

# ASAP - Adaptive QoS Support with Reduced Reservation Overhead in MANETs \*

Patrick Stuedi, Jianbo Xue, Gustavo Alonso

Swiss Federal Institute of Technology (ETHZ)  
Departement of Computer Science  
8092, ETH-Zentrum, Switzerland  
{stuedip, xue, alonso}@inf.ethz.ch

## ABSTRACT

Existing QoS approaches for mobile ad hoc networks focus on optimizing end-to-end bandwidth allocation with respect to, e.g., node mobility, link-failures and packet-loss. Typically this is achieved through over-reservation. This is specially true in loosely connected networks or in high mobility scenarios where reservations take place only on part of the path because no end-to-end connection is available. In this paper, we propose ASAP, an Adaptive ReReservation and Pre-Allocation Protocol for mobile ad hoc networks. ASAP provides high service quality and minimizes wasted reservations. It does so by using a simple soft/hard reservation mechanism and a flexible signaling schema.

## 1. INTRODUCTION

Quality of Service (QoS) support in Mobile Ad Hoc Networks (MANETs) is a challenging task. The dynamic nature of the network topology leads to frequent route changes and forces flows to dynamically compete for their resources. In addition, the wireless link is typically very unstable, as the available bandwidth may change over time due to interference. Even if the wireless link were stable and route changes negligible, it would still be difficult to control the resources on the shared medium. Ad hoc networks are inherently decentralized, so assigning a central controller to maintain connection state and reservations is not appropriate. Despite these difficulties, several solutions have been proposed using greedy reservations. Over-reservation occurs when, e.g., the path between two nodes is temporarily disconnected. Reservation requests move towards the destination node but do not reach it, leaving a partial reservation that is actually not used by any flow (Figure 1a). Another scenario is a path between two nodes where the bandwidth bottleneck is near the destination node (Figure 1b). The

\*The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

difference between the bandwidth that has been reserved on intermediate nodes and the bandwidth available on the bottleneck node is then wasted. Both cases get worse with mobility since over-reservations are propagated all over the network (Figure 1b).

In this paper we propose ASAP, a QoS framework in MANETs that provides high service quality while minimizing wasted reservations. It does so by using a soft/hard reservation concept and a flexible signaling schema. First of all, ASAP adopts a two-level QoS reservation mechanism (soft and hard reservation). In the phase of QoS probing and establishing, soft reservation is used to pre-reserve resources for future usage, but without preventing access to these resources. After this phase, the soft reservation is switched to a hard reservation which can then be used by the application. This soft/hard reservation mechanism reinforces fairness and minimizes over-reservations. Rather than requiring applications to choose between some discrete bandwidth levels, ASAP is able to deal with any QoS level within a given [MIN,MAX] range. This flexibility is crucial in many applications. For example, data streams of stock prices can be processed by the application at any QoS level within a certain range. New emerging multimedia applications can also support fine-grained throughput levels with adaptive codecs that can be dynamically reconfigured. ASAP is also highly adaptive. In case of resource loss due to movement or radio interference ASAP informs and assists both sender and receiver ends to adapt their behavior in a short period. The mechanism is also used in the situation when extra resources become available. Besides these adaptations, ASAP also deals with QoS reestablishment, restoration and bandwidth reclaiming caused by path changes or broken connections.

Another important feature of ASAP is its simple in-band signaling system. ASAP uses a header embedded in the packets. The headers are periodically placed into flow packets by specifying the refreshing interval. Intervals can be configured at run-time depending on the QoS situation. The shorter the interval, the faster the QoS protocol reacts to the time-variant network topology and QoS availability but also propagates more reservation states. The trade-off between service quality and over-reservation is investigated later in the paper.

In the paper, ASAP is described in detail and its performance evaluated by comparing it to the one of INSIGNIA[7] under varying network conditions. The experiments show improved throughput for QoS and best effort (BE) traffic and an efficient utilization of the available bandwidth. The rest of the paper is organized as follows. The next section discusses the related work. Section 3 presents the protocol. Section 4 evaluates the performance of ASAP and Section 5 concludes the paper.

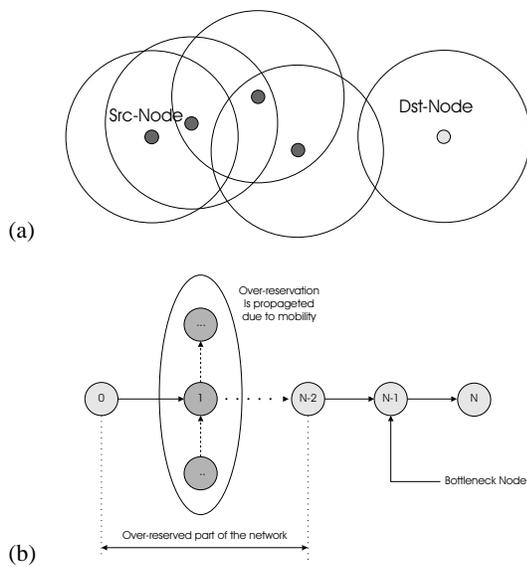


Figure 1: (a) partial reservations on a path, (b) propagation of over reservation

## 2. RELATED WORK

There are two generic approaches to bring QoS into MANETs: DiffServ[2] and IntServ[3]. In the DiffServ model, traffic is divided into one best-effort class and a few QoS classes. QoS treatment is provided depending on the priority of the traffic. The advantage of such an approach is its simplicity. No state information on intermediate nodes has to be maintained and no explicit signaling is needed. But the DiffServ model was originally developed for the Internet backbone where ingress and egress nodes are distinguishable. Ingress nodes typically are responsible for admitting flows and detecting contract violations. In an ad hoc network, every node is a potential sender and therefore an ingress node, so the DiffServ model cannot be applied. Moreover simply dividing the resources in several priority classes can not give any bandwidth guarantees to an individual flow. This is different in an IntServ based approach. Here, exclusive flow-based treatment is provided by reserving and allocating parts of the available bandwidth on each node. One drawback is that IntServ needs to store flow specific information on each node and therefore is not scalable. But we argue that this might not be an issue in MANETs as these networks typically are not bigger than 100 nodes. The challenge in making the IntServ model work for MANETs lies in including mechanisms which are adaptable to topology changes and bandwidth variations.

