QoS Issues in Ad Hoc Wireless Networks

Satyabrata Chakrabarti, Lucent Technologies
Amitabh Mishra, Virginia Tech

Abstract

Ad hoc wireless networks consist of mobile nodes interconnected by multihop communication paths. Unlike conventional wireless networks, ad hoc networks have no fixed network infrastructure or administrative support. The topology of the network changes dynamically as mobile nodes join or depart the network or radio links between nodes become unusable. This article addresses some of the quality of service issues for ad hoc networks which have recently started to receive increasing attention in the literature. The focus is on QoS routing. This is a complex and difficult issue because of the dynamic nature of the network topology and generally imprecise network state information. We present the basic concepts and discuss some of the recent results. The article concludes with some observations on the open areas for further investigation.

Introduction

Conventional wireless networks require as prerequisites some form of fixed network infrastructure and centralized administration for their operation. In contrast, the so-called *ad hoc* wireless networks, consisting of a collection of wireless nodes, all of which may be mobile, dynamically create a wireless network among themselves without using any such infrastructure or administrative support [1] (Fig. 1). Ad hoc wireless networks are *self-creating*, *self-organizing*, and *self-administering*. They come into being solely by interactions among their constituent wireless mobile nodes, and only such interactions are used to provide the necessary control and administration functions supporting such networks.

The ad hoc wireless networks offer unique benefits and versatility for certain environments and certain applications. No preexisting fixed infrastructure, including base stations, being prerequisite, they can be created and used "any time, anywhere." Second, such networks could be intrinsically fault-resilient, for they do not operate under the limitations of a fixed topology. Indeed, since all nodes are allowed to be mobile, the composition of such networks is necessarily time-varying. Addition and deletion of nodes occur only by interactions with other nodes; no other agency is involved. Such perceived advantages elicited immediate interest in the early days among military, police, and rescue agencies in the use of such networks, especially under disorganized or hostile environments, including isolated scenes of natural disaster and armed conflict. In recent days, home or small-office networking and collaborative computing with laptop computers in a small area (e.g., a conference or classroom, single building, convention center) have emerged as other major areas of potential application. In addition, people also recognize that ad hoc networking has obvious potential application in all the traditional areas of interest for mobile computing.

Numerous challenges must be overcome to realize the practical benefits of ad hoc networking. These include effective routing, medium (or channel) access, mobility management, power management, security, and, of principal interest here, *quality of service* (QoS) issues, mainly pertaining to delay and bandwidth management [1]. Cost-effective resolution of these issues at appropriate levels is essential for widespread general use of ad hoc networking.

The absence of fixed infrastructure means that the nodes of an ad hoc network communicate directly with one another in a peer-to-peer fashion. The mobility of these nodes imposes limitations on their power capacity, and hence on their transmission range; indeed, these nodes often must satisfy stringent weight limitations for portability. Assuming ubiquitous IP networking as the underlying model for our discussion, it is evident that each node must therefore be able to function as a router as well. As the nodes move in and out of range with respect to other nodes, including those operating as routers, the instantaneous topology changes must somehow be communicated to all other nodes as appropriate. In accommodating the communication needs of the user applications, the limited bandwidth of wireless channels and their generally hostile transmission characteristics impose additional constraints on how much administrative and control information may be exchanged, and how often. Ensuring effective routing is one of the great challenges for ad hoc networking.

0163-6804/01/$10.00 © 2001 IEEE

The lack of fixed base stations in ad hoc networks means that there is no dedicated agency to manage the channel resources for the network nodes. Instead, carefully designed distributed medium access techniques must be used for channel resources; hence, there must be mechanisms available to recover efficiently from the inevitable packet collisions. Traditional carrier sensing techniques cannot be used, and the "hidden terminal" problem may significantly diminish transmission efficiency [2].

All the challenges enumerated above are potential sources of service impairment in ad hoc networks, and hence may degrade the QoS seen by users of the network. As of now, the Internet has only supported best effort service; best effort in the sense that it will do its best to transport the user packets to their intended destination, although without any guarantee. With the Internet as the basic model, the same has also been true for ad hoc networks, especially given their peculiar challenges compared to traditional wireline or even conventional wireless networks. In recent years, however, QoS in ad hoc networks as a research topic has started to receive attention from a growing number of researchers [2–8], and major advances are expected in the next few years.

