A Survey of Mobile Ad Hoc Networks

By

Khurram Rafique
ELEN 6951

Presented to
Dr. Campbell
May 6th, 2002
1. Introduction:

Based on the recent progress and advances in the computing and communications technologies, a pervasive computing environment can be expected in the near future. Next generation of mobile communications will include both infrastructure-d wireless networks and infrastructure-less mobile ad hoc networks (MANET’s). Ad hoc networks will be one of the next storms in the wireless communications—large-area mobile multi-hop wireless and personal area networks.

A mobile ad hoc network is an autonomous system of mobile routers (and associated hosts) connected by wireless links. The routers and hosts are free to move randomly and organize themselves arbitrarily; thus the network’s wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or be connected to the larger Internet.

A mobile ad hoc network or MANET has the following features:

1. Autonomous Terminal: In MANET, each mobile terminal is an autonomous node, which may function as both a host and a router. So usually endpoints and switches are indistinguishable in MANET.

2. Distributed Operation: Since there is no background network for the central control of the network operations, the control and management of the network is distributed among the terminals. The nodes involved in a MANET should collaborate among themselves and each node acts as a relay as needed, to implement functions e.g. security and routing.

3. Multi-hop Routing: Basic types of ad hoc routing algorithms can be single-hop and multi-hop, based on different link layer attributes and routing protocols. Single-hop MANET is simples than multi-hop in terms of structures and implementation, with the cost of lesser functionality and applicability. When delivering data packets from a source to its destination out of the direct wireless transmission range, the packets should be forwarded via one or more intermediate nodes.

4. Dynamic Network Topology: Since the nodes are mobile, the network topology may change rapidly and unpredictably and the connectivity among the terminals may vary with time. MANET should adapt to the traffic and propagation conditions as well as the mobility patterns of the mobile network nodes. The mobile nodes in the network dynamically establish routing among themselves as they move about, forming their own network on the fly. Moreover, a user in the MANET may not only operate within the ad hoc network, but may require access to a public fixed network.
5. Fluctuating Link Capacity: The nature of high bit-error rates of wireless connection might be more profound in a MANET. One end-to-end path can be shared by several sessions. The channel over which the terminals communicate is subject to noise, fading, and interference, and has less bandwidth than a wired network. In some scenarios, the path between any pair of users can traverse multiple wireless links and the link themselves can be heterogeneous.

6. Light-Weight Terminals: In most cases, the MANET nodes are mobile devices with less CPU processing capability, small memory size, and low power storage.

The ad hoc architecture has many benefits, such as self-reconfiguration and adaptability to highly variable mobile characteristics such as power and transmission conditions, traffic distribution variations, and load balancing. However, such benefits come with some new challenges which mainly reside in the unpredictability of network topology due to mobility of nodes, which coupled with the local broadcast capability, cause a set of concerns in designing a communication system on top of wireless ad hoc networks. These challenges include:

1. Routing: Since the topology of the network is constantly changing, the issue of routing packets between any pair of nodes becomes a challenging task. Most protocols should be based on reactive routing instead of proactive. Multicast routing is another challenge because the multicast tree is no longer static due to the random movement of nodes within the network. Routes between nodes may potentially contain multiple hops, which is more complex than the single hop communication.

2. Security and Reliability: In addition to the common vulnerabilities of wireless connection, an ad hoc network has its particular security problems due to e.g. nasty neighbor relaying packets. The feature of distributed operation requires different schemes of authentication and key management. Further, wireless link characteristics introduce also reliability problems, because of the limited wireless transmission range, the broadcast nature of the wireless medium (e.g. hidden terminal problem), mobility induced packet losses, and data transmission errors.

3. Quality of Service (QoS): Providing quality of service levels in a constantly changing environment will be a challenge. The inherent stochastic feature of communications quality in a MANET makes it difficult to offer fixed guarantees on the services offered to a device. An adaptive QoS must be implemented over the traditional resource reservation to support the multimedia services.

4. Power Consumption: For most of the light-weight mobile terminals, the communication-related functions should be optimized for lean power consumption. Conservation of power and power-aware routing must be taken into consideration.
2. Routing in Ad hoc Networks:

The idea of forming an on-the-fly ad hoc network of mobile nodes dates back to DARPA packet radio network days. More recently the interest in this subject has grown due to availability of license-free wireless communication devices that users of laptop computers can use to communicate with each other. Several routing solutions have been proposed that leverage features from existing Internet routing algorithms.