SWAN[1] is based on the DiffServ model. Although SWAN is able to maintain some sort of QoS for admitted flows, it is not a sufficient solution as it treats all real-time traffic equally. Like other DiffServ architectures, it provides no QoS guarantees, but rough approximations. RSVP[4] is an IntServ flow-based QoS protocol in fixed networks. Currently, work on QoS in wireless networks mostly focuses on improving RSVP to suit last-hop wireless access networks. For instance, MRSVP[15] and HMRSVP[17] use excessive reservations in all neighboring cells so that QoS can be maintained in whichever cell the most hosts might move. An alternative solution is to find the nearest common router (NCR), locally repair the path from the NCR and the new access point, and restore the original QoS on the new path. Such schemes are used in LRSVP[8] and [10]. Although these attempts address some QoS issues in wireless environments, they cannot entirely cope with MANETs.

dRSVP[9] is an RSVP-extended protocol which aims at supporting dynamic QoS in wireless networks (including MANETs). By requesting bandwidth over a QoS range instead of a single value, dRSVP provides flexible QoS provisioning in a QoS-varying environment. The limitations of dRSVP are: excessive signaling, lack of an effective path reparation mechanism and slow QoS state setup. INSIGNIA[7] adopts efficient in-band signaling to piggy-back control information into the IP header of traffic so that resource reservation and QoS treatment can be provided along the flow, without the need of a pre-established path. INSIGNIA supports two QoS levels: base QoS and enhanced QoS. Flow traffic carries MIN/MAX bandwidth range requests in the packet headers. In each hop, the flow reserves bandwidth to meet the MIN/MAX request. The first node who is only able to support the minimum requirement marks the packet, so that all subsequent nodes will not try to allocate more than what is available on the bottleneck. Finally, base QoS or enhanced QoS traffic will be sent by the sender upon receiving a QoS report from the receiver indicating the total bandwidth reserved along the path. One important limitation of INSIGNIA is that it can only support two QoS levels, not enough to match the needs of fine-grained adaptive realtime applications. Another issue is that partial reservations may exist between the source and the bottleneck node. INSIGNIA allows to clear this reservation by sending MIN request, but this prevents the protocol from upgrading when new resources become available. The clearing mechanism is slow because it is triggered by a report message sent by the receiver node. In the meanwhile partial reservations might get propagated over the network due to node mobility.

QoS routing protocols such as CEDAR[13] and others [5][14] interact with resource management to establish paths through the network that meet end-to-end QoS requirements. This might be problematic because the time scales over which session setup and routing operate are distinct and functionally independent tasks. Also MANET routing protocols should not be burdened with the integration of QoS functionality that may be tailored toward specific QoS models. Integrating QoS in routing might even slow down the routing process itself.

## 3. ASAP FRAMEWORK

### 3.1 Soft- and Hard Reservation

In MANETs, the QoS level is determined by the hop with the worst QoS support on the path. Accordingly, a flow-based QoS reservation mechanism contains two parts: the QoS information collection, and the QoS reservation. Following this idea, ASAP uses a two-level reservation mechanism: soft reservation and hard reservation. This mechanism mirrors the passive/active reservation mechanism of MRSVP[15]. The difference is that ASAP integrates this concept into its architecture to minimize the waste of resources and maximize the protocol's agility. In one node, the total bandwidth  $R_{total}$  is divided into two parts: the QoS bandwidth  $R_{QoS}$  and the best-effort bandwidth  $R_{be}$ . As the QoS bandwidth takes a fixed portion of the total bandwidth, the ratio between the QoS bandwidth and the total bandwidth is  $\lambda = \frac{R_{QoS}}{R_{total}}$ .

When a reservation request arrives, the available bandwidth is checked. If enough bandwidth is available, ASAP marks these resources as soft reserved. Once the amount of bandwidth is soft-reserved by a flow, it cannot be booked by other flows but it is still accessible to other traffic (both best-effort and QoS traffic). As the soft reservation is only the booking of the resource, it is quite possible that the flow which requests it will not use the resource. For instance, the route may have changed by the time the reserved

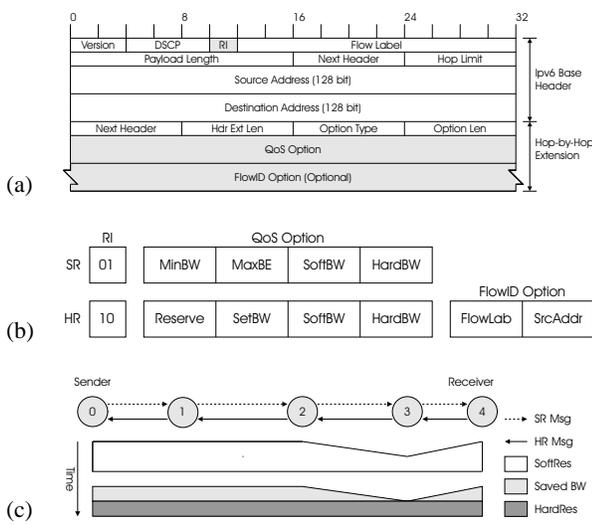


Figure 2: (a) ASAP Extension in IPv6 Header, (b) Structure of SR and HR Messages (c) SR/HR Signaling

bandwidth is to be used.

The use of soft-reservation has two effects: first, a soft reservation will not change the traffic characteristics in the node. The soft reserved resource can still be used by best effort traffic and some other QoS flows which temporarily need extra bandwidth, so this traffic will not be affected by a soft reservation. Second, a soft reservation will increase the efficiency of resource utilization both for QoS traffic and best-effort traffic. For example, if the ratio of QoS bandwidth  $\lambda$  is set to 60%, it implies 40% of total bandwidth is available to best-effort traffic. In case of using soft reservation, part of the best-effort traffic can borrow the soft reserved bandwidth. The amount of the unused soft reservation remains rather high when mobility is above a certain level. Therefore,  $\lambda$  can be adjusted accordingly. This will provide more bandwidth for QoS flows without affecting the best-effort traffic, thereby increasing bandwidth utilization. Once the soft reservation has been done on every hop of the flow path, the flow traffic can in principle get a certain level of QoS support on this path (the level of QoS is given by the node with the least resources). Before the admission of the flows, the soft reservation is switched to hard reservation state. If there is other traffic borrowing the soft-reserved resource, it is removed. After the hard reservation is completed, the flow is free to be admitted.

