



<b>Title</b>	<b>BGCA: bandwidth guarded channel adaptive routing for ad hoc networks</b>
<b>Author(s)</b>	<b>Lin, X; Kwok, YK; Lau, KN</b>
<b>Citation</b>	<b>IEEE Wireless Communications and Networking Conference Proceedings, Orlando, FL., 17-21 March 2002, v. 1, p. 433-439</b>
<b>Issued Date</b>	<b>2002</b>
<b>URL</b>	<b><a href="http://hdl.handle.net/10722/46286">http://hdl.handle.net/10722/46286</a></b>
<b>Rights</b>	<b>Creative Commons: Attribution 3.0 Hong Kong License</b>

# BGCA: Bandwidth Guarded Channel Adaptive Routing for Ad Hoc Networks

Xiao-Hui Lin, Yu-Kwong Kwok, and Vincent K. N. Lau

Department of Electrical and Electronic Engineering  
The University of Hong Kong

**Abstract --** To support truly peer-to-peer applications in ad hoc wireless networks, a judicious and efficient ad hoc routing protocol is needed. Much research has been done on designing ad hoc routing protocols and some well known protocols are also being implemented in practical situations. However, one major drawback in existing state-of-the-art protocols, such as the AODV (ad hoc on demand distance vector) routing protocol, is that the time-varying nature of the wireless channels among the mobile terminals is ignored, let alone exploited. In this paper, by using a previously proposed adaptive channel coding and modulation scheme which allows a mobile terminal to dynamically adjust the data throughput via changing the amount of error protection incorporated, we devise a new ad hoc routing algorithm that dynamically changes the routes according to the channel conditions. Extensive simulation results indicate that our proposed protocol is more efficient in that shorter delays and higher rates are achieved.

**Keywords:** ad hoc networks, routing protocols, on-demand routing, channel state dependent, adaptive.

## I. INTRODUCTION

To realize efficient information exchange in a peer-to-peer manner in an ad hoc wireless network, a judicious routing protocol is needed for the source to locate the destination in the network [9]. There are two major classes of ad hoc routing protocols: on-demand and table based. As many researchers have pointed out [9], table based algorithms are notoriously inefficient in that they require periodic update of the routing information stored in the routing tables, even when there is no data traffic. The major merit of table based algorithms, as compared with on-demand algorithms, is that the set up delay for a data transfer is expected to be shorter because a route is presumably stored in the table for use. However, such route may no longer exist or usable when the actual data transfer is to be taken place for at least two reasons. First, due to the mobility of the mobile terminals in the network, their geographical locations may have changed when a data transfer is required, rendering a previously set up route useless. The second reason, which, we believe, is a more important one, is that the quality of the channels among the mobile terminals is inevitably time-varying (due to shadowing and fast fading), and thus, the links in a route may no longer be usable even if the geographical locations do not change much. Indeed, this is a major consideration overlooked in previous researches on ad hoc routing protocols [1], [2], [7], [9], [10], [11], [12].

In this paper, we focus on on-demand routing algorithms for ad hoc networks. In particular, we are interested in studying the behavior and performance of routing protocols when the time-

varying nature of wireless channels is taken into account. In this regard, we propose a new ad hoc routing protocol, which is designed based on an improved version of the well-known ABR (associativity based routing) protocol, called the BGCA (Bandwidth Guarded Channel Adaptive) protocol. The proposed protocol works by adaptively changing the routes according to the current channel conditions. In BGCA, normally the change of a route (not a broken one) is due to the deterioration of links in the route, so the intermediate node must find a partial route to substitute the original one. This algorithm is a little "passive" to the change of the route. That is, only when the channel quality of the link drops below the bandwidth requirement of the traffics does it take actions to find a new route. Furthermore, in order to satisfy the bandwidth requirements of the traffic load, the algorithm tries to make the available link bandwidth greater or equal to the sum of bandwidth requirements.

This paper is organized as follows. In the next section, we briefly introduce the two well-known routing algorithms considered in our study, namely the ABR and AODV protocols. For an excellent survey on ad hoc routing protocols, the reader is referred to [11]. In Section III, we describe our proposed BGCA protocol in detail with illustrative examples. Section IV contains the performance results of a quantitative comparison of the four algorithms. We provide some concluding remarks in Section V.