RFC 2386 [9] characterizes QoS as a set of service requirements to be met by the network while transporting a packet stream from source to destination. Intrinsic to the notion of QoS is an agreement or a guarantee by the network to provide a set of measurable prespecified service attributes to the user in terms of transnetwork delay, delay variance (jitter), available bandwidth, probability of packet loss, and so on.

The Internet of today operates in a connectionless and stateless mode. The network of routers is not aware of any association between the source and destination except on a per-packet basis. Each packet is routed individually without any information about the state of the flow of packets between the source and destination. On the other hand, QoS is meaningful only for a flow of packets between the source and destination, and thus depends on the notion of a logical association, or logical *connection*, between them for the duration of the flow. Second, to attain and preserve the service attributes for such a logical connection, the network must guarantee the availability of a set of resources associated with the flow. Consequently, the routers must remain aware of the logical connection and state of the flow to ensure that adequate network resources (e.g., link bandwidth, nodal buffers, processing power) are available for the duration of the logical connection, and their underlying routes. QoS guarantees can be attained only with appropriate resource reservation techniques. The most important element among them is *QoS routing*, that is, the process of choosing the routes to be used by the flow of packets of a logical connection in attaining the associated QoS guarantee.

QoS for ad hoc networks is a new area of research; much remains to be done. A comprehensive reference is [7], which also contains an exhaustive review of the state of the art circa 1999.

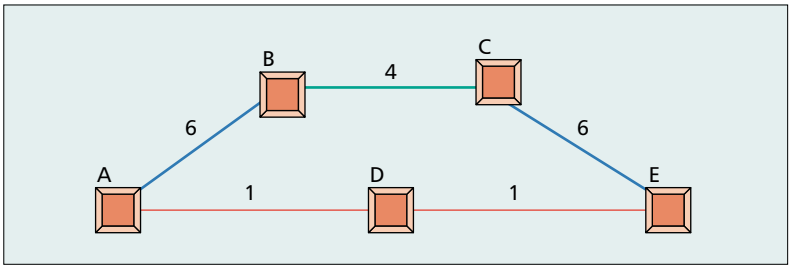

Figure 1. *An ad hoc network example.*

The organization of the rest of the article is as follows. We will present a brief review of the general operation of an ad hoc network and introduce some networking concepts pertinent to QoS. The general issue of QoS routing is reviewed, and we address the QoS routing issues for ad hoc networks and its current state of research. Concluding remarks and some thoughts on future research are included.

Ad Hoc Wireless Networks

Figure 1 depicts the peer-level multihop representation of an ad hoc network. Mobile node A communicates with another such node B directly (single-hop) whenever a radio channel with adequate propagation characteristics is available between them. Otherwise, multihop communication is necessary where one or more intermediate nodes must act as a relay (router) between the communicating nodes. For example, there is no direct radio channel (shown by the lines) between A and C or A and E in Fig. 1. Nodes B and D must serve as an intermediate router for communication between A and C, and A and E, respectively. Indeed, a distinguishing feature of ad hoc networks is that all nodes must be able to function as routers on demand.

An ad hoc network begins with at least two nodes broadcasting their presence (beaconing) with their respective address information. Preferably, they may also include their location information, obtained using a system such as the Global Positioning System (GPS). If node A is able to establish direct communication with node B in Fig. 1, verified by exchanging suitable control messages between them, they both update their routing tables. When a third node C joins the network with its beacon signal, two scenarios are possible. The first is where both A and B establish that single-hop communication with C is possible. The second is where only one of the nodes, say B, recognizes the beacon signal from C and establishes the availability of direct communication with C. The distinct topology updates, consisting of both address and route updates, are made in all three nodes immediately afterward. In the first case, all routes are direct. In the other, the route update first happens between B and C, then between B and A, and then again between B and C, confirming the mutual reachability between A and C via B. The mobility of nodes may cause the reachability relations to change in time, requiring route updates. Assume that for some reason the link between B and C is no longer available in Fig. 1. Nodes A and C are still reachable from each other, although this time

A mobile node may lose connectivity with the rest of the network simply because it has wandered off too far, or its power reserve has dropped below a critical threshold.

only via nodes D and E. All five nodes in Fig. 1 are required to update their routing tables appropriately to reflect this topology change, which will first be detected by nodes B and C, then communicated to A and E, and then to D. As more nodes join the network or some of the existing nodes leave, the topology updates become more numerous, complex, and usually more frequent, thus diminishing the network resources available for exchanging user information.