## 3.2 ASAP Signaling System

ASAP uses an in-band signaling system based on IPv6. It defines an extension which can be placed in IPv6's Hop-by-hop header (Figure 2a). The extension contains two Options: a QoS option and a FlowID option. The QoS option stores the QoS information related to the realtime flow. The FlowID option is the identifier of the flow and optional in the ASAP extension.

ASAP defines two signaling messages: **SR** (Soft Reservation Request) and **HR** (Hard Reservation Request). A two-bit RI field (Reservation Indicator)<sup>1</sup> is used to distinguish SR and HR messages: 01 as SR message and 10 as HR message, shown in Figure 2b. An SR message uses only a QoS option which contains four fields: **MinBW**, **MaxBW**, **SoftBW** and **HardBW**. MinBW and

<sup>1</sup>According to RFC2474[11], DSCP field is defined and the two-bit field between DSCP and Flow Label fields are unused. ASAP propose to define the RI field using these two bits.

Flow Label	Src Address	SoftResvBW	HardResvBW
AAAA	Host1	100	100
1111	Host2	100	50
BFFF	Host1	0	150

Table 1: ASAP Flow Table in an Intermediate Host

MaxBW indicate the minimum and maximum bandwidth to be reserved. These two fields can specify the desired QoS range for adaptive realtime applications to reply on. SoftBW and HardBW show how much bandwidth has been soft-reserved and hard-reserved on the flow path. The HR message has its QoS option with three fields: SetBW, SoftBW and HardBW. SetBW shows the amount of bandwidth the host should hard reserve. SoftBW and HardBW have the same meaning as in an SR message.

Additionally, the HR message optionally includes a FlowID to identify the flow. Normally a flow identifier is composed of several fields in the IPv6 common header: the Source/Destination Address and the Flow Label which is uniquely generated by the source node (the flow identifier can be shortened with only the Source Address and the Flow Label fields). As SR messages are always sent from sender to receiver (from source to destination), the flow which the SR messages belong to can be identified by the fields in its own packet's IPv6 header. In case of HR messages, which are often sent from the receiver to the sender, the IPv6 header in the HR packet contains no valid flow identifier. Therefore, it is necessary for the HR message to include a FlowID option with FlowLabel and SrcAddress fields to identify which flow this HR message belongs to.

SR messages can be used for several purposes: build up the flow path, monitor the QoS state, repair the broken path and so on. The usage of HR messages is relatively simple: to report the QoS state and to upgrade or downgrade the total QoS level on the flow path when the QoS varies.

SR messages are in-band signaling. Senders of real-time flows periodically insert SRs into the IP option header of the traffic packets sent to the receiver. The QoS monitoring and reservation is done while SR travels. The receiver then replies to the sender with a HR message (out-of-band) for QoS reporting and adaptation, as shown in Figure 2c.

## 3.3 QoS Connection Management

The ASAP protocol has three states: CLOSED, ESTABLISHING/UPDATING, and ESTABLISHING (Figure 3). These states include all the phases related to QoS control.

### 3.3.1 Fast Flow Setup

Initially, to setup a new QoS flow, the sender transmits an SR message to the receiver. In this SR message, MinBW and MaxBW are marked with the bandwidth range which the realtime application is willing to accept. SoftBW and HardBW are initially set to zero. Upon receiving this SR, the host considers it as a flow *setup\_request()*, and switches the state to ESTABLISHING, indicating it is ready for flow setup. The host creates a flow entry in its flow table as shown in Table 1, makes a soft reservation within the range MinBW-MaxBW and marks the SoftBW field with the reserved amount. It also updates the flow entry with the SoftResvBW and HardResvBW fields. These two fields in the flow table are used to show how much bandwidth is soft-reserved and hard-reserved in this hop respectively. In this case, the SoftResvBW should equal the SoftBW field. The next host repeats the same procedure, but only updates SoftBW if the reserved amount is less than the original amount in SoftBW.

When the receiver gets the SR message, it knows that the avail-

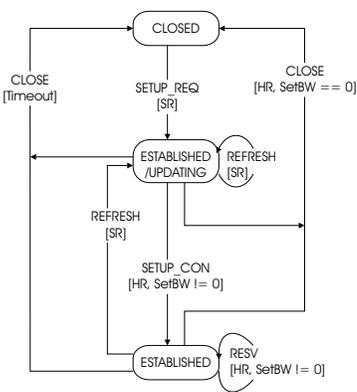


Figure 3: ASAP State Machine

able bandwidth equals the value of SoftBW. It then replies with an HR message with SetBW equal to the SoftBW in the SR message. This HR message is used as a method - flow *setup\_confirm()*, and it travels back to the sender along the incoming path. Upon receiving the HR message, each intermediate host switches its soft reservation to hard reservation. It also marks the SoftBW/HardBW fields in the message, releases the extra reservation if it exists, and updates its flow table. Now the state of the node is switched from ESTABLISHING to ESTABLISHED, which means the node is ready to provide QoS support for that flow. Finally, when the HR message arrives at the sender, the sender can start the flow with a speed matching the reserved bandwidth. A big advantage of this approach is that the combination of soft and hard reservation helps to avoid wasting resources in hosts other than the bottleneck, a problem which other QoS protocols fail to deal with.

### 3.3.2 Dynamic QoS Monitoring and Adaptation

Once established, the flow path can be affected in a multitude of ways: radio interference, movement of nodes, unavailability of routes, etc. At the same time, as flows complete their transmission, more bandwidth might become available and applications may have a chance to upgrade their QoS level. ASAP tries to adapt to these situations through monitoring.