## II. STATE-OF-THE-ART PROTOCOLS

### A. The ABR Protocol

ABR [7], [11], [12] is a source-initiated on-demand routing protocol—a mobile terminal in the network does not need to keep a route to every other terminals. The major distinctive characteristic of ABR is that the route is not chosen on a shortest-path basis as in other protocols, but on a long-lived basis. Selecting a long-lived route has much merit, such as the chosen route is more robust (not easy to break due to mobility), the maintenance of route is easier and the number of route reconstruct (RRC) is reduced (thus, the routing overhead is reduced and more bandwidth is saved). In ABR, longevity of a route is put at the first place. In other protocols, such as Link State and AODV, the primary goal is to find the shortest path. However, in these protocols, data flow transmission interruptions occur more often and more route reconstructions are needed.

The essence of ABR is that as a mobile terminal moves, its

associativity with the neighbor terminals also changes, and this associativity can be quantified by using associativity ticks. In ABR, each mobile terminal periodically transmits beacons (such as hello messages) to signify its existence. When these beacons are received by its neighbor terminals, these beacons cause the associativity of this terminal with its neighbor terminals increase. The greater the associativity, the more stable of this terminal will be. A high associativity of the terminal means a low mobility of a terminal. If a terminal A moves out of the transmission range of another terminal, the associativity record of A in this terminal will be reset. The most fundamental objective of ABR is to derive a long-lived route between the source and destination terminals.

### B. The AODV Protocol

AODV [9], [10], [11], is a kind of pure reactive routing protocol. In this protocol, each terminal does not need to keep a view of the whole network nor a route to every other terminal. Nor does it need to periodically exchange route information with the neighbor terminals. Furthermore, only when a mobile terminal has packets to send to a destination does it need to discover and maintain a route to that destination terminal. The major distinctive characteristics are:

- routes are discovered or maintained only when necessary (or on an as-needed basis);
- AODV is loop-free [3], [10] at all time, this is accomplished through the use of terminal sequence number which is increased monotonically (this technique also ensures that the most current route is always used when discovering a route);
- AODV is a bandwidth efficient routing algorithm, which greatly reduces the use of limited bandwidth (the number of route broadcasts are minimized on as-needed basis);
- AODV responds very quickly to the topology changes of the network and can recover a broken route in a timely manner; and
- AODV has a low storage requirement on the terminal because it only need to maintain the active neighbors information instead of a full route to the destination.

In AODV, each terminal contains a route table for a destination. Route table stores the following information: destination address and its sequence number, active neighbors for the route, hop count to the destination, and expiration time for the table. The expiration time is updated each time as this route is used. If this route has not been used for a specified period of time, it will be expired.

## III. BANDWIDTH GUARDED CHANNEL ADAPTIVE ROUTING

The main idea of BGCA is that, when a link is in deep fading, the upstream terminal will execute a local search to find a partial route to the destination. Before describing the protocol in detail, we introduce the channel model below. Throughout the paper, we assume a multi-code CDMA MAC layer is used

in all the protocols.

### A. Channel Model

The wireless channel between every two mobile terminals is time-varying and hence, the throughput of the channel is also a time-varying function. Specifically, using a channel adaptive coder and modulator proposed previously (called ABICM) [5], the transmitter/receiver can dynamically adjust the level of error protection in the data transmission according to the channel state (when the channel state is good, less protection is included and vice versa) and, as such, the effective throughput of the channel is dynamically changing according to the channel conditions. For the details of the ABICM scheme and its applications in MAC protocols, the reader is referred to [6].

We characterize the channel by modeling the fast fading and long-term shadowing components [8]. Based on the CSI (channel state information) of the channel, we divide the channel quality into 4 classes: A, B, C, and D, with a throughput of 60kbps, 40kbps, 20kbps and 10kbps respectively (after adaptive channel coding and modulation). We define one "hop" in the following manner: if a link between two terminals with channel quality of class A (with the throughput of 60kbps), then the distance between two terminal is defined as ONE hop. We use this "distance" as a baseline. Then, if a link between two terminals with a channel quality of class B (with a throughput 40kbps), the distance between 2 terminals is 1.5 hops because the transmission delay now is 1.5 times compared with a link of class A. Thus, for a link is with a throughput of 60kbps, 40kbps, 20kbps, and 10kbps, the distance will be 1, 1.5, 3, and 6 hops respectively.

