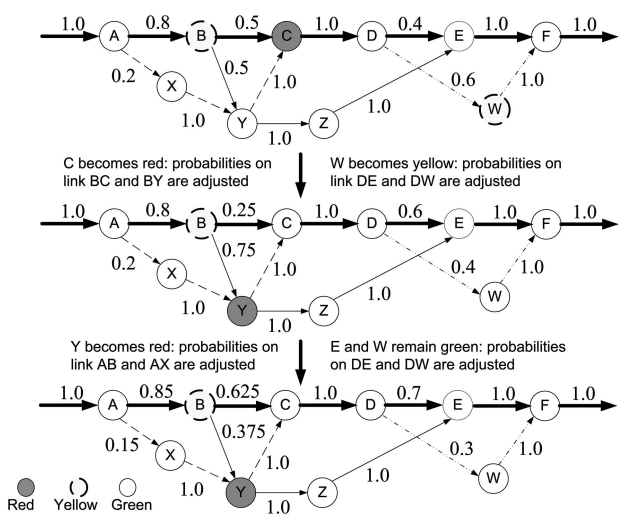


Fig. 2. Examples of splitting probability being adjusted adaptively to congestion.

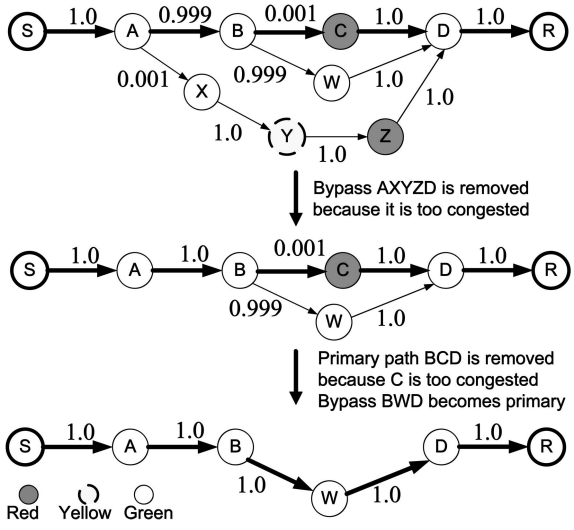


Fig. 3. Extreme cases of congestion adaptivity: Bypass removal and bypass-to-primary switch.

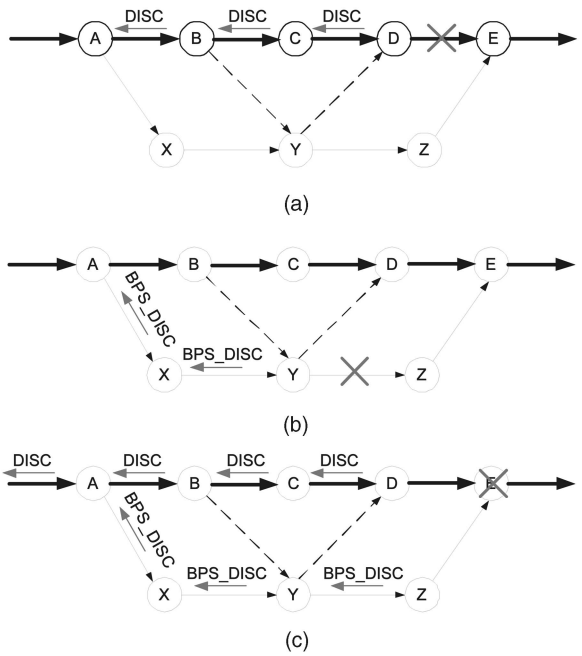


Fig. 4. Failure recovery: Bypass from B is $B \rightarrow Y \rightarrow D$ and from A is $A \rightarrow X \rightarrow Y \rightarrow Z \rightarrow E$. (a) Link DE on primary route fails: primary path ABCDE is removed, bypass BYD is removed, bypass AXYZE becomes primary. (b) Link YZ on bypass fails: bypass AXYZE is removed. (c) Node E on primary route fails: A forwards DISC upstream.

only one connection with sender S and receiver R. Fig. 4 demonstrates these cases, where the bypass from B is $B \rightarrow Y \rightarrow D$ and from A is $A \rightarrow X \rightarrow Y \rightarrow Z \rightarrow E$.

