CHAPTER – V

QOS AWARE ROUTING IN MANETS

5.1 INTRODUCTION

A typical text on Mobile Ad hoc network (MANETS)[48] describes such networks as simply being “a collection of mobile nodes, communicating among themselves over wireless links and thereby forming a dynamic, arbitrary graph” – listing wireless characteristics and graph dynamics as the main challenges for designing protocols and applications for this network.

Fig. 5.1 Mobile ad hoc networks

MANET has some characteristics:

- Rapidly deployable, self configuring.
- No need for existing infrastructure.
- Wireless links.
- Nodes are mobile, topology can be very dynamic.
- Nodes must be able to relay traffic since communicating nodes might be out of range.
- A MANET can be a standalone network or it can be connected to external networks (Internet).
While capturing important characteristics, this description does not make explicit how MANETs map into the Internet architecture – and does therefore not allow evaluation of existing IP protocols and their applicability on MANETs. Similarly, the lack of a clear architectural description within the context of the Internet has impeded the evaluation of the applicability of MANETs within the Internet. This fact became explicit during the chartering of the IETF AUTOCONF working group: in simple terms, the goal of the AUTOCONF working group is to provide automatic address configuration for MANET nodes. Most researchers and engineers familiar with MANETs shared the understanding that existing auto configuration approaches did not apply. The issue arose again within the context of routing and route optimization within nested NEMO networks, where a clear architectural description of MANETs leads to a poor general understanding of how MANETs might be a candidate technology. The purpose of this memorandum is to document the MANET architecture within the general Internet and IP architecture.

**TYPES OF MANET**

- **Vehicular Ad Hoc Networks (VANETs)** are used for communication between vehicles and between vehicles and roadside equipment.

- **Intelligent vehicular ad hoc networks (InVANETs)** are a kind of artificial intelligence that helps vehicles to behave in intelligent manners during vehicle-to-vehicle collisions, accidents, drunken driving, etc.
**Internet Based Mobile Ad hoc Networks (iMANET)** are ad hoc networks that link mobile nodes and fixed Internet-gateway nodes. In such type of networks normal ad hoc routing algorithms don't apply directly.

We can understand VANETs as a subset of MANET and the best example of VANET is the Bus System of any University. Connected together, these buses, moving in different parts of the city to pick or drop students if they are connected together, make an Ad hoc Network.

**APPLICATION OF MANET**

- Military scenarios
- Sensor networks
- Rescue operations
- Students on campus
- Free Internet connection sharing
- Conferences
- The main two characteristics are mobility and multihop.

**MECHANISMS REQUIRED IN A MANET**

- Multihop operation requires a routing mechanism designed for mobile nodes.
- Internet access mechanisms.
- Self configuring networks require an address allocation mechanism.
- Mechanism to detect and act on, merging of existing networks.
- Security mechanisms.
OVERVIEW OF AD HOC NETWORK

Since their emergence in 1970’s, wireless networks have become increasingly popular in the communication industry. These networks provide mobile users with ubiquitous computing capability and information access regardless of the users’ location. There are currently two variations of mobile wireless networks: infrastructured and infrastructureless networks. The infrastructure networks have fixed and wired gateways or the fixed Base-Stations which are connected to other Base-Stations through wires. Each node is within the range of a Base-Station. A “Hand-off” occurs as a mobile host travels out of range of one Base-Station and into the range of another and thus, mobile host is able to continue communication seamlessly throughout the network. Example applications of this type include wireless local area networks and Mobile Phone.

An ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of existing network infrastructure or centralized administration. Due to the limited transmission range of wireless network interfaces, multiple network hops may be needed for one node to exchange data with another across the network. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network, that may not be within the direct reach wireless transmission range of each other. Each node participates in an ad hoc routing protocol that allows it to discover
multi hop paths through the network to any other node. The idea of an ad hoc network is sometimes also called an infrastructure-less networking, since the mobile nodes in the network dynamically establish routing among themselves to form their own network on the fly.

The history of wireless ad hoc networks can be traced back to the Defense Advanced Research Project Agency (DAPRPA) packet radio networks (PRNet)[49], which evolved into the survival adaptive radio networks (SURAD) program. Ad hoc networks have played an important role in military applications and related research efforts, for example, the global mobile information systems (GloMo) program [12] and the near-term digital radio (NTDR) program. Recent years have seen a new spate of industrial and commercial applications for wireless ad hoc networks, as viable communication equipment and portable computers become more compact and available.

Applications of ad hoc network:

- Military communications.
- Law enforcement.
- Disaster situations e.g. earthquake.

Examples of ad hoc networks:

- MANET (Mobile Ad Hoc Networks) dealing with the routing aspects in Internet Network.
- RWN (Reconfigurable Wireless Network).
5.2 LITERATURE REVIEW

5.2.1 Quality of Service (QoS)[76]

Quality of Service (QoS) is the collective term used when talking about quality control of any system. In computer networks QoS involves adding mechanisms to control the network activity and a prediction of how routes will perform based on previously gathered statistics. Important principles, specifications and mechanisms need to be addressed during system design for it to have QoS that works satisfactory. The QoS needs to be implemented carefully. Some key QoS principles and concepts are listed below:

**Transparency principle**: Applications should be shielded from the complexity of underlying QoS management.

**Flow performance Specification**: Categorizes the flow performance requirements of the user i.e. throughput rate, latency, jitter and loss rate.

**Level of service**: Specifies the degree of end-to-end resource commitment required i.e. deterministic, predictive and best effort.

**Cost of service**: Addresses what the user is willing to pay to get the QoS it demands. If the service comes with no cost, the worst-case scenario is that everybody will ask for the best possible service.
**QoS Mapping**: This mechanism performs the translation of the QoS between the system layers, i.e. Application Layer, Transport Layer and Network Layer.