Routing protocols can be classified as proactive or on-demand. In proactive protocols, each node maintains paths at all times to all possible destinations. In on-demand protocols, paths are found only when they are required. Clearly, proactive protocols incur a large signaling overhead and low latency. On the other hand, on-demand protocols save the signaling overhead, but have higher latencies. Routing protocols can also be classified according to whether they find optimal (shortest) routes or sub-optimal routes. By not requiring routes to be optimal, it is possible to reduce the amount of control traffic (including routing updates) necessary to maintain the routes. However, optimal routes are desirable because they minimize delay and the amount of resources (e.g. bandwidth and power) consumed. The protocols can also be grouped into unicast and multicast routing protocols. Below is the description of some of the ad hoc routing protocols:

2.1 Destination Sequenced Distance Vector (DSDV):

Destination sequenced distance vector routing protocol is derived from a classical distance vector algorithm, Distributed Bellman-Ford (DBF) algorithm. Enhancements are made in order to avoid the looping problem present in the basic DBF. Formation of loops is avoided by tagging each route table entry with a sequence number to order the routing information.

In DSDV, each node maintains a routing table, which has an entry for each destination in the network. The attributes for each destination are the next hop, metric (hop counts) and a sequence number, which is originated by the destination node. To maintain the consistency of the routing tables, DSDV uses both periodic and triggered routing updates; triggered routing updates are used in addition to the periodic updates in order to propagate the routing information as quickly as possible when there is any topological change. The update packets include the destinations accessible from each node and the number of hops required to reach each destination along with the sequence number associated with each route.

Upon receiving a route update packet, each node compares it to the existing information regarding the route. Routes with old sequence numbers are simply discarded. In case of route with equal sequence number, the advertised route replaces the old one if it has a better metric. The metric is then incremented by one hop since incoming packet will require one more hop to reach the destination. Newly recorded routes are immediately advertised to its neighbors.

When a link to the next hop is broken, any route through that next hop is immediately assigned an infinity metric and an updated sequence number. This is the only case where sequence numbers are not assigned by the destination. When a node
receives an infinity metric, and it has an equal or later sequence number with a finite metric, a route update broadcast is triggered. Therefore, routes with infinity metric will be quickly replaced by real routes propagated from the newly located destination.

DSDV also employs a mechanism to damp out fluctuations in route table updates. In an environment where many independent nodes transmit routing information asynchronously, some fluctuations could develop. For example, a node could receive two routes to the same destination with the same sequence number, however, the one with the worst metric always arrives first. This could lead to continuous outbursts of route updates and fluctuations of the routing tables. DSDV solves this problem by using “settling time” data. Specifically, time duration until the route becomes stable (termed settling time) is predicted, and the settling time is allowed before advertising any new route information to the network. In other words, the settling time is used to decide how long to wait before advertising new routes. By delaying the advertisement of unstable routes, fluctuations of the routing tables are prevented, and consequently, the number of route updates are reduced.

One of the major advantages of DSDV is that it provides loop-free routes at all instants. However, it has a number of drawbacks. Optimal values for the parameters like maximum settling time for a particular destination are difficult to determine. This might lead to route fluctuations and spurious advertisements resulting in waste of bandwidth. DSDV also uses both periodic and triggered routing updates, which could cause excessive communication overhead. In addition, in DSDV, a node has to wait until it receives the next route update originated by the destination before it can update its routing table entry for that destination. This implicit destination-centered synchronization suffers from the latency problem. Furthermore, DSDV does not support multipath routing. DSDV is effective for creating ad hoc networks for small population of mobile nodes, but it is a fairly brute force approach because it depends for its correct operation on the periodic advertisement and global dissemination of connectivity information. Frequent system-wide broadcasts limit the size of ad hoc networks that can effectively use DSDV because the control message overhead grows as $O(n^3)$. DSDV also requires each mobile node to maintain a complete list of routes, one for each destination within the ad hoc network. This almost always exceeds the needs of any particular mobile node. Keeping a complete routing table does reduce route acquisition latency before transmission of the first packet to a destination. It is, however, possible to design a system whereby routes are created on-demand. Such systems must take steps to limit the time used for route acquisition; otherwise, users of the ad hoc nodes might experience unacceptably long waits before transmitting urgent information.

2.2 Ad Hoc On-Demand Distance Vector Algorithm (AODV):

This protocol proposal can be called a pure on-demand route acquisition system; nodes that do not lie on active paths neither maintain any routing information no participate in any periodic routing table exchanges. Further, a node does not have to discover and maintain a route to another node until the two need to communicate, unless the former node is offering its services as an intermediate forwarding station to maintain connectivity between two other nodes. The advantage here is that a smoothly functioning ad hoc system with on-demand routes could largely eliminate the need for periodic
broadcast of route advertisements. This protocol has the goals of minimizing broadcasts and transmission latency when new routes are needed.