Finding a loop-free path as a legitimate route between a source-destination pair may become impossible if the changes in network topology occur too frequently. Here "too frequently" means that the network topology changes before the last topology updates are propagated to all the pertinent nodes, or worse, before the completion of determining all loop-free paths accommodating the last topology changes. The ability to communicate degrades with accelerating rapidity, as the knowledge of the network topology becomes increasingly inconsistent. Given a specific time window, we call (the behavior of) 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, therefore, is a critical consideration for QoS in an ad hoc network. 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. Such is the case, for example, in a classroom setting for communication among laptop computers as ad hoc nodes. The routes among network nodes, in such cases, will change little or not at all. There are other cases (e.g., in rescue operations, refugee migrations) where the route updates do occur during the intervals of interest, but not sufficiently frequently to violate the limits of combinatorial stability. In such cases, it is possible that topology updating takes long enough so that by following the now unacceptable characteristics of the last used route, the QoS guarantees cannot be met. Indeed, the old route may even cease to exist during the topology update. This is entirely possible for geographically dispersed networks with a large number of nodes and sparse connectivity, where each route consists of many intermediate nodes like a string of beads.

The topology of an ad hoc network may be combinatorially just right so that QoS guarantees are maintained during any topology updating. It is just not the connectivity that affects the QoS, but equally essential is the availability of enough resources along the previous and new routes during and after the transition. We call an ad hoc network *QoS-robust* with respect to a specific set of QoS guarantees only if such guarantees are maintained *regardless* of the topology updates that may occur within the network. More narrowly, we call such a network *QoS-preserving* if it can continue to maintain the QoS guarantees *during* the interval spanning the end of a successful topology update until the occurrence of the next topology change event. A QoS-robust ad hoc network is, by definition, QoS-preserving; the converse is obviously false.

A mobile node may lose connectivity with the rest of the network simply because it has wandered off too far, or its power reserve has dropped below a critical threshold. Since the occurrence of such events cannot be controlled by the network, we must exclude them in considering QoS-guarantees. Topology update occurs when a new node joins the network or an existing node *deliberately* departs the network. One naturally expects that such topology updates should not affect the QoS for the rest of the nodes as long as the topology of the rest of the network (as a subnetwork) remains unchanged. So far, with the exception of [7], little has appeared on the preservation of QoS guarantees under various failure conditions in ad hoc networks as a specific area of study.

The mobile nodes use some form of multiple access technique with suitable collision avoidance and "hidden terminal" mitigation for accessing the radio resources as mentioned earlier. The larger the number of nodes contending for radio resources, the larger the delay (random variable) in accessing the radio channel for transmitting a packet. Enough reserved radio channel capacity must be available to ensure an upper bound on end-to-end delay as part of QoS.

Quality of Service

The notion of QoS, as mentioned before, is a guarantee by the network to satisfy a set of predetermined service performance constraints for the user in terms of the end-to-end delay statistics, available bandwidth, probability of packet loss, and so on. The cost of transport and total network throughput may be included as parameters. Obviously, enough network resources must be available during the service invocation to honor the guarantee. The first essential task is to find a suitable path through the network, or *route*, between the source and destination(s) that will have the necessary resources available to meet the QoS constraints for the desired service. The task of resource (request, identification, and) reservation is the other indispensable ingredient of QoS. By QoS routing, we mean both these tasks together.

Consider Fig. 1 where the numbers next to the radio links represent their respective bandwidth, say in megabits per second. To minimize delay and better use network resources, minimizing the number of intermediate hops is one of the principal objectives in determining suitable routes. However, suppose that the packet flow from A to E requires a bandwidth guarantee of 3 Mb/s. QoS routing will then select route A–B–C–E over route A–D–E, although the latter has fewer hops.