### B. Path Discovery

When a terminal has packets to send to a destination terminal, it first generates a RREQ (route request) packet, which includes the following fields: type of the packet, source and destination ID, broadcast ID, bandwidth requirement, hop count from the source (CSI based), and intermediate terminals list. Every time the source broadcast a RREQ to the same destination, broadcast ID increases by one. The source terminal appends its ID to the intermediate terminals list and broadcasts this RREQ throughout the network in search of the destination terminal.

When the RREQ is received by an intermediate terminal, the intermediate terminal checks its seen table to see whether it has processed this packet before. If yes, it will discard it. If it has not seen this packet before, it will check whether the available bandwidth of the link between it and the terminal from which the RREQ comes can satisfy the bandwidth requirement of the RREQ. If the link bandwidth can not satisfy the requirement, the terminal also discards the RREQ. Otherwise, if the link bandwidth can satisfy the requirement of the RREQ, the intermediate terminal will record this RREQ in its seen table including the following information: source and destination IDs, broadcast ID. Then the intermediate terminal will append its ID to the intermediate terminals list of the RREQ and set the

hop count field of RREQ to the original hop count plus the hop distance from the upstream terminal. After doing all these steps, this intermediate terminal rebroadcast the RREQ out. This process continues until the RREQ reaches the destination terminal. At last the destination terminal may receive several RREQs from all possible routes.

Figure 1 shows the broadcast of RREQ in the network. Each RREQ includes a full route from the source to the destination and related hop distance (CSI based). Then the destination terminal will choose the shortest route and sends a route reply (RREP) along this route to the source as shown in Figure 2 (three route with hop distance of 8.5, 11, 11.5 respectively). The RREP includes the full list of terminals along the route so that each terminal will know its upstream and downstream terminals together with the related PN codes to receive and transmit packets.

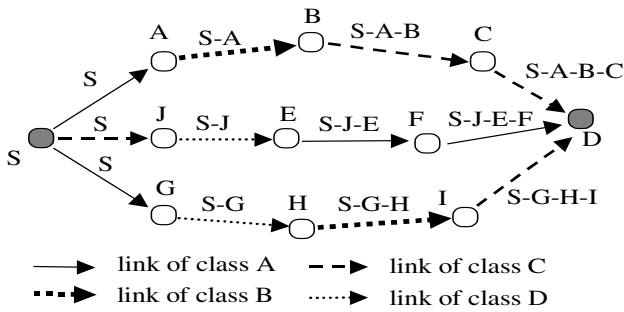


Figure 1: Broadcast of RREQ in BGCA.

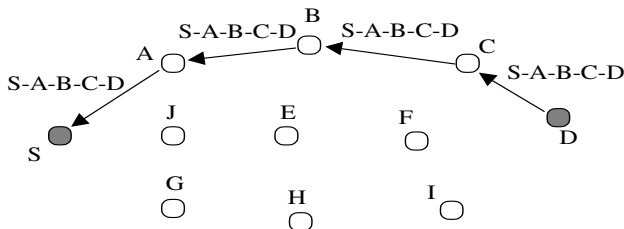


Figure 2: Generation of RREP in BGCA.

### C. Route Maintenance

Long deep channel fading will lead to the drop of data throughput and network congestion and data loss. This is what we try to avoid in the BGCA protocol. The terminal can adjust its data transmission rate according to the channel quality by using the adaptive channel coding and modulation scheme ABICM [5] as mentioned earlier.

Specifically, the terminal can get the CSI information from the feedback channel. When the channel quality drops, this will be noticed by the downstream terminal, then the downstream terminal will inform the upstream terminal this CSI change through the feedback channel. The upstream terminal will check whether the throughput of this link can still satisfy the bandwidth requirement of the data traffic. If the link can still satisfy the requirement, the upstream terminal ignores this warning. If the link deteriorates too much and cannot satisfy

the bandwidth requirement, the upstream terminal has to perform local search to find a partial route to the destination. If several routes share this link, then the upstream terminal will decide which routes will have to perform local search. We stipulate that the available bandwidth of the link must greater or equal to the bandwidth requirements of the traffics so the number of route through this link must be decreased.