2.7.1 A Primary Link Fails (Fig. 4a)

When a primary link fails, say link $D \rightarrow E$, its initial node sends a DISC packet upstream toward the sender along the primary route. DISC records the nodes it visits. DISC stops at nodes that have a bypass. If a node receives the DISC (node B) and finds that its bypass destination (node D) is included in DISC, the bypass cannot be used because it leads a node before the failed link. In this case, DISC is further forwarded upstream. Eventually, DISC will stop at a node (node A) with a bypass whose destination (node E) is after the failed link or DISC will reach the sender S. In the latter case, S finds a new primary route to R as in Section 2.3. If the former case occurs, the bypass will be used as the primary route.

2.7.2 A Bypass Link or Bypass Node Fails (Fig. 4b)

This case is handled simply. The bypass node (node Y) that detects this failure sends a BPS_DISC packet upstream along the bypass until it reaches a primary node (node A). The bypass will be removed. Note that, in Fig. 4b, the bypass $B \rightarrow Y \rightarrow D$ is not removed because the BPS_DSC only visits nodes that correspond to {destination R, bypass source A}.

2.7.3 A Primary Node Fails (Fig. 4c)

This case is considered as a combination of the two cases above. If a primary node detects this failure, e.g., node D, it sends a DISC upstream along the primary route. If a bypass node detects this failure (node Z), it sends a BPS_DISC upstream along the bypass until reaching a primary node. When a primary node with a bypass (node A) receives DISC, it waits a period to see if a BPS_DISC is coming. If BPS_DISC comes, the bypass is removed and DISC will be forwarded upstream along the primary route. From then onward, it is handled similarly to the first case above. If BPS_DISC does not arrive within the waiting period, the bypass will be used as the primary route. However, it is possible that BPS_DISC comes late. In this case, it will be ignored, but the bypass that we just converted into primary actually remains broken at the failed node. This failure will be detected and fixed shortly because another DISC packet will be sent back (from node Z), as in Case 1.

3 Performance Study

We implemented CRP using the Network Simulator Ns-2 version 2.27 [19] with the CMU Monarch wireless extensions [1]. We compared CRP's performance to that of DSR and AODV, two of the most popular MANET routing protocols. We present our observations in this section.

3.1 Simulation Configuration

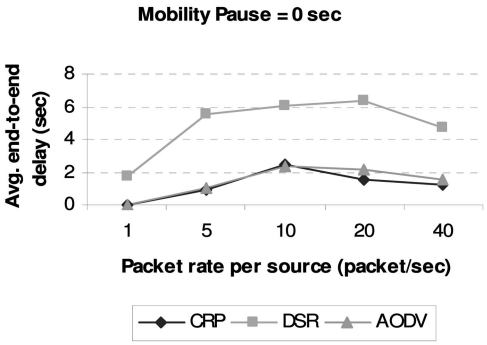
The network consisted of 50 nodes in a 1,500m $\times$ 300m rectangular field. The radio model used was Lucent's WaveLAN radio interface whose nominal bit rate is 2 Mbps and radio range 250 m. The MAC layer was based on IEEE 802.11 DCF (distributed coordination function). The channel propagation model we used combines both the free-space and 2-ray ground reflection models. The same configuration parameters were used as in Ns-2 version 2.27. An interface queue at the MAC layer could hold 50 packets before they were sent out to the physical link. Link breakage was detected from MAC layer feedbacks. A routing buffer at the network layer could store up to 64 data packets. This buffer keeps data packets waiting for a route, such as packets for which route discovery had started but no reply arrived yet.

We used the random waypoint mobility model [10]. Lu et al. [15] suggested that, by setting the maximum node speed to 4m/s, we could cover most mobility effects by just varying the pause period. Therefore, we used this maximum speed and considered different pause periods: zero second (highest mobility), 300 seconds (high mobility), 600 seconds (low mobility), and 900 seconds (zero mobility). For each of these cases, two simple ways can be used to illustrate different traffic loads: 1) fix the packet rate and vary the number of connections or 2) fix the number of connections and vary the packet rate. We employed the latter approach. Each simulation run lasted 900 seconds, during which 20 connections were generated and remained open until the simulation ended. For each connection, the source generated 512-byte data packets at a constant bit rate (CBR). This rate was varied among 1, 5, 10, 20, or 40 packets/s. The threshold ϵ used in CRP for multipath minimization is set to 0.1.

3.2 Performance Metrics

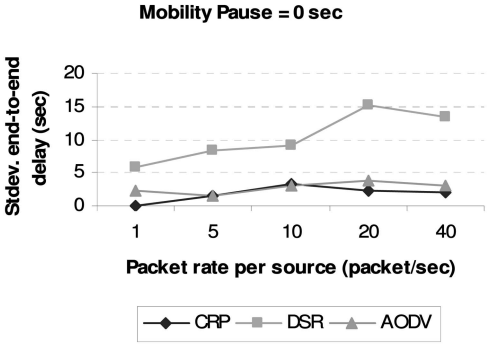
We considered the following important metrics for the evaluation:

1. **Packet Delivery Ratio**: Percentage of data packets received at the destinations out of the number of data packets generated by the CBR traffic sources.



