

Channel-aware Packet Scheduling for MANETs

Sridhar K N

Institute for Infocomm Research,
Singapore 119613
Email: sridhar@i2r.a-star.edu.sg

Mun Choon Chan

SoC, National University of Singapore
Singapore 117543.
Email: chanmc@comp.nus.edu.sg

Abstract

In this work, we present, CaSMA, a packet scheduling mechanism for mobile ad hoc networks (MANETs) that takes into account both the congestion state and the end-to-end path duration. We show that CaSMA approximates an ideal scheduling mechanism in terms of maximizing the goodput and sharing the throughput (losses) fairly among the contending flows. Further, the simulation results show that both average delay for CBR flows and throughput for TCP can be improved substantially compared to FIFO.

1 Introduction

Wireless medium is a shared and scarce resource, which is used by all nodes in the network. Efficiently controlling the access to this scarce resource is a complicated task. Resource management schemes play a major role in achieving this task. Packet scheduling is one such resource management scheme, which controls the allocation of bandwidth among multiple flows. Packet scheduling focuses in solving the problems associated with multiple sessions, within a single node, sharing the wireless link.

Our scheduling scheme considers challenges that are unique to MANETs (*changing topology, multihops and shared wireless medium*) by using the “channel-aware” approach. The term “channel-aware” in our work refers to having the knowledge of channel *conditions*. The term *condition* refers to the quality of the channel which can be measured in terms of suitable metrics. Terms “channel state” and “channel condition” are used interchangeably.

Channel conditions in wireless networks can be broadly classified as local and end-to-end channel conditions. For mobile ad-hoc networks, unlike wireless LANs, local and end-to-end channels are different. The difference between the local and end-to-end channel information can be better understood by considering their typical characteristics. We can consider 4 key categories as shown in Table 1: frequency (monitoring frequency of the channel state), gran-

ularity (representation of channel state), accuracy (correctness of the measurements that represent the channel state) and measured-time (time at which the monitoring of channel state is carried out) with respect to packet delivery.

Typical parameters that are used to represent the local channel information are received signal strength, signal-to-noise values, queue-length, burst-error mode, packet losses, single hop delay and link lifetime. Whereas, parameters that could possibly represent the end-to-end channel conditions are path lifetime, end-to-end packet delay and queue-length at every node. In our work, we focus on the end-to-end channel awareness and represent the end-to-end channel quality in terms of path lifetimes.

In this work, we present, *CaSMA*, a scheduling mechanism for mobile ad hoc networks (MANETs) that takes into account both the congestion state and end-to-end *path duration*. Our scheduling mechanism is termed Channel aware Scheduling for Mobile Ad hoc networks (CaSMA), where the term channel-aware is used to indicate both the *congestion state* and the *end-to-end path duration*. CaSMA is complementary to packet scheduling scheme that utilizes only local channel information, and can be added to these schemes.

During the path setup, the estimates of the path lifetimes are collected and stored. This path lifetime value is used as a parameter to represent the end-to-end channel condition. During packet scheduling, CaSMA selects packets, which has high probability of reaching the destination, and takes into account the cost of a link break by giving priority to flows that have a longer normalized (with path residual lifetime) backlog queue. We show that CaSMA approximates an ideal scheduling mechanism in terms of maximizing the goodput and sharing the throughput (losses) fairly among the contending flows. Further, the simulation results show that both average delay for CBR flows and throughput for TCP can be improved substantially compared to FIFO.

The remaining part of the paper is organized as follows. In Section 2, we discuss the related works, covering scheduling mechanisms in MANETs, and describing the contributions of our work. CaSMA is described in Section 3. We begin with describing the motivation for using channel aware-

| Category | Local | End-to-End |
|---------------|----------|-----------------|
| Accuracy | High | Low |
| Granularity | 2-Values | Multiple Values |
| Measured-time | Closer | Farther |
| Frequency | High | Low |

Table 1. Local versus end-to-end channel awareness

ness. Further, we describe the approach taken in CaSMA, the algorithm used for packet selection and the limitations of CaSMA. We conclude the Section 3 with the experimental evaluation of CaSMA. The paper ends with concluding remarks in Section 4.

2 Related Works

The term “scheduling” in multihop wireless networks usually refers to two problems - (1) Packet scheduling: which flow should be served among the set of backlogged flows within a node? (2) Channel access scheduling: which node should get access among the set of competing nodes in a “contention region”?

Majority of the previous proposals try to solve both the problems at the same time. In this process, however, the focus is more on the second problem (channel-access scheduling) rather than the first (packet scheduling). The standard approach used in all the earlier proposals for packet scheduling can be summarized as follows.

1. Choose a parameter (LC) that is locally computed and reflects only local conditions. For example, LC used by Kanodia et al. [9] in their scheme is termed as “priority index” or the “deadline” for each packet (considering a delay bound).
2. Choose a flow from a set of backlogged flows within a node using LC (minimum or maximum) values.
3. Modify the MAC protocol to approximate the global ideal scheduler. That is, priority to global minimum/maximum of LC is approximated by giving priority to the flow with local minimum/maximum of LC within a contention region.