Our strategy is to let the route whose partial route is shorter (with smaller hop count) to perform this local search (if the link is used by several routes so as to shorten the search time. The upstream terminal of the link in fading first generates an LQ (local query) packets which includes the following information: type of the packet, source and destination IDs, ORIG ID (the ID of the terminal originating the LQ), sequence number, intermediate terminals list, bandwidth requirement, hop count (originally is set to zero), and TTL (time-to-live) field. TTL field is set to the hop distance (not CSI based) from the originating terminal to the destination, each time the LQ is rebroadcast, the TTL will be decreased by one. When TTL reaches zero, the LQ is discarded. In this way, a full broadcast is avoided.

When this LQ is received by a neighbor terminal, this terminal first checks whether the packet comes from a link that can satisfy the bandwidth requirement in the LQ, if not, this LQ will be discarded. If the bandwidth requirement is satisfied, the terminal further checks whether it has seen this packet before, if it has seen this LQ, LQ will also be discarded; otherwise, the neighbor terminal will append its ID to the intermediate terminals list in the LQ, reset the hop count (CSI based) and rebroadcast the LQ out. This process continues until the LQ reaches the destination. At last, the destination may receive several LQs from the originating terminal through all possible routes. Each LQ includes a full route from the originating terminal to the destination and the hop distance of the route. Now the destination terminal can select the shortest partial path and send a RREP including the all the terminals in path to the originating terminal.

On receiving the RREP, the originating terminal will use this route to substitute the original partial route. If the timeout period has passed and the originating terminal has not yet received the RREP, it will try another time. If still fail, it will unicast an unreachable message to the source. On receiving this unreachable message, the source will initiate another RREQ in search of the destination. In Figure 3, the original route from source to destination is S-A-B-C-D. Link B-C is in deep fading and can not satisfy the bandwidth requirement, so terminal B performs the local search. Now the new full route is S-A-B-E-D.

### D. Link Break

When an intermediate terminal or the destination moves and causes the break of a link, the downstream terminal will notice that (because it can not receive the ACKs of the data packets), it will unicast a route error (REER) to the source terminal. On receiving this REER, the source will initiate a RREQ in search

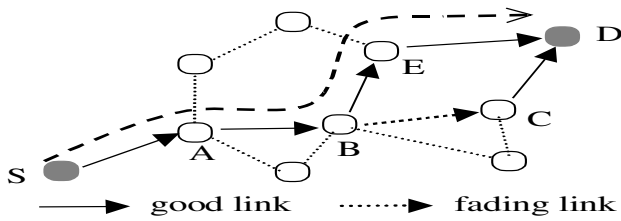


Figure 3: Change of links in BGCA.

of a new route to the destination. If the source terminal moves and causes the break of the link, it will initiate a RREQ to find a new route to the destination.

#### IV. RESULTS

In this section, we present the results obtained in our extensive simulations comparing the three algorithms considered in this paper. We first introduce the simulation environment.

##### A. Simulation Environment

The simulation parameters we used are as follows:

- number of terminals: 50;
- testing field: 1000m × 1000m ;
- mobile speed: uniformly distributed between 0 and MAX-SPEED (will be elaborated later);
- mobility model: when the terminal reaches its destination, it pauses for 3 seconds, then randomly chooses another destination point;
- radio transmission range: 250m;
- channel model: characterized by fast fading and long-term shadowing components, with throughput 60kbps, 40kbps, 20kbps, 10kbps respectively;
- bandwidth of the common channel: 100kbps, we suppose this channel is robust that can withstand deep fading and interference; (the common channel is protected with the maximum level of FEC)
- MAC of common channel: unslotted CSMA/CA based on CDMA;
- traffic load: 10 terminal pairs, in each pair, we change the traffic load for 10, 15, 20 packets/sec respectively.

Furthermore, the length of the data packet is 1000 bits and the capacity of data buffer is 10. The transmission of packet is described as a store-and-forward process. When packet reaches an intermediate terminal, it waits in the queue for service (FCFS). Each packet is allowed to be kept in the buffer for no more than one second, if it has not been transmitted in this period, it will be discarded. The generation of data packets in each source terminal is a Poisson arrival process, i.e., the inter-arrival of two packets is exponential distribution.