**Flow Construction**: The flow discovery is based on the flow performance metrics. In addition to concepts and principles, there have to be mechanisms that enforce and maintain the initial frame that were sketched. The mechanisms have to be chosen depending on the QoS system that is currently being developed. Important QoS mechanisms are listed here:

**QoS monitoring**: The levels of a system may track the QoS achieved by lower levels.

**QoS maintenance**: Compares the monitored QoS against the expected performance and then exerts tuning operations on resource modules to sustain the delivered QoS.

**QoS degradation**: Indicates to the upper layer that the lower layers have failed to acquire or maintain the demanded QoS. The application can try to adapt to the QoS or reduce the QoS i.e. renegotiate the QoS with the destination.

**QoS availability**: Allows an application to specify how often and which QoS parameters it wants feedback about from the QoS monitoring mechanism.
5.2.2 The Effects of Shadow-Fading on QoS-Aware Routing and Admission Control Protocols Designed for Multi-Hop MANETs [55],[64]

For all but the least-demanding applications, which have no critical time, reliability, or throughput-related constraints, a system for providing Quality of Service (QoS)[21] assurances is required. Two of the most crucial components in such a system are a QoS- aware routing (QAR) and an admission control (AC)[50] protocol. The QAR protocol is required to find nodes with adequate resources for supporting the QoS[51] requested by applications. It is the task of the AC protocol to estimate the residual resources of the network and to make decisions about whether new application data sessions can or cannot be admitted, given their own QoS constraints, as well as those of previously admitted sessions. At the time of writing, the most critical network resource is usually considered to be the bandwidth, although power efficiency is also of prime concern. A specific minimum throughput has to be maintained in most practical applications. Hence, in this work, the author also focuses on throughput- constrained data sessions. An AC protocol often has to perform a balancing act between admitting too much traffic, promising more resources, such as network capacity, than are available, and thereby causing congestion, and blocking too many admission requests, thereby wasting resources that could be allocated to more users. This paper presents a comparative study, utilizing a realistic shadow fading
channel, of the performance of several state-of-the-art QAR and AC protocols, which are designed for providing throughput guarantees to applications. This is done because providing quality-of-service (QOS) assurances in a mobile ad hoc network (MANET) is difficult due to node mobility, contention for channel access, a lack of centralized coordination, and the unreliable nature of the wireless channel. QOS-aware routing (QAR) and admission control (AC) protocols comprise two of the most important components of a system attempting to provide QOS guarantees in the face of the above-mentioned difficulties. Many QAR and Ac based solutions have been proposed, but such network layer schemes are often designed and studied with idealized lower layer models in mind.

This means that the existing solutions are not designed for dealing with practical phenomena such as shadow fading and the link quality-dependent fluctuation of link transmission rates. The advantages and drawbacks of their particular features are highlighted, and based on the lessons learnt, design guidelines for protocols operating in such an environment are presented.

5.2.3. Multipath Admission Control for Mobile Ad hoc Networks [59]

As wireless networks become more prevalent, users will demand the same applications that are currently available in wired networks. Further, they will expect to receive a quality of service similar to that
obtained in a wired network. Included in these applications are real-time applications such as voice over IP and multimedia streams. To enable the support of applications that require real-time communication in ad hoc networks, congestion must be prevented so that the needed quality of service can be provided.

An admission control mechanism is an essential component of the quality of service solution. Unfortunately, current admission control solutions encounter problems during mobility, often resulting in unacceptable disruptions in communication. To solve this problem, apply multi-path routing mechanisms that maintain alternate paths to the destination and propose a new admission control protocol. This work shows that the solution is able to prevent communication disruptions and meet the QoS needs of applications better than previous solutions.

5.2.4. PAC: Perceptive Admission Control for Mobile ad hoc Networks [58]

Traditional approaches guarantee quality of service (QOS) work well only with predictable channel and network access. In wireless mobile networks, where conditions dynamically change as nodes move about the network, a stateless approach is required. As wireless networks become more widely used, there is a growing need to support advanced services, such as multimedia streaming and voice over IP. Since shared wireless resources are easily over-utilized, the load in the
network must be controlled so that an acceptable QoS for real-time applications can be maintained.

If minimum real-time requirements are not met, these data packets waste bandwidth and hinder other traffic, compounding the problem. To address this issue, they proposed the Perceptive Admission Control (PAC) protocol. PAC[58] monitors the wireless channel and dynamically adapts admission control decisions to enable high network utilization while preventing congestion. Through discussion, simulations show that PAC achieves this goal and ensures low loss and delay for all admitted flows.

5.2.5. Ad-hoc On-Demand Distance Vector Routing [54]

Ad hoc On-Demand Distance Vector Routing is also an on demand routing algorithm, but in contrast to DSR it is a not source based routing scheme. Rather every hop of a route maintains the next hop information by its own. Operation of the protocol here is also divided into two functions – route discovery and route maintenance. At first all the nodes send Hello messages on its interface and receive Hello messages from its neighbors. This process repeats periodically to determine neighbor connectivity. When a route is needed to some destination, the protocol starts route discovery. The source sends Route Request Message to its neighbors. If a neighbor has no information on the destination, it will send messages to all of its neighbors and so on. Once the request reaches a node that has
information about the destination (either the destination itself or some node that has a valid route to the destination), that node sends Route Reply Message to the Route Request Message initiator. In the intermediate nodes (the nodes that forward Route Request Message), information about the source and destination from the Route Request Message is saved. The address of the neighbor that the Route Request Message came from is also saved. In this way, by the time the Route Request Message reaches a node that has the information to answer Route Request Message, a path has been recorded in the intermediate nodes. This path identifies the route that the Route Request Message took and is called reverse path. Since each node forwards Route Request Message to all of its neighbors, more than one copy of the original Route Request Message can arrive at a node. When a Route Request Message is created at the initiator, it is assigned a unique id. When a node receives the Route Request Message, it will check this id and the address of the initiator and discard the message if it had already processed that request.