When the local connectivity of the mobile node is of interest, each mobile node can become aware of the other nodes in its neighborhood by the use of several techniques, including local (not system-wide) broadcasts known as hello messages. The routing tables of the nodes within the neighborhood are organized to optimize response time to local movement and provide quick response time for requests for establishment of new routes. The algorithm’s objectives are:

1. To broadcast discovery packets only when necessary.
2. To distinguish between local connectivity management (neighborhood detection) and general topology maintenance.
3. To disseminate information about changes in local connectivity to those neighboring mobile nodes which are likely to need the information.

One distinguishing feature of AODV is its use of a destination sequence number for each route entry. The destination sequence number is created by the destination for any route information it sends to requesting nodes.

Using destination sequence numbers ensure loop freedom and is simple to program. Given the choice between two routes to a destination, a requesting node always selects the one with the greatest sequence number. AODV uses a broadcast route discovery mechanism, as is also used (with modifications) in the Dynamic Source Routing (DSR) algorithm. Instead of source routing, however, AODV relies on dynamically establishing route table entries at intermediate nodes. This difference pays off in networks with many nodes, where a larger overhead is incurred by carrying source routes in each data packet. To maintain the most recent routing information between nodes, the concept of destination sequence numbers is borrowed from DSDV. Unlike in DSDV, however, each ad hoc node maintains a monotonically increasing sequence number which is used to supersede stale cached routes. The combination of these techniques yields an algorithm that uses bandwidth efficiently (by minimizing the network load for control and data traffic), is responsive to changes in topology, and ensures loop-free routing.

Although AODV does not depend specifically on particular aspects of physical medium across which packets are disseminated, its development has been largely motivated by limited range broadcast media such as those utilized by infrared or radio frequency wireless communications adapters. Using such media, a mobile node can have neighbors, which hear its broadcasts yet do not detect each other (the hidden terminal problem). There is no attempt to use specific characteristics of the physical medium in this algorithm, nor to handle specific problems posed by channelization needs of radio frequency transmitters. Nodes that need to operate over multiple channels are presumed to be able to do so. The algorithm works on wired media as well as wireless media, as long as links along which packets may be transmitted are available. The only requirement placed on the broadcast medium is that the neighboring nodes can detect each other’s broadcasts. AODV uses symmetric links between neighboring nodes. It does not attempt to follow paths between nodes when one of the nodes cannot hear the other one.

2.3 Dynamic Source Routing (DSR):
Dynamic source routing is based on source routing, where the source specifies the complete path to the destination in the packet header, and each node along this path simply forwards the packet to the next hop indicated in the path. It utilizes a route cache where source routes it has learned so far are cached. Therefore a source first checks its route cache to determine the route to the destination. If a route is found, the source uses this route. Otherwise the source uses a route discovery protocol to discover a route. A route is sought only when desired by a source. In route discovery, the source floods a query packet through out the ad hoc network, and the reply is returned by either the destination or another host, which can complete the query from its route cache. Each query packet has a unique ID and an initially empty list. When receiving a query packet, if a node has already seen this ID (i.e. duplicate) or it finds its own address already recorded in the list, it discards the copy and stops flooding; otherwise, it appends its own address in the list and broadcasts the query to its neighbors. If a node can complete the query from its route cache, it may send a reply packet back to the source without propagating the query packet further. Furthermore, any node participating in route discovery can learn routes from passing data packets and gather this routing information into its route cache. This route discovery protocol is similar to the Internet’s Address Resolution Protocol (ARP), except that ARP requests do not propagate beyond a router. It is also similar to the route discovery protocol used in source routing bridges in IEEE 802 LAN’s.

A route failure can occur since mobile hosts move from place to place. A route failure can be detected by the link-level protocol (i.e. hop-by-hop acknowledgment), or it may be inferred when no broadcasts have been received for a while from a former neighbor. When a route failure is detected, the node detecting the failure sends an error packet to the source, which then uses route discovery protocol again to discover a new route. Note that in DSR, no periodic control messages are used for route maintenance.

The major advantage of DSR is that there is little or no routing overhead when a single or few sources communicate with infrequently accessed destinations. In such situation, it does not make sense to maintain routes from all sources to such destinations. In DSR, only the sources which desire communication with such destinations need to discover routes. Furthermore, since communication is assumed to be infrequent, a lot of topological changes may occur without triggering new route discoveries (i.e. little or no communication overhead).