Rate	CRP/DSR	CRP/AODV
1	0.02	0.85
5	0.17	0.94
10	0.41	1.04
20	0.24	0.69
40	0.26	0.81

Fig. 5. Pause 0: Average end-to-end delay.



Rate	CRP/DSR	CRP/AODV
1	0.01	0.01
5	0.18	1.04
10	0.37	1.14
20	0.14	0.58
40	0.14	0.67

Fig. 6. Pause 0: Uniformity end-to-end delay.

2. **End-to-End Delay**: The accumulative delay in data packet delivery due to buffering of packets, new route discoveries, queuing delay, MAC-layer retransmission, and transmission and propagation delays.
3. **Normalized Routing Overhead**: The ratio of the amount in bytes of control packets transmitted to the amount in bytes of data received.
4. **Normalized Power Consumption**: The ratio of the amount in bytes of both control and data packets transmitted to the amount in bytes of data received.

A desirable routing protocol should offer a high packet delivery ratio, small end-to-end delay, small routing overhead, and low power consumption.

3.3 Simulation Results

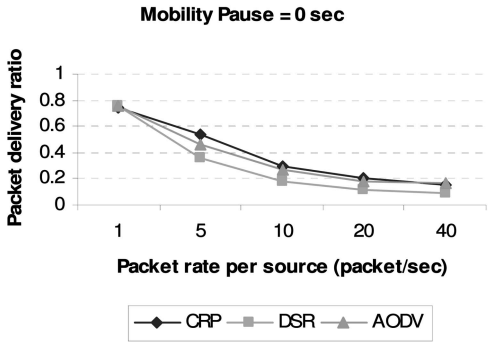
The results were collected as average values over 10 runs of each simulation setting. In what follows, the numeric results are demonstrated in figures. Each figure is associated with a table showing the ratio between CRP's performance measure to that of DSR and AODV. For instance, the second column of the table in Fig. 5 shows the ratio between CRP's average packet delay to DSR's average delay, while the third column shows the ratio between CRP's average delay to AODV's average delay.

3.3.1 Highest Mobility

Nodes move continuously in this simulated network, where packets are lost or dropped not only because of congestion but also mobility. Fig. 5 and Fig. 6 show the average delay and the delay standard deviation, respectively. DSR suffered the worst delay in all measures. This is because every data packet in DSR carries the entire route information, thus making the network severely congested. CRP and AODV resulted in less congestion because neither data nor control packets need to include the entire route information. They offered similar average delay and delay deviation. CRP was slightly better than AODV when packets were sent at a high rate (20 or 40 packets/sec). For instance, when the packet rate was 20, the average delay and delay deviation of CRP were only 69 percent and 58 percent of that of AODV.

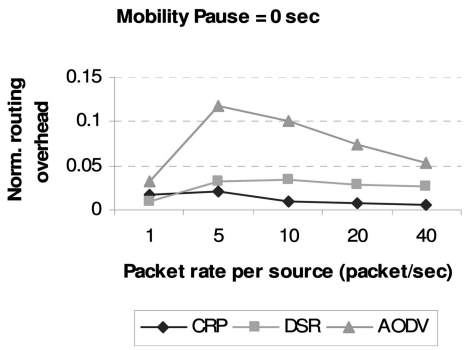
In regard of data packet delivery ratio (Fig. 7), both AODV and CRP outperformed DSR. Packets lost due to congestion in DSR were more than in the other protocols. When the packet rate was small (1 packet/s), CRP and AODV delivered similar loads of packets. This was because network traffic was not yet heavy. But, if more data were transmitted from the source, CRP delivered more successfully. Indeed, CRP successfully delivered at least 11 percent more data than AODV when the packet rate was 5, 10, or 20 packets/s. CRP and AODV converged to a similar performance when the rate was too high (40 packets/s).