A node that has information about the route to the destination sends Route Reply Message to the neighbor from which it received Route Request Message. This neighbor then does the same. This is possible because of the reverse path created by the Route Request Message. While the Route Reply Message travels back using reverse path, that path is being transformed into forward path, by recording the node that the Route Reply Message came from (i.e. Same procedure
as mentioned above just in the opposite direction). WhenRoute Reply Message reaches the initiator, the route is ready, and the initiator can start sending data packets. If one of the links on the forward path breaks, the intermediate node just above the link that failed sends new Route Reply Message to all the sources that are using the forward path to inform them of the link fails. It does this by sending the message to all neighbors using the forward path. In turn, they will send to their neighbors until all upstream nodes that use forward path are informed. The source nodes can then initiate new route request procedures if they still need to route packets to the destination.

5.2.6. Physical Carrier Sensing and Spatial Reuse in Multirate and Multihop Wireless Ad Hoc Networks [69]

Physical carrier sensing is an effective mechanism of medium access control (MAC) protocols to reduce collisions in wireless networks, and the size of the carrier sensing range has a great impact on the system performance. Previous studies have shown that the MAC layer overhead plays an important role in determining the optimal carrier sensing range. However, variable transmission ranges and receiver sensitivities for different channel rates and the impact of multihop forwarding have been ignored.

This work investigates the impact of these factors as well as several other important factors, such as SINR (signal to interference plus noise ratio), node topology, hidden/exposed terminal problems
and bidirectional handshakes, on determining the optimum carrier sensing range to maximize the throughput through both analysis and simulations. The results show that if any one of these factors is not addressed properly, the system performance may suffer a significant degradation. Furthermore, considering both multirate capability and carrier sensing ranges, this paper proposes to use the bandwidth distance product as a routing metric, which improves end-to-end throughput by up to 27% in the simulated scenario.

5.2.7. QOS aware routing Protocols[77]

The primary goal of the QoS-aware routing protocols is to determine a path from a source to the destination that satisfies the needs of the desired QoS. The QoS-aware path is determined within the constraints of bandwidth, minimal search, distance, and traffic conditions. Since path selection is based on the desired QoS, the routing protocol can be termed QoS-aware. In the literature, numerous routing protocols have been proposed for finding QoS paths. The QoS routing protocols are described below.

**CEDAR** — The Core Extraction Distributed Ad Hoc Routing (CEDAR) algorithm is proposed as a QoS routing scheme for small to medium-sized ad hoc networks consisting of tens to hundreds of nodes. It dynamically establishes the core of the network, and then incrementally propagates the link states of stable high-bandwidth links to the core nodes. Route computation is on demand, and is
performed by the core nodes using only local state. CEDAR has three key components:

**Core extraction:** A set of nodes is elected to form the core that maintains the local topology of the nodes in its domain, and also to perform route computations. The core nodes are elected by approximating a minimum dominating set1 of the ad hoc network.

**Link state propagation:** QoS routing in CEDAR is achieved by propagating the bandwidth availability information of stable links to all core nodes. The basic idea is that the information about stable high bandwidth links can be made known to nodes far away in the network, while information about the dynamic or low bandwidth links remains within the local area.

**Route computation:** Route computation first establishes a core path from the domain of the source to the domain of the destination. Using the directional information provided by the core path, CEDAR iteratively tries to find a partial route from the source to the domain of the furthest possible node in the core path satisfying the requested bandwidth. This node then becomes the source of the next iteration. In the CEDAR approach, the core provides an efficient low-overhead infrastructure to perform routing, while the state propagation mechanism ensures availability of link state information at the core nodes without incurring high overheads.
Multipath Routing Protocol (MRP)

MRP is a reactive on-demand routing Protocol which extends DSR protocol to find multipath routing coupled with bandwidth and reliability constraint. It consists of three phases: routing discovery, routing maintenance and traffic allocation. In routing discovery phase, the protocol selects several multiple alternate paths which meet the QoS requirements and the ideal number of multipath routing is achieved to compromise between load balancing and network overhead. In routing maintenance phase, it can effectively deal with route failures similar to DSR. Furthermore, the per-packet granularity is adopted in traffic allocation phase.

Predictive Location-Based QoS Routing in Mobile Ad Hoc Networks (PLBQR)

It is a location aware QoS routing protocol in which a location-delay prediction scheme, based on a location-resource update protocol has been performed. The location updates contain resource information pertaining to the node sending the update. This resource information for all nodes in the network and the location prediction mechanism are together used in the QoS routing decisions. There are dynamic changes in topology and resource availability due to the high degree of mobility of nodes in the ad hoc network. Due to these changes, the topological and routing information used by current network protocols is rendered obsolete very quickly. The advantage of this systems is the prediction of new location based on previous
location when there is variation in the geographical location. QoS routing based on the resource availability at the intermediate nodes in the source to destination route is performed which is rare in other location based routing scheme. But accurate prediction on velocity and direction is not made when there are dynamic changes in the direction. The transmission is made only in linear pattern (i.e., angular velocity is kept as zero).