There are various disadvantages with these approaches. To begin with, the flow model considered by previous works does not consider the validity-period of the flow, which depends on the quality of the path it is taking. The LC chosen by previous works are typically used for Internet (tagging), with slight modifications, which may not be suited for ad hoc networks. For example, majority of the works do not show how the chosen LC helps in achieving the objective (either fairness or throughput/delay bounds). Further, the approximation of the “local minimum/maximum of LC ” by modifying the MAC protocol to achieve “global (within a contention

region) maximum/minimum of LC ” is less accurate as LC does not reflect end to end behavior. Earlier works on channel quality aware scheduling [5] have considered only the local channel states. This is mainly because the channel state dependent research has focused more on wireless LANs (single hop networks), and less on multihop wireless networks. Finally, the impact of mobility is not investigated in earlier packet scheduling schemes. Mobility directly affects residual path lifetime, which is an end-to-end parameter.

Our work is different in the following ways. We focus only on the problem of packet scheduling. We do not propose any channel-access mechanisms. Any of the existing channel-access mechanisms can be used with our scheme. We propose a novel LC that reflects end-to-end conditions. According to our knowledge there is *no work considering end-to-end channel information in packet scheduling*.

3 Congestion and Path Lifetime Aware Packet Scheduling for Mobile Ad-hoc Network

In this section, we describe *CaSMA*, the scheduling mechanism for mobile ad hoc networks (MANETs) that takes into account local congestion information and end-to-end *path duration* information. We begin with describing the importance of considering channel awareness in general and path durations in particular. Further, we formally define the problem and describe the approach to solve the problem in detail.

3.1 Motivation for Considering Path Residual Lifetime

There are three aspects that we would like to investigate. First, how mobility affects the duration of the period between link breakage? We call this duration as the *flow on-times*, which is also termed as continuous period in our flow model. Second, how often does link breakage result in end-to-end route repair instead of local route repair? Finally, how expensive is end-to-end recovery as compared to local recovery?

3.1.1 Impact of Mobility on Flow On-Times

We studied the duration of link lifetimes for various mobility models, with maximum speed varying from 1 to 40 m/s. We found that the lifetime of the links can vary widely. This means that, at any node, if there are n flows, the lifetimes of those flows are unlikely to be similar and can vary over a large range. Therefore, considering these lifetime values of the routes as a parameter for scheduling can be useful, and can play a significant role in improving the performance.

3.1.2 Ratio of End-to-end and Local Route Repair

Due to dynamic nature, many routing protocols like AODV have in-built mechanisms for local route-recovery (route re-

pair) and end-to-end recovery. Typically local recoveries are triggered when routing protocol at any intermediate node gets packet transmission failure message from its MAC layer. Whereas, end-to-end recovery is triggered when routing protocol at source node receives error-message from any intermediate node.

We considered different mobility models, with maximum node speeds varying from 1 to 40 m/s. We found that for lower mobility (1 m/s) there were more local route-recoveries compared to end-to-end recoveries. As the node mobility increases, however, end-to-end recovery dominates, and the number of local recoveries becomes small for high speeds.

3.1.3 Impact of End-to-end Recovery

The impact of end-to-end recovery comes in two forms. First, recovery time is longer, in the order of milliseconds to tens of seconds. Second, the packets buffered by the nodes on the path before any link breakage will be lost. Therefore, the cost of a link breakage is also determined by the amount of data buffered by the nodes as these packets are lost, and depending on the application may have to be retransmitted. Hence, it is important to include queue size in the scheduling decision so that the backlog for each flow is reduced.

The conclusion we can draw from the study in this section is that, packet scheduling must take into account the end-to-end channel conditions. In addition, in cases where end-to-end recovery is common, it is also important to minimize the amount of backlog data in the flow.

3.2 End-to-End Channel State Representation in CaSMA

One of the key ideas in CaSMA is to represent end-to-end channel quality in terms of path lifetimes. The residual lifetime of a path reflects the current end-to-end channel state. Since the channel state keeps changing continually, the end-to-end path has temporal interval for which they are valid. We use the term *path lifetime* to define the time interval for which the path associated for a flow is valid or exists. If the lifetime of each and every link of path \mathbf{P} from node i to node j is estimated as l_1, l_2, \dots, l_n , then the path lifetime $P_{ij} = \min(l_1, l_2, \dots, l_n)$.

We consider path lifetime value in our scheduling process by using the shortest-path-lifetime-first approach. Such a scheduling process can be viewed as a variation of earliest-lifetime-first (ELF) approach. It can be easily proved that a scheduling process, which chooses packets based on shortest-residual-lifetime first, minimizes the maximum of the late-ness experienced [11]

The lifetime value is typically obtained by estimation techniques. The residual lifetime estimation techniques can be broadly classified as: measurement-based [2, 4] and

probabilistic-based [1]. In our work, we incorporate a measurement-based lifetime estimation technique. The exact estimation technique is out of the scope of this paper. *Our scheme assumes existence of at least one of the estimation techniques.* The details of the technique used in this work, for experimental studies, can be found in the works [4].