Each simulation is run for 300 seconds and repeated 15 times. We average the results of these 15 iterations. To evaluate the three routing algorithms, we compare them in three met-

rics:

- **Average End-to-End Delay:** Measured in ms, the end to end delay includes the transmission time and queuing time of packet in each terminal in the route.
- **Packet Delivery Rate:** This is the ratio of packets reaching the destination to total packets generated in the sources. A packet may be dropped if there is not enough data buffer due to the congestion, or has stayed in the buffer for more than one second.
- **Routing Overheads:** This is measured in bps. We count the total routing packets in each round of simulation. Each time the common channel is used to transmit a routing packet, this will be counted as one transmission. We average the amount of routing overheads (in bits) to the whole simulation time.

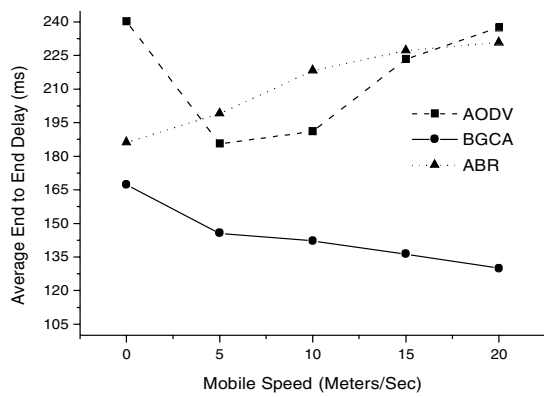
##### B. Average End-to-End Delay

The first set of results is average delay vs. mobile speed while adding the traffic load from 10 packets/sec to 20 packets/sec. There results are shown in Figure 4. We varied the mean mobile speeds from 0 to 20 m/sec, and thus, the value of MAX-SPEED was varied from 0 to 40m/sec. As can be obviously seen from the plots, taking the CSI into consideration can greatly shorten the transmission delay from the source to destination as in the BGCA protocol.

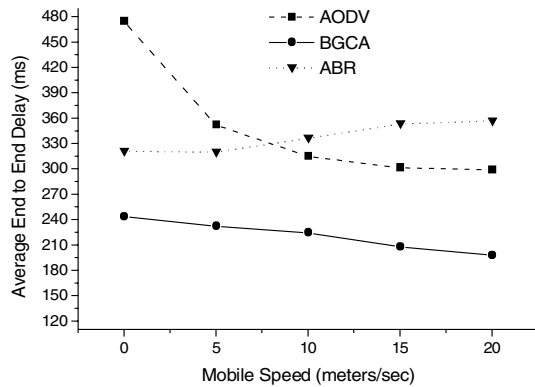
In BGCA, the delay decreases with the increase of the mobile terminal. This is because when the mobile speed increases, the long queue is not easy to form (because link break happens more often), thus decreasing the queueing delay, but at the same time the number of dropped packets also increases as will be seen later. In ABR, however, delay increases with the mobile speed because of the local search. When the link breaks, the packets will accumulate in the upstream terminal performing the local search until a partial route is found, thus the long queue will form and queuing time increases.

We also observe one interesting phenomenon that when in low mobility ABR outperforms AODV, but in high mobility AODV outperforms ABR in end to end delay. This is because ABR takes the load and transmission delay of the link into consideration when selecting the route (by choosing a link with high throughput), thus balancing the link load and decreasing the delay. While in AODV, the destination responds only the first RREQ and chooses the path this RREQ has gone through although this route is usually not the shortest one or the some links in the route may be congested. But as the mobility increases and the link will be easier to break due to the mobility, in AODV, the source terminal will perform a full broadcast in search of a new route, and packets in the original broken route usually will be discarded, so the long queue wont be easy to form.

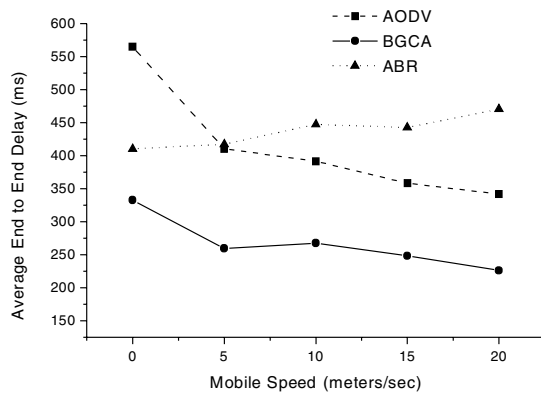
However, in ABR, a LQ will be implemented to find a partial route and data packets will have to wait in the terminal performing LQ, so the long queue is formed, and this will increase