**QoS Multicast Routing Protocol with Dynamic group topology (QMRPD)**

The QMRPD is a hybrid protocol which attempts to significantly reduce the overhead of constructing a multicast tree with multiple QoS constraints. In QMRPD, a multicast group member can join or leave a multicast session dynamically, which should not disrupt the multicast tree. It satisfies the multiple QoS constraints and least cost’s (or lower cost) requirements. Its main objective is to construct a multicast tree that optimizes a certain objective function (e.g., making effective use of network resources) with respect to performance-related constraints (e.g., end-to-end delay bound, inter-receiver delay-jitter bound, minimum bandwidth available, and maximum packet-loss probability) and design a multicast routing protocol with dynamic group topology. It attempts to minimize the overall cost of the tree. The dynamic group membership has been handled by this protocol with less message processing overhead.
**QoS Optimized Link State Routing (QOLSR)**

QOLSR protocol is an enhancement of the OLSR routing protocol to support multiple-metric routing criteria. OLSR is a proactive routing protocol, which inherits the stability of a link state algorithm. The basic QoS metrics considered here are throughput and delay. The routes are immediately available when needed. The OLSR protocol uses a kind of Dijkstra’s shortest path algorithm to provide optimal routes in terms of number of hops. It minimizes the control overhead involved in flooding routing information in which MAC protocol is required to notify the routing protocol when it transmits a packet. QOLSR does not rely on the MAC protocol to provide residual channel capacity or delay information. These values are estimated statistically, using the periodic HELLO messages.

**Ad hoc QoS on-demand routing (AQOR)**

This protocol uses limited flooding to discover the best route available in terms of smallest end-to-end delay with bandwidth guarantee. A route request packet includes both bandwidth and end-to-end delay constraints. Let $T_{max}$ denotes the delay constraint. If a node can satisfy both constraints, it will rebroadcast the request to the next hop and switch to explore status for a short period of $2T_{max}$. If multiple request packets arrive at the destination, it will send back a reply packet along each of these routes. Intermediate nodes will only forward the reply, if they are still in explored state. However, the bandwidth reservation for each flow is only activated by the arrival of
the first data packet from the source node. Delay is measured during route discovery. The route with the least delay is chosen by the source. No mechanism for connection tear-down is needed or integrated, since all reservations are only temporary. Timers are reset every time a route is used. So there is an upper time bound after which broken routes are detected. To further reduce communication overhead during route discovery, AQOR can work with some location aided routing protocols. For delay violation detection, the estimated time offset between the systems clocks of source and destination node has to be known.

5.3 EXISTING SYSTEM

StAC utilizes three stages of AC. The first stage consists of capacity-constrained route discovery, wherein each node forwards the flooded route request (RReq) or the route reply (RRep) if and only if it has sufficient residual capacity to support the session. Residual capacity is estimated using the CITR, a fixed transmission rate, and a “CS-range = two maximum length hops” model, as with most previous works. The session’s capacity requirement at a particular node $B_{req}$ is expressed as its requested throughput $b_{req}$ times the protocol overhead weighting factor $w_{req}$ times the contention count $c_{cont}$. This is expressed as $B_{req} = b_{req}w_{req}c_{cont}$, [19] where we have

$$W_{req} = \left( T_{DIFS} + T_{RTS} + T_{CTS} + T_{Ack} + 3T_{SIFS} \right) / T_{Data} + \left( T_{bloff} + T_{MAChdr} + T_{IPHdr} + T_{SRhdr} + T_{Data} \right) / T_{Data}$$

(15)
and the terms denoted by $T_x$ represent the transmission times of the packet or header (hdr), or the relevant interframe space indicated by the specific subscript $x$ in $T_x$. Furthermore, the subscript SRhdr represents the source route header whose length depends on the route length and $T_{bkoff}$ is the minimum amount of time that is always wasted by the 802.11 back-off algorithm before transmissions. Additionally, $C_{cont} = |N_{cs} \cap R_{prim}|$, where $N_{cs}$ represents the CS neighbour set, and $R_{prim}$ is the set of transmitter/traffic forwarding nodes on the (potential) primary/current route of the session.

StAC [20] allows all routing information that was opportunistically discovered and cached by DSR to be utilized. To test such routes, a second stage of AC also performs the above test at each node by exchanging session request (SREQ) and session reply (SREP) packets between source and destination nodes along a previously discovered route. At this stage, the SREQ is also cached at each node, while its CS neighbours are tested for adequate capacity using a method similar to CACP multihop. If the SREP [21],[22] is received at the source node (the route’s CS neighbourhood has sufficient residual capacity), the reliability of the route is also tested in the third stage of AC. During this stage, which lasts a few seconds, the session is partially admitted, its packet generation and transmission rate is gradually ramped up and the achievable throughput is tested along the route. Any node detecting a lower than expected throughput at any of the staggered rate stages rejects the session, informing the source
node. If the session is not rejected immediately after reaching its desired packet sending rate, it is fully admitted.

5.4 PROPOSED SYSTEM

The objective of this work is to propose and evaluates new solutions for improving the performance of QAR and AC protocols in the face of mobility, shadowing, and varying link SINR[64]. This is done by proactively maintaining backup routes for active sessions, adapting transmission rates, and routing around temporarily low-SINR links can noticeably improve the reliability of assuring throughput services. The existing solutions are not designed for dealing with practical phenomena such as shadow fading and the link quality-dependent fluctuation of link transmission rates. So in this a StAc-backup protocol is used. The protocol was evaluated via NS-2 simulations.

The studies in that work showed that maintaining backup routes and testing their CS neighbors for the availability of adequate capacity to support the needs of requesting sessions may require an excessive overhead that is counterproductive to the maintenance of throughput QoS guarantees. This was shown in the case, where the fixed node transmission rate was relatively low at 2 Mbps. However, the overhead that is introduced for capacity testing per data session is fixed for a given network size. Thus, it was also shown that, when the node transmission rate is increased, the effect of the overhead
diminishes proportionately, and the maintenance of backup routes may greatly increase the throughput QOS requirement satisfaction ratio in the face of high levels of node mobility.