QoS routing offers serious challenges even for today's Internet [9]. Different service types (e.g., voice, live video, and document transfer) have significantly different objectives for delay, bandwidth, and packet loss. Determining the QoS capability of candidate links is not simple for such scenarios; for multicast services, the difficulties are even larger. We have already noted that the route computation cannot take "too long." Consequently, the computational complexity of route selection criteria must also be taken into account. More than one QoS con-

straint often make the QoS routing problem NP-complete [7, references therein]. Suboptimal algorithms such as sequential filtering are often used, especially for large networks, where an optimal path based on a single primary metric (e.g., bandwidth) is elected first, and a subset of them are eliminated by optimizing over the secondary metric (e.g., delay), and so on, until all the metrics have been taken into account. A random selection is made if there are still more than one choice after considering the network throughput as the last metric. All else remaining the same, as long as the QoS constraints are satisfied, the same route is used for all packets in the flow.

Once a route has been selected for a specific flow, the necessary resources, (bandwidth, buffer space in routers, etc.) must be reserved for the flow. These resources will not available to other flows until the end of this flow. Consequently, the amount of remaining network resources available to accommodate the QoS requests of other flows will have to be recalculated and propagated to all other pertinent nodes as part of the topology update information.

Minimization of routing updates is a principal objective of network engineering, for routing updates consume network bandwidth and router CPU capacity. Second, frequently changing routes could increase the delay jitter experienced by the users. This objective is extremely difficult to attain in wireless networks because of involuntary network state changes as nodes join or depart, traffic loads vary, and link quality swings dramatically. To accommodate real-time traffic needs such as voice or live video, both the overall delay and delay variance must be kept under a certain bound which is accomplished primarily by minimizing as far as possible the number of hops, or intermediate routers, in the path. With potentially unpredictable topology changes in an ad hoc network, this objective is difficult to attain.

QoS routing being dependent on the accurate availability of the current network state, we briefly consider the nature of such information. The first is the *local state* information maintained at each node, which includes queuing delay and the residual CPU capacity for the node, as well as the propagation delay, bandwidth, and some form of cost metric for each of its outgoing links. The totality of local state information for all nodes constitutes the *global state* of the network which is also maintained at each node. The instantaneous network connectivity is part of the global state information. While the local state information may be assumed to be always available at any particular node, the global state information is constructed by exchanging the local state information for every node among all the network nodes at appropriate moments. The process of updating the global state information is also loosely called *topology updates*, and as we have observed already, may significantly affect the QoS performance of the network. The global state update may be done by broadcasting the local state of each node to every other node (*link-state protocol*), or by exchanging suitable "distance vector" information among adjacent nodes only (*distance-vector protocol*) [7]. Since topology updates throughout the network cannot happen instantaneously, the global state information may only be an approximation of the true current network state. For ad hoc networks with highly mobile nodes, the global state information may never be accurate.

Practical considerations for large networks with many nodes and high connectivity sometimes compel the use of so-called *aggregated* global state information, by first partitioning the network into a hierarchical cluster of some form, and then only considering a suitable state information associated with these clusters. Such information is necessarily a partial representation of the true global state. See [8] for an excellent discussion of the use of hierarchically organized clustering for QoS support in ad hoc networks.

Three distinct route-finding techniques are used for determining an optimal path satisfying the QoS constraints. These are *source routing*, *destination routing*, and *hierarchical routing*. In source routing, a feasible path is locally computed at the source node using the locally stored global state information, and then all other nodes along this feasible path are notified by the source of their adjacent preceding and successor nodes. In distributed or hop-by-hop routing, the source as well as other nodes are involved in path computation by identifying the adjacent router to which the source must forward the packet associated with the flow. Hierarchical routing, as the name suggests, uses the aggregated partial global state information to determine a feasible path using source routing where the intermediate nodes are actually logical nodes representing a cluster; for more details see [8]. *Flooding* is not an option for QoS routing, except for broadcasting control packets under appropriate circumstances (e.g., for beaconing, or at the start of a route discovery process).

One may reasonably expect that all packet exchanges will not be treated with equal priority in a QoS network. The exchange of control packets should receive higher priority than user data packets in a network designed for QoS. Indeed, except for instances of "thin" low-traffic (relative to the network capacity) networks, control packets should receive preemptive priority over user data packets. Second, the QoS policy may allow different priorities to exist even among different flows of user packets. Clearly, in accommodating packets with preemptive priorities, the network may not be able to preserve the QoS guarantee for ordinary flows. Indeed, QoS routing admitting preemption is an open area for further research.