After a flow path is established, SR messages are periodically inserted into the traffic flow. As each SR message collects QoS information along the flow path, the receiver can keep track of the latest QoS situation and give the sender feedback using HR messages, so that real-time applications can adapt their QoS profile accordingly. The algorithm for processing SR messages is shown in Alg. 1. Each host on the flow path is now in the state ESTABLISHED. Upon receiving SR messages related to a specific flow in its flow table, the host considers it as a flow *refresh()* and changes its state from ESTABLISHED to UPDATING (the same state as ESTABLISHING), as shown in Figure 3. It checks whether the existing total reservation for this flow (SoftResvBW and HardResvBW) meets MaxBW. If not, the host tries to make an additional soft reservation out of its free bandwidth (AvailBW in the flow table) to fill the gap. If successful, it updates SoftBW/HardBW in the packet accordingly. If every host on the path manages to make an additional soft reservation, the receiver will notice that the total bandwidth reserved (sum of SoftBW and HardBW) has increased. Therefore it sends an HR message back to change the reservation state (to hard reservation), and the sender can adjust the bandwidth to the new QoS level. Again, after this process, the host's state is switched from UPDATING back to ESTABLISHED (Alg. 2).

Similarly, the sender can scale down the real-time flow transmis-

### Algorithm 1 SR Processing (Intermediate Node)

---

```

1: receiving_message(sr);
2:  $totalResvBW = softResvBW + hardResvBW$ ;
3: if  $totalResvBW < SR.Max$  then
4:   if  $availBW \geq sr.max - totalResvBW$  then
5:      $softResvBW = sr.max - hardResvBW$ ;
6:      $soft\_reserving(sr.max - totalResvBW)$ ;
7:   else if  $availBW > 0$  then
8:      $softResvBW += availBW$ 
9:      $soft\_reserving(availBW)$ ;
10:  end if
11: if  $sr.hardBW \neq hardResvBW$  then
12:    $res = \min(sr.hardBW -$ 
13:      $hardResvBW, softResvBW)$ ;
14:    $softResvBW - = res$ ;
15:    $hardResvBW + = res$ ;
16: end if
17: if  $softResvBW < sr.softBW$  then
18:    $sr.softBW = softResvBW$ ;
19: end if
20: if  $hardResvBW < sr.hardBW$  then
21:    $sr.hardBW = hardResvBW$ ;
22: end if

```

---

sion in case the flow path changes and less guaranteed bandwidth can be maintained. Intermediate hosts release extra reservations upon receiving the HR message and the sender adapts its flow speed accordingly.

### 3.3.3 Local Path Repairing

Besides bandwidth variations, mobility can cause paths to break. ASAP has a simple but efficient local path repairing mechanism. When a host receives an SR that it has not seen before, and the SRs SoftBW and HardBW are not zero, it realizes that the flow is away from the original path, and it triggers local path reparation. The host will create a new flow entry in the flow tables and make a soft reservation according to MinBW/MaxBW in the SR. This procedure is the same as in the flow setup stage. Then, the host will switch its soft reservation to hard reservation by as much as it is indicated in the HardBW field of the SR, and keep the rest as soft reservation (lines 11 to 15 in Alg. 1). After that, the SR is modified and relayed to the next hop. When the SR message arrives at the receiver, the broken path has been repaired. The state switching of the host is as described in the previous sections. It is quite possible for the QoS provisioning on the new path to differ from the original QoS level. Depending on the information collected by the incoming SR messages, the receiver will decide whether to send an HR back to the sender to adapt the QoS levels.

### 3.3.4 Reservation Releasing

Releasing a reservation can be done either by implicit time-out of the flow path or by explicitly sending an HR message. By setting the SetBW field of the HR message to zero, each host which receives the message releases all its reserved bandwidth for the flow and removes the flow entry from the flow table. Then the host switches its ASAP state to CLOSED. The releasing by explicit signaling is fast, efficient but not robust. If the signaling is lost, the resource will not be released. In case the host maintaining the reservation state moves away from the flow path, it is impossible for the QoS framework to notice this wasted reservation, and to send signaling to release it. With soft-state mechanism, the un-

### Algorithm 2 Adaptation (Destination Node)

```
1: receiving_message(sr);
2: if sr.softBW > 0 then
3:   create_message(sr);
4:   hr.setBW = sr.softBW + sr.hardBW;
5:   end2endBW = sr.softBW + sr.hardBW;
6:   hr.softBW = 0;
7:   hr.hardBW = 0;
8:   sending_message(HR);
9: end if
10: if sr.hardBW < end2endBW then
11:   create_message(sr);
12:   hr.setBW = sr.hardBW;
13:   end2endBW = sr.hardBW;
14:   hr.softBW = 0;
15:   hr.hardBW = 0;
16:   sending_message(HR);
17: end if
```

used reservation state will time-out and be released if no signaling arrives within the time-out interval to refresh the state. So ASAP adopts both explicit signaling releasing and implicit time-out releasing mechanisms. Each SR and HR messages refreshes soft-/hard reservation states. If both hard and soft reservation states for a flow expire in a host, the host switches its state immediately to CLOSED by releasing the resources and erases the flow entry.