Also it describes the combination of a rate-switching mechanism with a Multirate 802.11[62] model and proposes a new Multirate-aware version of stars called StAC-Multirate. The StAC-backup protocol, proposed and the StAC-Multirate protocol, described above, can have their features combined into a protocol, we call the stars-Multirate-backup protocol. A further optional extension of this protocol is the functionality to pre-test multiple backup routes per session, as opposed to the single pretested backup route employed by StAC-backup. However, even with the “multiple backup routes” an extension having more than one backup route is not mandatory.

5.5 STAGGERED ADMISSION CONTROL BACKUP PROTOCOL

5.5.1 Introduction

QAR and AC protocols are generally unsuited for use in environments, where the link quality may fluctuate quickly due to, for example, shadow fading. Previously proposed scheme, the staggered admission control (StAC) protocol is able to uphold admitted data sessions’ throughput guarantees more reliably than other advanced related protocols mostly due to its fast rerouting of data sessions after route failures, and its gradual admission of traffic while directly testing the achievable QOS prior to session admission. The fact that the QOS
is tested directly as part of the AC process allows not only the capacity of routes to be tested, but also the reliability of their composing links, which may fluctuate due to phenomena such as shadow fading. However, further measures were deemed to be required to improve the achievable throughput-QOS in the face of link quality fluctuations. With this aim in mind, the StAC backup protocol is introduced which evaluated their performance in the face of increasingly severe shadowing attenuation fluctuations.

5.5.2 StAC – Backup Protocol

In this section, we describe an extended version of StAC termed as StAC-backup, which exploits the knowledge of alternative or backup routes to a source’s destination in order to improve the robustness of throughput-QOS assurances in the face of route failures. The version of StAC-Backup described here is an improved relative of the protocol described briefly in the previous work. The studies in that work showed that maintaining backup routes and testing their CS neighbors for the availability of adequate capacity to support the needs of requesting sessions may require an excessive overhead that is counterproductive to the maintenance of throughput QoS guarantees.

This was shown in the case, where the fixed node transmission rate was relatively low at 2 Mbps. However, the overhead that is introduced for capacity testing per data session is fixed for a given
network size. Thus, it was also shown that, when the node transmission rate is increased, the effect of the overhead diminishes proportionately, and the maintenance of backup routes may greatly increase the throughput QOS requirement satisfaction ratio in the face of high levels of node mobility.

5.6 PROPOSED PROTOCOL
5.6.1 Backup Route Discovery

In the newly proposed protocol, once a session being admitted by StAC has found a suitable route and its CS neighbors have been tested during the SREQ/SREP exchange a backup route for the session must be found. There are two possible cases.

- Either more than one route to the destination of the session is already known.
- The backup route must be discovered.

If a new backup route must be discovered, the search packet, referred to as RReq backup carries a copy of the session’s primary route. To avoid fully flooding the network with the RReq-backup, once the RReq backup has travelled at least half of the length of the primary route, the disjointness condition is enforced and the packet is dropped if the partially discovered route does not comply. Also, the RReq backup time to live (TTL) is only one greater than the primary route length, again to reduce the extent to which the network is flooded.
5.6.2 Backup Route Maintenance

No periodic residual capacity query packets are needed, since the status of backup routes can be kept up-to-date as follows: If a node receives a rejection packet, while promiscuously processing all received packets, and the node is part of the rejected backup route, it erases the backup route record from its session’s table. If the session’s primary route fails, the backup route will come into use anyway and its constituent nodes will learn of this through the first data packet’s header, and then, again, the backup route testing will stop. By contrast, if the backup route is rejected and a node does not receive the rejection message, it will continue passively monitoring the CS-neighborhood residual capacity, but this monitoring is overhead free. In this case, it might still reject the route, but the packet delivering the “reject” command will have no effect, as it will be dropped at the first node that did learn of the backup route’s prior rejection.

5.6.3 Comparison of Protocols

Compared to StAC the protocol extension described above does its best to ensure that a capacity-tested backup route is available for each session at all times. This potentially ensures that no drop in throughput occurs after a primary route failure, as opposed to the case when no backup route is established. Note also that StAC-backup eliminates the on-demand update packets employed by StAC to maintain source nodes’ information about residual node capacity.
values, owing to the backup routes being maintained by the mechanism described above.

This reduces the overall protocol overhead. Since StACs mechanism could not maintain information about all of the nodes, such as a route’s CS neighbor, that would be affected by a cache-based reroute, StAC-backup additionally reduces the risk of excessive interference being imposed on the routes of other sessions. The careful, on demand CS neighbor capacity querying method of starch is invoked for initial session admission. However, for backup route testing, a passive, overhead free, low-latency capacity testing method is employed. Our design goal is to have a higher chance of finding backup routes than in MACMAN due to a lower route-disjointness requirement.

Furthermore, the backup route maintenance overhead is theoretically much lower, since only one backup route is tested, and this only introduces overhead once, when it is first established, in contrast to the periodic testing of all backup routes by MACMAN. StAC-backup also maintains all of StAC’s other advantages compared to other state-of-the-art protocols.

**5.7 STAGGERED ADMISSION CONTROL MULTIRATE PROTOCOL**

A further enhancement of both the achievable protocol performance and of the grade of realism may be achieved by employing
adaptive modulation, since it is unrealistic to expect that a fixed-throughput modem maintains a near constant bit error rate (BER), especially in the presence of shadow fading and in the absence of power control. Naturally, adaptive modulation imposes new challenges on the upper layers. This section describes the combination of a rate-switching mechanism with a multirate 802.11 model and proposes a new multirate-aware version of StAC called StAC-multirate.

