Springer Series on
SIGNALS AND COMMUNICATION TECHNOLOGY

SIGNALS AND COMMUNICATION TECHNOLOGY

Advances in Wireless Ad Hoc and Sensor Networks
M.X. Cheng and D. Li (Eds.)
ISBN 978-0-387-68565-6

Wireless Sensor Networks and Applications
Y. Li, M. Thai, and W. Wu (Eds.)
ISBN 978-0-387-49591-0

Multimodal User Interfaces: From Signals to Interaction
D. Tzovaras (Ed.)
ISBN 978-3-540-78344-2

Handover in DVB-H: Investigations and Analysis
X. Yang
ISBN 978-3-540-78629-0

Passive Eye Monitoring: Algorithms, Applications, and Experiments
R.I. Hammoud (Ed.)
ISBN 978-3-540-75411-4

Digital Signal Processing: An Experimental Approach
S. Engelberg
ISBN 978-1-84800-118-3

Digital Video and Audio Broadcasting Technology: A Practical Engineering Guide
W. Fischer
ISBN 978-3-540-76357-4

Foundations and Applications of Sensor Management
A.O. Hero III, D. Castañón, D. Cochran, and K. Kastella (Eds.)
ISBN 978-0-387-27892-6

Three-Dimensional Television: Capture, Transmission, Display
H.M. Ozaktas and L. Onural (Eds.)
ISBN 978-3-540-72531-2

Human Factors and Voice Interactive Systems, Second Edition
D. Gardner-Bonneau and H. Blanchard
ISBN 978-0-387-25482-1

Wireless Communications: 2007 CNIT Thyrrenian Symposium
S. Pupolin
ISBN 978-0-387-73824-6

Blind Speech Separation
S. Makino, T.-W. Lee, and H. Sawada (Eds.)
ISBN 978-1-4020-6478-4

Continuous-Time Systems
Y. Shmaliy
ISBN 978-1-4020-6271-1

Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems
H. Arslan (Ed.)
ISBN 978-1-4020-5541-6

Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches
T. Ogunfunmi
ISBN 978-0-387-26328-1

Wireless Network Security
Y. Xiao, X. Shen, and D.Z. Du (Eds.)
ISBN 978-0-387-28040-0

Satellite Communications and Navigation Systems
E. Del Re and M. Ruggieri
ISBN 0-387-47522-2

Terrestrial Trunked Radio – TETRA: A Global Security Tool
P. Stavroulakis
ISBN 978-3-540-71190-2

Multirate Statistical Signal Processing
O. Jahromi
ISBN 978-1-4020-5316-0

Face Biometrics for Personal Identification: Multi-Sensory Multi-Modal Systems
R.I. Hammoud, B.R. Abidi, and M.A. Abidi (Eds.)
ISBN 978-3-540-49344-0

Positive Trigonometric Polynomials and Signal Processing Applications
B. Dumitrescu
ISBN 978-1-4020-5124-1

Wireless Ad Hoc and Sensor Networks
A Cross-Layer Design Perspective
R. Jurdak
ISBN 978-0-387-39022-2

Ad-Hoc Networking Towards Seamless Communications
L. Gavrilovska and R. Prasad
ISBN 978-1-4020-5065-7

Cryptographic Algorithms on Reconfigurable Hardware
F. Rodriguez-Henriquez, N.A. Saqib, A. Díaz Pérez, and C.K. Koc
ISBN 978-0-387-33883-5

Acoustic MIMO Signal Processing
Y. Huang, J. Benesty, and J. Chen
ISBN 978-3-540-37630-9

Algorithmic Information Theory: Mathematics of Digital Information Processing
P. Seibt
ISBN 978-3-540-33218-3

Multimedia Database Retrieval
A Human-Centered Approach
P. Muneesawang and L. Guan
ISBN 978-0-387-25627-6

(continued after index)

Edited by
Maggie Xiaoyan Cheng
Deying Li

Advances in Wireless Ad Hoc and Sensor Networks

Springer

Editors
Maggie Xiaoyan Cheng
Department of Computer Science
Missouri University of Science
and Technology
Rolla, MO 65409
USA

Deying Li
School of Information
Renmin University of China
100872 Beijing
China

ISSN: 1860-4862
ISBN: 978-0-387-68565-6 e-ISBN: 978-0-387-68567-0
DOI: 10.1007/978-0-387-68567-0

Library of Congress Control Number: 2007938692

Printed on acid-free paper

springer.com

Contents

Foreword

Wireless ad hoc networks, mobile or static, have special resource requirements and different topology features, which make them different from classic computer networks in resource management, routing, media access control, and QoS provisioning. Some issues are unique to ad hoc wireless networks and sensor networks, such as self-organization, mobility management, energy efficient design, and so on. The purpose of this book is not to provide a complete survey of the state-of-the-art research on all areas of ad hoc and sensor networks, but rather to focus on the theoretical and experimental study of a few advanced topics. We carefully selected papers around the following four topics: security and trust, broadcasting and multicasting, power control and energy efficency, and QoS provisioning.

Chapters 1–3 are about QoS routing in Mobile Ad hoc NETworks (MANET): Chapter 1 discusses QoS routing for heterogeneous mobile ad hoc networks; Chapter 2 proposes a link state QoS routing protocol for ad hoc networks using bandwidth and delay as routing metrics; Chapter 3 studies the interworking between a mobile ad hoc network and the Internet, extending the Differentiated Services (DiffServ) model to a wireless environment.

Chapters 4–6 are related to security and trust issues in ad hoc environments: Chapter 4 addresses secure communication in ad hoc networks, providing a threshold decryption scheme that allows different mobile nodes to use public keys of several different cryptosystems; Chapter 5 proposes a secure group communication protocol for ad hoc wireless networks and maintenance processes for topology changes; Chapter 6 addresses routing in ad hoc networks from the trust and security perspectives, and proposes a direct trust model that establishes and manages trust without using cryptographic mechanisms, which is suitable in ad hoc networks.

Chapters 7–9 address power control and energy efficient design: Chapter 7 proposes a power optimization scheme that improves both power consumption and throughput of multihop wireless networks; Chapter 8 presents mechanisms to save energy in sensor networks without losing sensing area, which control the network density based on the Voronoi diagram, and deterministically deploy sensors after the initial network has been used; Chapter 9 addresses the self-organization of MANET from the message optimality perspective and proposes an algorithm that is message-efficient for initial configuration and message-optimal for self-configuration under mobility.