3.3 Flow Model

A flow, along with its span (start and end times), is also defined by its *breakpoints* and *continuous period*, as indicated in the flow model (Figure 1). A breakpoint is the duration of time during which an attempt to transmit a packet of that flow will result in failure. In this work, neglecting packet loss due to congestion, we will consider mobility as purely the reason behind the loss of link between the transmitter and the receiver. This duration can be in the order of seconds to minutes. The occurrence of breakpoints is more frequent in ad hoc networks compared to wireless LANs. CaSMA is designed exactly to handle these breakpoints. Therefore, the channel aware scheduling in ad hoc networks plays a more significant role compared to wireless LANs. The continuous period denotes the period during which the flow can receive the service. It is to be noted that after a continuous period, the source pauses for some time and start transmitting on a different path. Therefore it is important for all the packets of a flow to reach its destination before the end of continuous period.

For any given moment of time, we can only deal with single value of breakpoint and continuous period for any given flow, as we do not have any information about the future values. We consider the cost of not completing the service before the breakpoint as one or more of the following: (a) Packet queued at the intermediate node may not reach the destination after the continuous period. They may be dropped or may have to be retransmitted. (b) Any attempt to transmit these queued packets at the intermediate nodes may result in wastage of resources. (c) Packets might reach the destination unordered.

3.4 Problem Formulation

We formalize our problem as follows. Each request or flow i running through a path is described by a 6-tuple $(T_i, C_i, s_i, e_i, \{o_i\}, \{b_i\})$, where T_i is the minimum packet inter-arrival time, C_i is the maximum packet transmission time over a link, s_i and e_i are the start and termination period of a flow, finally $\{o_i\}$ and $\{b_i\}$ are the sets of continuous (duration for which the path exists or duration between link breakages) and breakpoint periods (duration for which path does not exist), respectively. We use o_i to represent single continuous period of flow i . The relation between s, e and o_i, b_i is as shown in Figure 1. Let us denote o_i^s and o_i^t as

starting and termination of a continuous period of flow i .

Let us define the *span* of a flow f as the interval $[s, e]$. The flow f can only be served within this span. Let us also define a schedule instance I , as a sequence (f_1, f_2, \dots, f_n) . How the flows are served is described by the schedule.

Formally a *schedule* for I can be seen as a function H , which can be defined as

$$H : R \rightarrow \{f_1, f_2, \dots, f_n\} \cup \{\emptyset\}$$

where $H(t \subseteq \text{span}(f_k)) = f_k$. That is, k 'th flow is served at time t . Further $H(t) = \emptyset$ means no flow is being served.

At any moment t_i , if a packet belonging to flow f receives a service ($H(t_i) = f$) at any of the nodes (except the penultimate node) in the path p , is said to be *partially served*. If this service is at the penultimate node of the path p , then the packet is said to be *completely served*. We also define $cs(f)$, which is the finite union of service (completely served) received by all the packets of flow f . $cs(f)$ is directly related to the goodput of a flow.

Further, to denote the *pending* state of any flow f , indicating the amount of workload remaining to be served for the queue at any time moment t , at any node, we define residue of flow as $\gamma(f, t)$.

Lastly, we define the important optimizing factors called *merit* and *backlog* of a schedule. The *backlog* is defined as the amount of packets that remain in the network at the end of their respective continuous period of all the flows.

$$\sum_{j=1}^n \gamma(j, o_j^t)$$

The merit $M(H)$ of a schedule is defined as:

$$\sum_{j=1}^n cs(j)$$

For each flow, the scheduler gains the merit based on the number of completely served packets. Packets which do get transmitted for a few hops and get dropped at any of the intermediate nodes will not contribute for the merit of the schedule.

The problem is to design a schedule, which over a period attains maximum *merit* and minimum *backlog*, and also fairly distributes the achieved merit among all flows. Minimizing backlog can serve two purposes. First, it reduces the delay, second, it reduces the loss due to link breakages.

In the remaining part of this work, we will focus only on the important flow parameters, and in this regard we will reduce the flow representation from 6-tuple to 3-tuple: (T_i, C_i, o_i) . This is mainly because at any given time the scheduler is aware of a single continuous period value, and the other three parameters s, e, b_i are not accessible to the scheduler. Further, important notations used in the remaining

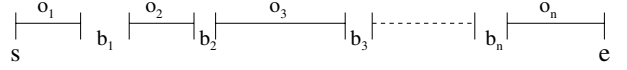


Figure 1. Flow model

| Notation | Description |
|----------|---|
| o | Continuous period |
| n, m | No. of flows |
| r | No. of packets existing in a queue |
| X | No. of packets served for a flow |
| α | Proportion of r of service received by a flow |
| ζ | Schedulable set |
| Γ | Flow set |

Table 2. Important Notations

part of this work is summarized in the Table 2, to make the reading easier.

