

Credit-Based Fair Scheduling in Ad Hoc Wireless Networks

Hsi-Lu Chao and Wanjiun Liao

Department of Electrical Engineering

National Taiwan University

Taipei, Taiwan

Email: wjliao@cc.ee.ntu.edu.tw

Abstract This paper studies fair scheduling in ad hoc wireless networks. We propose a credit-based mechanism to ensure fairness for best effort flows. Each flow is associated with a credit value, and is scheduled based on the concept of “the less excess in usage value, the higher the transmission priority.” The network is logically partitioned into clusters, each with a scheduler, and a unique code. The schedulers first assign time slots to mobiles in the affiliated clusters. The mobile scheduled to send at the next time slot then in turn assign the time slot to a flow. Each scheduler operates independently, and together, they allow flows in the network to share bandwidth in a fair way. We have also performed simulations to validate the proposed mechanism. The results show that the proposed mechanism provides fair share of bandwidth to all flows, and also improves overall system throughput.

Keyword: credit-based, fair scheduling, ad hoc networks

1. Introduction

This paper studies fair scheduling for best effort flows in an ad hoc network. An ad hoc network is a self-organizing wireless network comprised only of mobile nodes without the support of any pre-existing wired infrastructure. In such a network, each node plays both roles of a terminal and a router. Due to the fully-distributed characteristic of the network, collisions may occur and the system throughput may be low without proper coordination among the transmitting nodes.

Fair scheduling is a mechanism which allows all participating flows to share resource fairly. While this issue has been intensively studied in the context of wired networks, very little work has been proposed for ad hoc networks. [1,2] are the only two mechanisms proposed for ad hoc networks that we are aware of. Both are timestamp-based and need to convert a node graph into a flow graph first as in [3].

In a flow graph, a vertex indicates a flow, and an edge must be added when two flows are contending for resource. For each newly arrived packet, two timestamps are assigned, namely, the start tag and finish tag. The start tag is set either to the system time at which the packet arrives, or to the finish tag of its previous packet, depending on which value is larger. The finish tag is set to the predicted finishing time, which is equal to the start tag plus the estimated packet transmission time. Either timestamp can serve as the service tag. The packet with the least service tag will be transmitted first.

Spatial channel reuse is an important technique to improve system throughput for wireless networks. In ad hoc networks, each node functions as both a terminal and a router, and works independently. Without proper coordination, all simultaneous transmissions within the same transmission range will collide and be destroyed. However, flows located far away will not interfere with each other. This is called spatial channel reuse. The difference between [1] and [2] is in the mechanism used to implement spatial channel reuse. In [1], graph coloring theory is adopted to partition all vertices into several disjoint sets. Those flows in the same set will not contend for resource and can be transmitted simultaneously without collisions. In [2], a backoff mechanism is used instead. For each flow, the number of neighboring flows with smaller service tags is used as its backoff value. When this backoff value becomes zero, the packet can be transmitted.

The timestamp-based algorithms suffer two problems. One is the need to sort all packets in the queue in ascending order of their service tags. The other is the virtual clock cannot be reinitialized to zero unless the system is empty [4].

In this paper, a credit-based fair scheduling mechanism is proposed. Each flow simply maintains a counter to record the transmission credit, instead of using two tags as in timestamp-based mechanisms. The basic scheduling concept is “the less excess in usage value, the higher the transmission priority.” In addition, it makes use of clustering to implement spatial channel reuse.

The rest of this paper is organized as follows. Section 2 describes the proposed mechanism. Section 3 shows the simulation results. Finally, the conclusion is drawn in Section 4.

2. Proposed Mechanism

2.1 Assumptions

1. In this paper, we assume that errors are caused only by collisions.
2. We assume a TDMA-based system on a single channel shared by all hosts. To avoid collisions, CDMA can be overlaid on the top of the TDMA system. We assume a code assignment algorithm is running in the lower layer of our system.
3. We consider packet-switched multihop wireless networks, but do not consider host mobility as in existing work [1, 2].

2.2 Cluster Formation and Scheduler Selection

The network is logically partitioned into clusters. Each cluster has exactly one scheduler. A cluster can be formed by approaches such as [5,6]. Briefly speaking, we can predefine the radius of a cluster. Those nodes within the circle defined by this radius are in the same cluster. The mobile node with the smallest node ID, or the node staying in the cluster the longest then can be elected as the scheduler. The scheduling done in circles will be described in Sec. 2.4.