(a) 10 packets per second



(b) 15 packets per second



(c) 20 packets per second

Figure 4: End-to-end delays of the all the protocols.

the end to end delay, but on another side, the packet delivery rate of ABR is also greater than that in AODV as shown later (usually in AODV a great portion of data packets is dropped due to link break as observed in our experiments). Another reason is that usually, the link in ABR is robust than that in AODV, so the long queue is easier to form in the link with low throughput (10kbps or 20kbps). Long queue is also formed in link with low throughput in AODV, but frequent link break will often eliminate these long queues. Normally, route in ABR is longer than that in AODV because different route selection criteria, this also makes the delay in ABR longer.

As the traffic load increase, AODV behaves differently in end to end delay. This is because in low load (10 packets/sec), buffers of terminals normally are not saturated, packets will have to wait in source for a new route (original route has broken), thus increasing mobility will increase this waiting time because link breaks happens more frequently. But when the traffic load is high (15 or 20 packets/sec), buffers begin to saturate in terminals with deep fading links (long queues are formed in intermediate terminals), so increasing the mobility will cause more packets be dropped, thus the delay will decrease, but the delivery rate also drops. As we have expected, in all 4 algorithm, the end to end delay increased as we add the traffic load from 10 packets/sec to 20 packets/sec.

### C. Packet Delivery Rate

From the simulation results shown in Figure 5, we can see that taking CSI into consideration will contribute to the reliability of packet delivery. Again in delivery rate, BGCA outperforms the other two algorithms. The gain is more obvious as we increase the traffic loads. Note that in BGCA, the update of route does not happen very often. The route update in BGCA only happens when it is broken (i.e., in deep fading) and the source has to find a new route to substitute it, so the packet queues in BGCA can be long and when the link breaks, data loss will be more serious. But in BGCA, measure (LQ) is taken to ensure the bandwidth requirement be satisfied, so the congestion is avoided and when link breaks, data loss will be reduced. ABR and AODV do not take the CSI into consideration, so their routing can not adapt to the change of link throughput which fluctuates with time and long queue is easier to form.

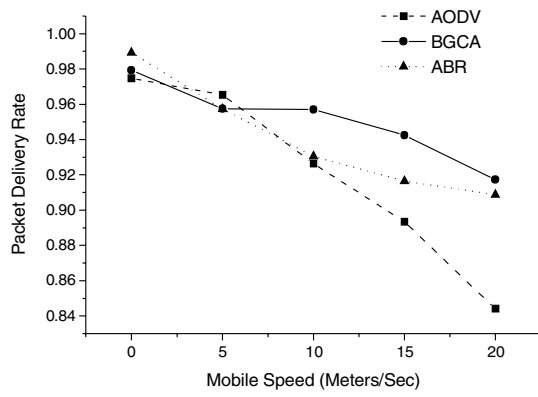
Normally the main causes of data loss are: link congestion and not enough data buffer; and link break. In these two algorithms, long queue is very easy to form in the link with low throughput especially when the traffic load is high (for example 15 packets/sec or 20 packets/sec). We have observed the saturation of the data buffers in this circumstance for many times in our experiments. To ensure the reliability of packet delivery, long queue should be avoided. As seen from the results of delivery rate, ABR performs better than AODV because:

- the routes in ABR are more robust than those in AODV.
- ABR takes the link load and transmission delay into consideration when choosing the route; and
- ABR performs LQ to find a partial route at the broken point, so the probability of packets being dropped in the upstream route is reduced.

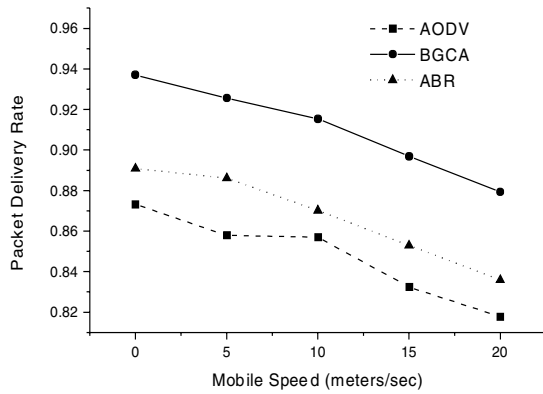
As what we have expected, packet delivery rate decreases with increase of the mobility and traffic loads because the link break happens more often and congestion and long queue are easier to form.