Chapters 10–13 focus on broadcast and multicast in MANET: Chapter 10 considers energy-efficient multicast with mobility support for ad hoc networks by using two multicast trees; Chapter 11 proposes novel approaches to construct multicast trees in mobile ad hoc networks and to maintain the trees under rapid topology changes. The resulting multicast trees have the LAST property; that is, the cost of each path from the source to any terminal in the multicast tree does not exceed a given constant factor α from the corresponding shortest-path cost in the original graph, and the total cost of the multicast tree does not exceed a given constant factor β from the total cost of the Minimum Spanning Tree (MST); Chapter 12 proposes three techniques to improve the counter-based broadcasting scheme in mobile ad hoc networks; Chapter 13 addresses energy-efficient broadcasting and multicasting schemes in ad hoc networks that balance the energy consumption in broadcasting and multicasting by minimizing the maximum energy consumption.

We would like to thank all the authors whose contributions made this book possible, and all the anonymous reviewers whose valuable suggestions ensure the high quality of this book. We hope this book will serve as a useful reference for studying mobile ad hoc and sensor networks.

January, 2008

Maggie Cheng
Deying Li

Chapter 1

Backbone Quality-of-Service Routing Protocol for Heterogeneous Mobile Ad Hoc Networks

Xiaojiang Du
Department of Computer Science
North Dakota State University, Fargo, ND 58105, USA
E-mail: `xdu@web.cs.ndsu.nodak.edu`

1 Introduction

Mobile Ad hoc NETworks (MANETs) form a class of dynamic multihop networks consisting of a set of mobile nodes that intercommunicate on shared wireless channels. MANETs are self-organizing and self-configuring multihop wireless networks, where the network structure changes dynamically due to node mobility. Quality-of-Service (QoS) routing is important for a mobile network to interconnect wired networks with QoS support (e.g., Internet). QoS routing is also needed in a standalone mobile ad hoc network for real-time applications, such as voice, video, and so on.

QoS routing requires not only finding a route from a source to a destination, but a route that satisfies the end–to–end QoS requirement, often given in terms of bandwidth or delay. QoS routing in wired networks has been well studied. Some of the recent works are listed below. In [5], Xue proposed an efficient approximation algorithm for minimum-cost QoS multicast routing problems and an efficient heuristic algorithm for unicast routing problems in communication networks. In [25], Orda and Sprintson proposed efficient precomputation schemes for QoS routing in networks with topology aggregation by exploiting the typical hierarchical structure of large-scale

M.X. Cheng, D. Li (eds.) *Advances in Wireless Ad Hoc and Sensor Networks.*
Signals and Communication Technology, doi: 10.1007/978-0-387-68567-0_1.

networks. In [16], Cao et al. studied QoS for Voice-over-IP, and they proposed measurement-based call admission control to guarantee the QoS. In [21], Li and Mohapatra proposed QoS-aware Routing protocols for Overlay Networks (QRONs).

Quality of service is more difficult to guarantee in ad hoc networks than in most other types of networks, because the network topology changes as the nodes move and network state information is generally imprecise. This requires extensive collaboration between the nodes, both to establish the route and to secure the resources necessary to provide the QoS. In recent years, several researchers have studied QoS support in ad hoc networks [1–4,11–14,25]. QoS needs a set of service requirements to be met by the network while transporting a packet stream from source to destination. The ability to provide QoS heavily depends on how well the resources are managed at the MAC layer. Some QoS routing protocols [4,25] use generic QoS measures and are not tuned to a particular MAC layer. Some QoS routing protocols [2,3,14,15] use CDMA to eliminate the interference between different transmissions. In this chapter, we develop a QoS routing protocol: B-QoS for heterogeneous mobile ad hoc networks using pure TDMA. In [2,3], CDMA is overlaid on top of the TDMA to reduce interference; that is, multiple transmissions can share TDMA slots via CDMA. With some minor modification, our B-QoS routing protocol can also be applied to ad hoc networks with a MAC layer using CDMA/TDMA.

In MANETs, QoS issues include delay, delay jitter, bandwidth, probability of packet loss, and so on. In this chapter, we are mainly concerned about bandwidth. The goal is to establish bandwidth-guaranteed QoS routes in MANETs. Our B-QoS is an on-demand routing protocol, and builds QoS routes only as needed. A flow specifies its QoS requirement as the number of transmission time slots it needs from a source to a destination. For each flow, the B-QoS routing protocol will find both the route and the transmission time slots for each node on the route.

The rest of the chapter is organized as follows. In Section 2, we review the related work of QoS routing, best-effort backbone routing, and routing with location information. In Section 3, we discuss the algorithm that calculates the available maximum bandwidth in a given path. We describe our B-QoS routing protocol in Section 4 and give a routing example there. In Section 5, we discuss the simulation experiments performed with the B-QoS routing protocol and we compare the performance with another QoS routing protocol [1] and the best-effort AODV routing protocol [17]. In Section 6, we estimate the probability of having backbone nodes in one cell by both simulation and computation. And we conclude the chapter in Section 7.

2 Related Works

Most existing QoS routing protocols assume homogeneous MANETs: all nodes have the same communication capabilities and characteristics. They have the same (or similar) transmission power (range), bandwidth, and processing capability, and the same reliability and security. However, a homogeneous ad hoc network suffers from poor scalability. Recent research has demonstrated its performance bottleneck both theoretically and through simulation experiments and testbed measurement [7]. In many realistic ad hoc networks, nodes are not homogeneous. For example, in a battlefield network, there are soldiers carrying portable wireless devices, there are vehicles and tanks carrying more powerful and reliable communication devices, and there may be aircraft and satellites flying above, covering the whole battlefield. They have different communication characteristics in terms of transmission power, bandwidth, processing capability, reliability, and so on. So it would be more realistic to model these network elements as different types of nodes. Also there are many advantages that can be utilized to design better routing protocols when nodes in heterogeneous MANETs are modeled as different types.

The major difference between our B-QoS routing protocol and other QoS routing protocols [1–4,11–14,25] is: B-QoS considers heterogeneous MANETs, whereas other QoS routing protocols consider homogeneous MA-NETs. B-QoS routing takes advantage of the different communication capabilities of heterogeneous nodes in many ad hoc networks. Some physically more powerful nodes are chosen as backbone nodes for routing. The idea of using backbones in routing has appeared in several previous works. The CEDAR [13] algorithm establishes and maintains a routing infrastructure called core-in ad hoc networks. And routing is based on the core. There are several differences between CEDAR and our B-QoS routing protocol. We list some of the differences in the following. (1) CEDAR considers homogeneous nodes, whereas B-QoS considers heterogeneous nodes. The heterogeneous node model is more realistic and provides efficient routing. (2) In CEDAR, a complex algorithm is used to generate and maintain the core nodes, and the algorithm introduces large overhead, because every node needs to broadcast messages to its neighbors periodically. While in B-QoS, the election of backbone nodes is very simple, the first backbone-capable (more powerful) node that sends out a claim message becomes the backbone node. (3) In addition, CEDAR needs to broadcast a route probe packet to discover the location of a destination node. While in B-QoS, a Global Positioning System (GPS) is used to provide node location information, and an efficient algorithm is used to disseminate node location information. The idea of using backbone nodes in routing has also appeared in [8], where Butenko et al. proposed to compute a virtual backbone (a minimum connected dominating set) based on physical topology. In addition, Butenko et al. [8] consider homogeneous node models.