3.5 Ideal Global Scheduler and Approximation

3.5.1 Global Ideal Scheduler

Let us consider a simple model with multiple flows over a single bottleneck link where we have a single scheduler. After the single shared link (with infinite lifetime), these flows use different (non-shared) links with different lifetime.

Let us assume a global scheduler - S_i , which schedules these flows ("m" flows). Let us consider a single continuous period o of "m" flows, with arrivals within this continuous period, and no further arrivals. That is, let us take a single snapshot in time of m flows with each flow having single continuous period of varying durations. For simplicity, let all flows have same $T = 1$, and $C = 1$. Therefore, S_i can schedule at most r_{max} packets in o_{max} period, where o_{max} is the maximum continuous period of any flow (or total interval of the snapshot). r_i represents the number of packets existing for flow i .

We, however, know that maximum number of packets existing in all the queues is $\sum_{i=1}^n r_i$. Let us call this value as r_{sum} .

Therefore, percentage ratio of throughput would be $\frac{r_{max}}{r_{sum}}$. Now, we adopt a fairness criterion, where this ratio is maintained across all the flows. In other words, the losses/backlog is proportionately distributed across all the flows. The idea here is that all sharers constrained by the same problem are treated fairly by assigning the proportionally equal throughputs. That is, for each flow i , the throughput it would receive is

$$r_i * \frac{r_{max}}{r_{sum}}$$

Further, losses at each queue would be

$$r_i \left(1 - \frac{r_{max}}{r_{sum}}\right)$$

The rationale behind having this formulation for an ideal scheduler is based on the argument that shorter continuous periods of flows are purely due to the inherent property of ad hoc networks. Therefore, we believe in not penalizing flows which suffer due to the inherent property of the network. Further, the scheduler will not be aware of the amount of service a flow has received in the previous continuous period (if existed) or the amount of service a flow will receive in the next continuous period. Therefore, we go by the assumption that providing equal proportion of service in the current set of continuous periods would probably prove to be advantageous.

3.5.2 Approximation

Our approximation to the global scheduler has two steps. First, we show that use of the parameter $\frac{QS}{RLT}$, where QS is queue size and RLT is the residual life time, approximate the ideal-scheduler described above. Second, we describe the schedulability list technique, which shows how the decision made at first node would be sufficient enough, and encompasses the decision for the whole path. Schedulability list technique also results in maximizing the merit of the scheduler.

3.5.3 Use of $\frac{QS}{RLT}$ to Approximate Ideal Scheduler

We have to show that when we schedule using $\frac{QS}{RLT}$, and when scheduling mechanism can schedule at most r_{max} packets, the number of packets served for each flow i is approximately $r_i * (\frac{r_{max}}{r_{sum}})$. The important idea here is to solve two problems:

- provide higher priority to flows which take short-lived paths, and
- proportion of service received for each flow will remain similar.

Let us consider the scheduling approach where queues with maximum value of $\frac{QS}{RLT}$ is always chosen, where QS is queue size and RLT is residual lifetime (also called continuous period o). Serving every queue considering $\frac{QS}{RLT}$ is similar to rate monotonic scheduling (RMS). RMS is an optimal, static-priority scheduling used in hard real-time systems [6]. Higher priority is given to a flow which has higher request rate. RMS aims at maximizing the number of tasks meeting its deadlines. In CaSMA, $\frac{QS}{RLT}$ acts as a request rate. Therefore, serving queues which has higher $\frac{QS}{RLT}$ values first will result in providing higher priority to flows which take short-lived paths.

Now we have to solve the second problem of providing equal proportion of service. This is because, considering only $\frac{QS}{RLT}$ may not provide equal proportion of service. Let us consider a simple model, which is a single snap-shot in time,

where we have n flows with each flow i having workload (number of packets) as r_i . Let the service time for all packets be same (1 time unit). Further, as all flows have T and C set to 1, then $o_i = r_i$, i.e., RLT for each flow will be the same as r_i (to begin with all flows have same request rate). Let the maximum number of packets the scheduler serves in the given time duration (or the maximum duration of time snapshot) be maximum of r_i values, termed as r_{max} . Let us term the number of packets served for flow i be X_i , and use r_{sum} to represent sum of all r_i s.

We know that QS either decreases or remain the same (as we consider single snapshot in time and no further arrivals), and RLT is strictly decreases. Therefore request rate ($\frac{QS}{RLT}$) can either remain the same or increase. Using the above model and notations we can rewrite request rate as

$$\frac{r_i - X_i}{r_i - \sum_n X_j}$$

Let us define another parameter α ($0 < \alpha < 1$), which is the proportion(percentage) of r_i of service that any flow i receives at any given time within the considered time duration. The important point to note here is that, there is no one-to-one mapping between the request rate considered and proportion of service received (α). That is, if a flow i has greater request rate than the other flow j , then it may not mean that amount of the service (proportionately, α) received by the flow i is lesser than j . In fact, when the o_i s varies to a larger extent, the proportionate amount of services received by flows can also vary to a larger extent (flows with shorter continuous periods (o_i) will receive proportionately greater service).