### D. Routing Overhead

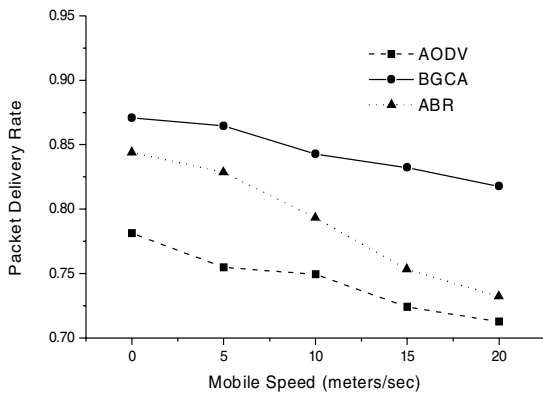
The results on routing overhead are shown in Figure 6. Tak-



(a) 10 packets per second

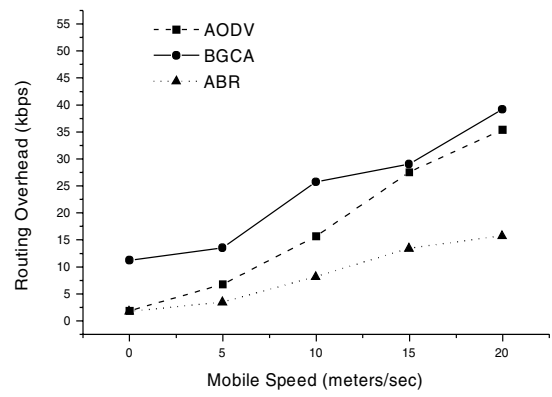


(b) 15 packets per second

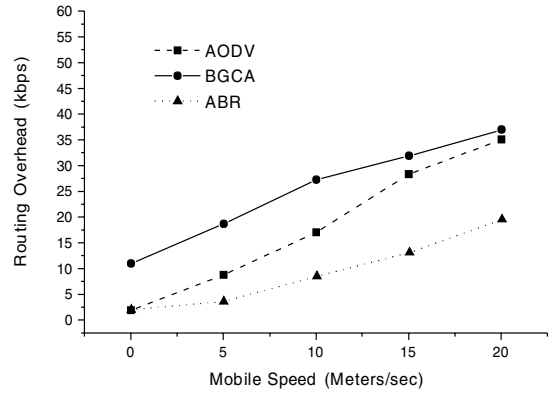


(c) 20 packets per second

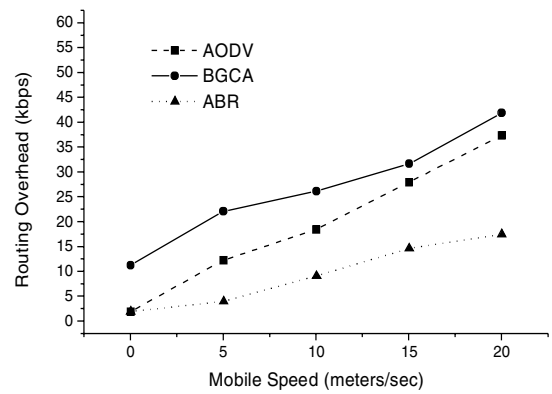
Figure 5: Packet delivery rates of the all the protocols.



(a) 10 packets per second



(b) 15 packets per second



(c) 20 packets per second

Figure 6: Routing overheads of all the protocols.

ing CSI into consideration when choosing a route can improve the network performance in the sense of delay and packet delivery rate, but the cost is that it will also increase more routing overhead. If use the amount of routing overhead in AODV as baseline, BGCA generates about 2 and 4 times of overhead respectively. The reason is obvious: in BGCA, in order to ensure the bandwidth requirement of traffic is satisfied, the

intermediate terminals have to perform local search, this will also increase the routing overhead. As seen from the plots, ABR generates the least of routing overhead because:

- the route in ABR is long-lived so the break of link happens not so frequent as in other routing algorithm; and
- even when the link breaks, the intermediate terminal perform local search instead of a full broadcast.