The difference between CRP and AODV in terms of delay and delivery ratio may not seem significant, but, in comparing their routing overheads, CRP was clearly better as shown in Fig. 8. CRP was much more lightweight than AODV in all scenarios. When the traffic load was small (1 packet/s), CRP's



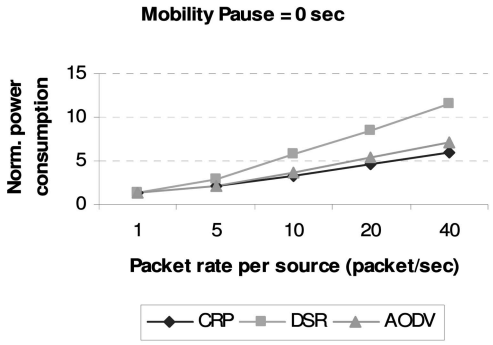
Rate	CRP/DSR	CRP/AODV
1	0.98	0.99
5	1.50	1.14
10	1.67	1.11
20	1.75	1.14
40	1.82	0.94

Fig. 7. Pause 0: Data packet delivery ratio.



Rate	CRP/DSR	CRP/AODV
1	1.80	0.53
5	0.63	0.17
10	0.29	0.10
20	0.27	0.10
40	0.24	0.12

Fig. 8. Pause 0: Protocol overhead.



Rate	CRP/DSR	CRP/AODV
1	0.97	0.98
5	0.73	0.96
10	0.57	0.86
20	0.54	0.85
40	0.52	0.84

Fig. 9. Pause 0: Energy efficiency.

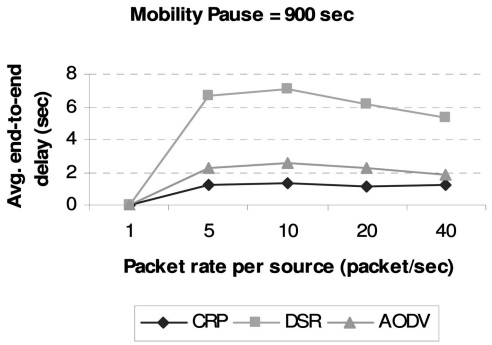
routing overhead was only half of that incurred by AODV. More impressively, when the traffic was heavier, the routing overhead of CRP was no more than 17 percent of that of AODV. The reason is as follows: Upon link breakage, while AODV tried to establish a new route to the destination by broadcasting a route request, CRP tried to make use of an available bypass. Therefore, route requests were sent less often in CRP. One could argue that it costs some overhead to maintain bypass paths in CRP. However, this overhead is kept small because of the way we minimize the use of multiple paths, as discussed in Section 2.6. DSR incurred the least routing overhead when the traffic was light, but, when more packets were generated into the network, the overhead of CRP was 64 percent of DSR and less than 30 percent in more stressful network scenarios.

Energy efficiency is crucial to any MANET. AODV and CRP were both competitive and more efficient than DSR. The gap between the first two became more noticeable in densely loaded networks, when DSR consumed even greater power (Fig. 9). As mentioned earlier, CRP carried less control traffic. Therefore, the fact that CRP's energy efficiency was higher than AODV's implies that CRP forwarded less data traffic. Interestingly, the amount of data received in CRP was no worse than in AODV. This finding convinced us that being adaptive to congestion helps increase both the effectiveness and efficiency of routing.

3.3.2 Zero Mobility

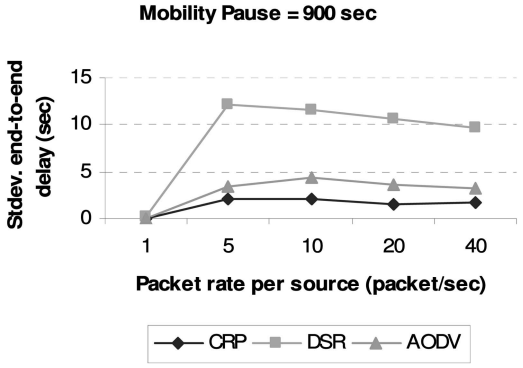
By setting the pause period to 900 seconds, loss in the network was only due to channel error and network congestion, not mobility. Therefore, since CRP distributes congestion in the network dynamically with the use of bypass paths, we expected that CRP would offer the best delay. This hypothesis indeed tested true in our simulation. The average delay and delay deviation are shown in Fig. 10 and Fig. 11, respectively. It is understandable, as explained above, that DSR suffered the worst delay. Between CRP and AODV, in contrast to their similarity in the case of highest mobility, the delay gap between them was much more noticeable in steady networks. CRP outperformed AODV not only in terms of average delay but also in delay deviation. In most traffic scenarios, CRP's data packets arrived at their destinations with an average end-to-end delay about half the delay if we use AODV. Not only that, the delay deviation in CRP was only 40 percent-60 percent of the deviation in AODV. These significant improvements make CRP more suitable than AODV and DSR for real-time and multimedia applications. There was one case (1 packet/sec rate) where CRP's delay was longer than AODV's and DSR's. However, the difference was negligible because all of these delays were very small; most data packets arrived instantly.