#### 3.3.5 Robustness

It is important for protocols in MANETs to cope with unpredictable message loss. As ASAP has only two signaling message: SR and HR, this issue is easier to solve. As SR messages are constantly inserted into the flow and included in the traffic packets, the losses of SR messages implies a traffic loss as well. The state switching of the host is not affected by SR losses either. Unlike SR messages, HR messages are sent only in response to the reception of SR messages with newly updated QoS states. Although HR messages are sporadically sent and the losses of HR messages prevent the hosts from changing their state from ESTABLISHED to ESTABLISHING/UPDATING, or to CLOSED, the system is not severely affected from the loss of HR messages. The reason is that if an HR message is lost, the reservation state on the flow path cannot be switched to hard reservation. The next incoming SR message can immediately discover the unchanged states, and inform the receiver end to re-send an HR message. The re-sending continues until the HR message is received and all the states of each host on the flow path have been switched to hard reservation. One issue related to the sending of HR messages is that it is assumed HR messages travel to the sender back along the path from the sender to the receiver so that each host on the path can switch its reservation state. This assumption is not always true in MANETs, where it is quite possible that the reverse path does not map to the original path. The asymmetry of the incoming and outgoing paths can invalidate the HR message's reservation mechanism: even though HR messages are received by the sender end, it is still possible that several hosts on the path have not received the messages and their reservation states have not been switched. In order to solve the HR loss issue, ASAP uses a hop counting mechanism: a HC (hop count) field is added to the HR message. Additionally, a constant value MAX\_HC (maximum hop count) is defined. Initially HC is set to MAX\_HC. If the HR message passes one host which is not on the path, HC is decreased by 1. If the HR message passes one host on the path, HC is reset to MAX\_HC. Obviously, when HC in the

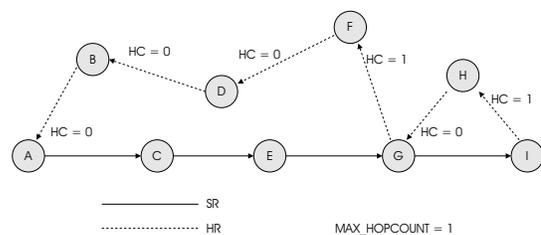


Figure 4: HopCount Mechanism for HR messages.

HR packet is zero, that means this packet has travelled MAX\_HC number of nodes which are not on the path, and far away from the right route. Then, the HR packet is forwarded directly to the sender, regardless of which route the message travels. Temporarily there may exist a partial reservation on the path while some hosts stay untouched. But once the HR message is received by the sender, the sender updates the flow table by setting the SoftResvBW and HardResvBW to the corresponding values within the message. Now the hard reservation state is done at the sender end. The next incoming SR message will bring the new reservation state information, and trigger a local repair procedure on the host which the HR message hasn't visited, as described in Section 3.3.3. Figure 4 is an example showing how the HC mechanism works.

## 4. EVALUATION

We have studied ASAP using the ns simulator version 2.27 [16]. The goal is, on the one hand, to show the behavior and the key features of ASAP and, on the other hand, to emphasize the performance and efficiency of the protocol. We therefore divide this evaluation in two parts. While in the first part topologies comprise only a few nodes and the scenarios are manually created, we use large topologies and random traffic and movement patterns in the second part. Throughout the evaluation we compare ASAP to IN-SIGNIA.

In order to guarantee equal treatment of both protocols, a common framework has been developed (Figure 5). It has four components: a *packets scheduler* responsible for forwarding the packets according to the bandwidth reservation of the associated flow, an *adaptation controller* providing a unified interface to the application, a measurement system and a bandwidth monitoring component dynamically estimating the actual available bandwidth on a given node. The packet scheduler is actually a deferred weighted fair queue (WFQ) with exact bandwidth warranties and contract violation detection. That means a flow is allowed to use its reserved bandwidth but no more, every additional fraction will be treated as best-effort. There is a special interface from the packet scheduler to the measurement system that allows collecting throughput information. Bandwidth estimation on each node is done by averaging the throughput of recent packets destined to a neighboring node. The throughput of each transmitted packet is computed as:  $TP = \frac{S}{t_r - t_s}$ , where S is the size of the packet,  $t_s$  is the time-stamp when the packet is ready to be sent at the MAC layer, and  $t_r$  is the time-stamp when an ACK is received. AODV [12] is used as a routing protocol.

### 4.1 System Behavior

#### 4.1.1 Reservation Mechanism and Localrepair

This first experiment is to provide evidence that ASAP is able to establish QoS flows in networks already saturated with BE background traffic. We also want to show the localrepair behavior of

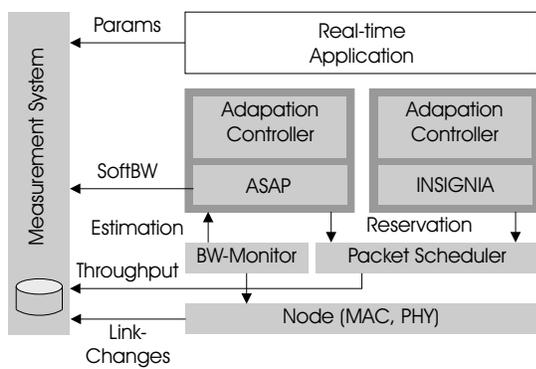


Figure 5: Simulation framework

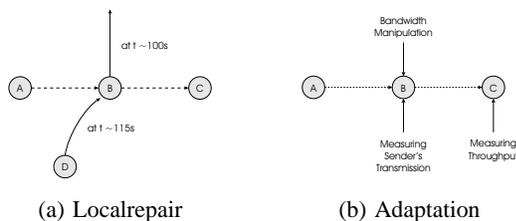


Figure 6: Scenarios

ASAP. Figure 6a illustrates the scenario we created comprising four nodes. In order to trigger a localrepair event, we introduced some simple movement. At time 100s node B starts moving upward as shown in Figure 6a breaking the path between nodes A and C. With a delay of around 15s node D also starts moving, closing the gap roughly at time 125s. We set the available link bandwidth on node B and D to not exceed 150 Kbps and also placed some BE background between nodes A and C, transmitting data at a mean rate of 250 Kbps. Figure 7a shows the traffic measured on node C when we add some CBR (Constant bit rate) flow ( $flow0$ ) with a data rate of 100 Kbps. The CBR flow and the BE background traffic compete for the available bandwidth which is actually less than required. As a result  $flow0$  suffers from a bad quality (Figure 7a). Figure 7b shows the same scenario but with added QoS for  $flow0$ . The CBR flow now benefits from a 100 Kbps reservation along the path and therefore pushes back BE background traffic to what is left. We also observe that localrepair works. Immediately after the path is rebuild reservation takes place on the new node D, leaving almost no room for BE traffic. The experiment was performed with an SR interval of 4.