### 5.7.1 Rate-Switching Mechanism

We implemented a rate-switching mechanism inspired by the hybrid auto-rate fallback (HARF) scheme proposed to adopt the transmission rate based on the channel’s condition. Our scheme inherits HARF’s main mechanisms. These include first that the transmission rate is increased if a given number of ACK frames acknowledging data frames are successfully received in a row. By contrast, the rate is decreased if a given number of ACK timeouts occur in a row (ACK misses). Furthermore, the rate is not only increased or decreased by one level, but the received signal strength indicator (RSSI) corresponding to the last received packet is used to determine whether to keep increasing or decreasing the rate. In our scheme, as opposed to HARF, the last received packet’s power is compared to the receive thresholds stipulated for the various rates. In the case of rate increase, the rate continues to be increased while the received power is higher than the threshold to be exceeded for switching to the next highest rate. A similar scheme is used for
decreasing the rate, until it has been reduced to a value that has a lower receive threshold than the last received packet’s signal strength. This enables fast adaptation to the varying received signal strength resulting from shadowing. As an extension to HARF’s above mentioned behavior, we also use missed RTS frames (lacking a CTS response) to increase the so-called ACKs missed count. This ensures that the rate can still be adapted even if no data frames are transmitted due to a failed RTS-CTS handshake.

In our implementation, each node stores the rate that was last used for transmission to each of its neighbors with which it has communicated, as well as the amount of contiguous missed or received ACKs. Since the transmission rate is likely to change multiple times per second, following the fluctuations due to shadowing, it is impractical to report every change to the network layer protocols. Instead, the rate in use by each packet is recorded, and the average rate is calculated in a 1 s sliding window. This average rate is rounded off to the nearest supported rate, which is reported to the routing protocol when it queries that particular link rate. Note that despite the different transmission ranges achieved by the different modulation schemes, the optimal CS range does not vary. Therefore, a fixed CS range is maintained for simulations in this work.
5.7.2 Calculating a Session’s Capacity Requirement

For each CS neighbor of the current node on the route, the channel time occupied by the MAC layer overhead must first be calculated assuming the basic rate and the standard expression of \( C_{\text{cont}} = |N_{\text{cs}} \cap R_{\text{prim}}| \), for the contention count, where R is the set of transmitters on the route being tested. Then, the channel occupation time of the higher layer data portion of a session’s traffic must be calculated using the current link rate for that hop. In our proposed method, we arranged for this effect to modify the contention count. The contention count thus becomes a non integer multiplier of the session’s capacity requirement at a particular node as follows:

\[
C_{\text{cont}} = \sum_{j=0}^{[N_{\text{cs}} \cap R]} \frac{\beta_0}{\beta_j}, \forall j \in N_{\text{cs}} \cap R
\]  

(16)

Where \( \beta_0 \) represents the basic rate and \( \beta_j \) denotes the rate of link j, which is being considered. Links j includes all links originating at transmitters that are in both NCs and R. This formula reflects the fact that the effective contention count applied to data payloads will be reduced by the ratio of the link rate to the basic rate, since the amount of channel time they will occupy is reduced by that proportion. Therefore, the full expression for the session’s capacity requirement on a link j becomes as shown by the below equation, each T is first calculated using the basic transmission rate. More specifically, the first term in the two part sum of the following equation represents the
fact that the network and higher layer overhead and data may be transmitted with a higher transmission rate than the basic rate, depending on the channel conditions. Therefore, the contention count for those components is calculated according to the above equation. Meanwhile, the MAC [25], [26] layer control frames are transmitted at the basic rate, and therefore, their contention count is expressed using the single rate contention count formula given by

\[ C_{\text{cont}} = |N_{cs} \cap R_{\text{prim}}| \]  \hspace{1cm} (17)

\[ B_{\text{req}} = b_{\text{req}} \frac{T_{\text{IPhdr}} + T_{\text{SRhdr}} + T_{\text{Data}}}{T_{\text{Data}}} \left( \frac{|N_{cs} \cap R| \beta_0}{\sum_{j=0}^{\infty} \beta_j} \right) + b_{\text{req}} \frac{T_{\text{MAChdr}} + T_{\text{DIFS}} + T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{Ack}} + 3T_{\text{SIFS}} + T_{\text{bloff}}}{T_{\text{Data}}} |N_{cs} \cap R| \]  \hspace{1cm} (18)

### 5.7.3 Route Discovery

The route discovery procedure is similar to StACs with a few notable differences: aside from each node’s free capacity, the RReq/RRep packets carry a 3-bit value for each IP address in the source route, representing the index of the link rate currently in use to the next hop. At the time of the route discovery, there may not be any information about the link rate average over the aforementioned 1s interval, and hence, the RReq/R Rep packets’ received signal strengths are used to select the appropriate rate. Symmetric rates in both directions on a link are initially assumed, until it is found otherwise.
Link rate information collected during route discoveries occurring in the course of normal network operation (after a network warms up) will typically be more reliable. If multiple routes are stored in the cache, the “fewest hops” metric utilized by StAC for choosing between them may no longer be optimal. Therefore, StAC-Multirate selects the route that minimizes the channel time utilization of the session in the network, which is proportional to |R|/βR, where βR is the average link rate used by the transmitters on the route R. This metric thus encourages the selection of short routes with high average link rates. A high βR also implies a higher chance of shorter and more reliable hops, compared to a route having the same number of hops with a lower βR. Therefore, the aim is to strike a balance between the number of hops and the internodes distances.