There are several best-effort (non-QoS) routing protocols that consider heterogeneous MANETs. One obvious difference is that B-QoS is a QoS support routing protocol, whereas these routing protocols do not consider the QoS issue. Besides, there are some other differences. We compare our B-QoS with some of these best-effort routing protocols in the following.

In [7], Xu, Hong, and Gerla proposed an MBN routing protocol with backbone nodes. Besides the above difference, the major differences between our B-QoS routing and MBN are the way to deploy backbone nodes and the routing algorithm for backbone nodes. In MBN, a multihop clustering scheme is used to form clusters in the network, and the cluster heads become the backbone nodes. However, the multihop clustering algorithm is complex. In B-QoS routing, the backbone node deployment is based on node location information. The entire routing area is divided into several small equal-size squares—cells—and one backbone node is elected in each cell. A simple algorithm is used for backbone node election. In MBN, routing among backbone nodes is based on another routing algorithm, LANMAR [19], which is not trivial. Furthermore, LANMAR uses a logical group concept to aid routing. However, the logical group is not applicable to all MANETs. In B-QoS routing, routing among backbone nodes is based on node location information and the cell structure: some cells between source and destination are chosen as routing cells, and a route is discovered among backbone nodes in the routing cells. Details are given in Section 3.

Several papers have discussed the node heterogeneity problem [14,15,30]. However, they mainly discuss how to solve the unidirectional link problem in ad hoc networks. In B-QoS routing, we consider how to take advantage of the different communication capabilities of heterogeneous nodes and provide a better QoS routing strategy. The unidirectional link problem also exists in B-QoS routing, that is, the connection from source or destination to a nearby backbone node. Usually the source (or destination) is close to the nearest backbone node, and it is only a small number of hops to the backbone node.

We solve the unidirectional link problem as follows. When there is a packet that needs to be sent, the source (or destination) node floods the packet within a small area to find a path to the nearest backbone node.

In [9], Ye et al. proposed a scheme to build a reliable routing path by controlling the positions and trajectories of some reliable nodes. Ye et al. [9] mainly consider how to build reliable best-effort routes. In our B-QoS routing, we do not assume control of the positions and trajectories of backbone (reliable) nodes, and our goal is to establish efficient and effective QoS routes.

Research has shown that geographical location information can improve routing performance in ad hoc networks. Routing with assistance from geographic location information requires each node to be equipped with a GPS device. This requirement is quite realistic today because such devices are inexpensive and can provide reasonable precision. Several routing algorithms based on location information have been proposed. The well-known location-based routing algorithms are the Location-Aided Routing (LAR) protocol [10], Distance Routing Effect Algorithm for Mobility (DREAM) [28], and Greedy Perimeter Stateless Routing (GPSR) [27], among others.

B-QoS routing utilizes node location information to simplify the routing strategy. The entire routing area is divided into several cells. The cell or grid structure has been utilized in some routing algorithms such as GRID [20], GAF [24], and so on. There are several differences between B-QoS and these algorithms. The major difference is that B-QoS considers QoS routing in heterogeneous MANETs, whereas GRID and GAF consider best-effort routing in homogeneous MANETs. The design of B-QoS is based on the following assumptions.

1. In B-QoS routing, we assume the routing area is fixed (i.e., nodes move around in a fixed territory). This is true for many MANETs, such as ad hoc networks in military battlefields, disaster relief fields, conferences, convention centers, and so on. They all have a fixed routing territory.

2. We consider MANETs whose topologies do not change very quickly. We also assume the routing area is fixed in 1; this means that we mainly consider MANETs where nodes do not move very quickly. If the topology of an ad hoc network changes too quickly, the provision of the QoS can be even impossible [4]. In [22], the authors called an ad hoc network combinatorially stable if and only if the topology changes occur sufficiently slowly to allow successful propagation of all topology updates as necessary. Combinatorial stability follows directly when the geographical distribution of the mobile nodes do not change much relative to one another during the time interval

of interest. In this chapter, we only study the type of ad hoc networks whose topologies do not change so quickly that they make the QoS routing meaningless.

3. We assume there are a reasonable number of backbone-capable nodes in the network: for example, the number of backbone-capable nodes is close to (or larger than) the number of cells in the network.

3 The Path Bandwidth Calculation Algorithm

In a time-slotted network (e.g., TDMA), to provide a bandwidth of B slots on a given path P, it is necessary that every node along the path find at least B slots to transmit to its downstream neighbor, and that these slots do not interfere with other transmissions. Because of these constraints, the end-to-end bandwidth on the path is not simply the minimum bandwidth on the path.

In general, to compute the available bandwidth for a path in a time-slotted network, one not only needs to know the available bandwidth on the links along the path, but also needs to determine the scheduling of the free slots. To resolve slot scheduling at the same time as available bandwidth is searched on the entire path is equivalent to solving the Satisfiability Problem (SAT) which is known to be NP-complete [23]. In [1], Zhu and Corson developed a heuristic algorithm — Forward Algorithm (FA) — to compute the available bandwidth in a path. In [3] Lin and Liu also proposed a heuristic approach to calculate the path bandwidth. In this chapter, the focus is not on developing a new bandwidth calculation algorithm. Instead, we use the existing bandwidth calculation algorithms. Most bandwidth calculation algorithms that are developed for time-slotted networks can be incorporated into B-QoS. In the current design and simulation, our B-QoS routing protocol adopts the FA algorithm in [1] to calculate the available path bandwidth and slot scheduling at each node in the path. The FA algorithm is a greedy scheme that finds the local maximal bandwidth from the source to the next hop, given the sets of slots used on the three links closest to the current node. We briefly state the FA algorithm in the following. Consider a given path $P = (n_m \rightarrow, \ldots, n_{k+3} \rightarrow n_{k+2} \rightarrow n_{k+1} \rightarrow n_k \rightarrow, \ldots, n_1 \rightarrow n_0)$, where n_m is the source, and n_0 is the destination. Based on the input from the upstream node n_{k+2}, an intermediate node n_{k+1} computes the slot allocations at links $n_{k+3} \rightarrow n_{k+2}$ and $n_{k+2} \rightarrow n_{k+1}$, and determines the available bandwidth from the source to itself. Then node n_{k+1} passes the two slot allocations and its free transmission slots as the input to the next node, n_k. Node n_k computes the slot allocations at links $n_{k+2} \rightarrow n_{k+1}$ and $n_{k+1} \rightarrow n_k$, and determines the available bandwidth from the source to itself. Note the slot allocation at link $n_{k+2} \rightarrow n_{k+1}$ is computed twice, by both node n_{k+1} and n_k. Only the one computed at n_k is used to determine the final slot allocation. (The one computed at n_{k+1} is just used as an input to node n_k.) Node n_k stores the slot allocation at link $n_{k+2} \rightarrow n_{k+1}$. Then node n_k passes the input to the next node, and the process continues until the destination n_0 is reached. In the QoS routing, the destination n_0 will send a route reply message via the reverse path to source n_m, and each intermediate node with reserve the slot according to the computed slot allocation. The details of the FA algorithm can be found in [1].