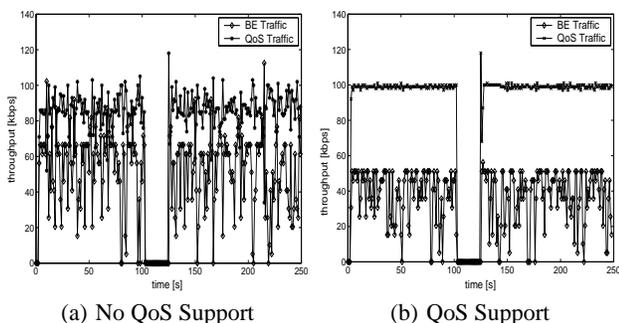
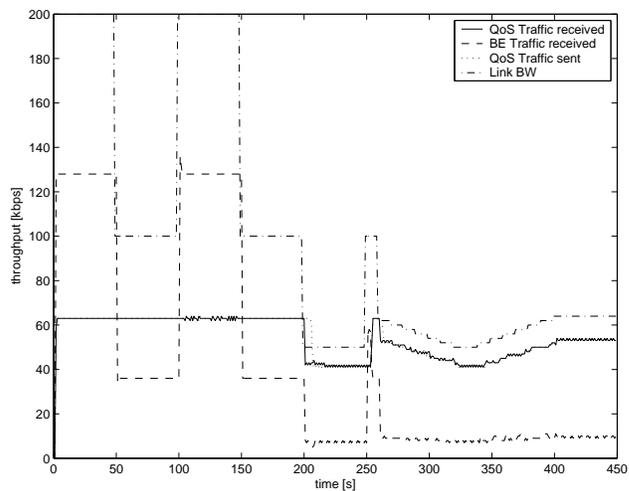
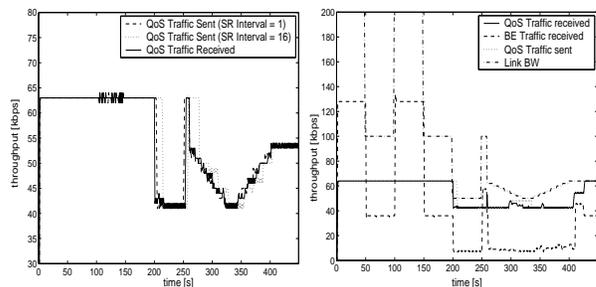


Figure 7: Reservation Mechanism and Localrepair



(a) ASAP



(b) SR Interval impacts agility (c) INSIGNIA

Figure 8: Adaptation

#### 4.1.2 Adaptation

One key feature of ASAP is its adaptive behavior. In order to measure how ASAP reacts to changing bandwidth we constructed a simple topology containing only three nodes (Figure 6b). Node A acts as a source node for both, a CBR traffic without QoS support ( $flow0$ ) and one including QoS support ( $flow1$ ). Both flows have destination node C. While  $flow0$  is sent at 128 Kbps,  $flow1$  is sent at 64 Kbps. We vary the available bandwidth on node B, and measure throughput on nodes B and C (referred to as  $t_B$  and  $t_C$ ). In an optimal case  $t_B$  and  $t_C$  should match, meaning that any change in bandwidth is immediately detected and signaled to the sender node. By using a  $\lambda$  of 0.85 the critical threshold for the QoS flow to run smoothly is roughly 75 Kbps ( $75 * 0.85 \sim 64$ ). Figure 8a illustrates the result using an SR interval of 4. We observe that bandwidth variations above 85 Kbps do not affect QoS flow, but BE traffic. Though as the available bandwidth on node B drops below the critical threshold even the QoS flow has to downgrade. But in contrast to  $flow0$  that keeps sending at its original bit-rate what leads to a huge drop-rate,  $flow1$  adapts its reservation and signals to the sender node who then adjusts its bit-rate to match the current reservation. Figure 8 proves the obvious assumption that the agility of the signaling protocol is strongly dependant on the SR interval. Using a bigger SR interval (16 in figure 8a) leads to longer reaction times. When running the same experiment with INSIGNIA, it turns out that the protocol is less agile than ASAP. As INSIGNIA can only support two QoS levels, it cannot follow fine grained bandwidth variations (Figure 8c).

## 4.2 Performance Analysis

We have set the topology to a rectangle of size 1000x1000m containing 35 nodes, each of them having a transmission range of 250m. Calculated according to the connectivity theorem shown in [6], the probability for two random chosen nodes to be connected over multiple hops in such a scenario is 99.8%. The soft state timeout is specified to be three times that of the SR interval (time between two SR messages). This is a reasonable compromise between providing a stable reservation and not propagating too much reservation in the network in case of high mobility. During all the simulations QoS traffic is allowed to use 85% ( $\lambda = 0.85$ ) of the current link bandwidth. So while adjusting the current link bandwidth, the actual available bandwidth for best-effort is adjusted as well. If the value for current link bandwidth drops below what is actually reserved for QoS traffic, then the existing reservation will be changed to meet the capacity constraint. We are running the exact same amount of BE and QoS traffic (10 CBR flows each, transmitting at a max data of 100 Kbps, QoS min rate is set to 0).

#### 4.2.1 Throughput

These experiments investigate the overall throughput. Analyzing QoS frameworks is best done in under-provisioned networks (obviously if the network has enough resources to cope with all the traffic, any QoS framework is not of much use). In such networks, QoS support typically goes at the expense of BE traffic, since the network capacity is limited. We distinguish four sorts of packets: QoS packets that arrive in a proper manner, QoS packets that are degraded (packets that get no QoS treatment but are forwarded as if they were best-effort), BE packets, and packets that have been dropped. For a packet to be counted as degraded it is enough be degraded by one hop on the path. We measure the fraction of degraded and dropped packets compared to the total packets sent (100%) for both, BE and QoS traffic. Figure 9 shows the results for ASAP and INSIGNIA. Note that what we totally send (BE and QoS, or 200%) is much more than what we finally receive (sum of black and grey parts of both, BE and QoS traffic). We observe that QoS treatment is generally much better than BE treatment, roughly double in case of ASAP. As mentioned, any QoS improvement happens at the cost of BE traffic if the network is saturated. In Figure 9a we see a slight degradation with increasing node movement in both diagrams. This makes sense as mobility leads to frequent route changes and eventually even to disconnections, making reservation establishment difficult. The simulation results also demonstrates ASAP's excellent QoS performance, compared to the one of INSIGNIA. One possible explanation is, that ASAP is rather efficient in terms of bandwidth utilization due to its soft/hard reservation mechanism and its adaptive behavior, and therefore leaves more room for other traffic, be it BE or QoS. As we will see, this assumption is proofed later in section 4.2.3.