For a special case where o_i s are same, if a flow has received lesser proportion of service than the other flow, then its request rate will always be higher than the other flow. Under these conditions (similar o_i s), it can be shown that serving by $\frac{QS}{RLT}$, results in fair distribution of service.

We have seen in the preceding section (Section 3.1) that o_i values can vary to a great extent. Therefore, we need to avoid the condition where short-lived flows can receive proportionately greater service. We achieve this by having an additional parameter termed as *eligible - service*, for each flow. This *eligible - service* for any flow i is equivalent to $\frac{r_{max}}{r_{sum}}$, and is computed by considering the r_i s, which is given as follows:

$$\frac{(r_i * \frac{C_i}{T_i})}{\sum_{j=1}^n r_j * \frac{C_j}{T_j}} * (r_{max} * \frac{C_{max}}{T_{max}})$$

C_{max} and T_{max} indicates maximum possible C and T , respectively. The first term indicates the ratio of the work to be performed for a flow i and the total amount of work considering all flows. Whereas, the second term indicates the

maximum work that can be done, and this term, in practice, is related to the maximum wireless link rate.

We update this parameter (*eligible – service*) only when new flows arrive or existing flows leave. The priority is given to flows considering both the request rate and eligible-service. Higher priority is given to flows whose request rate is high, and which has not yet received its eligible-service. This parameter will ensure that flows do not receive greater service (in proportion) at the cost of other flows.

In the remaining part of this section, we will describe how we enhance the approximation of ideal scheduler by considering end-to-end packet scheduling.

3.5.4 Schedulability

A set of flows Γ is said to be “schedulable” (S) if none of the flows has packets queued in the intermediate nodes at the end of their respective continuous periods. Any set of flows at a node that are schedulable over a link is termed as “schedulable set”.

We consider the following two problems related to flow schedulability.

- First, we have to consider that given a set of n flows $\Gamma = (T_i, C_i, o_i), i = 1, 2, \dots, n$, how many of them ($m, m \leq n$) are schedulable over a link? (schedulable set).
- Second, suppose there are n flows $\Gamma = (T_i, C_i, o_i), i = 1, 2, \dots, n$, of which m flows form a schedulable set ζ . Now, given a new flow j , what is the maximum value of its continuous period (o_j), such that the new flow will be subset of the schedulable set (may result in preemption of a flow existing in the current schedulable set).

We will provide the solution for the above two problems, which will be used in our scheduling algorithm. First, let us begin with the schedulable set (ζ). A schedulable set (set of flows that are schedulable) is derived as follows. Let us assume that a node has n flows, of which it has to choose m flows to form a schedulable set. We use the classic result of real-time scheduling [7], and define the necessary condition for a set of flows to be schedulable over a link is given as

$$\sum_{i=1}^m \left(\frac{C_i}{T_i} \right) \leq 1 \quad (1)$$

We can also rewrite the above necessary condition in terms of the packets scheduled. Since the minimum packet inter-arrival time of a flow i is T_i , there are at most $\frac{(o_i)}{T_i}$ packets arrived over channel i during the interval, and which need at most $\frac{(o_i)}{T_i} C_i$ units of time to transmit. Now the summation of this time for all the m flows should be less than the r_{max} (maximum number of packets the scheduler serves), which is written as

$$\sum_{i=1}^m \left(\frac{o_i}{T_i} \right) C_i \leq r_{max}$$

In addition, we know that there are different combinations that are possible in choosing m flows out of n flows (C_n^m). We know that the value of m is dependent on the C_i and T_i values. For example, value of m becomes smaller for smaller values of T_i . Hence, we have to decide on a specific way to choose m flows out of n flows.

In our work, we choose the m flows considering the residual lifetime values of the flows. Scheduling based on residual lifetime is similar to earliest deadline scheduling (EDF). Therefore, based on the results from EDF scheduling [6] and adhering to the approach of choosing smallest residual lifetime first, we sort all the n flows in terms of the increasing residual lifetime, and from this sorted set we choose the first m flows. These m flows form our schedulable set ζ .

To simplify the understanding, let us take an example, as shown in Figure 2. Let $\{a, b, c, d\}$ be the flows at node ‘D’. Let $\{2, 4, 4, 6\}$ and $\{1/2, 1/4, 1/4, 1/6\}$ be their continuous periods and rates ($\frac{C_i}{T_i}$), respectively. SCH indicates scheduler at node ‘D’. Node ‘D’ chooses flows $\{a, b, c\}$ as schedulable following the condition given by equation 1. Flows $\{a, b, c\}$ are chosen considering their continuous periods and the rates. We can see that an addition of flow d will violate the condition, that is, summation of the rate values ($\frac{C_i}{T_i}$) will be greater than 1.