Even though DSR is suitable for the environment where only a few source communicate with infrequently accessed destinations, it may result in large delay and large communication overhead in highly dynamic environment. Furthermore, DSR may have scalability problem. As the network becomes larger, control packets and message packets also become larger since they need to carry addresses for every node in the path. This may be a problem since ad hoc networks have limited available bandwidth.

2.4 Temporally-Ordered Routing Algorithm (TORA):

TORA (Temporally-Ordered Routing Algorithm) is a highly adaptive, loop-free, distributed routing algorithm based on the concept of link reversal. TORA is proposed to operate in a highly dynamic mobile networking environment. It is source-initiated and provides multiple routes for any desired source/destination pair. The key design concept
of TORA is the localization of control messages to a very small set of nodes near the occurrence of a topological change. To accomplish this, nodes need to maintain routing information about adjacent nodes. The protocol performs three basic functions:

1. Route creation
2. Route maintenance
3. Route erasure

During the route creation and maintenance phases, nodes use a “height” metric to establish a directed acyclic graph (DAG) rooted at the destination. Thereafter, links are assigned a direction (upstream or downstream) based on the relative height metric of neighboring nodes. In times of node mobility, the DAG route is broken and route maintenance is necessary to re-establish a DAG rooted at the same destination. Upon failure of the last downstream link, a node generates a new reference level which results in the propagation of that reference level by neighboring nodes, effectively coordinating a structured reaction to the failure. Links are reversed to reflect the change in adapting to the new reference level. This has the same effect as reversing the direction of one or more links when a node has no downstream links.

Timing is an important factor for TORA because the “height” metric is dependent on the logical time of a link failure; TORA assumes all nodes have synchronized clocks (accomplished via an external time source such as Global Positioning System). TORA’s metric is a quintuple comprised of five elements, namely:

1. Logical time of a link failure
2. Unique ID of the node that defined the new reference level
3. A reflection indicator bit
4. A propagation ordering parameter, and
5. Unique ID of the node

The first three elements collectively represent the reference level. A new reference level is defined each time a node loses its last downstream link due to a link failure. TORA’s route erasure phase essentially involves flooding a broadcast “clear packet” (CLR) throughout the network to erase invalid routes.

In TORA, there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other. Because TORA uses internodal coordination, its instability problem is similar to the “count-to-infinity” problem in distance-vector routing protocols, except that such oscillations are temporary and route convergence will ultimately occur.

2.5 Zone Routing Protocol (ZRP):

In Zone Routing Protocol, each node has its own “routing zone” which includes the nodes whose distance (hops) is at most some predefined number. Each node is required to know the topology of the network within its routing zone only, and route updates are propagated only within the routing zone. A proactive protocol such as DSDV is used within the routing zone to learn about its topology. To discover a route to an out-of-zone node, a reactive protocol such as DSR is used. Note that ZRP exhibits hybrid behavior of proactive and reactive through the use of the zone radius. For large zone radius, ZRP is more proactive, and for small zone radius, ZRP is more reactive.
The advantage of ZRP is that it significantly reduces the communication overhead as compared to the pure proactive protocols since in ZRP each node needs to know the topology of its zone only. In addition, ZRP discovers routes faster than the pure reactive protocols, since only the peripheral nodes are queried in the route discovery process. It is also noted that the ZRP path, which consists of nodes spaced approximately by distance of zone radius, is more sable than the full path, which consists of all the nodes between the source and the destination. This is because there are some topological changes that affect the full path, but not the ZRP path.

Apart from the unicast routing protocols described above, a lot of work has been done in the area of multicast protocols for ad hoc networks. Some of the multicast routing protocols for ad hoc networks are:

1. Distance Vector Multicast Routing Protocol (DVMRP) for a wireless environment
2. Core-Based Trees (CBT) based protocol
3. Protocol Independent Multicast (PIM) based protocol
4. Forwarding Group Multicast Protocol (FGMP)
5. Hyper Flooding
6. On-Demand Multicast Routing Protocol (ODMRP) etc.

3. Security in Mobile Ad Hoc Networks:

Security in mobile ad hoc networks is particularly difficult to achieve, because of the vulnerability of the links, the limited physical protection of each of the nodes, the sporadic nature of connectivity, the dynamically changing topology, the absence of a certification authority, and the lack of a centralized monitoring or management point. Security requirements depend very much on the kind of mission for which the mobile ad hoc network has been conceived, and the environment in which it has to operate. For example, a military mobile ad hoc network certainly will have very stringent requirements in terms of confidentiality and resistance to denial of service attacks. Mechanisms to encourage cooperation between nodes can be highly desirable in a civilian context, whereas they do not make much sense in their military counterpart. Moreover, anonymity will usually be desirable in both military and civilian contexts, but with different flavors: in the case of the battlefield, it is important to hide the location of the headquarters, whereas in a commercial scenario, a consumer may wish to protect his privacy with respect to a given service provider or merchant. Also, a network of sensors will generally have security requirements that are quite different from ad hoc networks comprised of personal communication devices.