#### 4.2.2 Reservation Efficiency

A QoS framework should not occupy bandwidth that is actually not needed. We measured the reserved bandwidth under different mobility conditions and observed that the part of the whole reservation that is actually soft reserved does not drop below 80% (10a). This proves the need to be careful with over-reservation. Also note the increasing behavior of the curve in Figure 10a. The faster the nodes move the more soft-reservation we measure in the whole network. There are two explanations for this: either it gets very difficult to turn soft reservation into hard reservation under high node mobility or the soft reservation messages can not pass through, leaving partial soft-reserved bandwidth.

We ran another experiment to clarify this issue. In particular, we consecutively collected information about the state of a real-

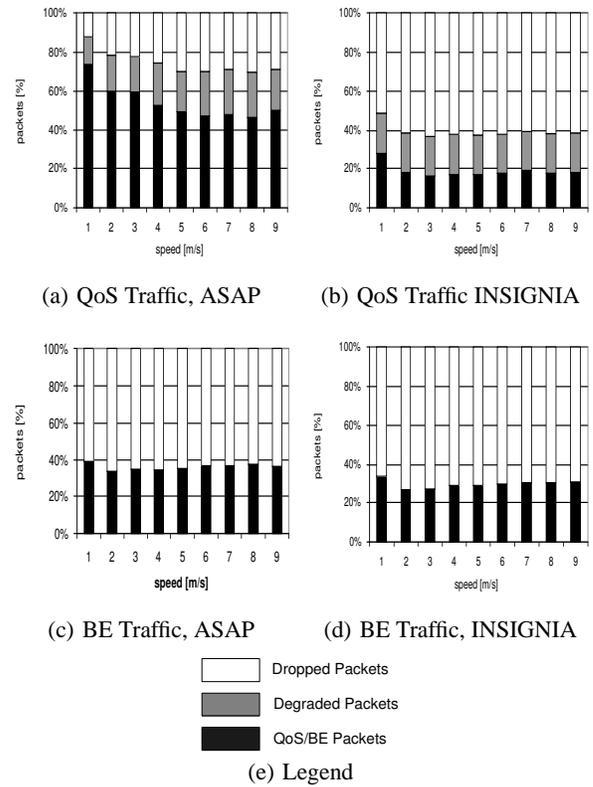


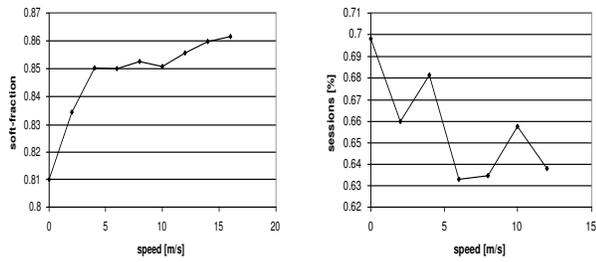
Figure 9: Throughput

time flow, whether it is soft-established or hard-established. A flow is called soft-established if a soft-reservation has been performed along the path, but no hard-reservation could be performed yet or has been timed out. In contrast, a flow is hard-established if hard-reservation is currently established. Figure 10b shows what we measured: The fraction of hard-established flows stays above 63 % within each simulation (this is an upper bound because our measuring counts sessions being disconnected but holding partial hard reservations as soft-established). Moreover it seems a slight node movement improves the process of establishing a hard reservation. However, if mobility increases, the number of hard-established flows decreases slowly. This is what we expected because node movement induces path changes and new routes have to be built.

To get back to the question raised in the beginning of this section, Figure 10b shows that an increasing soft-reservation with increasing mobility, as indicated in Figure 10a, is not due to an inefficient reservation process but to the partial reservation left by soft-reservation messages not reaching the destination node. But as soft-reservation is only a pre-booking, this does not affect best-effort traffic.

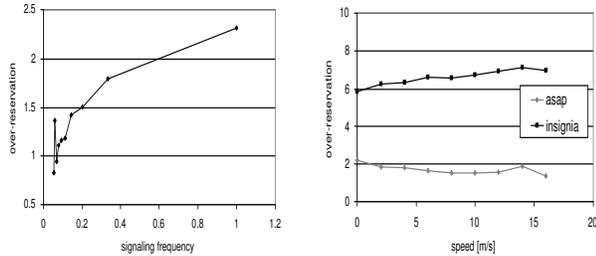
#### 4.2.3 Reservation Surplus

We define the relation of what is reserved and what is actually consumed as reservation surplus and investigate its behavior under different signaling frequencies and mobility conditions. In the best case, reservation surplus equals 1, when no reservation goes unused. Figure 11a shows how over-reservation is affected by the SR signaling frequency. The curve states that the higher the frequency (or the smaller the SR interval), the bigger the over-reservation. The reason is because a high signaling frequency together with node mobility causes reservation requests to be propagated all over



(a) Part of Reservation that is Soft-Reserved (b) Fraction of Soft-Established Sessions that can be turned in a Hard-Established state

Figure 10: Reservation Efficiency under different Mobility Conditions



(a) The higher the signaling frequency, the bigger the over-reservation (b) Over-reservation and Mobility

Figure 11: Reservation Surplus

the network. Figure 11b shows the over-reservation in the case of ASAP compared to the one of INSIGNIA, plotted against node mobility. The simulation result shows INSIGNIA having a much higher over-reservation than ASAP. This is for two reasons: First ASAP, by using its soft/hard reservation mechanism, actually saves a lot of bandwidth (we previously showed the soft-reserved part of bandwidth is up to 80%). Second, ASAP has a flexible signaling. We used an SR interval of 4 while INSIGNIA inserts control information in every packet causing more over-reservation (Figure 11a).