The solution to the second problem follows the solution of the first problem. If a node has a set of flows Γ passing through it, we define a schedulable set ζ ($\zeta \subseteq \Gamma$) where ζ is the set of flows which are schedulable at that particular node. Let the maximum continuous period in the set ζ be o_j of some flow j . The schedulable set ζ also satisfies the necessary condition provided above. Now the maximum value of continuous period for a new flow, say k to be schedulable is to be lesser than o_j , and the arrival rate is lesser than or equal to j 's arrival rate. That is, a new flow k with continuous period o_k will be schedulable, iff $o_k < o_j$ and $\frac{C_k}{T_k} \leq \frac{C_j}{T_j}$. This is because, the schedulable set is built considering two conditions - residual lifetime and the necessary condition as given above (equation 1).

If the continuous period of the new flow (k) is lesser than the continuous period of a flow (j), where flow j is both a member of the existing schedulable set and has a maximum continuous period in the set, then the new flow (k) will be added into the schedulable set at the expense of this existing flow (j , which had maximum continuous period will be preempted). In addition, the second condition ($\frac{C_k}{T_k} \leq \frac{C_j}{T_j}$) is important to make sure that the new schedulable set does not violate the condition given by the equation 1. Therefore, for a flow to become eligible as a member of the existing schedulable set is that its continuous period be lesser than the maximum continuous period in the existing schedulable set.

Considering the example in Figure 2, if a new flow has to become schedulable then its continuous period has to be < 4 , and request rate has to be $\leq \frac{1}{4}$.

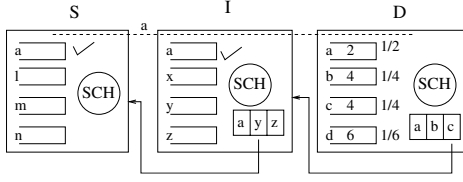


Figure 2. Scheduling example

The solution to the second problem leads to the notion of a flow i being “schedulable” (S) at node l . This notion provides an important parameter in our analysis, as it is used in two ways: (1) An end-to-end measure of this value during the path set-up helps the source to decide on initiating the traffic (2) Intermediate nodes make their scheduling decision based on these values, which can be updated by the downlink neighbors whenever value changes.

We know that if a flow is schedulable at all the intermediate nodes, then it is schedulable over the path. The idea is analogous to the *series of traffic lights*. It is useful to turn the first light green when all the remaining lights will turn green within some acceptable duration. This technique helps in increasing the *merit* (as described in the problem formulation section) of a scheduler, as priorities are given to packets which will be “completely served”.

The notion of schedulability takes on only binary values (TRUE/FALSE). When we use this parameter in the algorithm, the mechanism just makes the decision for given values and existing conditions. This decision process is used to build the schedulable-list message, as described below, following the example in Figure 2.

Consider three nodes S , I , and D as shown in Figure 2. We will focus on a single flow ‘a’ starting at node ‘S’, with intermediate node ‘I’ and terminating at node ‘D’. Let $\{a, b, c, d\}$ be the flows at ‘D’. Let $\{2, 4, 4, 6\}$ and $\{1/2, 1/4, 1/4, 1/6\}$ be their continuous periods and rates ($\frac{C}{T}$), respectively. Node ‘D’ chooses flows $\{a, b, c\}$ as schedulable following the condition given by equation 1, and creates a schedulable-list message (list of flows schedulable), which is transmitted to the upstream neighboring nodes. When ‘I’ receives this message, marks flow ‘a’ as schedulable at downstream (sets the schedulability value to TRUE), and builds its own schedulable-list (let it be $\{a, y, z\}$) and transmits it to its upstream neighbors. In this manner, the schedulable-list message flows upstream until it reaches source node ‘S’, which upon receiving will mark flow ‘a’ as schedulable (at downstream). If either the destination node or any of the intermediate nodes does not include flow ‘a’ in their schedulable list message, then the source node will not set flow ‘a’ as schedulable (at downstream).

3.6 Algorithm and Limitations

A single queue is maintained for every destination of the flows that a node carries, i.e., different flows to the

same destination are enqueued in the same queue. The Dynamic Source Routing (DSR) [8] is enhanced to implement schedulability-list technique. The algorithm is as shown in **Algorithm 1**.

Algorithm 1 Packet selection in CaSMA

- 1: **repeat**
- 2: Consider a set of high-priority (real-time) queues
- 3: From this set of queues.
- 4: Select the set of queues, such that for every queue q , $q.schedulability = \text{TRUE}$
- 5: **if** No queue satisfies the condition ($q.schedulability = \text{TRUE}$) **then**
- 6: Select all the queues.
- 7: **end if**
- 8: From these selected queues:
- 9: Select queue q such that the value $\frac{QS(q)}{RLT(q)}$ is the maximum, and who have not yet received *eligible-service*
- 10: In case of tie select flow that has received least *throughput*
- 11: **until** all queues are empty

Some of the limitations of CaSMA are: CaSMA assumes a path/link lifetime estimation technique. As no standard technique exists till date, CaSMA’s performance varies as the accuracy of link estimation varies. Further, neighbor management and schedulable-list scheme can add overhead with respect to bandwidth consumption, especially for high-mobility scenarios.