2.3 Scheduling Parameters

There are four parameters used in this mechanism, including *Num*, *Credit*, *Usage*, and *Excess*.

1. *Num*: the number of flows scheduled by a node. This parameter appears only in the Node

Allocation Table (NAT) used in Sec. 2.4.

2. *Credit*: the remaining slot quota for a request per iteration. The corresponding *Excess* value is updated when this value becomes zero. Once the *Excess* value has been updated, the *Credit* value is reset to a *Num* value, and a new iteration starts.
3. *Usage*: cumulative time slots assigned to a request. For a scheduler, it records a *Usage* for each individual node. For a node, it records a *Usage* for a flow. The *Usage* value may be a zero or any positive integer.
4. *Excess*: this value is used to determine to whom the next time slot is assigned. The next slot is assigned to the request with the smallest *Excess*. The initial *Excess* value is zero. The *Excess* value increases by one at each update.

2.4 Service Registration

A flow sender must first determine a flow path before a transmission starts. The sender broadcasts a Route REQuest (RREQ) message including a flow ID, a source ID, a destination ID, a previous-hop node ID, a next-hop node ID, and a *Num* to its neighbors. Upon receiving a new request, say, *i*, to relay the flow's packets, the node, say, *N*, will temporarily create an entry to record the flow information in its Flow Allocation Table (FAT). Node *N* then writes its own ID in the “previous-hop node ID” field of the RREQ, and rebroadcasts to its neighbors. A duplicate RREQ is discarded upon receipt.

When the destination node receives the first RREQ, it sends a RREP message in reply. The destination copies all information in the RREQ message, and further writes its own ID in the “next-hop node ID” field. It then unicasts this RREP message to the source node along the reverse path of the corresponding RREQ. Upon receipt of an RREP message, each node on the path records the node ID from which the RREP received in its routing table. The node then switches the state in the corresponding entry in its FAT from temporary to regular. The node also needs to register with a scheduler. According to its next-hop node, it determines to whom the registration is made. The rules to determine the scheduler for a node are listed as follows. For node *N*,

- (1) If node N and its next-hop node can hear the same scheduler S , node N sends a registration request to S ;
- (2) Otherwise, node N sends a registration request to its scheduler.

Node N will update the number of flows under its service in the “Num” field of the registration request. Upon receipt of the registration request, the scheduler creates/updates an entry in its Node Allocation Table (NAT). Once the RREP message reaches the sender, the shortest path is constructed.

2.5 Slot Allocation Algorithm

In our protocol, the slot is allocated based on a two-tier hierarchical mechanism. The scheduler assigns the next time slot to a node. The node then in turn assigns the time slot to a flow. Both the scheduler and the node allocate slots only to regular entries of their respective tables. Such allocation is repeated as long as there is a flow relayed through any ad hoc node.

Suppose that there are m and n entries in both NAT and FAT, respectively. The slot allocation rules used by the scheduler are described as follows.

- (1) The scheduler assigns the next time slot to the node with the smallest Excess value in the NAT at decision time

Assume node x is scheduled to transmit at the next time slot. The scheduler then updates the values of Usage, Credit, and Excess in the corresponding entry of the NAT, as follows. Other entries remain intact.

- (2) Increment the Usage value of node x , i.e., $Usage(x) \leftarrow Usage(x) + 1$.
- (3) Decrement the Credit value of node x , i.e., $Credit(x) \leftarrow Credit(x) - 1$.
- (4) If the Credit value becomes zero, the Excess value is increased by one and the Credit value is reset to the Num value, i.e., if $Credit(x) = 0$, $Excess(x) \leftarrow Excess(x) + 1$, and $Credit(x) \leftarrow Num(x)$. Otherwise, the Excess value stays unchanged

Node x will further assign this time slot to a flow in its FAT using the rules listed as follows.

- (1) Node x cannot schedule any flow packet unless a time slot is assigned to it. Once a time slot is

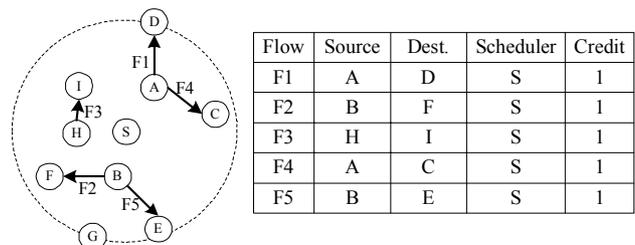
assigned to itself, node x assigns the time slot to the flow with the smallest Excess value in the table, say flow F .

Node x then updates the values of Usage, and Excess of flow F 's entry in its FAT as follows, and leaves the others intact.

- (2) Increment the Usage value of flow F , i.e., $Usage(F) \leftarrow Usage(F) + 1$.
- (3) Update the Excess value of flow F , i.e., $Excess(F) \leftarrow Usage(F)$.

Figs. 1 and 2 explain how the proposed mechanism works. Fig. 1 (a) shows the node topology and Fig. 1 (b), the flow information. This is a single cluster example. All nodes are located within the transmission range of node S, the cluster scheduler. Node A sends a registration request to node S with a Num value of two. Similarly, both nodes B and H send registration requests to S with Num values of two and one, respectively. Scheduler S records the states of nodes A, B, and H in its NAT.

Fig. 2 (a) shows the slot allocation by node S to nodes A, B, and H, and the updates on the NAT in a 12-slot time span. Fig. 2 (b) shows the slot allocation by node A to its flows, again in a 12-slot time span. Note that each entry is stored in the corresponding node's FAT. We can see that node A is scheduled at time slots 1, 2, 6, 7, 11, and 12. Nodes B and H use four and two time slots, respectively.



(a) Node topology (b) Flow information
Figure 1. The flow information for Fig. 2.

So far, we have described how to ensure per-cluster fairness to all flows. In [1,2], the authors focused on global fairness. That is, no matter where the flow is, each can share an equal portion of the bandwidth. To achieve this goal, they set a parameter called “*scheduling window*” to ensure global fairness. The basic idea is as follows. Based on a sliding window, denoted as η , they limit the maximum number of packets each flow can send in a period of

Time \ Node	1			2			3			4			5			6		
	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E
A	1	1	0	2	2	1	2	2	1	2	2	1	2	2	1	1	3	1
B	2	0	0	2	0	0	1	1	0	2	2	1	2	2	1	2	2	1
H	1	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	1

Time \ Node	7			8			9			10			11			12		
	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E
A	2	4	2	2	4	2	2	4	2	2	4	2	1	5	2	2	6	3
B	2	2	1	1	3	1	2	4	2	2	4	2	2	4	2	2	4	2
H	1	1	1	1	1	1	1	1	1	1	2	2	1	2	2	1	2	2

(a) The updates in NAT at scheduler S

Time \ Flow	1			2			3			4			5			6		
	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E
F1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
F4	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Time \ Flow	7			8			9			10			11			12		
	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E
F1	1	2	2	1	2	2	1	2	2	1	2	2	1	3	3	1	3	3
F4	1	2	2	1	2	2	1	2	2	1	2	2	1	2	2	1	3	3

(b) The updates in FAT at nodes A

Figure 2. An example of the two-tier slot allocation mechanism

period of time, i.e., a flow can send up to η packets in a time interval. The proposed mechanism can be combined with the scheduling window mechanism to achieve global fairness as well. Due to space limitations, we will not describe it in detail in this paper.

3. Simulation

This section evaluates the performance of the credit-based scheduling mechanism by simulation. We consider two scenarios: single cluster and multi-clusters.

There are 20 mobile nodes randomly distributed in a 1000m-by-1000m area. We randomly selected nodes, some as flow senders and some as flow receivers. Each packet has fixed packet length and is assumed to occupy one time slot. The simulation time is set to 10000 time slots.

We first randomly generated five best effort flows in a single cluster network, i.e., the transmission range of each node is 1000m. The slot

allocation is shown in Fig. 3. In this scenario, only one scheduler operates so that it can schedule one node to transmit at a time. As a result, all flows share the bandwidth fairly, and each has an equal share of $1/5=0.2$.

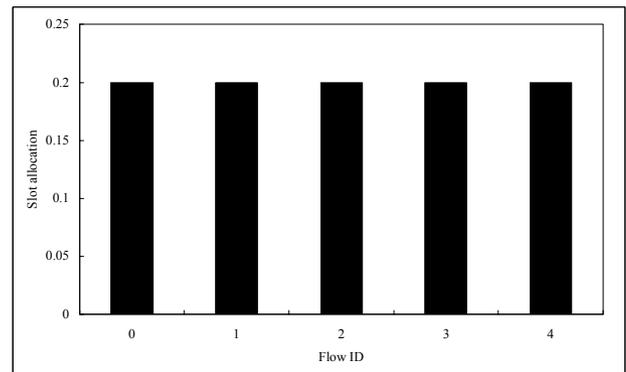


Figure 3. Slot allocation in the single-cluster case

Next, we randomly generated 10 best effort flows in a multi-cluster network. The transmission range of each node at this time is set to 250m. The nodes are logically partitioned into three clusters. Flows 0, 1, 2, 6, and 9 are in cluster 1 and managed by scheduler node 2. Flows 3, 5, 7, and 8 are in cluster 2 and

managed by scheduler node 0. Finally, flow 4 is in cluster 3 and managed by scheduler node 1.

For cluster 1, the expected share fraction of each flow should be $1/5=0.2$. Similarly, the expected share fraction for each flow in clusters 2 and 3 should be $1/4=0.25$, and $1/1=1$, respectively. The slot allocation for this case is shown in Fig. 4. The bars with the same color represent flows in the same cluster. Thus, flows 0, 1, 2, 6, 9 are in one cluster, flows 3, 5, 7, 8 are in another, and flow 4 is in a cluster by itself. As expected, the simulation result shows that flows in each cluster equally share the bandwidth. In this experiment, spatial channel reuse is implemented by way of clustering. Thus, the network throughput are three times more than the single cluster case. Note that here only the per-cluster fairness to each flow is demonstrated. In fact, this mechanism can be easily extended to achieve global fairness, at the expense of lower network throughputs.

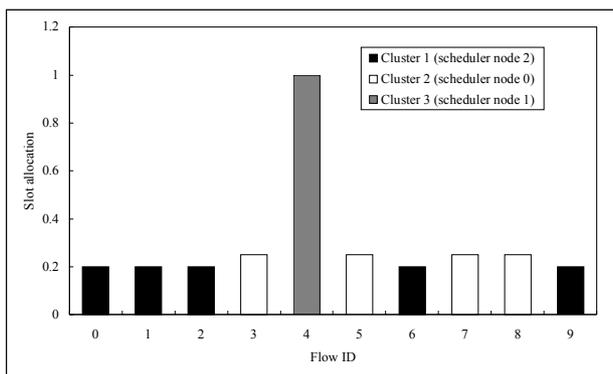


Figure 4. Slot allocation in the multi-cluster case

4. Conclusion

In this paper, we have proposed a credit-based fair scheduling mechanism, which provides fair scheduling for ad hoc wireless networks. This mechanism is based on the concept of “time-slot usage,” instead of “service tag,” to make scheduling decisions. It is a two-level hierarchical scheduling algorithm. The first level is for schedulers to assign time slots to mobile nodes in their respective clusters. The second one is for mobile nodes to assign such time slots to the flows passing through. Besides, spatial channel reuse is implemented by logically partitioning the network into several clusters. We have also performed simulations to verify our

mechanism. The results show that this mechanism improves system throughput and ensures per-cluster fairness. This mechanism can be extended to provide QoS support for guaranteed flows while ensuring fair share of residual bandwidth among both best effort and guaranteed flows. In the future, we will further extend the mechanism to consider host mobility in ad hoc wireless networks.

Acknowledgement

This work was supported by the Communications Software Technology project of the Institute for Information Industry (III) and sponsored by Ministry of Economic Affairs (MOEA), R. O. C.

References

- [1] Haiyun Luo and Songwu Lu, “A topology-independent fair queueing model in ad hoc wireless networks,” *Proc. IEEE ICNP’00*, pp. 325-335.
- [2] Haiyun Luo and Songwu Lu, “A self-coordinating approach to distributed fair queueing in ad hoc wireless networks,” *Proc. IEEE INFOCOM’01*, pp. 1370-1379.
- [3] N. H. Vaidya and P. Bahl, “Fair scheduling in broadcast environments,” *Microsoft Research Tech. Rep. MSR-TR-99-61*.
- [4] E. Hossain and V. K. Bhargava, “A centralized TDMA-based scheme for fair bandwidth allocation in wireless IP networks,” *IEEE J. Select. Areas Commun.*, Vol. 19, No.11, Nov. 2001, pp.2201-2214.
- [5] M. Gerla and J. T-C. Tsai, “Multicluster, mobile, multimedia radio network,” *ACM Wireless Networks*, vol. 1, no. 3, 1995, pp. 23-29.
- [6] H.-C. Lin and Y.-H. Chu, “A clustering technique for large multihop mobile wireless networks,” *IEEE VTC 2000*, pp. 1545-1549.