3.1 Framework for Security Goals:

Security is an important issue for ad hoc networks, especially for security-sensitive applications. To secure an ad hoc network, the following attributes can be considered: availability, confidentiality, integrity, authentication, and non-repudiation.
Availability ensures the survivability of network services despite denial-of-service attacks. A denial-of-service attack could be launched at any layer of an ad hoc network. On the physical and media access control layers, an adversary could employ jamming to interface with communication on physical channels. On the network layer, an adversary could disrupt the routing protocol and disconnect the network. On the higher layers, an adversary could bring down high-level services. One such target is the key management service, an essential service for any security framework.

Confidentiality ensures that certain information is never disclosed to unauthorized entities. Network transmission of sensitive information, such as strategic or tactical military information, requires confidentiality. Leakage of such information to enemies could have devastating consequences. Routing information must also remain confidential in certain cases because the information might be valuable for enemies to identify and locate their targets in a battlefield.

Integrity guarantees that a message being transferred is never corrupted. A message could be corrupted because of benign failures, such as radio propagation impairment, or because of malicious attacks on the network.

Authentication enables a node to ensure the identity of the peer node with which it is communicating. Without authentication, an adversary could masquerade as a node, thus gaining unauthorized access to resource and sensitive information and interfering with the operation of other nodes.

Non-repudiation ensures that the origin of a message cannot deny having sent the message. Non-repudiation is useful for detection and isolation of compromised nodes. For example, when node A receives an erroneous message from node B, non-repudiation allows node A to accuse B using this message and to convince other nodes that B is compromised.

3.2 Vulnerabilities and Challenges:

The salient features of ad hoc networks pose both challenges and opportunities in achieving these security goals.

First, use of wireless links renders an ad hoc network susceptible to link attacks ranging from passive eavesdropping to active impersonation, message replay, and message distortion. Eavesdropping might give an adversary access to secret information, violating confidentiality. Active attacks might allow the adversary to delete messages, to inject erroneous messages, to modify messages, and to impersonate a node, thus violating availability, integrity, authentication, and non-repudiation.

Second, nodes roaming in a hostile environment, with relatively poor physical protection, have non-negligible probability of being compromised. Unlike nodes of conventional (wire-line) networks, nodes of ad hoc networks cannot be secured in locked cabinets; they risk being captured and compromised. Therefore, we should not only consider malicious attacks from outside a network, but also take into account the attacks launched from within the network by compromised nodes. Therefore, to achieve high survivability, ad hoc networks should have a distributed architecture with no central entities. Introducing any central entity into a security solution could lead to significant vulnerability; that is, if this centralized entity is compromised, the entire network is subverted.
Third, an ad hoc network is dynamic because of frequent changes in both its topology and its membership (i.e., nodes frequently join and leave the network). Trust relationships among nodes also change, for example, when certain nodes are detected as compromised.

Finally, algorithms used in ad hoc networks are assumed to be cooperative. For example, in a MAC layer, nodes are expected to cooperate. In a contention-based mechanism, nodes must follow the predefined rules to avoid collisions or recover from them. In a contention-free mechanism (which is better suited to ad hoc networks), each node must obtain an agreement from all the others for an exclusive use of the channel resource. In both cases, if a node does not respect the rules, the allocation of the communication channel will be unfair and the performance of the network can be severely affected.

To protect an ad hoc network, we can use intrusion detection as our first line of defense. Then we have to secure the network from the attacks on basic mechanisms, such as routing. Prevention of these attacks requires security mechanisms that are often based on cryptographic algorithms. We also have to protect the security mechanisms of the network, such as the key management system.

### 3.3 Intrusion Detection in Ad hoc Networks:

When an intrusion (defined as “any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource”) takes place, intrusion prevention techniques, such as encryption and authentication (e.g., using passwords), are usually the first line of defense. However, intrusion prevention alone is not sufficient. Intrusion detection can be used along with intrusion prevention as a second wall to protect network systems because once an intrusion is detected, e.g., in the early stage of a denial-of-service attack, response can be put into place to minimize damages, gather evidence for prosecution, and even launch counter-attacks.

Intrusion detection assumes that users and program activities are observable; and more importantly, normal and intrusion activities have distinct behavior. Therefore, intrusion detection involves capturing audit data and reasoning about the evidence in the data to determine whether the system is under attack.