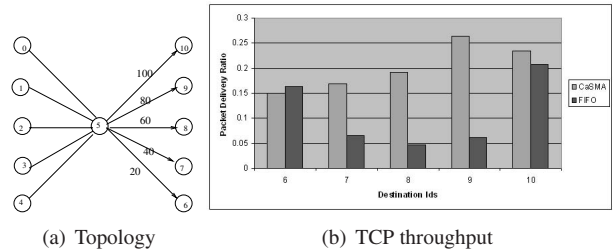


Figure 3. Packet delivery ratio for different flows

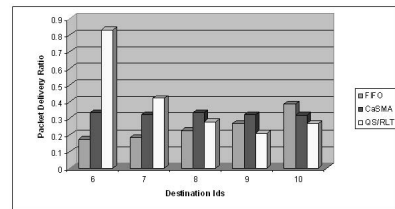


Figure 4. Packet delivery ratio for different flows

3.7 Experimental Evaluation

In this section, we describe the experimental evaluation of CaSMA. In the first part of the simulation we consider a scenario where the scheduler has perfect knowledge of the link lifetimes. The goal of this part is to provide the reader a better understanding of the advantages of CaSMA, when there are no lifetime estimation errors. In the second part of the simulation, we consider scenarios where link lifetime is estimated, and we compare the performance of various scheduling mechanisms. All our evaluations are carried out on NS-2 [3] simulator. Each mobile host has a transmission range of 250 m and shares a 2 Mbps radio channel with its neighbors. The simulation includes a two-ray ground reflection model and IEEE 802.11 MAC protocol.

3.7.1 Performance Comparison with Known Path Lifetime

In this section, we focus on understanding the significance of the parameters considered (QS , RLT and eligible-service). We considered a simple topology of 11 nodes, and simulation duration of 100 seconds. The topology is as shown in Figure 3(a). The source-destination pairs are [(0,6),(1,7),(2,8),(3,9),(4,10)], with single intermediate node 5. In Figure 3(a), the numbers shown on links between node 5 and {6, 7, 8, 9 and 10} indicate the respective link lifetimes.

We consider CBR flows transmitting at 400 kbps. The packet delivery ratios are shown in Figure 3(b). The delivery ratios for CaSMA are both even and higher compared to FIFO. For flow [0-6], FIFO has slightly better delivery ratio than CaSMA, but it performs badly for other flows. The delivery ratios are higher for CaSMA because CaSMA does not make an attempt to transfer those flows, whose link lifetime has expired. This shows that CaSMA is designed to provide service to the flows within their “lifetime” and not beyond that.

To focus on the importance of *eligible-service*, we slightly modified the source-destination pairs. Now, all the 5 flows initiate from node 5, flowing towards same destination, with same RLTs. The transmitting rate, however, is increased from 400 kbps to 600 kbps. Figure 4, shows the packet delivery ratios for FIFO, CaSMA and $\frac{QS}{RLT}$ (without eligible-service). We can see that CaSMA, achieves both better packet delivery ratio and proportionate share. Though $\frac{QS}{RLT}$ (without eligible-service) performs better than FIFO, the division of share is not fair (flow [5-6] gets proportionately greater share). This is precisely the case for which eligible-service is included to handle, which results in providing fair share. It can also be seen that performance trend of FIFO and $\frac{QS}{RLT}$ tend to be opposite. That is, FIFO’s performance increases with link lifetimes, whereas $\frac{QS}{RLT}$ ’s performance decreases with link lifetime.

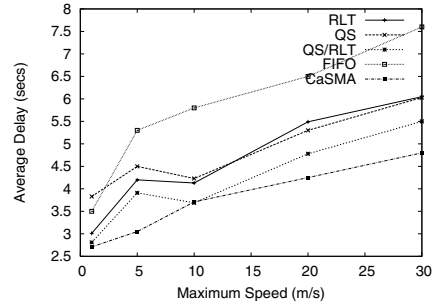


Figure 5. Average delay versus maximum speed

In summary, CaSMA is designed to perform such that flows with lesser residual lifetime get higher preference, and the losses (throughput) will remain proportionately same for all contending flows.

3.7.2 Performance Comparison with Estimated Path Lifetime

In our second part of the simulation, we consider scenarios where lifetimes are estimated. This mobile environment is considered to emphasize on the advantage of using schedulable-list scheme along with the other parameters ($\frac{QS}{RLT}$ and eligible-service). We consider a network with 50 mobile nodes, with area 1000 m x 1000 m. All the simulations are run for 1000 seconds, with 8 replications. In this part of evaluation, maximum speed of the node is varied from 1 m/s to 20 m/s.