Handling of user data with multiple priorities presents difficulties as well. When a user requests QoS with a certain priority, the network first needs to authenticate such a request by exchanging appropriate control packets. (Too many authentication requests, by themselves, may degrade the operational performance of a large QoS network). Next, the network must find a route with the requested QoS for a higher priority against all other flows with lesser priority, even if they are allocated identical QoS parameters in all other respects. In heavy traffic situations, guaranteeing QoS for lesser priority traffic may be extremely difficult or impossible. The development of QoS routing policies, algorithms, and protocols for handling user data with multiple priorities is also an open area.

Minimization of routing updates is a principal objective of network engineering, for routing updates consume network bandwidth and router CPU capacity. Second, frequently changing routes could increase the delay jitter experienced by the users.

The beaconing mechanism lies at the heart of ad hoc networking, for otherwise, a node will not even know its adjacent neighbors which change dynamically in an ad hoc network.

Similar challenges exist in designing QoS routing schemes supporting multiple service classes. For more discussion, see [9]; for additional details, [7; Ch. 3].

We conclude this section with a word on security issues for QoS routing. The objective of a robust security policy is to maintain the operational integrity of a routing protocol against unintended or deliberate attacks. The attacks may appear in the form of flows making too many invalid requests, or requests for inappropriate allocation of network resources, or attempting to mimic or preempt network control functions. This is also an area for further investigation.

Our discussion, up to this point, has been limited to *unicast* routing. The essential problem here is to find a feasible path from a source node to a single destination node that satisfies a set of QoS constraints, and possibly some other additional optimization criteria such as minimal cost and maximum network throughput. The multicast routing problem, on the other hand, is distinguished by more than one destination node, where the objective is to find not a single path, but a feasible *tree* rooted at the source. Each path from the source to one of the destination nodes in the tree is required to satisfy the specified set of QoS constraints; these paths are required to satisfy the additional optimization criteria, if any, simultaneously. As observed in [7], many of the associated optimization problems are NP-complete.

We have presented only a broad-brush overview of the QoS routing. Many issues such as the effect of imperfect knowledge of network state information on routing, and hierarchical aggregation of routing information for scalability, have not been mentioned. All these issues profoundly affect the QoS in ad hoc wireless networks, and are considered in the next section.

QoS Routing in Ad Hoc Networks

The basic concepts of QoS routing discussed in the previous section constitute the foundation for QoS routing for ad hoc networks as in Fig. 1. We assume that each node carries a unique identity recognizable within the network. Our basic reference is [7]. Following [7], we assume the existence of all necessary basic capabilities, such as suitable protocols for medium access control and resource reservation, resource tracking, and state updates. Each node periodically broadcasts a *beacon* packet identifying it (and its pertinent QoS characteristics), thus allowing each node to learn of its adjacent neighbors (i.e., with which it can communicate directly).

The beaconing mechanism lies at the heart of ad hoc networking, for otherwise a node will not even know its adjacent neighbors which change dynamically in an ad hoc network. The knowledge of adjacent neighbors is, of course, indispensable for routing.

Two routing techniques are considered in [7], both limited to combinatorially stable QoS-preserving networks. One is based on the availability of only local state information, and the other assumes possibly inaccurate knowledge of global states. When an existing feasible route becomes unavailable, a new feasible path is determined, and the flow is rerouted to the new feasible path. During the interval immediately following the disappearance of the existing path and the establishment of the new route, data packets are sent as best-effort traffic.

For QoS routing using only the local state information, [7] introduces two different distributed routing algorithms, the so-called *source-initiated routing* and *destination-initiated routing*. Both rely on the use of *probe* packets with appropriate nodal identity and QoS information in identifying a feasible route with the desired QoS characteristics.[1] The probe packets are sent by the source and intermediate routers using a form of flooding. Various mechanisms are considered in [7] to mitigate the penalties of flooding, and the advantages of destination-initiated routing over the other methods established under certain conditions.