The differences between ad hoc and wired networks make it difficult to apply intrusion detection techniques developed for a fixed network wired network to an ad hoc wireless network. Today’s networks based IDSs, which rely on real-time traffic analysis, can no longer function well in the new environment. Compared with wired networks where traffic monitoring is usually done at switches, routers, and gateways, an ad hoc network does not have such traffic concentration points where the IDS can collect audit data for the entire network. Secondly, the traffic patterns in an ad hoc network are very different than a wired network, and it is hard to detect anomalies.

Intrusion detection and response systems should be both distributed and cooperative to suit the needs of wireless ad hoc networks. In [1], Zhang and Lee propose a new architecture, where every node in the wireless ad hoc network participates in intrusion detection and response. Each node is responsible for detecting sign of intrusion locally and independently, but neighboring nodes can collaboratively investigate in a broad range. In the systems aspect, individual IDS agents are placed on each and every node. Each IDS agent run independently and monitors local activities (including user an
systems activities, and communication activities within the radio range). It detects intrusion from local traces and initiates response. If anomaly is detected in the local data, or if the evidence is inconclusive and a broader search is warranted, neighboring IDS agents will cooperatively participate in global intrusion detection actions. These individual IDS agents collectively form the IDS system to defend the wireless ad hoc network.

3.4 Protecting Ad Hoc Routing Mechanism:
Routing mechanisms are more vulnerable in ad hoc networks than in conventional networks because in ad hoc networks each device acts as a relay. This means, for example, that an adversary who hijacks an ad hoc node could paralyze the entire network by disseminating false routing information. A less dramatic but subtle malicious behavior is node selfishness: some nodes may be tempted to not relay packets (e.g., in order to save their own battery).

There are two sources of threats to routing protocols. The first comes from external attackers. By injecting erroneous routing information, replaying old routing information, or distorting routing information, an attacker could successfully partition a network or introduce excessive traffic load into the network by causing retransmission and inefficient routing. The second and more severe kind of threat comes from compromised nodes, which might advertise incorrect routing information to other nodes. Detection of such incorrect information is difficult: merely requiring routing information to be signed by each node would not work, because compromised nodes are able to generate valid signatures using their private keys.

To defend against the first kind of threat, nodes can protect routing information in the same way as they protect data traffic, that is, through the use of cryptographic schemes such as digital signature. However, this defense is ineffective against attacks from compromised servers. Detection of compromised nodes through routing information is also difficult in an ad hoc network because of its dynamically changing topology: when a piece of routing information is found invalid, the information could be generated by a compromised node, or it could have become invalid as a result of topology changes. It is difficult to distinguish between the two cases.

On the other hand, certain properties of ad hoc networks can be exploited to achieve secure routing. Routing protocols of ad hoc networks have to cope with outdated routing information to accommodate the dynamically changing topology. False routing information generated by compromised nodes could, to some extent, be considered outdated information. As long as the number of correct nodes remains high enough, the routing protocol should be able to find routes that circumvent the compromised nodes. As routing protocols can discover multiple routes, nodes can switch to an alternative route when the primary route appears to have failed.

3.5 Protecting the Key Management System:
Protecting the security mechanisms in an ad hoc network is a major challenge. One of the most complex issues is key establishment. Key establishment can be achieved by key transport or key agreement. In key transport, one party creates or otherwise obtains a secret value, and securely transfers it to the others. In key agreement, a shared key is derived by two (or more) parties as a function of information contributed by, or
associated with, each of these, in such a way that no party can predetermine the resulting value. Both approaches can be based on symmetric or asymmetric techniques.

Asymmetric key is an appropriate concept for the case of ad hoc networks. However, there is a drawback to this approach. The usual scalable mechanism to achieve this is that an authority maintains a list of revoked keys on a server, a solution clearly not adapted to ad hoc networks. An alternative would be to request the public key directly from its owner. But this would have to be realized for each new interaction in a secure way, and therefore the expected benefits of the asymmetric technique would be lost.

In [2], the authors propose a system, which focuses on key establishment using asymmetric key cryptography. In their system, each node has a public/private key pair. Public keys can be distributed to other nodes, while private keys should be kept confidential to individual nodes. A crucial problem for a given node A is how to obtain the authentic public key of a node B. The most important threat is an intruder-in-the-middle attack. A way to use asymmetric key cryptography for the transport of the symmetric keys is to encrypt a symmetric key generated by one party with the public key of the other party.

4. Quality of Service for Ad Hoc Networks:

Quality of service refers to the ability of a network to provide a more reliable (or guaranteed) service to selected network traffic. We speak of quality of service in terms of many network parameters such as dedicated bandwidth, controlled jitter, end-to-end network delay, probability of packet loss, and so on. Quality of service also requires traffic differentiation and making sure that giving priority to one class would not make others fail.