Fig. 12 shows the results for the data delivery ratio. In a steady network with little traffic, all protocols performed well when 80 percent of packets were delivered successfully. However, CRP delivered more as the packet rate was increased: at least twice and 24 percent more than DSR and AODV did, respectively, when the packet rate was 20 packets/sec or 40 packets/sec. Obviously, no matter which



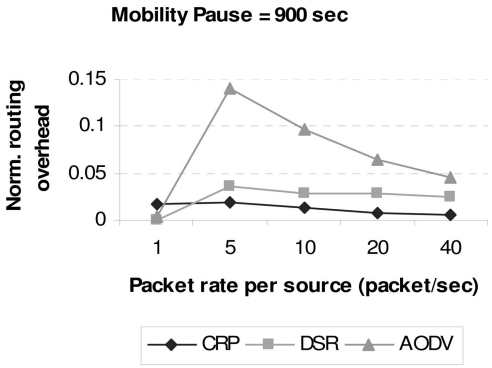
Rate	CRP/DSR	CRP/AODV
1	1.62	1.66
5	0.19	0.56
10	0.18	0.52
20	0.18	0.51
40	0.23	0.65

Fig. 10. Pause 900: Average end-to-end delay.



Rate	CRP/DSR	CRP/AODV
1	0.13	1.32
5	0.18	0.61
10	0.18	0.47
20	0.14	0.41
40	0.18	0.52

Fig. 11. Pause 900: Uniformity of end-to-end delay.



Rate	CRP/DSR	CRP/AODV
1	23.1	3.73
5	0.53	0.14
10	0.45	0.13
20	0.30	0.13
40	0.27	0.14

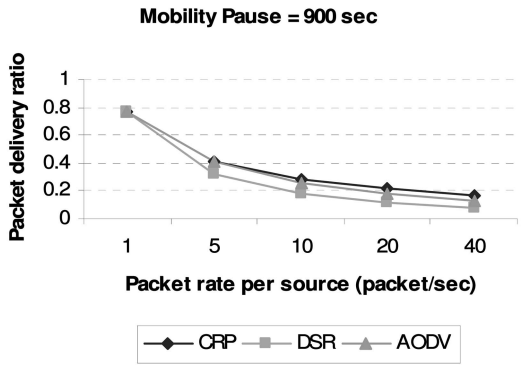
Fig. 13. Pause 900: Protocol overhead.

protocol was the best, a delivery rate of about 20 percent was considered too low to be acceptable. However, our simulation was intended to consider highly stressful networks and our purpose was to show that CRP still performed much better than AODV and DSR in such networks.

Similarly to the case of highest mobility, CRP consistently outperformed both the other protocols in terms of routing overhead and energy efficiency. AODV incurred the heaviest routing overhead, whereas CRP required the least and at most twice as much as the overhead of AODV and DSR (see Fig. 13). CRP seemed unaffected by increasing traffic. Again, resolving congestion by predicting its occurrence and adaptively distributing it over the primary and bypass paths was the reason why the routing overhead of CRP changed very little. We observed that the overhead decreased when the packets were generated too fast. This is because, in this case, congestion occurred early on the delivery path and just a few nodes participated in the recovery process; hence, not many control packets were sent out. Overall, CRP was very lightweight because it required a routing overhead only 13 percent-14 percent of that required by AODV and 27 percent-53 percent of that required by DSR in most scenarios. CRP was the most energy-efficient in steady networks, too, as shown in Fig. 14. The improvement of CRP over the other protocols especially grew larger as more traffic was injected into the network.

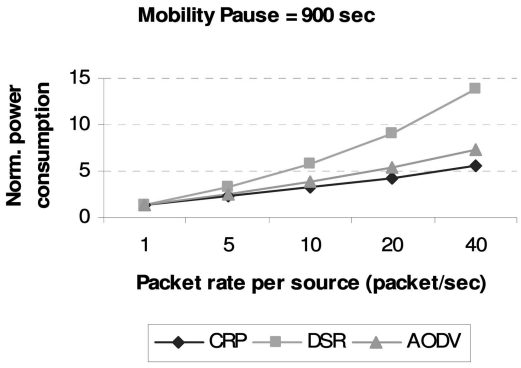
3.3.3 Other Levels of Mobility

We ran simulation with 300-second and 600-second pause periods to experience different levels of mobility. The differences in performance between the three protocols are



Rate	CRP/DSR	CRP/AODV
1	0.99	0.99
5	1.32	1.00
10	1.64	1.11
20	2.02	1.23
40	2.38	1.29

Fig. 12. Pause 900: Data packet delivery ratio.