4 QoS Routing Based on Backbone Nodes

Many real-world ad hoc networks are heterogeneous MANETs, where physically different nodes are present. Thus it would be more realistic to model nodes in such networks as different types of nodes. For simplicity, we consider there are only two types of nodes in the network. One type of node has a larger transmission range (power) and bandwidth, better processing capability, and is more reliable and robust than the other type. We refer to the more powerful nodes as Backbone-Capable nodes, in short as BC-nodes. In B-QoS routing, BC-nodes can be elected to serve as Backbone nodes (B-nodes). Other nodes are referred to as general nodes. For example, in a battlefield MANET, tanks and vehicles can be considered as BC-nodes, and soldiers can be considered as general nodes. There might be more than two types of nodes in a heterogeneous MANET. It is possible to extend B-QoS routing to consider more than two types of nodes, and this will be our future work. In this chapter, we only consider the two types of node model.

The main idea of B-QoS routing is to find a QoS route mainly based on B-nodes. There are several advantages of using B-nodes in QoS routing.

- B-nodes have larger bandwidth than general nodes. The large bandwidth of B-nodes increases the chance of satisfying the QoS requirement.
- B-nodes have larger transmission range than general nodes, which reduces the number of hops in routing, and thus reduces the routing overhead and latency.

- B-nodes have better processing capability than general nodes. Routing packets via B-nodes is more efficient than via general nodes.
- B-nodes provide better reliability and fault tolerance, because they are more reliable than general nodes.

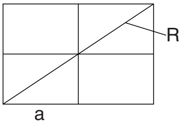

Figure 1: The relationship between a and R.

Usually the transmission range of B-nodes, R, is much larger than that of general nodes, r. For simplicity, we assume the transmission ranges of B-nodes and general nodes are fixed. The routing area is divided into several small, equal-sized squares referred to as cells. An example with nine cells is shown in Figure 2.

If the side length of a cell is set as $a = R/2\sqrt{2}$, as shown in Figure 1, where it is the worst case (longest distance) between B-nodes in two nearby cells, then a B-node can always directly communicate with B-nodes in all nearby cells, including the diagonal one. Because most of the time, two nearby B-nodes are not in the two opposite corners, a larger cell size can be used (i.e., $a > R/2\sqrt{2}$), and still usually ensure the connection of nearby B-nodes. A more detailed discussion of cell size is given in Section 5.6. All the cells form a grid structure, and the grid structure is fixed for a given cell size. One and only one B-node is elected and maintained in each cell if there are BC-nodes available in the cell. In B-QoS routing, we assume the routing area is fixed, thus for a given cell size a, the position of each cell is also fixed. Given the location (coordinates) of a node, there is a predefined mapping between the node location and the cell in which it lies. For simplicity, we assume the routing area is a two-dimensional plane. The grid is created starting from the left-top point of the routing area. The B-QoS routing protocol is presented below.

4.1 The Backbone QoS Routing Protocol

The basic operation of the B-QoS routing protocol is now described.

1. There is a unique ID for each cell. In Figure 2, the number is the ID for each cell. One (and only one) B-node is elected and maintained in each cell, and each B-node has a second address, which is the same as the ID of the cell where it stays. So a B-node can send a packet to a B-node in a nearby cell by using the second address, even though the identity of that B-node may change.

2. There is no routing table maintained among B-nodes and general nodes. The QoS route is discovered on demand. When a B-node moves out of a cell, it initiates a B-node election process in the cell and a BC-node will be elected as the new B-node. The B-node election algorithm is described in Section 4.2.

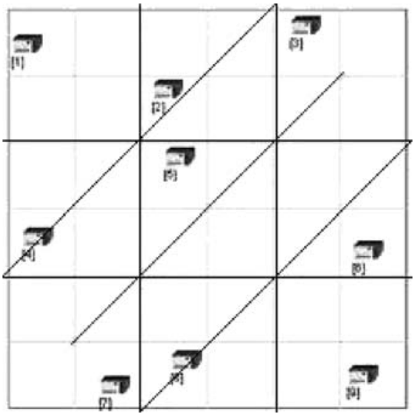

Figure 2: Routing cells.

3. Routing among B-nodes. In this step, we discuss the scheme by which B-nodes find routes (without QoS requirement) to other B-nodes. This scheme is used in step 10 for the dissemination of node location information, and it is also the base for QoS routing in step 4. B-nodes use their second addresses to communicate with each other. Assume B-node B_s (in cell C_s) wants to send a packet to the B-node in cell C_d (denote the B-node as B_d). Although nodes move around, the cells are fixed. B_s knows B_d's second address because it is the same as the ID of cell C_d. A straight line H is drawn between the centers of cell C_s and cell C_d. An example is given in Figure 2, where B-node 7 wants to send a packet to the B-node in cell 3. The center line is line H.