By the intrinsic make-up of ad hoc networks, offering a quality of service (or QoS) is a challenge. The absence of an infrastructure, limited bandwidth of wireless channels and their susceptibility to errors, and the mobility of nodes impose many difficulties in achieving QoS in ad hoc networks.

4.1 QoS Routing for Ad Hoc Networks:

Many researches have focused on the topic of QoS routing for ad hoc networks. Routing has a great affect on QoS due to the dynamic nature of network topology and very volatile state information. In [4], the authors have given a survey of issues in ad hoc routing supporting QoS. The authors have described the scenario where the network topology changes too often. They use the term 'combinatorial stability' for a state of the network where topology of a network changes sufficiently slow to allow successful propagation of all topology updates as necessary.

Combinatorial stability is a very important factor in QoS guarantees in an ad hoc network. Combinatorial stability is seen in scenarios where nodes are geographically moving slowly relative to each other. One instance of such a case would be a classroom setting. QoS guarantees cannot be met in networks that are changing too quickly. For many ad hoc networks, combinatorial stability can be achieved. At the same time, its just not the updates in routing tables that affect QoS, but equally important are the availability of enough resources along the old and new routes, during and after a transition.
In setting up a QoS-bounded flow, the first task is to calculate a suitable route in the network that will have the required resources available to meet the QoS guarantees. Then, QoS routing has to reserve resources. In this sense, QoS routing is more than merely calculating the path to a destination. We also have to take into account the computational complexity of a route calculation. One choice is to use sub-optimal algorithms, where paths are chosen based on one metric, and then a subset of them are optimized based on another metric.

Once a path has been reserved for a flow along with the resources (e.g., buffers, bandwidth etc.), these resources will then be reserved for the whole duration of the flow. They would not be available to other flows in the network. Other flows in the network would then have to recalculate their resources.

QoS is also dependent on the precise knowledge of the state of the network. We can divide the information into local and global. Local state information is kept at each node, which includes queuing delay, CPU capacity, some metric for outbound links etc. The sum of all local states forms the global state. Each node also has a picture of the global state. The global state is maintained by exchanging messages between nodes. Since topology updates cannot be propagated throughout the network instantaneously, each node would only have an approximate knowledge of the global state. In ad hoc networks, the global state can never be accurate across all the nodes.

We can use source routing, destination routing, or the hierarchical routing to satisfy the QoS needs. The prominent difference between these approaches is that in source routing, computations are done locally; while in destination and hierarchical routing, the source as well as other nodes are involved in route computing.

In [4], the authors have put forward a routing mechanism for ad hoc networks. Each node sends a periodic beacon to its neighbors informing them of its existence and its QoS attributes. The beaconing system is crucial for QoS support, as the nodes would not have any knowledge of their neighbors without it. The authors mention two main types of routing algorithms: one uses only the local state information, while the other algorithm assumes inaccuracies in the global state information. During a flow, if an existing link becomes unusable, a new route is constructed and the traffic is routed through that link. In the middle period between switching, the packets are routed on a best-effort basis.

The route failure between two nodes is discovered by absence of the beacon message. After a QoS route has been formed, each link uses refresher messages to affirm its continued existence. If a refresher message is not received within a timeout period, the route is considered unusable. Redundant routes are also used to support QoS. There are many ways of using redundant routes. After the first establishment of redundant routes, they can be used simultaneously, or they can be put on hold and used in case of high priority link failures.

5. Power Efficiency in Ad Hoc Networks:

Saving battery power is a major design criteria for wireless systems. Both cellular and paging networks are built with power-efficient protocols. For any wireless handheld device, battery power is consumed mainly by the core processor, the receiver, and the transmitter (along with the synthesizers that generate modulating carriers). Transmission
usually takes the most amount of battery power, followed by reception and the core. At the same time, transmissions occur a lot less than receptions. Most cell phones and paging devices are designed around this assumption that it is not necessary to keep the transmitter and the receiver on all the time. Transmitters only need to be turned on at the time of sending messages, while the receiver needs to be on at periodic intervals. On all other times, the device can be in a batter-save or sleep mode.

Wireless ad hoc networks pose a different challenge for designing power efficient systems. Due to the absence of an infrastructure, each node in an ad hoc network also acts as a router. For an ad hoc network to exist, nodes have to be at least in the reception mode most of the time. The protocol designers have to choose of either operating the network at peak bandwidth performance at the expense of short-lived network, or below average performance for a longer living network. But even in an ad hoc network, we can see that all nodes need not be awake at all times. If a node has the global information that it would not be used as a router, or it won’t be transmitting itself, it can go into a sleep mode. Ad hoc networks should be able to balance traffic load among nodes such that power constrained nodes can be put to sleep mode while traffic is routed through other nodes.