Rate	CRP/DSR	CRP/AODV
1	0.96	0.99
5	0.72	0.93
10	0.58	0.86
20	0.47	0.79
40	0.40	0.76

Fig. 14. Pause 900: Energy efficiency.

TABLE 4
Improvement of CRP over DSR and AODV: Pause Period = 300 Seconds

	Avg. delay		Delivery ratio		Routing overhead		Energy	
Rate	CRP/DSR	CRP/AODV	CRP/DSR	CRP/AODV	CRP/DSR	CRP/AODV	CRP/DSR	CRP/AODV
1	0.99	**1.37**	**0.98**	**0.99**	**2.58**	0.66	0.98	0.99
5	0.22	0.76	**0.98**	**0.72**	0.43	0.12	0.96	**1.30**
10	0.28	0.67	1.67	1.12	0.31	0.10	0.57	0.86
20	0.30	0.75	1.85	1.17	0.26	0.11	0.51	0.83
40	0.33	0.78	1.84	1.21	0.33	0.15	0.52	0.81

TABLE 5
Improvement of CRP over DSR and AODV: Pause Period = 600 Seconds

	Avg. delay		Delivery ratio		Routing overhead		Energy	
Rate	CRP/DSR	CRP/AODV	CRP/DSR	CRP/AODV	CRP/DSR	CRP/AODV	CRP/DSR	CRP/AODV
1	0.91	**1.74**	**0.99**	**0.79**	**4.87**	**1.15**	0.97	0.99
5	0.27	0.72	1.24	1.02	0.49	0.12	0.77	0.93
10	0.33	0.74	1.70	1.17	0.31	0.09	0.56	0.82
20	0.27	0.78	2.08	1.24	0.28	0.10	0.46	0.79
40	0.26	0.72	2.32	1.28	0.28	0.14	0.40	0.76

shown in Table 4 for 300-second pause and Table 5 for 600-second pause. The results were consistent with our findings in the two previous cases (pause 0 second, pause 900 second) that CRP was indeed a more desirable routing protocol than AODV and DSR in traffic-intense networks. In these two tables, **bold** numbers indicate when CRP was worse than the others. Such occurrences, however, were rare and mostly took place when the traffic was light, in which most data packets were delivered successfully, nearly instantly, and with little control overhead and power consumption.

3.3.4 Comparison Remarks

The following highlights were concluded from our performance evaluation:

- End-to-end delay: Consistently in simulation runs, CRP provided an average delay shorter than did AODV and DSR. In addition, delay standard deviation was smaller in CRP than in the other protocols, making CRP more suitable for real-time and multimedia applications.
- Data packet delivery ratio: Both CRP and AODV successfully delivered more data packets than DSR. However, when the network was heavily loaded, whether the network was steady or highly mobile, CRP performed better than AODV. In the other cases (only a few), they performed similarly.
- Protocol overhead: Both CRP and DSR were more lightweight than AODV. CRP was significantly better when the network traffic became heavier.
- Energy efficiency: CRP and AODV were consistently better than DSR. CRP was more efficient than AODV, especially when the network traffic was heavier.

4 Related Work

Congestion is a dominant reason for packet drops in ad hoc networks [15]. Lu et al. [15] found that AODV is ineffective under stressful network traffic situations. They therefore proposed a modified version of AODV (called CADV) which favors nodes with short queuing delays in adding into the route to the destination. While this modification may improve the route quality, the issues of long delay and high overhead when a new route needs to be discovered remain unsolved. Furthermore, CADV is not congestion adaptive. It offers no remedy when an existing route becomes heavily congested. This is probably the reason that CADV improves AODV in delivery ratio by only 5 percent in highly loaded networks. (CRP improves by 10 percent-28 percent.) A dynamic load-aware routing protocol (DLAR) was proposed in [12]. DLAR is similar to CADV, the difference being that a node with low routing load is favored to be included in the routing path during the route discovery phase.

CADV, DLAR, as well as most on-demand routing protocols, are single-pathing. Multipath protocols may be used to shorten the delay due to new-route discoveries. Some of these protocols are multipath versions of existing on-demand single-path protocols, such as [13], [16] (extensions to AODV) and [18] (extension to DSR). Another multipath protocol, named MDVA, was proposed in [29]. MDVA operates proactively and requires heavy overheads and, therefore, it may not perform as well as an on-demand protocol does for MANETs.