Two border lines (outside lines in Figure 2) which are parallel to line H with distance of W from H are drawn from cell C_s to C_d. The set of all the cells that are (fully or partially) within the two border lines is defined as routing cells. The value of W determines the width of the routing cells. The proper value of W depends on the density of BC-nodes in the network. If there are enough BC-nodes in the network (i.e., with high probability there is at least one BC-node in each cell), then W can be small. For QoS routing, W also depends on the available bandwidth of the B-nodes and the bandwidth requirement of the QoS session. If the B-nodes have enough available bandwidth, or if the QoS bandwidth requirement is low, then a QoS route can be found easily, and W can be small. Otherwise, large W should be used. After determining the routing cells, source B-node B_s can start sending packets to B_d. If the packet is a short one (like the location request/update packet in step 10), B_s will flood the packet to all B-nodes in the routing cells, and the packet will be forwarded to B_d. If it is a long packet, like a data packet, first a route request packet is flooded to all B-nodes in the routing cells, then the data packet is sent via the discovered route. In both cases, some B-nodes in the routing cells form a route from $B_s \rightarrow B_d$. Consider the example in Figure 2: if W is set to zero, then the routing cells are only the cells that intercept with red line H, cells 7, 5, 3. The B-nodes from the routing cells form a route: $B_7 \rightarrow B_5 \rightarrow B_3$. And if W is set as $a\sqrt{2}/2$, where a is the side length of a cell, then the routing cells are cells 7, 5, 3; 4, 2; and 8, 6. The routing cells are used to balance the chance of finding a (QoS) route and the overhead from route discovery. The width of the routing cells is based on the network state. In the current B-QoS, W is based on the number of BC-nodes in the network and the QoS requirement of a session. The information of the available bandwidth in other B-nodes may also be used to determine W, with some scheme to disseminate such information among B-nodes. However, for simplicity, we do not use such information in our current design. A proper W should provide a high probability of finding the (QoS) route while limiting the routing overhead.

4. QoS route discovery starting from B-nodes. Assume a source node S (in cell C_s) wants to set up a QoS route for a flow to a destination node D (in cell C_d). We first discuss the case where S is a B-node. And we discuss the QoS routing scheme when S is a general node in

step 9. S is the first B-node in the QoS route, and is referred to as the starting B-node. In B-QoS routing, the starting B-node S needs to know the current location of the destination node D. The scheme by which S obtains D's location is described in step 10. With D's location information, S knows the cell C_d in which D stays, and S knows the B-node in cell is B_d (using the second address of the B-node). First S determines the width of the routing cells (2W), based on the number of BC-nodes in the network and the QoS requirement from the flow. Then S determines the routing cells between cell C_s and C_d as in step 3. The routing cells may also include the circle that centers at node D with the radius being the expected moving distance, like the scheme used in the LAR routing protocol [10]. Based on assumption 2 that nodes do not move very fast, usually node D is within the transmission range of the B-node in C_d.

5. The starting B-node S floods Route Request (RR) packets to all the B-nodes in the routing cells. The RR packet includes the following fields: starting B-node, sequence-n, route, routing-cells, slot-set-list, destination-cell, RB, where RB is the required bandwidth. Each B-node maintains a sequence-n, and the sequence-n increases for each RR flooding. Starting B-node plus sequence-n uniquely determines a route request session. The route field records the path that the RR packet traversed. At each node, the slot-set-list records the free slots at the node and the slot allocations at the two upstream links. For example, consider a route $(n_m \rightarrow \ldots, n_{k+3} \rightarrow n_{k+2} \rightarrow n_{k+1} \rightarrow n_k \rightarrow \ldots, n_1 \rightarrow n_0)$. At node n_{k+1}, the slot-set-list records the free slots at n_{k+1}, plus the slot allocations at link $n_{k+3} \rightarrow n_{k+2}$ and $n_{k+2} \rightarrow n_{k+1}$, which are computed by the FA algorithm. When receiving a RR packet, an intermediate B-node uses the FA algorithm to compute the maximum bandwidth from the source to itself, based on its free slots and the slot-set-list from its upstream node. If the maximum bandwidth is less than the required bandwidth, then the QoS cannot be satisfied, and the RR packet is dropped. Otherwise, the B-node n_k computes the slot allocations at two upstream links $n_{k+3} \rightarrow n_{k+2}$ and $n_{k+2} \rightarrow n_{k+1}$, stores the slot allocation of link $n_{k+2} \rightarrow n_{k+1}$, updates the slot-set-list, and forwards the RR packet to neighbor B-nodes that are in the routing cells (except the incoming B-node). When a B-node receives duplicate RR packets of the same route request session, it also processes the RR packet in the same way. This is to increase

the chance of finding a QoS route. It is possible that a detour path may have larger bandwidth than a direct link, such as the example in Figure 3. The direct link on the top only has two slots, whereas the detour path below has a bandwidth of three slots.

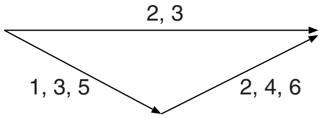

Figure 3: Detour path with larger bandwidth.

6. When the RR packet arrives at the B-node B_d in cell C_d, B_d will first send a probe packet to search the destination node D. The probe packet includes B_d's location. Because B_d has a large transmission range, the transmission of the probe packet can reach all nodes in the neighbor cells. If node D is still in the cell, or in a neighbor cell, D will receive the probe packet. And D will send an Ack (acknowledge) packet including its free slots to B_d. If D is a general node, there is a unidirectional link problem here. D may not be able to send the Ack packet to B_d in one hop. Instead, based on the location of B_d and itself, node D knows the direction to node B_d and the distance between itself and B_d. Node D sends the Ack packet to B_d via limited-hop, small-area directional flooding. When B_d receives the Ack (with D's free slots) from D, B_d computes the maximum bandwidth from source S to destination D and the slot allocations at links $B_{d-1} \rightarrow B_d$(where B_{d-1} is the upstream node of B_d) and $B_d \rightarrow D$. Note the last bandwidth computation is done by B-node B_d, not by D. This approach is more efficient because of the possible unidirectional link between B_d and D. If the required bandwidth cannot be satisfied, B_d will send a route-failure (RF) packet to source S via the reversed route. If the required bandwidth is satisfied, B_d will reserve the slots, and send a route reply (RP) packet (including the uplink slot allocation) along the reversed route back to source node S. Consider the example in step 5: the RP packet from node n_k to upstream node n_{k+1} includes the slot allocation of link $n_{k+1} \rightarrow n_k$, calculated in step 5. An intermediate node n_{k+1} reserves the slots for this route, according to the slot allocation at link $n_{k+1} \rightarrow n_k$ and the required bandwidth. Note the reserved slots (referred to as slot assignment) at n_{k+1} is a subset of the slot allocation.

For example, if the slot allocation is 1,2,3 and the required bandwidth is 2, then only two slots are reserved. The starting B-node B_1 needs to send its slot assignment to source node S, which may be used in route repairing when source S moves away (details are discussed in Section 4.2). When the RP packet arrives at the source node S, the QoS path is set up and the bandwidth is reserved. If node B_d receives multiple RR packets from the same route discovery session, node B_d will reply to two or three of them, and discard the rest. This is to set up one or two backup QoS routes in addition to the primary route. In the case when the primary route is broken, the backup route can be used. If source node S does not receive any RP packet for a route request timeout (or S receives a route-failure packet), S assumes the route discovery within the routing cells failed. S will then flood the RR packet to all B-nodes in the network, and try to find a QoS route based on B-nodes. If even the above scheme fails, S will flood the RR packet to all nodes in the network, which is similar to the QoS routing protocol based on AODV in [1].