5.7.4 Resource State Maintenance

StAC uses a source route header extension to carry a source node’s view of the available residual capacity at the nodes on a session’s route. If, upon forwarding a data packet, a node detects that the source’s view differs from its own estimate of the residual capacity by a given amount, and no update has been sent to that source node recently, an update packet is sent. All nodes forwarding the update packets add their own updates, if required, in order to avoid having to send separate update packets. This means that a given node always has up-to-date information about the residual capacity of all nodes that lie along any active routes passing through it. This aids StAC in
its rerouting procedure, since the delivery of a single data packet on a route is enough to trigger an update. In StAC multirate, this behaviour is merely extended to enable updates to be triggered by a change in a link rate. The link rates are averaged over 1 s, and hence, averages are unlikely to change due to the 1 Hz (or faster) shadowing fluctuations we model. Instead, they are likely to change only due to mobility, which alters the Internode distance, and hence, the mean path loss. As such, overly frequent updates are avoided. When StAC-Multirate is combined with starch-backup (which does not use the update packets), the backup route maintenance scheme detects any change in link rates that would render a backup route’s capacity inadequate for its session.

5.8 ROUTING AROUND TEMPORARILY LOW-QUALITY LINKS

All related protocols discussed later report a broken link, if the MAC protocol cannot deliver a packet to the next node. Moreover, when experiencing shadowing induced link quality fluctuations, this can result in highly inefficient behaviour due to more route searches being triggered. If operating at a higher rate, switching down can help in this situation, but only if both the received power and the SINR stay above the basic rate’s requirements. If the fraction of time for which the SINR is below a usable level is much less than the fraction of time for which reliable communications are possible, it seems more efficient to deem the link intact but temporarily route packets around it.
There is also more sense in waiting for the link quality to improve than in attempting further retransmissions, which cause additional interference. For this purpose, a 0 bps transmission mode is introduced for short periods of low link quality. If three ACKs (or CTS frames) are missed in a row in BPSK mode, the rate is switched to 0 bps. The rate is only increased again when a timeout period has passed. This timeout period should be based on the shadowing frequency, which can be estimated by a simple level-crossing test, or by monitoring the average time between periods of receiving many contiguous ACKs and periods of missing many ACKs in a row. For this work, we statically set the 0 bps mode timeout period of 0.1 s, noting that a long period causes less needless interference if the channel quality has not yet improved, but wastes transmission opportunities during relatively short fades, and vice versa.

The average amount of time a link spends in the 0 bps transmission mode per second is recorded and reported to the routing protocol. This is used to estimate the “channel’s usable time ratio” (CUTR), i.e., the percentage of time that particular link can be used. A link is reported broken, if the usable time is below 70 percent (a tunable parameter). A node’s CUTR is multiplied by the next hop link’s rate and the CUTR to yield an estimate of the link’s usable free capacity. During StAC’s rate ramp up stage of admission, an insupportable session is more promptly rejected because aside from monitoring the session’s throughput, the amount of capacity required
for the next hop also compares to the link’s usable capacity. This mechanism can additionally be employed during the periodic testing of the backup route nodes’ capacity, when StAC-backup and StAC-multirate are combined. In most cases, the network will have been operating for some time, and thus, both the average link rates and usable time ratios will already be known.

For links at the center of the network, which are the most likely to be used, given a high enough data sending rate to “sample” the channel, these average values will already be meaningful after a few seconds. While the transmission rate is set to 0 bps, we do not wish to report a broken link. However, the buffering of packets would simply impose high latency, and buffers may also become full. Instead, we propose a mechanism, which attempts to find an intermediate relay node to the next hop. For this purpose, the DSR route cache is used, which, as previously stated, stores all overheard routing information.

A brief description of the algorithm follows:

[1] A packet is returned to the routing protocol by the MAC protocol as “undeliverable” due to the retransmission count limit being exceeded or the transmission rate of 0 bps is currently selected.

[2] If this is the first time this particular packet (identified by its packet ID or source time stamp) has been returned to this node by the MAC protocol, a record of its original next hop (the unreachable) node’s address is made.

[3] An alternative route to the unreachable node is sought in the route cache. This will usually be two hops long, adding one extra
relay node. This relay node is appended to a list of alternative next hops tested. The packet’s source route header is modified to include the new relay node and it is sent down to the MAC protocol to be retransmitted. If the same packet is returned once again by the MAC protocol, meaning that the channel between the current node and the alternative next hop is currently also bad, a third alternative is sought, avoiding previously attempted next hops. Before each new attempt, the packet’s original source route is reconstructed, to avoid the route length growing by more than one hop at each forwarding node.

[4] This process continues from step 1, until the packet is either successfully delivered to the new next hop or no further routes to the unreachable node are known. In the latter case, the protocol reverts to the default operation of reporting a broken link.

This scheme utilizes a form of link diversity, following on from the assumption that transmissions to different neighbours will suffer independent shadowing and there is a good chance that the signal paths to two different neighbours are not both unusable at the same time. An alternative scheme for selecting relay nodes during periods of bad link quality was previously suggested. However, that scheme relied on node location information and signal quality measurements to select the relay node. Our scheme does not rely on location information, and is thus, cheaper to implement and simpler, and uses the existing information in the DSR route cache combined with the 0 bps transmission mode.
5.9 STAGGERED ADMISSION CONTROL MULTIRATE BACKUP PROTOCOL

The StAC-backup protocol and the StAC-multirate protocol, described above, can have their features combined into a protocol, we call the StAC multirate- backup protocol. A further optional extension of this protocol is the functionality to pretest multiple backup routes per session, as opposed to the single pretested backup route employed by StAC-backup. However, even with the “multiple backup routes” extension having more than one backup route is not mandatory, as detailed below.

It makes no sense to make multiple backup routes a requirement for the following reasons. In DSR, used as the underlying basic routing protocol of StAC, all detected link failures are reported to affect source nodes by Route Error (RErr) packets. These RErrs are received by all nodes observing the channel, thereby informing them of route failures. By this mechanism, a session causing a link failure detection can also cause the discarding of another session’s backup route. This can happen particularly often in the presence of link quality fluctuations caused by shadowing variations.