Recently, [28] proposed CHAMP, a cache-based on-demand multipath routing protocol. CHAMP balances network routing load better than other on-demand multipath protocols because it sends packets on multiple paths simultaneously in a round-robin manner. CRP is similar to CHAMP in that CRP also sends packets on both bypass paths and primary routes simultaneously. However, CRP distributes incoming traffic over the bypass and primary routes dynamically based on the current network congestion situation. Congestion is subsequently better resolved. In addition, CHAMP only works effectively if storage space is available for caching packets, which is not a requirement in CRP. Another feature of CRP is that bypass paths have short lengths. Since a bypass is established from a node to the next noncongested node on the primary route, it is not costly to maintain and not time-consuming to discover. On the contrary, an alternate path in other multipath routing

schemes is longer because it is destined all the way for the destination.

To reduce the routing length, [6] proposed a routing optimization technique called SHORT. SHORT monitors existing routing information and detects situations where a subpath can be replaced with a short-cut. SHORT can also be adopted to reduce energy consumption or optimize residual battery power. Similarly to SHORT, a neighborhood-aware source routing protocol NSR was proposed in [25]. In NSR, a shortcut is found between a node and a two-hop neighbor and used when a link breakage occurs. The concept of "shortcut" is similar to our "bypass" concept in that they both do not cross the main routing path. However, because a bypass is more flexible and not necessarily a shorter path (like a shortcut), it is more likely to be found. Moreover, unlike shortcuts, bypass paths are used in parallel with the primary route, thus better dealing with traffic congestion. Nevertheless, since SHORT was designed as an optimization technique, it can absolutely be used with CRP to better improve the routing efficiency.

Other routing protocols such as secure-aware routing [8], [32], gossip routing [7], geographic routing [11], power-aware routing [4], [24], [26], and reliable routing [31] were also proposed. Surveys on different routing protocols and their performance comparisons can be found in [3], [14], [23].

5 Conclusions

We have proposed CRP, a congestion-adaptive routing protocol for MANETs. CRP enjoys fewer packet losses than routing protocols that are not adaptive to congestion. This is because CRP tries to prevent congestion from occurring in the first place, rather than dealing with it reactively. A key in CRP design is the bypass concept. A bypass is a subpath connecting a node and the next noncongested node. If a node is aware of a potential congestion ahead, it finds a bypass that will be used in case the congestion actually occurs or is about to. Part of the incoming traffic will be sent on the bypass, making the traffic coming to the potentially congested node less. The congestion may be avoided as a result. Because a bypass is removed when the congestion is totally resolved, CRP does not incur heavy overhead due to maintaining bypass paths. The bypass maintenance cost is further reduced because a bypass is typically short and a primary node can only create at most one bypass. A short end-to-end delay is also provided by CRP. Indeed, since CRP makes the network less congested, the queuing delay is less. Furthermore, since recovery of a link breakage is realized gracefully and quickly by making use of the existing bypass paths, the delay due to new-route establishment is also low. Our ns-2-based simulation has confirmed the advantages of CRP and demonstrated a significant routing and energy efficiency improvement over AODV and DSR.

References

[1] CMU, "The CMU Monarch Project: Wireless and Mobility Extensions to ns," http://www.monarch.cs.cmu.edu/cmu-ns.html, Oct. 1999.

[2] S. Corson and V. Park, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," *Mobile Ad Hoc Network (MANET) Working Group, IETF,* Oct. 1999.

[3] S.R. Das, C.E. Perkins, E.M. Royer, and M.K. Marina, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *IEEE Personal Comm. Magazine,* special issue on ad hoc networking, pp. 16-28, Feb. 2001.

[4] S. Doshi and T.X. Brown, "Minimum Energy Routing Schemes for a Wireless Ad Hoc Network," *Proc. IEEE INFOCOM,* 2002.

[5] J.J. Garcia-Luna-Aceves and M. Spohn, "Source-Tree Routing in Wireless Networks," *Proc. IEEE Int'l Conf. Network Protocols (ICNP),* 1999.

[6] C. Gui and P. Mohapatra, "SHORT: Self-Healing and Optimizing Routing Techniques for Mobile Ad Hoc Networks," *Proc. ACM MobiHoc,* 2003.

[7] Z. Haas, J. Halpern, and L. Li, "Gossip-Based Ad Hoc Routing," *Proc. IEEE Infocom,* 2002.