The techniques based on imprecise knowledge of global states in [7] uses the notion of *ticket-based* probing for identifying a feasible route. Each probe from the source toward the destination carries at least one ticket to control how many alternate paths to be searched, thus minimizing the routing overhead. The lower the likelihood of finding a route with the desired QoS requirements, the larger the number of tickets carried by the probe. The probes are attempted to be sent along links the QoS characteristics of which are relatively constant (or slowly varying) in time. The basic routing mechanism is distributed or hop by hop; in [7], the information for multiple feasible routes is stored in the probes instead of the intermediate routers.

Multiple mechanisms are considered in [7] for QoS-preserving QoS routing by detecting broken routes and then either repairing the broken route or rerouting the flow on an alternate route with the desired QoS. The likelihood of QoS violation is reduced further by using redundant routes of various kinds. A broken route is detected by using the beaconing protocol for detecting adjacent neighbors. Consider Fig. 2. If node B determines that C is no longer its neighbor because the link between B and C (in red) is broken, it may attempt to repair the route by finding another node E such that by replacing segment B–C with segment B–E–C, the QoS requirement is satisfied between the source *s* and the destination *d*. If no such route segment can be found, B notifies the source that the route is broken. Depending on the network policy, B may send the notification of route unavailability to *s* without attempting to repair the route.

When the source receives the notification of route unavailability, it seeks an alternate route with the same QoS characteristics, as shown in Fig. 3. The unusable route is shown in red, and the new alternate route is shown in blue. If such a route can be found, the flow is rerouted to it after the necessary route updates among the pertinent nodes.

The existence of the QoS route between a source-destination pair needs to be reaffirmed periodically when routing with imprecise information by sending suitably constructed control

[1] *Preestablished network policies should determine the steps to be taken in case no feasible route could be found during the route establishment phase. The service request may be rejected, and the node blocked, or the network may negotiate for a service with lower QoS by exchanging control packets using best-effort routing, assuming that such alternative QoS is available. Such considerations are beyond the scope of this article.*

packets, called *refresher* packets in [7], from the destination back to the source. If such a packet fails to arrive within a predetermined timeout interval, the QoS route is declared unavailable and the associated resources released. This also accommodates the failure to reach various unavailability notifications to their intended recipients using additional timeout mechanisms.

Multiple redundant routing mechanisms are also considered in [7] for minimizing the likelihood of QoS violation due to route failures. Consider Fig. 4. At the highest level of redundancy, multiple alternate routes with the same QoS guarantee are established for the flow, and are used simultaneously. The alternate routes should be preferably disjoint, although this may not always be possible. Duplicate packets are discarded at the destination. At the next lower level of redundancy, the routes and associated resources are reserved and rank-ordered, but not used unless the primary route fails, or the first choice for the alternate route fails while the primary is unavailable, and so forth. When not in use for the QoS-guaranteed flow, the alternate route is used to carry best-effort packets. At the lowest level of redundancy, only the route is identified; no resource is reserved. When the primary path fails, the alternate paths are checked to determine whether the necessary resources are still available. Rerouting is initiated if none of the alternate routes are found to be able to support the desired QoS.

A bandwidth-constrained QoS routing algorithm using a distance vector protocol was proposed in [10], but without accommodating the effects of imprecise network state information. Using concepts from multilayer adaptive control, [8] presented a highly sophisticated approach for controlling QoS in large ad hoc networks by using hierarchically structured multiclustered organizations. The role of cluster dynamics and mobility management, as well as resource reservation and route repair and router movement on QoS are addressed in detail. Two new QoS routing schemes, both based on link state protocols as the underlying mechanism, appear in [3]. Both attempt to reduce routing update overhead, one by selectively adjusting the frequencies of routing table updates, and the other by reducing the size of the update messages by using a hierarchical addressing approach. Another novel approach for QoS routing is advanced in [5] using the notion of a *core* as a self-organizing set of nodes for routing. The overhead of routing update is reduced by decreasing the number of nodes doing route computation and limiting the propagation of link state information for highly transient nodes. The emphasis of [2] is on the medium access mechanism, while [4] proposes a distance-vector-based routing mechanism with focus on bandwidth control, with explicit consideration of broken routes.