7. If B_d does not receive Ack from node D for a certain time, it means D is no longer in the neighbor cells of C_d. Node B_d will obtain the current location information of node D (described in step 10), then it will forward the RR packet to a B-node close to node D, and that B-node will process as above. Because we mainly consider MANETs without high mobility, most of the time, node D will not be far away from B_d. In the case where node D moves to a new location far away from B_d, after having D's new location, B_d will send a route-failure packet (with D's new location) to source node S, and S will start a new QoS route discovery process.

8. If there is no B-node in the destination cell C_d that can be detected by a B-node (say B_{d-1}) in a neighbor cell of the destination cell (i.e., B_{d-1} does not overhear the transmission of a probe packet from B_d after it sends the RR packet to the destination cell for a certain time), then B_{d-1} will flood the RR packet in its cell and the destination cell, and try to find a QoS route via general nodes to destination D. The general nodes (possibly including D) will compute the available bandwidth and slot allocation according to the FA algorithm. If a QoS route is found, D will send the Ack via the reverse route to B_{d-1}, and then node B_{d-1} will send the Ack to source S.

9. QoS route discovery from general nodes. If the source node S is not a B-node, S will first find a route to a nearby B-node with enough bandwidth. Node S floods a Route Discovery (RD) packet to all the nodes in its cell C_s. The RD packet includes the following fields: source, source-cell, sequence-n, path, slot-set-list, destination, and RB, where source-cell is the cell in which source stays, and RB is the required bandwidth. Only nodes in the same cell as S will process and forward the RD packet. This reduces the routing overhead from route discovery. When other general nodes receive the RD packet, they will calculate the available bandwidth from the source to itself and compare it with the required bandwidth. If QoS is satisfied, the node stores the slot allocation, updates the slot-set-list, and forwards the RD packet to its neighbors. Otherwise, the RD packet is dropped. When the B-node in cell C_s receives the first RD packet and if it has enough bandwidth, it will flood the route request packet to all B-nodes in the routing cells, and proceed as in step 4. Because B-nodes have much larger bandwidth than general nodes, most of the time the B-node will have enough bandwidth. In the case where the B-node in cell C_s does not have enough bandwidth, this B-node will send a route-failure packet to the source node directly. (Recall a B-node can directly reach all nodes in its cell.) It is also possible that there is no B-node in cell C_s when S wants to discover a QoS route. The source node S can detect no B-node in C_s if S does not overhead the transmission of RR from a B-node in C_s after S sending out RD for a certain time. If the B-node in cell C_s does not have enough bandwidth, or if there is no B-node in cell C_s, source node S will flood the RD packet to all neighbor cells and find a nearby B-node with enough bandwidth. In the worst case, if all the nearby B-nodes do not have enough bandwidth to continue the QoS route discovery, source node S will flood RD packets to all nodes in the network, and find a QoS route if possible. In the worst case, B-QoS is similar to the QoS routing protocol in [1], which combines AODV with the FA algorithm.

10. Dissemination of node location information. As mentioned in step 4, in B-QoS routing, a starting B-node needs to know the current location of the destination node D. Because nodes move around, an algorithm is needed to disseminate updated node location information. We propose an efficient dissemination scheme, and it is described in the following. If a node moves within the same cell, there is no need to update its

location information. When a node moves out of its previous cell, it sends a location update packet (with its new location) to the B-node in the new cell (or the nearest B-node). The location update packet can be sent out via broadcast within a small hop count. And all B-nodes periodically send aggregated node location information to a special B-node B_0, for example, B_0 could be the command headquarters in a battlefield. The period of updating location information should not be too long, because this will cause the location information to not be accurate. Also the period should not be too short, because updating the location information too often will cause large overhead. The special B-node B_0 is preferred to be a fixed B-node, or a B-node only moving within one cell. If B_0 is fixed or within one cell, the dissemination algorithm is very simple. When a starting B-node S needs to know the location of a node D, S sends a location request packet to B_0; then B_0 sends the location of D to S. Because both S and B_0 are B-nodes, they know how to communicate with each other (step 3). If B_0 also moves around, then if needs to multicast its current location to all B-nodes when it moves from one cell to another. Then all B-nodes know the current location of B_0, and they are able to request location information from B_0. In many MANETs, it is possible to choose a static or slowly moving B-node as B_0. And in many (as with military) MANETs, it worthwhile to deploy a static B-node as B_0.

4.2 More Protocol Details

Route maintenance and election of B-nodes is discussed here.

4.2.1 QoS Route Maintenance

Node mobility can cause an established QoS route to be broken. Route maintenance is very important for QoS routing in MANETs. Assume a QoS route $R : S \to g_1 \to B_1 \to B_2, \ldots, \to B_k \to g_2 \to D, \ldots, n_1 \to n_0)$ is set up between source node S and destination node D, where g_1 and g_2 are general nodes, and $B_j (j = 1, \ldots, k)$ are B-nodes. Usually B-node B_k can send a packet directly to destination D. We add node g_2 to cover a more general case when there is no B-node in the cell of destination node D, and general nodes are used to form the QoS route.

Detection of Broken Route

Moving away (or failure) of any node in the route can cause the route to be broken, and the broken route is detected by the upstream node (closer to the source). That is, after node i sends a packet to its downstream node j, if node i does not overhead a transmission of the packet from node j for a certain time, node i assumes node j moves away or fails. And node i will start the route repairing process; if route repairing does not work, node i will notify source node S to discover a new route. If source S is the node moving away, S itself will detect the broken link. The route repairing and rerouting processes are discussed in the sequel. We refer to the node that moves away and causes a broken link as the leaving node L. The upstream node and downstream node of L are denoted as up-L and down-L respectively.

Route Repairing and Rerouting

There are two different cases of broken routes, depending on whether the leaving node is source node S.

1. The leaving node is source S. If S is still in the same cell, or in a nearby cell, S will flood RE (Route rEpair) packets to nodes in the cell (or plus the nearby cell) and try to find a new QoS path to the starting B-node; the RE packet includes the bandwidth requirement and the slot assignment at the starting B-node B_1. The slot assignment at the upstream node of B_1 must not conflict with the slot assignment at B_1. So if a new QoS path is found between S and B_1, the slot assignments from B_1 to D do not need to change. If S moves far away from its previous cell, or if the route repairing fails, rerouting will be used: S will use the B-QoS routing protocol to discover a new QoS route to destination D.