[8] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking (MobiCom),* 2002.

[9] J. Broch, D. Johnson, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet draft, Oct. 1999.

[10] D. Johnson and D. Maltz, *Ad Hoc Networking.* Addison-Wesley, 2001.

[11] B. Karp and H.T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom '00),* pp. 243-254, Aug. 2000.

[12] S.-J. Lee and M. Gerla, "Dynamic Load-Aware Routing in Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm.,* pp. 3206-3210, June 2001.

[13] S.-J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm.,* pp. 3201-3205, June 2001.

[14] S.-J. Lee, E.M. Royer, and C.E. Perkins, "Scalability Study of the Ad Hoc On-Demand Distance Vector Routing Protocol," *ACM/Wiley Int'l J. Network Management,* vol. 13, no. 2, pp. 97-114, Mar. 2003.

[15] Y. Lu, W. Wang, Y. Zhong, and B. Bhargava, "Study of Distance Vector Routing Protocols for Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom),* pp. 187-194, Mar. 2003.

[16] M. Marina and S. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks," *Proc. IEEE Int'l Conf. Network Protocols (ICNP),* pp. 14-23, 2001.

[17] A. Nasipuri, R. Castaneda, and S.R. Das, "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks," *ACM/Baltzer Mobile Networks and Applications J. (MONET),* vol. 6, pp. 339-349, 2001.

[18] A. Nasipuri and S.R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Computer Comm. and Networks (IC3N),* Oct. 1999.

[19] NS-2, Network Simulater, http://www.isi.edu/nsnam/ns/, 2006.

[20] C.E. Perkins, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM,* pp. 234-344, 1994.

[21] C.E. Perkins, E.M. Belding-Royer, and I. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing," IETF Internet draft, Oct. 2003.

[22] H. Raghavendra and D.A. Tran, "Congestion Adaptive Routing in Ad Hoc Networks (Short Version)," *Proc. ACM Int'l Conf. Mobile Computing and Networking (MOBICOM),* Oct. 2004.

[23] E.M. Royer and C.E. Perkins, "Evolution and Future Directions of the Ad Hoc On-Demand Distance Vector Routing Protocol," *Ad Hoc Networks J.,* vol. 1, no. 1, pp. 125-150, July 2003.

[24] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proc. ACM Mobicom,* 1999.

[25] M. Spohn and J.J. Garcia-Luna-Aceves, "Neighborhood Aware Source Routing," *Proc. ACM MobiHoc,* 2001.

[26] I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 10, Oct. 2001.

[27] D.A. Tran and H. Raghavendra, "Routing with Congestion Awareness and Adaptivity in Mobile Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC),* Mar. 2005.

[28] A. Valera, W. Seah, and S. Rao, "Cooperative Packet Caching and Shortest Multipath Routing in Mobile Ad Hoc Networks," *Proc. IEEE Infocom,* Apr. 2003.

[29] S. Vutukury and J. Garcia-Luna-Aceves, "MDVA: A Distance-Vector Multipath Routing Protocol," *Proc. IEEE Infocom,* pp. 557-564, 2001.
[30] G. Yashar and A. Keshavarzian, "Load Balancing in Ad Hoc Networks: Single-Path Routing vs. MultiPath Routing," *Proc. IEEE INFOCOM '04,* Mar. 2004.
[31] Z. Ye, S.V. Krishnamurthy, and S.K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," *Proc. IEEE Infocom,* Apr. 2003.
[32] S. Yi, P. Naldurg, and R. Kravets, "Security-Aware Ad Hoc Routing for Wireless Networks," *Proc. ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc),* 2001.

Duc A. Tran received the BS degree in computer science from Vietnam National University in 1996 and the PhD degree in computer science from the University of Central Florida in May 2003. He has been an assistant professor of computer science at the University of Dayton since Fall 2003. Dr. Tran currently does research on multimedia systems, P2P and grid computing and networking, and mobile ad hoc and sensor networks. His professional service includes serving as a guest editor for the *Journal on Wireless and Mobile Computing*, a member of eight technical program committees, and a reviewer for numerous conferences, journals, and magazines including *ACM Multimedia Systems Journal*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Knowledge and Data Engineering*, and *IEEE Network*. Dr. Tran is a member of the IEEE, the IEEE Computer Society, and the ACM.

Harish Raghavendra is currently a computer science graduate student at the University of Dayton, working in the Multimedia and Collaborative Systems Group under Dr. Duc A. Tran's supervision. His main research interest is networking and information retrieval in mobile ad hoc and sensor networks.