Thus, ABR is a bandwidth efficient algorithm. Just as expected, in all algorithms, routing overhead increases with the mobility because link break will be more frequent and this will increase the load of route maintenance. It is also observed that, increasing the load of data traffic only has little influence on the routing overhead because route needs repair only when it is broken.

Finally, as shown in Figure 7, we also measured the packets reaching destination terminals in every 1.2 seconds. As can be seen, in BGCA, CSI of links is taken into consideration when routing the packets and this will significantly enhance the instantaneous throughput of the network. Thus, it is useful to incorporate CSI in determining routes dynamically in an ad hoc network.

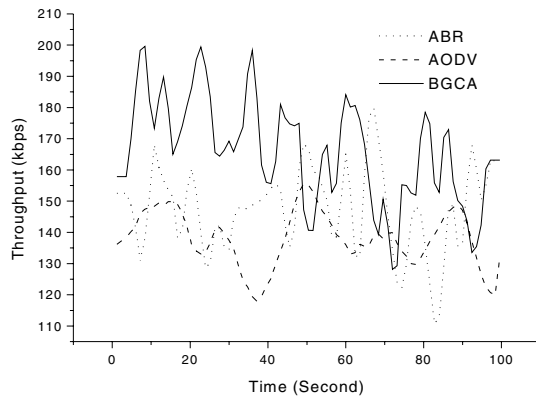


Figure 7: Throughput variations of all the protocols over the testing time period (20 packets/sec; mobile speed: 20 m/sec).

## V. CONCLUDING REMARKS

In this paper, we study the behavior and performance of ad hoc routing protocols under a more realistic channel model. We have proposed a new routing protocol, called BGCA (Bandwidth Guarded Channel Adaptive) protocol, which takes into account the time-varying nature of the channel and incorporate an adaptive channel coding and modulation scheme for dynamically adjusting the throughput (the amount of error protection) according to the channel conditions. In our extensive simulations study, we found that the BGCA protocol significantly outperforms the well-known ABR and AODV protocols. We are currently studying the performance of the BGCA protocol by incorporating preemptive routing [4].

## ACKNOWLEDGMENTS

This research is supported by HKU URC seed grants under contract numbers 10203010 and 10203413, and by a RGC research grant under contract number HKU7024/00E.

## REFERENCES

- [1] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. MOBIKOM'98*, pp. 85–97, July 1998.
- [2] M. S. Corson, J. P. Macker, and G. H. Cirincione, "Internet-Based Mobile Ad Hoc Networking," *IEEE Internet Computing*, July/Aug. 1999, pp. 63–70.
- [3] J. J. Garcia-Luna-Aceves and S. Murthy, "A Path-Finding Algorithm for Loop-Free Routing," *IEEE/ACM Trans. Networking*, vol. 5, no. 1, pp. 148–160, Feb. 1997.
- [4] T. Goff, N. Abu-Ghazaleh, D. Phatak, and R. Kahvecioglu, "Preemptive Routing in Ad Hoc Networks," *Proc. MOBIKOM'2001*, pp. 43–52, July 2001.
- [5] V. K. N. Lau, "Performance of Variable Rate Bit-Interleaved Coding for High Bandwidth Efficiency," *Proc. of VTC'2000*, vol. 3, pp. 2054–2058, May 2000.
- [6] V. K. N. Lau and Y.-K. Kwok, "On Integrating Multiple Access Control and Adaptive Channel Coding for Cellular Wireless Voice and Data Services," *Computer Communications*, vol. 24, no. 10, pp. 903–916, May 2001.
- [7] S.-J. Lee, M. Gerla, and C.-K. Toh, "A Simulation Study of Table-driven and On-demand Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, vol. 13, no. 4, pp. 48–54, July-Aug. 1999.
- [8] J. D. Parsons, *The Mobile Radio Propagation Channel*, Second Edition, Wiley, 2000.
- [9] C. E. Perkins (Ed.), *Ad Hoc Networking*, Addison-Wesley, 2000.
- [10] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing, Mobile Computing Systems and Applications," *Proceedings of WMCSA'99*, pp. 90–100, 1999.
- [11] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, April 1999.
- [12] C.-K. Toh, "A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing," *Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pp. 480–486, 1996.