2. The leaving node is a node other than the source. The upstream node up-L broadcasts a RE packet, which includes its slot assignment and address, with a TTL (Time-To-Live) set as two hops. The address of up-L is used to solve the unidirectional problem. It is possible to have the unidirectional link problem during route repairing. We solve the problem using the similar approach as found in step 6 of Section 4.1. A general node will use limited-hop, small-area directional flooding to send a packet back to a B-node. When down-L receives a RE packet, there are two cases depending on whether the RE packet comes directly from up-L or via an intermediate node.

   1. If the RE packet is directly from up-L, because the original slot assignments at up-L and down-L are conflict-free with each other, a repaired route is found. Node down-L will send a route-repaired packet

to up-L, and the repaired route is: up-L →down-L.

2. If the RE packet is from an intermediate node K, K will add its free slots and address to the RE packet, and down-L will try to find a slot assignment at the intermediate nodes that satisfies the QoS requirement and does not conflict with the slot assignments at nodes up-L and down-L. If found, down-L will send a route-repaired packet to up-L via node K. Node K will reserve the slots for the QoS flow, and route repairing is done. Otherwise, the RE packet is discarded. For example, assume the bandwidth requirement is two slots, and assume the slot assignments at nodes up-L, L and down-L are 1,2; 3,4; 7,8 respectively. When node L moves away, there is another path that connects up-L to down-L, for example, up-L down-L. And node K has free slots 5, 6, which satisfy the QoS requirement and do not conflict with the slot assignments at nodes up-L and down-L. Then a repaired route is found.

If up-L does not receive any route-repaired packet for a certain time, it assumes the route repairing failed. And up-L will send a route-failure packet to source S. Then S will start a new QoS route discovery process. In the above route repairing process, the TTL can be set to a value larger than 2, that is, allow more than one intermediate nodes to relay the QoS flow between up-L and down-L. This will increase the chance of successful route repairing, but it will increase the routing overhead.

4.2.2 Election of B-node

Initially, one B-node is elected in each cell if there are BC-nodes available in the cell. Because B-nodes also move around, an algorithm is needed to elect a new B-node. When a B-node moves out of its current cell, it initiates the B-node election process. When a general node discovers there is no B-node in the cell (as stated in step 8 of the B-QoS routing protocol), for example, because the B-node failed, it initiates the B-node election process. Initially nodes know which node is the B-node in the cell. The election process works as follows. The leaving B-node or the general node floods an election message to all the nodes in the cell. When a BC-node receives the election message, it broadcasts a claim message that claims it will become the B-nodes to all nodes in the cell. Because there is a delay in propagating the claim message to neighbor nodes, several BC-nodes may broadcast during this period. To reduce such concurrent broadcasts, a random timer is used.

Each BC-node defers a random time before its B-node claim. If it hears a claim message during this random time, it then gives up its broadcast. And then one of the BC-nodes T becomes the new B-node in the cell, and T will start using the second address, which is the same as the cell ID. Because all nodes in the cell can hear the claim message, they know that T is the new B-node. This idea is similar to the cluster-head election scheme proposed in [7].

4.2.3 A Routing Example

We present a routing example by using the B-QoS routing protocol in Figure 4, where the routing area is divided into nine cells, and the black boldface number is the cell ID. In Figure 4, the larger grey nodes are B-nodes, and the smaller green nodes are general nodes. In the example, node 2 (in cell 7) wants to set up a QoS session and sends packets to node 41 (in cell 3).

1. Node 2 floods RD packets (black arrows) to all the nodes in its cell. When a general node receives the RD packet, it forwards the RD packet to its neighbors. Only nodes in the same cell as node 2 will process the RD packet as in step 9 of B-QoS routing.

2. When B-node B7 receives the RD packet, first it checks if it has enough bandwidth. In the example assume B7 has enough bandwidth. Then B7 requests the location of destination node 41 from (B4 in the example), and then B7 knows that node 41 is in cell 3 with B-node B3. B7 determines the width of the routing cells to be $a\sqrt{2}/2$, where a is the side length of a cell. The routing cells are the cells between the two blue border lines, that is, cells 4, 2; 7, 5, 3; and 8, 6. Then B7 floods RR packets (red arrows) to all B-nodes in the routing cells, and tries to find a QoS route to destination.

3. When B3 receives a RR packet, it sends a probe packet (blue arrow) to find destination node 41. In this example, node 41 is close to B3, and node 41 sends the Ack back to node B3. Then B3 sends RP back to B7, and to source node 2. A QoS route is set up, and it is $2 \rightarrow 1 \rightarrow B7 \rightarrow B5 \rightarrow B6 \rightarrow B3 \rightarrow 41$.

Routing Latency and Overhead

An important feature in B-QoS routing is to permit most of the transmissions based on B-nodes. Because B-nodes have large bandwidth, it increases

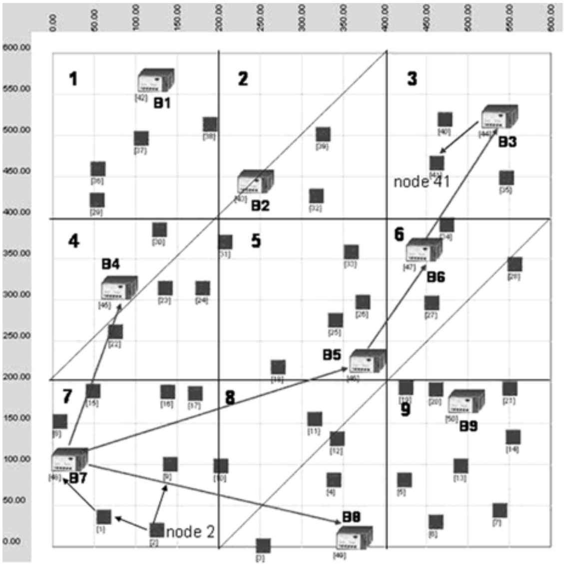

Figure 4: A routing example.

the chance of finding the route that satisfies the QoS requirement. Also B-nodes have a long transmission range, which greatly reduces the hop number in the route.

Based on node location information and cell structure, routing among B-nodes is very efficient (step 3 in Section 4.1). Small hop number and efficient B-node routing ensure B-QoS has low routing latency. Low latency is very important for routing in MANETs, because nodes in MANETs are constantly moving. Low routing latency means the intermediate nodes will not move far away from previous locations when the data packet comes, and this reduces the chance of a broken link. It also means the destination node will not be far away from its previous location when the data packet arrives, which also reduces routing overhead.

The routing overhead in B-QoS routing includes a small-area flooding from the source (or destination) to a nearby B-node, plus B-node route discovery among routing cells. And usually the nearby B-node is close to the source (or destination), for example, in the same or neighbor cell. So the overhead from small-area flooding is not large. We want to point out that usually the number of B-nodes is small (although the number of BC-nodes may be large). Because the transmission range of a B-node, R, is large, the side length of a cell $a = R/2\sqrt{2}$ is also large. Then the number of cells in a fixed routing area is small. Recall that only one B-node is maintained in each cell, thus the number of B-nodes is small. So the overhead from B-node route discovery is limited. The routing overhead from disseminating node location information is also not large. Providing the location of all nodes to a B-node B_0 does not incur much overhead. And other B-nodes request node location information from B_0 only when a QoS route needs to be discovered.

5 Performance Evaluation

The B-QoS routing protocol is implemented in QualNet, a scalable packet-level simulator with an accurate radio model. TDMA is used as the MAC protocol. The transmission rate of the general node and the B-node are 1 Mbps and 4 Mbps, respectively. There are 50 slots in a TDMA frame. For B-nodes, each TDMA slot is further divided into 4 subslots. So there are 200 subslots in the TDMA frame of B-nodes. For the transmission between two B-nodes, a subslot can handle the data transmitted by one slot of a general node. Note: for the transmission from a B-node to a general node, 1 Mbps data rate and slot (not subslot) should be used to avoid overflow at the general node. A subslot is only used between two B-nodes. The simulation testbed that we used consists of 35 general nodes and 15 BC nodes uniformly distributed at random in an area of 600 m × 600 m, which is divided into 9 cells. The radio transmission ranges of the general node and the B-node are 80 m and 320 m respectively. The side length of a cell is set as $a = R/1.6 = 200$ m. The detailed discussion of cell size is given in Section 5.6. Each simulation was run for 600 simulated seconds. The mobility in the environment was simulated using a random-waypoint mobility model. In our simulations, the pause time was set to 1 millisecond (close to 0 second), which correspondsto constant motion. We control the node mobility by varying the maximum node velocities. The maximum velocities range from 0 m/s to 20 m/s. In all simulations, B-nodes have the same mobility as general nodes, also the special B-node B_0 has the same mobility as general nodes.

User traffic is generated with CBR sources, where the source and the destination of a session are chosen randomly among the nodes. The default parameter settings are given below. A particular parameter is varied when we test the QoS routing performance according to the parameter. During its lifetime of 100 seconds, a CBR source generates 20 packets per second. A CBR source does not adjust its transmission according to the network congestion, and all 2000 packets are always transmitted irrespective of how many of them get through. The size of a CBR packet is 256 bytes. The starting time of a session is randomly chosen between 0 and 500 seconds, so a session always ends naturally by the end of the simulation. The offered traffic load is varied by increasing the number of CBR sessions generated during the simulation from 20 to 300. For each simulation configuration, we generate 20 different traffic patterns and get the average results.

We compare our B-QoS routing protocol with the QoS routing protocol proposed in [1], which is referred to as A-QoS. A-QoS has also been implemented in QualNet. We chose A-QoS as the routing protocol for comparison because it has a similar route discovery mechanism as B-QoS. One of the performance metrics is the "serviced session," which is used in [1]. A session is called "serviced" if at least 90% of the packets are received by the destination. This is an approximate measurement of the quality-of-service provided to the end-user. AODV [17] is also used in the performance comparison because it is a widely used benchmark for MANET routing protocols. AODV is an on-demand best-effort routing protocol that uses flooding to discover the route. The following metrics are used to compare routing performances.

1. *Routing overhead.* Routing overhead is the number of routing-related packets (RR, RE, RP, RF packets, etc.) for each QoS session request. This metric is used to measure the efficiency of the routing protocols. In all the tests, the routing overhead of B-QoS includes the overhead of disseminating node location information. Section 5.1 presents the result of routing overhead comparison.

2. *Success ratio.* The success ratio is the ratio between the number of accepted sessions and the number of session requests. This metric measures the effectiveness of finding QoS routes, and the result is discussed in Section 5.2.

3. *Session good-put.* Session good-put is the number of sessions that are serviced. The session good-put is not the same as the success ratio. Even after a QoS route has been set up, it may become broken during the session because some intermediate nodes move away. Such a session can be counted as an accepted session for the success ratio, but it cannot be counted as a serviced session for good-put if less than 90% of the packets are delivered. Whether the session can be serviced depends on the route repairing or rerouting. Also, a session may be serviced by a best-effort route even if a QoS route is not found. Session good-put is discussed in Section 5.3.

For the simulations presented in Section 5.3, we modified the B-QoS and A-QoS routing protocols so that if B-QoS (or A-QoS) cannot find a QoS route on the first try, a best-effort route will be discovered and used to deliver packets.

4. *Throughput and delay.* These two metrics are used to measure the effectiveness of the routing protocols. Sections 5.4 and 5.5 present the results of throughput and delay comparison.

5.1 Routing Overhead

Routing overhead is the number of routing-related control messages (RR, RE, RP, RF packets, etc.) per QoS connection request. Sending a control packet over one link is counted as one message. If a control packet traverses a route of k hops, k messages are counted. The control packet from route repairing is also included in the overhead. We compare the routing overhead of B-QoS, A-QoS, and AODV for different node mobility. Figure 5 is the average routing overhead per QoS connection request when node maximum speed varies from 0 m/s (the actual value is a small number close to 0) to 20 m/s. Although AODV does not process QoS request, the routing overhead from best-effort AODV routing is still presented for comparison purposes. Figure 5 shows that all the routing overheads increase as node speed increases. Higher mobility causes more broken links, and thus increases

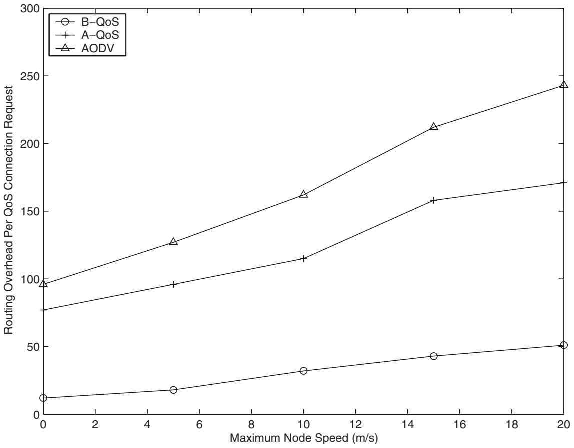

Figure 5: Routing overhead versus mobility.