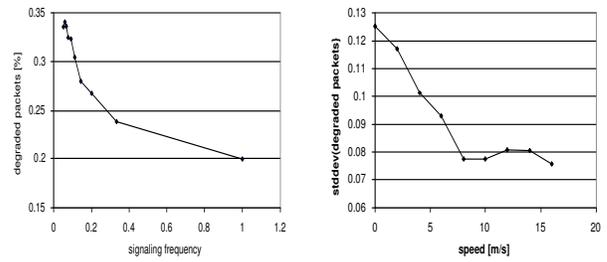
#### 4.2.4 Sensitivity

Not only signaling frequency impacts over-reservation, a high signaling frequency also leads to a smoother and more stable reservation, as shown in Figure 12a. So there is clearly a trade-off between service quality and reservation waste. This experiment is to show how sensitive QoS traffic is to changing signaling frequencies under various node mobility conditions. We compute the following value as a function of node speed:

$$\Phi^s = \sqrt{\frac{1}{20} \sum_{k=1}^{20} \left( \frac{d_k^s}{q_k^s} - \mu^s \right)^2}$$

Where  $s$  is the node speed,  $k$  denotes the SR interval,  $d$  is the total number of degraded packets,  $q$  is the total number of realtime packets, and  $\mu^s$  is the mean of  $\frac{d_k^s}{q_k^s}$  for a given speed  $s$ . This is the standard deviation of the degradation ratio viewed under different SR intervals.

Figure 12 illustrates the simulation result. Interestingly, QoS



(a) The higher the signaling frequency, the less degraded packets (b) Sensitivity to SR Interval

Figure 12: Sensitivity

provisioning is less sensitive to changing signaling frequencies under high mobility. This is the result of the reservation mechanism used in ASAP which in all cases produces deviations of less than 10%. At low mobility, there are topologies that are very favorable and situations that are less convenient and will remain so for some time. At high mobility the topology changes more frequently and, thus, the average results are more stable.

## 5. CONCLUSION

In this paper, we have presented ASAP, a framework for QoS in MANETS.

ASAP is intended as a flexible and highly adaptive signaling protocol capable of providing different QoS levels through a fair reservation mechanism. ASAP is rather efficient in terms of bandwidth utilization as it minimizes wasted resources during the establishment of QoS flows. This improvement is achieved by using a two phase reservation procedure based on two type of messages, one to establish a *soft reservation*, and one to convert the soft-reservation into a *hard reservation*. In the paper we have discussed in detail how ASAP works and how it deals with problems like adaptation to changing bandwidth conditions (both in cases of a diminution of resources as well as increases in the available bandwidth), local path repairing, and fair use of the available bandwidth.

The paper also includes an extensive empirical analysis of the behavior of ASAP. In particular, the simulations study ASAP under varying network conditions (node mobility, signaling frequency). We also compare the behavior of ASAP with that of INSIGNIA. The results presented demonstrate that ASAP surpasses INSIGNIA in terms of throughput while reserving less bandwidth. This feature of ASAP plays an important role in situations where the network must support both QoS and BE traffic since the available bandwidth with ASAP is larger due to the more efficient reservation mechanism.

ASAP is designed to be independent of routing protocols. As part of future and ongoing work, ASAP could be extended to increase synergies with the underlying the routing protocol. For example, one could imagine the routing protocol to notify ASAP whenever a route changes, reducing the localrepair time. Similarly, we are also looking into exploiting and adapting the MAC layer to improve QoS signaling, thereby making ASAP more streamlined and efficient.

## 6. REFERENCES

- [1] G. Ahn, A. Campbell, A. Veres, and L. Sun. Swan: Service differentiation in stateless wireless ad hoc networks. *IEEE INFOCOM'2002.*, 2002.

- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. *RFC2475*, Dec 1998.
- [3] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. *RFC1663*, June 1994.
- [4] R. Braden, L. Zhang, S. Berson, S. Herzogg, and S. Jamin. Resource reservation protocol (rsvp) - version 1 functional specification. *RFC2205*, Sep 1997.
- [5] Shigang Chen and Klara Nahrstedt. A distributed quality-of-service routing in ad-hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8), August 1999.
- [6] Madhav Desai and D. Majunath. On the connectivity in finite ad hoc networks. *IEEE Communications Letters*, 2002.
- [7] S. Lee, G. Ahn, X. Zhang, and A. Campbell. INSIGNIA: An IP-based quality of service framework for mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 60(4):374–406, 2000.
- [8] J. Manner and K. Raatikainen. Extended quality-of-service for mobile networks. *IWQoS*, 2001.
- [9] M. Mirhakkak, N. Schult, and D. Thomson. Dynamic quality-of-service for mobile ad hoc networks. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking and computing*, pages 137–138, 2000.
- [10] B. Moon and H. Aghvami. Reliable rsvp path reservation for multimedia communications under an ip micromobility scenario. *IEEE Wireless Communications*, 9(5):93–99, Oct 2002.
- [11] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. *RFC2474*, Dec 1998.
- [12] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.
- [13] Prasun Sinha, Raghupathy Sivakumar, and Vaduvur Bharghavan. Cedar: a core-extraction distributed ad hoc routing algorithm. *IEEE INFOCOM*, pages 202–209, 1999.
- [14] S. Shah and K. Nahrstedt. Predictive location-based qos routing in mobile ad hoc networks. *IEEE ICC*, April 2002.
- [15] A. Talukdar, B. R. Badrinath, and A. Acharya. MRSVP: A resource reservation protocol for an integrated services network with mobile hosts. *Wireless Networks*, 7(1):5–19, 2001.
- [16] The VINT Project. The NS network simulator.
- [17] C. Tseng, G. Lee, R. Liu, and T. Wang. Hmrsvp: a hierarchical mobile rsvp protocol. *Wirel. Netw.*, 9(2):95–102, 2003.