If having multiple backup routes was a requirement, their frequent re-discovery would result in many flooding based route searches, and thus, much extra protocol overhead. On the top of this, all existing backup routes are likely to be found by the already existing
mechanism for finding backup routes in StAC-Multirate-backup, especially since multiple routes may be learned as the result of a single route discovery operation, or routes may be “snooped” from “overheard” packet headers. This implies that there is no sense in triggering new route searches when there is already a primary route and a backup route for a session as no further routes are likely to be found.

Design for a tertiary route feature is as follows: All routes discovered by a source node in the above mentioned ways are evaluated for suitability to be used as tertiary routes. Only sessions that already have designated primary and (secondary) backup routes can be assigned tertiary routes. All discovered routes that are sufficiently linked disjoint (according to the “minimum disjointness” parameter) from both the primary and backup routes are stored in a session’s tertiary route list. As soon as they are stored in the list, a capacity test probe packet is created.

The tertiary route and its CS neighbours are then tested for adequate capacity to serve the testing session in the same manner as backup routes are tested, as described above. State information about them is stored, again, as for secondary backup routes, and the available capacity of their constituent nodes’ CS neighbourhoods is continually tested again, as described above. Now, if a session’s primary route fails, it is replaced by its backup route. The backup
route is then replaced by the first protested the tertiary route in the list, which is removed from the list. If a session’s primary route fails, and there is no backup route, a tertiary route is selected to be the new primary route, and so on for all other possible cases. Note, however, that this optional extension is not part of the StAC-Multirate-backup protocol by default, rather the protocol is first evaluated without it, than with this feature separately afterwards.

5.10 SIMULATION MODEL

In this section, we evaluate the performance of throughput by Simulations. We first describe the simulation environments and performance evaluation metrics, then evaluate the performance with given environments and parameters. NS-2 is the network simulator used here. NS-2 stands for Network Simulator version 2. It is a discrete event simulator for networking research. NS-2 provides substantial support to simulate a bunch of protocols.

SIMULATION CONDITIONS

In the simulation, we randomly selected source node and destination node to simulate our scheme on NS-2. Detailed simulation parameters are listed below:

- NS Version : 2.34
- Channel Type : Wireless Channel
- Network Interface Type : Wireless Physical
- Propagation Model : Two Ray Ground, Ricean Shadowed
• MAC :  802.11
• Interface Queue Type : CMUPriQueue
• Antenna : Omni Antenna
• Link Layer Type : LL
• Interface Queue Length : 64
• Number of Nodes : 70
• Terrain Range (m²) : 1660 × 1660
• Routing Algorithm : DSR (Extended)

**EVALUATION METRICS**

The primary concern in this thesis is how well the promised throughput QOS guarantees are upheld.

• **Session Admission Ratio (SAR):** - The total number of admitted sessions divided by number of session admission requests.

• **Session Rejection Ratio (SRR):** - The total number of rejected sessions divided by number of session admission requests.

• **Packet Loss Ratio (PLR):** - The fraction of generating application layer data packets that were not delivered to their destination nodes.
5.11 RESULTS AND OBSERVATION

Call Acceptance Ratio (Mobile)

The call acceptance ratio of mobile nodes is getting lower for Stars Multirate backup protocol when compared with stock, Stac backup and StacMultirate protocol, which is represented by the Fig.5.2.
Call Acceptance Ratio (Static)

Fig.5.3 The session admission ratios in a network of static nodes

The call acceptance ratio of static nodes is better for StAC Multirate backup protocol when compared with StAC, Stac backup and StacMultirate protocol, which is represented by the Fig.5.3.
Call Rejection Ratio (Mobile)

The Call rejection ratio of mobile nodes is lower for StAC Multirate backup protocol when compared with StAC, Stac backup and StacMultirate protocol, which is represented by the Fig.5.4.
Call Rejection Ratio (Static)

Fig. 5.5 The session rejection ratios in a network of static nodes

This fig. 5.5 shows the session rejection ratio of static nodes where the increment in the shadow fading and call acceptance gives more or less the StACmultirate back up protocol getting improvements than the other protocols such as StAC, StAC backup, StAC Multirate protocols.
Packet Drop Ratio (Mobile)

Fig. 5.6 The packet drop ratios in a network of mobile nodes

The packet drop ratio of mobile nodes is lower for StAC Multirate backup protocol when compared with StAC, Stac backup and StacMultirate protocol, which is represented by the Fig.5.6.
Packet Drop Ratio (Static)

Fig. 5.7 shows that Packet drop ratio of static nodes with the increment in shadowing environment is lower for StAC Multirate backup protocol when compared with StAC, Stac backup and StacMultirate protocol.

5.12 CONCLUSIONS AND FUTURE ENHANCEMENTS

The proposed protocols relate to the StAC protocol, and their performances are evaluated in the face of increasingly severe shadowing attenuation fluctuations. The StAC backup protocol added
a feature that attempts to provide a pre-capacity-tested backup route to each active data session. The novelty lay in the method of maintaining the status of backup routes regarding their capacity at data source nodes without incurring any test packet overhead, as well as in the combination with StAC.

Use of such backup routes allowed the elimination of “available capacity” status update packets used by StAC while reducing the risk of rerouting to routes for which there is no knowledge of their free capacity. However, it was found that with severe shadowing induced signal strength fluctuations, the protesting of backup routes was less significant, although merely proactively seeking backup routes still improved the achieved QoS.

We describe the combination of a rate-switching mechanism with a multirate 802.11 model and propose a new multirate-aware version of StAC called StAC multirate. The StAC-backup protocol, proposed and the StAC-multirate protocol, described above, can have their features combined into a protocol, we call the StAC multirate-backup protocol. A further optional extension of this protocol is the functionality to pretest multiple backup routes per session, as opposed to the single pretested backup route employed by StAC-backup. However, even with the “multiple backup routes” an extension having more than one backup route is not mandatory.