The five mechanisms chosen are: FIFO (First In First Out), RLT (considering only residual life time), QS (considering only queue size), QS/RLT (considering both queue size and RLT), CaSMA (considering, queue size, RLT and schedulable-list).

For the first set of plots, we use 10 CBR flows with the transmission rate of 500 kbps. Figure 5 shows the plot of average delay values for all the five mechanisms. Considering the Figure 5, CaSMA performs best among all the schemes. This can be attributed to both the parameter chosen and the schedulable-list technique. Considering only the performance between QS/RLT and CaSMA, we can see the advantage of using schedulable-list technique.

To have a better understanding of the advantage we note the maximum and minimum of delay values, considering only FIFO and CaSMA. We found that the maximum values of CaSMA are also lesser compared to FIFO, whereas minimum values are almost the same. The main reason behind the reduction in delay values (average and maximum) is due to a reduction in the backlogs (or γ values, as described in preceding sections). The increase in backlogs can result in transmissions after a route-recovery delay. The backlog increase also has effect on the losses.

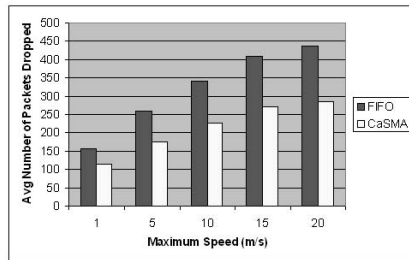


Figure 6. Number of packets dropped at queue due to link breakage versus maximum node speed

We also found that, with CaSMA the sharing of bandwidth is more fairer and better and had 25% less packet loss compared to FIFO. Figure 6 shows the number packets that are dropped at the queue due to link breakages (CaSMA in comparison with only FIFO). This parameter is directly related to the amount of backlog. From the figure, we can see that the backlogs using CaSMA is reduced by more than 30% - 40%. Further, we can see that increasing the frequency of topology changes, the amount of backlog also increases.

We further consider 10 TCP flows, and study the TCP performance in such scenarios. TCP flows are considered because, if the scheduler attempts to schedule a packet whose path residual lifetime has expired, with high probability, it will result in dropping. This dropping will force TCP to reduce the congestion window, and in turn reduce the throughput. We studied the throughput performance of various schemes. We found that the TCP throughput for CaSMA increased in some cases up to 50% over FIFO. The reasons behind better TCP performance are the same as provided in the first part of this section.

4 Summary

In this paper, we proposed a novel scheduling mechanism considering the inherent feature (existence of multihops) of MANETs. We consider end-to-end channel condition represented as residual lifetime for channel-awareness, and also included a queue size parameter to make the scheduling scheme congestion-aware. This combination of parameters avoids the congestion and reduces the accumulation of packets (backlogs) at the end of flow on-times. Further, we included a schedulable-list technique, which apart from providing better end-to-end co-ordination and approximation to an ideal scheduler, also increases the merit (number of completely served packets) of the scheduler.

References

[1] J. Shengming, H. Dajiang, and R. Jianqiang, "A prediction-based link availability estimation for mobile

ad hoc networks," in *Proceedings of INFOCOM 2001*, Anchorage, Alaska, April 2001, pp. 1745–1752.

- [2] M. Gerharz, C. de Waal, M. Frank, and P. Martini, "Link stability in mobile wireless ad hoc networks," in *Proceedings of LCN 2002*, Tampa, Florida, November 2002, pp. 30–42.
- [3] F. K. and V. K., "NS notes and documentation," The VINT Project, UC Berkely, LBL, USC/ISI, and Xerox PARC, Available from <http://http://www.isi.edu/nsnam/vint/>, 1997.
- [4] K. N. Sridhar and C. M. Choon, "Stability and hop-count based approach for route computation in MANET," in *Proceedings of ICCCN 2005*, San Diego, California, October 2005, pp. 25–31.
- [5] P. Bhagwat, P. P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs," *ACM Wireless Networks*, vol. 3, no. 1, pp. 91–102, 1997.
- [6] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.
- [7] J. P. Lehoczky, L. Sha, and Y. Ding, "The rate monotonic scheduling algorithm: Exact characterization and average case behavior," in *Proceedings of IEEE RTSS*, IEEE. Santa Monica, California: Computer Society Press, Dec 1989, pp. 166–171.
- [8] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: the dynamic source routing protocol for multihop wireless ad hoc networks," *Ad hoc networking Boston, MA: Addison-Wesley Longman Publishing Co., Inc.*, pp. 139–172, 2001.
- [9] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. W. Knightly, "Distributed priority scheduling and medium access in ad hoc networks," *ACM Wireless Networks*, vol. 8, no. 5, pp. 455–466, 2002.
- [10] B.-G. Chun and M. Baker, "Evaluation of packet scheduling algorithms in mobile ad hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 3, pp. 36–49, July 2002.
- [11] T. M. Chen, J. Walrand, and D. G. Messerschmitt, "Dynamic priority protocols for packet voice," *IEEE JSAC, special issue on mobile and wireless networks*, vol. 7, no. 5, 1989.