Rapid topology changes militate against QoS guarantees. Let τ_u and T_{uc} denote the interval between two consecutive topology change events and the time it takes to complete the calculation and the propagation of the topology updates

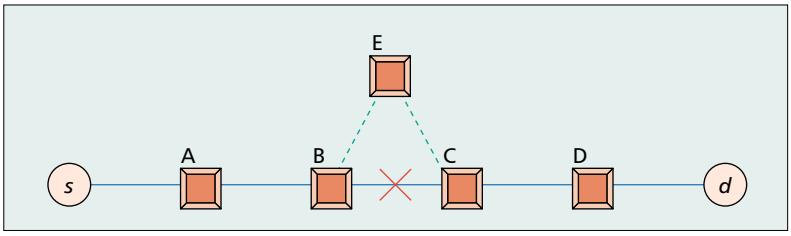

Figure 2. *Route repair.*

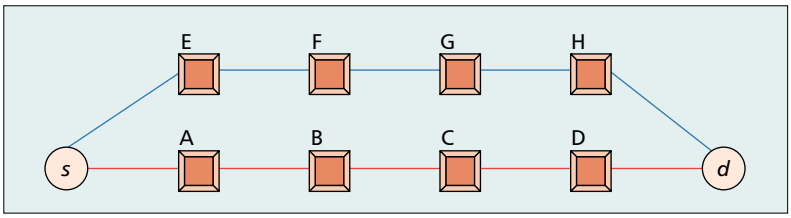

Figure 3. *Alternate routing.*

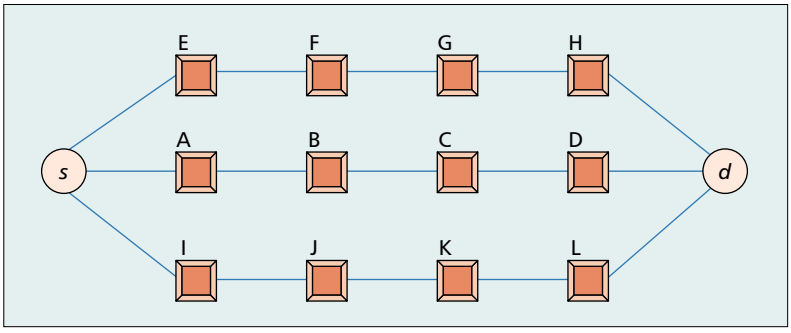

Figure 4. *Redundant routing.*

resulting from the last topology change, respectively. Recall that an ad hoc network is combinatorially stable only if $T_{uc} < \tau_u$.[2] If the just computed feasible route ceases to exist during the corresponding topology update, the QoS guarantee becomes meaningless. Maintaining bounds on delay jitter may also become impractical even in a combinatorially stable network if τ_u remains "close" to T_{uc}. It may be necessary to investigate more rigorous criteria for different *degrees* of combinatorial stability for different QoS constraints.

Consider now the possibility of making the network QoS-robust for a particular flow and its associated QoS constraints so that the guarantees can be maintained during any topology updating. This is clearly impossible as a deterministic objective for an arbitrary ad hoc network; QoS robustness needs to be specified as a probability bound for QoS violation during a topology update the duration of which does not exceed a fixed upper bound. We limit our attention exclusively to combinatorially stable networks. We also assume that connectivity between a node and the rest of the network is never lost because of low battery power, or because the mobile node has wandered far enough away. The smaller the value of T_{uc}, the smaller the probability of QoS violation. In addition, resources must remain available for use whenever necessary. Redundant routing as in Fig. 4 clearly could help accomplish both. Further

[2] *In practice, these are random variables.*

The general issue of QoS-robustness is yet uncharted territory. The same is also true for accommodating traffic with multiple priorities, including preemptive priorities.

studies are necessary to identify completely all the quantitative benefits of this approach. Use of preemptive priority, class of service mechanisms, and segregation of dedicated resources for QoS-robust ad hoc networking also offer promising areas of investigation.

CONCLUSION

We have attempted a brief introduction to the new but rapidly growing area of research on guaranteeing QoS in ad hoc mobile wireless networks. We refer the reader to [7] for a comprehensive treatment of the state of the art circa 1999. The issues are challenging; many of the underlying algorithmic problems are currently perceived as generally intractable (NP-complete). The issues are complicated by the lack of sufficiently accurate knowledge, both instantaneous and predictive, of the states of the network (e.g., the quality of the radio links, and availability of routers and their resources). Indeed, guaranteeing QoS in such a network may be impossible if the nodes are *too mobile*. Even the size of the ad hoc network becomes an issue beyond a certain level, because of the increased computational load and difficulties in propagating network updates within given time bounds. Will the network have to be treated, as some have already suggested [8], as some form of hierarchically ordered collection of subnetworks where at each level the pertinent size is not an issue? Is such an ordering always possible? The challenges increase even more for those ad hoc networks that, like their conventional wireless counterparts, support both best-effort services and those with QoS guarantees, allow different classes of service, and are required to interwork with other wireless and wireline networks, both connection-oriented and connectionless. Algorithms, policies, and protocols for coordinated admission control, resource reservation, and routing for QoS under such models are only beginning to receive attention. The general issue of QoS robustness is yet uncharted territory. The same is also true for accommodating traffic with multiple priorities, including preemptive priorities. We have not even mentioned the issue of network management for ad hoc networks with QoS in the main text; to our knowledge, neither has anyone else. Much work remains to be done on cost-effective implementation issues to bring the promise of ad hoc networks within the reach of the public.

REFERENCES

[1] Z. J. Haas *et al.*, "Guest Editorial," *IEEE JSAC*, Special Issue on Wireless Networks, vol. 17, no. 8, Aug. 1999, pp. 1329–32.

[2] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-Service in ad hoc Carrier Sense Multiple Access Wireless Networks," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1353–1414.

[3] A. Iwata *et al.*, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1369–79.

[4] C. R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1426–38.

[5] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1454–65.

[6] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp. 1488–1505.

[7] S. Chen, "Routing Support For Providing Guaranteed End-To-End Quality-Of-Service," Ph.D. thesis, Univ. of IL at Urbana-Champaign, http://cairo.cs.uiuc.edu/papers/SCthesis.ps, 1999.

[8] R. Ramanathan and M. Steenstrup, "Hierarchically Organized, Multihop Mobile Wireless Networks for Quality of Service Support," *Mobile Network and Apps.*, vol. 3, 1998, pp. 101–19.

[9] E. Crawley *et al.*, "A Framework for QoS-Based Routing in the Internet," RFC 2386, http://www.ietf.org/rfc/rfc.2384.txt, Aug. 1998.

[10] T. Chen, M. Gerla, and J. T. Tsai, "QoS Routing Performance in a Multi-Hop, Wireless Network," *Proc. IEEE ICUPC '97*.

BIOGRAPHIES

SATYABRATA CHAKRABARTI [SM] (schakrabarti@lucent.com) is a technical manager of switching software development at Lucent Technologies. His research interests include multimedia communications over wireless networks with emphasis on service architectures, as well as design, performance analysis, simulation, and software implementation of the associated protocols. After receiving his D.Sc. in network theory from the University of Calcutta he worked as a post-doctoral fellow at the University of California and Concordia University, Montréal, Canada, and as an assistant professor of electrical engineering at Rutgers University, before joining Bell Telephone Laboratories. At Bell Labs he has worked on analysis, design, and software implementation of packet-switched data networks, common channel signaling, ISDN and intelligent network protocols and services, and "video dial tone" technology. His recent focus is on wireless position location, IP telephony, and secure wireless ad hoc networking. He is a life member of Sigma Xi and MAA and a member of ACM.

AMITABH MISHRA [SM] (mishra@vt.edu) is an associate professor of electrical and computer engineering at Virginia Tech. where his main thrust of research is in the area of computer-communication networks architecture and performance. At present he is looking at the architectures of 3G wireless networks (UMTS and CDMA2000), next -generation switch and router architectures, scalability in communication networks, and QoS issues in packet data networks. From 1987 to 2000 he was a member of technical staff at Lucent Technologies – Bell Laboratories, Naperville, Illinois, where his focus was on application architecture performance. POTS, IN/AIN, ISDN, ATM, GPRS, CDMA200, and UMTS were the major areas he worked on while with Bell Laboratories. He received his B.Eng. and M.Tech. degrees in electrical engineering from Jabalpur University and Indian Institute of Technology, Kharagpur, respectively. He obtained an M.Eng. and a Ph.D., also in electrical engineering, from McGill University in Montreal, Canada, and an M.S. in computer science from the University of Illinois at Urbana-Champaign. He is a vice chair of the Communications Software Technical Committee and a member of ACM and SIAM.