Several solutions have been presented to this problem. One approach is to use power-aware routing algorithms, that balance the load of a network onto nodes that are not power-constrained (SCR and NSR). Another approach is to use distributed and randomized algorithms that run on each node, and determine if the node needs to go into sleep mode or not.

5.1 SCR and NSR:

In [6], the authors propose a power conservation scheme for mobile ad hoc networks. Nodes in an ad hoc network can deploy a sleeping or dozing scheme. A node can choose to go into dozing if it knows that it would not be needed during an amount of time. It can listen on a channel at the beginning of a fixed period of time. If there is data for it, it can stay awake to receive it, otherwise it can turn its receiver off, and start dozing for the next time period. It can keep checking for data at periodic intervals. Such a scheme would work if other nodes are also made aware of a node’s dozing period. A node can disseminate its dozing state to other nodes with a routing algorithm. Based on the global state received from other nodes, each node can compute metrics for route calculations.

Synchronous Collision Resolution (or SCR) is a MAC scheme that makes dozing predictable. SCR asks each node to contend for the channel simultaneously and synchronously. The SCR uses a signaling scheme followed by an RTS-CTS handshake. Nodes can wake up prior to the contention signaling and then immediately return to the doze state after the contention if they will not participate in a data exchange. Signaling not only identifies which nodes win a contention but also whether dozing nodes need to remain awake.

The default energy conservation mode of SCR is for nodes to doze on a slot-by-slot basis. Nodes wake prior to each slot, listen to the signaling, listen to RTS-CTS exchanges and can enter a low energy state as soon as they determine they are not participating in the following data exchange.
Node State Routing (or NSR) is a routing algorithm that assigns metrics to links based on their energy constraint. If a node is constrained with respect to energy, it gets a high metric. After assigning the metrics, the next step is to use Dijkstra’s algorithm to compute the routes. NSR defines two processes of disseminating nodal information and calculating routes. NSR uses a diffusion process to disseminate states.

CSR and NSR complement each other. NSR enables the dissemination process of dozing information to other nodes, while CSR makes the sleep times predictable for nodes.

5.2 SPAN—An Energy-Efficient Coordination Algorithm for Ad Hoc Networks:
Consider an ad hoc network with several nodes. It is possible for a network topology to have many nodes on the edges and a few in the center. The center nodes then can act as a backbone to the other nodes. Using this backbone, a node on one edge of the network can communicate with a node on the other edge. For this communication to occur, only the essential nodes in the backbone have to be awake. Some nodes can take advantage of this situation and go to sleep mode as they would not be needed for some amount of time. For this to work, every node needs to make periodic, local decisions on whether to stay awake or sleep as a coordinator and participate in the forwarding backbone topology.

SPAN is a distributed algorithm that runs on every node in an ad hoc network. The objective of the computation is to determine if a node is on a critical path between two nodes. For example, a node decides to volunteer to be a coordinator if it discovers that two of its neighbors cannot communicate with each other directly or through an existing coordinator. To keep the number of redundant coordinators low and rotate this role amongst all nodes, each node delays announcing its willingness with a random delay that takes two factors into account: the amount of remaining battery energy, and the number of pairs of neighbors it can connect together. This combinations ensures, with high probability, a capacity-preserving connected backbone at any point in time, where nodes tend to consume energy at about the same rate. SPAN does all this using only local information, consequently scaling well with the number of nodes.

6. Conclusion:
Mobile ad hoc networks constitute an emerging wireless networking technology for future mobile communications. In ad hoc networks, every mobile host is a router by itself and can communicate with other hosts without an infrastructure. As the network topology changes, the routing tables in each node need to be updated. This requires routing protocols which are sensitive to mobility. Routing tables can be constructed locally, or a group of nodes can be involved in their computation. Like any other network, ad hoc networks need to be secure. The wireless links are more prone to eavesdropping and security attacks. Attacks can be made at the routing level or at the key management levels. Ad hoc networks need protection from both attacks.

Providing QoS in ad hoc networks is an open research area with many challenges. Robust routing algorithms are required to meet QoS needs. There can be times between a QoS flow, where only best-effort can be provided. Power efficiency is a basic design goal for any wireless network. In ad hoc networks, nodes have to go into sleep mode, but this
information needs to be disseminated to the rest of the network. Distributed algorithms can be used to decide locally if a node can go to sleep mode.

7. References: