Understanding HTTP Traffic Performance in TDMA Mesh Networks

Vishal Sevani
Dept. of Computer Science & Engg.
Indian Institute of Technology-Bombay
Email: vsevani@cse.iitb.ac.in

Bhaskaran Raman
Dept. of Computer Science & Engg.
Indian Institute of Technology-Bombay
Email: br@cse.iitb.ac.in

***Abstract*—TDMA based wireless mesh networks have gained prominence as some of the recent standards such as WiMAX, 802.11s have proposed the use of TDMA based MAC protocol for mesh networks. But as of yet there have been no attempts to study the performance of HTTP based web browsing traffic in TDMA mesh networks. HTTP web browsing traffic has different characteristics compared with other types of traffic. In particular, as HTTP traffic consists of large number of small sized file transfers (median file size is 10KB), it can impose high scheduling overhead. As we highlight, HTTP traffic requires that RTT (round trip time) be small and also it requires that large sized flows be allocated higher share of bandwidth. Given these characteristics of HTTP traffic, it is not clear what protocol design for TDMA mesh networks performs best. In this work we have studied the comparison of four different TDMA MAC protocols for HTTP web browsing traffic. Two of these protocols follow distributed scheduling, one centralized and the other naive static fixed schedule approach. Comparison of the different protocols enables us to understand as to how the different scheduling mechanisms used by the different protocols affect the HTTP traffic performance.**

A particularly crucial aspect that our results point out is that the performance of the two distributed protocols specified by the recent WiMAX and 802.11s standard is poor in comparison with the naive static approach for some commonly arising conditions. This implies that there is need for further improvement of these standard protocols. Specifically we observed that the two distributed protocols perform well under high load and single channel operation. But, in comparison with the static approach, their performance is quite poor in presence of wireless packet loss or co-existing large sized HTTP file downloads. Likewise we observed that none of the protocols perform well under all the dimensions that we have considered implying a need to devise a better protocol that can efficiently support HTTP traffic. We believe that our results lay foundation for further efficient protocol design, for TDMA mesh networks.

I. Introduction

It is well known that the 802.11 CSMA/CA based MAC protocol performs poorly for wireless mesh networks [1]. So researchers have considered the use of TDMA MAC protocol for mesh networks [2], [3]. Recent standards such as WiMAX based 802.16 [4] and WiFi based 802.11s [5] also specify the use of TDMA based MAC protocol for mesh networks.

Most of the work that has been done for TDMA based mesh networks is for improving the QoS of applications such as real time voice, video, etc [6]. However mesh networks are also used to provide internet connectivity to a small town, rural regions, etc [7], [8]. Since HTTP traffic constitutes a significant percentage of the internet traffic [9], it is likely to be the dominant traffic in such mesh networks. For improving the user experience for HTTP web browsing traffic, it is essential that the response time for HTTP web page transfer be as small as possible [10], [11], [12]. But, despite the development of recent standards, there have been no studies so far that evaluate in detail the performance of HTTP web browsing traffic in TDMA mesh networks. Typically HTTP traffic is considered as Best Effort traffic and the objective to improve the performance of such traffic is to maximize the throughput. However as our measurements results show that merely maximizing the throughput may not be sufficient to improve the performance. HTTP web browsing traffic imposes few other requirements as well that we describe below.

978-1-4673-5494-3/13/$31.00 © 2013 IEEE

HTTP web browsing traffic is different from other types of traffic such as real time voice, video, etc. HTTP web browsing traffic consists of large number of small sized file transfers. We analyzed the HTTP file distribution outlined in [13] and observed that the median cumulative file size (i.e. sum of file sizes of HTTP requests and replies for a URL), was less than 10 kilobytes while about 88% sessions had cumulative file size less than 50 kilobytes.

Given such large number of small sized files, the rate of incoming HTTP requests can be quite high especially with a large user population. This can impose high scheduling overhead. So it is not clear if dynamic demand based scheduling can improve the performance in comparison with static scheduling. Even with dynamic scheduling, it is essential to understand how the performance of the distributed scheduling compares with that of centralized scheduling.

To minimize the response time of small sized file transfers it is essential to minimize the RTT (round trip time). As we illustrate in Sec. IV if response time of small sized flows is not minimized then number of simultaneous flows in the network increases. As number of simultaneous flows increases the available bandwidth per flow decreases thereby affecting performance. So it is essential to take into account this requirement of minimizing the RTT for improving performance.

Based on our analysis presented in Sec. V, we observe that another requirement for improving the performance of HTTP web browsing traffic is that large sized flows need to be given higher share of the available bandwidth.

	u-distributed	MCCA	Centralized	Static
Performance under increasing load	very good	very good	very poor	poor
Performance under packet loss	poor at moderate and high load	poor at moderate and high load	very poor at moderate load and high load	very good at moderate and high load
Performance under single channel operation	good	good	very poor	poor
Performance with co-existing large sized HTTP file downloads	poor	poor	good	very good
Efficiency in utilizing available throughput	7-41%	5-32%	2-14%	15-32%

TABLE I: Comparison of different protocols: summary of our results

Given these varying requirements of the HTTP web browsing traffic, it is essential to study how the different scheduling mechanisms used by different MAC protocols affect the performance. So, in this regard our primary contribution is that we have studied in detail the behaviour of following four different protocols for HTTP web browsing traffic, (i) Uncoordinated distributed protocol specified by WiMAX [14] (henceforth referred as *u-distributed*), (ii) MCCA protocol [5] described in the 802.11s standard; this is a distributed protocol too (henceforth referred as *MCCA*), (iii) Centralized LiT MAC [2] (henceforth referred as *centralized*), and (iv) Static slot allocation (henceforth referred as *static*).

We have compared the performance of these protocols for different aspects such as performance under increasing load, performance under packet loss, etc. Table I gives comparison of the protocols for different aspects. As we can see from Table I, none of the protocol choices are ideal across all dimensions, and each has shortcomings. Particularly the Table I highlights the key aspect that the two distributed protocols, specified by recent standards, perform poorly under commonly arising conditions such as wireless packet losses and in presence of co-existing large sized HTTP downloads. Even a naive static protocol performs better under these conditions.

Our results thus point toward various open issues. (1) The design of the distributed approaches need enhancements to better handle wireless losses and large HTTP flows. (2) On the other hand, both the centralized and static approaches need improvements under high load and single channel operation, and centralized approach, in addition, needs improvement under wireless losses too. (3) All of the approaches have significant room for improvement in terms of efficiency.

The organization of the rest of the paper is as follows, Sec. II presents related work, while Sec. III describes the different TDMA MAC protocols that we have compared. Sec IV discusses the background for simulation based experiments, while Sec. V presents comparative results for the performance of the different protocols. Finally Sec. VI discusses the improvements that can be made to different protocols.

II. Related Work

Quite a few distributed MAC protocols have been proposed and evaluated for TDMA mesh networks in literature. These protocols can be largely classified into two categories, priority based slot reservation and request based slot reservation.

Priority based slot reservation: These protocols typically employ a pseudo-random hash function to decide their transmission slots in collision free manner. Accordingly in NAMA [15], each node has information about the nodes in its two-hop neighbourhood. Thereby making use of a common hash function, each node can decide its transmission slot in contention free manner. Likewise NAMA has been evaluated for throughput and delay performance.

Overlay MAC [3] implements a weighted slot allocation based mechanism wherein slots are alloted to the nodes based on their weights. Overlay MAC has been evaluated for throughput performance and fairness. TRAMA [16] makes use of pseudo-random hash function with the primary objective of providing energy efficiency. Accordingly TRAMA has been evaluated for queuing delay and energy savings. Coordinated distributed protocol of WiMAX [17] makes use of distributed leader election procedure for deciding the allocation of control slots in contention free manner. Once a node has finished transmission in the control slot, it then transmits again after a certain time interval which depends on the value of the parameter XmtHoldOffExponent. Accordingly there are some studies that have carried out modeling of the coordinated distributed protocol [14], or adaptively adjusted the value of XmtHoldOffExponent for improving performance [18].

Request based slot reservation: In these protocols the nodes explicitly reserve slots by exchanging slot reservation requests. Accordingly FPRP [19] makes use of contention based mechanism for slot reservation that consists of five phases and is free of hidden terminals. The authors have evaluated the efficiency of FPRP in terms of minimizing the number of slots required for all nodes in the network to transmit. In [20] the authors outline the protocol DRAND that also makes use of contention based access for slot reservation and evaluate it for convergence time, message complexity as well as throughput performance. LMAC [21] is primarily designed to minimize energy usage. In LMAC each node is alloted one control slot in the frame wherein the node transmits the control message which consists of the information about the intended receivers and the slots that are occupied by one hop neighbours. Likewise LMAC has been evaluated for energy efficiency.

Likewise few protocols have been proposed for TDMA mesh networks that follow centralized design. Accordingly LiT MAC [2], makes use of slot reservation request packets, transmitted probabilistically, to convey flow information to the root node. On receiving the flow information, root node allocates time slots to the new flow and the new schedule is then conveyed to network via control messages. LiT MAC has been evaluated in detail for real time voice.

WiMAX standard also proposes the use of centralized scheduling [17] for mesh mode. Several algorithms have been proposed for centralized scheduling, primarily to improve the routing and QoS performance [6], [22]. In [23], the authors have implemented the centralized backpressure algorithm for TDMA mesh networks alongwith evaluating the throughput and fairness performance of the same.

Key differences: None of the above mentioned works have carried out performance evaluation specifically for HTTP traffic. Though several of the above protocols have been evaluated for throughput performance but as mentioned in Sec. I, throughput optimization is not the only goal for improving the HTTP traffic performance. HTTP traffic can also impose high scheduling overhead and requires that RTT be minimized. It is not clear as to how the different protocols mentioned above perform under high scheduling overhead or as regards minimizing the RTT. Further none of the above mentioned works have carried out evaluation under wireless losses. This is significant, since, as we shall see, protocol behaviour can critically change under wireless losses; such loss can lead to loss of both the control as well as data packets.

III. TDMA MAC PROTOCOLS

For the distributed protocols, we have compared the performance of the two protocols that are specified by the recent standards, i.e. the uncoordinated distributed protocol specified by 802.16 WiMAX [14] and the MCCA protocol specified by 802.11s mesh standard [5].

For the centralized scheduling we have compared the performance of LiT MAC protocol [2]. There are few other protocols in literature that follow centralized scheduling particularly backpressure algorithm [23], centralized scheduling of WiMAX [17], etc.

Backpressure scheduling algorithm [23] is a throughput optimal scheduling algorithm for centralized TDMA mesh networks. However the backpressure algorithm requires the information about queue length for each of the flows in the network. Since for HTTP traffic the flows are frequently entering and leaving the network the queue information at a node can change very frequently. Conveying this queue information frequently to the central node can incur high overhead. Moreover, the queue information may be outdated by the time it reaches the central node.

As against this, the LiT MAC protocol supports flow based slot reservation. A flow is a TCP connection that is used for transfer of HTTP requests and replies. For flow based slot reservation slots are allocated to a flow only once when the flow initiates. So the overhead of scheduling is small compared with backpressure scheduling. Another aspect of LiT MAC is that it makes use of soft-state mechanism which makes it more resilient to packet loss in comparison with the centralized scheduling of WiMAX as we describe below.

For comparison, we have also studied the performance of a naive static protocol wherein the allocation of the slots to the links is fixed. The static protocol is not adaptive to dynamic traffic conditions, but does not incur any scheduling overhead. We describe the four protocols below.

Uncoordinated distributed (u-distributed): Uncoordinated distributed or the u-distributed protocol follows distributed scheduling and is specified by WiMAX standard for use for occasional traffic [14]. The TDMA frame structure of the u-distributed protocol consists of control and data slots. The slot reservation request is transmitted in the data slots itself. A node requests a neighbour for reserving as many slots as required to transmit the packets in its queue. The slot reservation follows a three way message exchange. First the slot reservation request is transmitted by requester to requestee, then reply is transmitted by requestee to requester followed by confirmation message which is transmitted by requester to requestee. This three way message exchange ensures that all the nodes in the neighbourhood of requester and requestee are informed about the data slots that have been reserved.

WiMAX standard specifies that the data slots in the TDMA frame be divided into three groups, one to be used for centralized scheduling, other for coordinated distributed scheduling and third group for u-distributed scheduling. However since we are interested in evaluating only the performance of the u-distributed protocol, we assume that all the data slots are used for u-distributed.

For multi-channel operation, one of the channels is used as control channel. So if channels 1,6 and 11 are the three channels in use with channel 1 as control channel, a node first selects the free slots on channel 6 and 11 and if none are available then on channel 1.

MCCA: The MCCA (Mesh Coordinated Channel Access) protocol is specified by 802.11s mesh standard to provide collision free access in 802.11 CSMA/CA based mesh networks. In MCCA the nodes reserve the slots in the DTIM (delivery traffic indication message) interval using the 802.11 CSMA/CA contention based access. For reservation, the nodes specify the number of slots and the time period after which the slots will repeat.

We slightly modify the MCCA protocol for use in TDMA based mesh networks. We make use of the frame structure which consists of control, contention and data slots. The control slots are used for time synchronization, while the contention slots are used for reserving the data slots using the contention based access. In the contention slots the nodes, as in u-distributed protocol, make use of three way message exchange for reserving the slots. A node reserves as many data slots as there are packets to be sent in its queue. Also the reserved slots do not repeat i.e. if the node has packets pending in its queue after its allocated slots are over, then it has to reserve slots again.

The main difference between MCCA and u-distributed protocol is that in former the slot reservation request packets are transmitted in the dedicated contention slots, while in the latter the request packets are transmitted in the data slots itself.

Centralized LiT MAC: LiT MAC [2] is a centralized protocol wherein central root node carries out the allocation of the data slots to the different links. The TDMA frame structure of LiT MAC protocol consists of control, contention and data slots. The tree connecting the root node to the rest of the nodes in the network is termed as the control tree. The slot reservation requests are transmitted by a node probabilistically in the contention slots. These slot reservation requests propagate up the control tree towards the root node.

On receiving slot reservation request, the root node allocates slots for a new flow and then propagates the new data schedule down the control tree via control packets. The data schedule is the set of slots allocated to the links, that repeats itself. Thus the nodes in network learn the new data schedule via control packets that they hear from their neighbours.

When the flow terminates the node transmits flow termination request so that the root node can remove the slots allocated to that particular flow from the data schedule.

To handle wireless packet loss, LiT MAC makes use of soft-state mechanism. Once the slots have been allocated for a new flow, the node periodically transmits flow renewal requests which inform the root node that a given flow is active. If the root node does not receive the flow renewal request for a particular flow for certain timeout period, then it assumes that the flow has terminated and removes the slots allocated to that particular flow from the schedule. Thus the use of periodic flow renewal requests ensure that the slots do not remain allocated to a flow indefinitely in the event of loss of flow termination request.

Static slot allocation: In static slot allocation protocol the slot allocation is carried out only once during initialization and thereafter the schedule remains fixed. One slot is allocated to each link along the tree connecting the gateway node to all other nodes in the network. A node stores the packets of different flows in different queues. In its allocated data slot, a nodes transmits the packets from the different queues using round robin scheduling. The TDMA frame consists of control and data slots. The control slots, as for other protocols, are used for periodic time synchronization.

IV. Simulation Methodology

A. Preliminaries

We have carried out simulation based analysis to study the behaviour of different protocols. For this we have implemented a custom simulator. To simulate the TCP flows we use NSC (Network Simulation Cradle) library [24] that makes use of real linux kernel stack TCP implementation.

We are interested in studying the behaviour of the protocols for community/rural mesh networks that provide internet connectivity to a small town or rural regions. Accordingly, we have carried out measurements for five different topologies. Three of the topologies consist of 30 mesh nodes and one gateway node while the other two topologies consist of 45 mesh nodes and one gateway node. The gateway node is connected to the Internet via a wired connection. The HTTP requests originating at the clients pass through the gateway node to the internet. Each topology consisted of a maximum of 5-6 hops from gateway node. We have used two-hop interference model wherein the contending transmitter does not interfere with the signal being received by a node, if it is two hops or farther away from that node.[1]

We observed that some of results were different for different topologies. We describe the results in detail for one of the topologies of 30 mesh nodes and briefly highlight the different results for different topologies in Sec. V-F.

Each node has three clients connected to it. These clients issue HTTP request to a server over the internet. We make use of HTTP/1.1 which reuses the TCP connection for different HTTP requests and replies for the same web server. However we assume that each request for a new URL is for a different web server. We note that internet traffic has evolved over time. But due to lack of availability of detailed statistics from any recent model we have taken the file sizes for HTTP requests and replies from the HTTP model outlined in [13]. However the characteristics of the model outlined in [13] are similar to those observed by the recent studies [25] which note that web traffic is heavy tailed with most HTTP responses being short.

We have also studied the behaviour of the protocols under different traffic loads. For varying the traffic load we keep the number of clients in the network fixed, but vary the think time between the HTTP requests. Varying the think time enables us to vary the number of simultaneous flows in the network and hence the traffic load. Likewise we have carried out measurements for three different loads,

- Low load: Think time is from the HTTP model specified in [13] and can be as high as few hundred seconds.
- Moderate load: Think time is taken from the uniform distribution in the range 0.1 to 50 seconds.
- High load: Think time is taken from the uniform distribution in the range 0.1 to 20 seconds.

We have kept number of channels as three and used a wireless link data rate of 6Mbps for our experiments unless otherwise specified. We make use of link-level ARQ wherein lost packet is retransmitted immediately. We have kept maximum number of link-level retransmissions as 15.

We have used HTTP response time as a metric to compare the different protocols. Response time is the time taken for the download of entire HTML web page, including the embedded objects, after the first HTTP request is issued. For improving the user experience it is essential that the HTTP response time be small [10], [11]. As the HTTP response time increases, more users tend to abandon the web page download. For e.g. [12] lists the example of a website for which the page abandonment rate dropped from 30% to 6-8% when its response time was improved just by 1 sec.

For reporting the results, we carry out an experiment of half hour duration. We repeat the experiment three times by varying the seed of the random number generator and report the average across these three experiments. As the variation across the three experiments was small, we do not report the standard deviation for the measurements.

Three factors, namely the TDMA frame parameters, contention probability scheme, and design of slot allocation algorithm have a significant effect on the performance. So we first carried out measurements to determine these.

B. TDMA frame parameters

We carried out measurements to determine the appropriate values for the various TDMA frame parameters. The different TDMA frame parameters are number of control, contention and data slots in a frame along with the duration for these different types of slots. We discuss our results below.

[1] We do not model RF propagation or other PHY layer details

u-distributed and static: For both u-distributed and static protocol there are no contention slots. So to ensure that nodes are time synchronized periodically we keep number of control slots in TDMA frame as 2 and number of data slots as 30.

MCCA : For MCCA the number of contention slots should be large enough so that the nodes are successfully able to reserve the data slots for a particular frame or else the data slots in the frame will be unutilized leading to inefficiency. So, first keeping the data slots as ten, we varied the number of contention slots. We observed that for number of contention slots more than five, the performance does not improve. So we fix the number of contention slots to five.

We then varied the number of data slots in the TDMA frame from three to ten. We observed that the performance improves as the number of data slots in increased to five, but then degrades. For e.g. we observed that at high load the 90th percentile response time for the case of five data slots is about 2 secs less than that for ten data slots.

Now as the number of data slots is increased the overhead of contention and control slot decreases, so the available bandwidth should increase. But as the data slots is increased the RTT for the packets also increases. This is since in MCCA, a data packet traverses at-most one hop in a TDMA frame. So, in case of number of data slots in the frame as ten, a data packet has to wait for longer time to reach the next hop which in turn increases the RTT.

The effect of increasing the RTT is that the response time of small sized flows decreases. This in turn increases the number of simultaneous flows in the network thereby decreasing the available throughput per flow and hence affecting the performance. Fig. 1 shows the probability distribution of number of simultaneous flows in the network at high load for the two parameter values of five and ten data slots.[2] As can be seen from Fig. 1 the probability of having more number of simultaneous flows in the network is more for the case of ten dataslots in the frame.

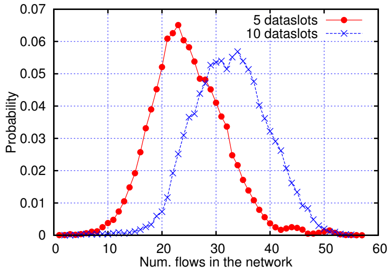


Fig. 1: PDF of number of flows for MCCA protocol

We keep number of control slots in TDMA frame as one, which is sufficient to periodically resynchronize the nodes.

Centralized: For the centralized protocol we empirically observed that the setting of 8 control, 40 contention and 30 data slots gives best performance. Increasing the number of data slots to more than 30 increases the flow set-up time affecting the performance. Decreasing the number of data slots decreases the flow set-up time but increases the overhead of control and contention slots thereby reducing the available bandwidth and hence affecting the performance.

The Table II below summarizes the parameter values for the different protocols along with overhead of control and contention slots. We have chosen the slot duration for different slots accordingly to ensure that these are sufficiently large to transmit the different packets. As can be seen from Table II, the per-frame overhead is maximum for MCCA protocol.

	Num. slots in frame (slot duration in μs)			
	Control	**Contention**	**Data**	**Overhead**
MCCA	1(300)	5(600)	5(1800)	36.7%
u-distributed	2(300)	-	30(1800)	1.08%
Centralized	8(1200)	40(200)	30(1800)	24.5%
Static	2(300)	-	30(1800)	1.08%

TABLE II: TDMA frame parameter values

C. Contention probability scheme

For slot reservation the nodes transmit slot reservation request packets probabilistically in the contention slots. We cannot follow the 802.11 CSMA/CA based scheme for transmitting the request packets, since to perform carrier sensing based backoff the contention slot will have to be further divided into smaller sub-slots and it cannot be ensured that packet transmission will finish before the slot ends. So to identify the appropriate probability scheme, we compared the performance of following two schemes,

Scheme 1: A node transmits a slot reservation packet with probability 1/2. In case of collisions, the node first reduces the probability to 1/3, then to 1/4, etc.

Scheme 2: A node transmits a slot reservation packet with probability 1/2. In case of collisions, the node reduces the probability exponentially i.e. first to 1/4, then to 1/8, etc.

We observed that for the case of MCCA and centralized protocol both the probability schemes performed similarly under all the traffic loads. However for the case of u-distributed protocol, the first scheme performed better especially under high load. We observed that at high load the 90th percentile response time for scheme 1 was about 2 secs less than that for scheme 2.

The difference of u-distributed protocol compared with MCCA protocol is that in the former the slot reservation request packets are transmitted in the data slots itself. The result of this is that slot reservation request packets can collide with the data packets. At high load as the number of data packets in the network is also more, the number of slot reservation request collisions increases. The increase in the number of slot reservation request collisions increases the flow set-up time. The flow set-up time, for the distributed protocols, is the time taken by the node to successfully reserve the data slots after it first receives a data packet for which no slot is reserved, while for the centralized protocol, flow set-up time is the time taken by the slot reservation request to reach the root node after it originates. The result of increase in flow set-up time is that the RTT increases which in turn increases the response time of small sized flows thereby increasing the number of simultaneous flows in the network and hence affecting the performance, as mentioned before.

[2] The number of flows in the network is sampled at the times the TCP data/ack packet reaches the end destination

Thus these results illustrate that for the u-distributed protocol wherein slot reservation request packets can collide with data packets, a more aggressive contention probability backoff scheme performs better.

D. Slot allocation algorithm

For the centralized and the static protocol, the protocol does not specify the slot allocation algorithm. So we carried out measurements to understand what should be the optimization goal for algorithm design for these two protocols.

As mentioned in Sec. I to minimize the response time of small sized flows RTT needs to be minimized while to improve the response time of large sized flows throughput needs to be maximized. To understand how these two different optimization goals affect the performance we studied the performance of following two slot allocation algorithms, (i) algo 1 - follows a greedy strategy to minimize RTT, (ii) algo 2 - follows a greedy strategy to maximize throughput. More details regarding the two algorithms are given in [26].

We observed that for the case of static protocol algo 2 performs better, while for the case of centralized protocol algo 1 performs better. The algo 1 performs better for the static protocol as it minimizes the response time of large sized flows. However the reason for poor performance of algo 2 for the centralized protocols is as follows. Firstly it increases the RTT which in turn increases the response time of small sized flows causing the number of simultaneous flows in the network to increase. Secondly, it does not utilize the data slots efficiently. This is since the centralized protocol employs flow based slot reservation wherein a reserved slot can be used only by a particular flow. Now for algo 2, RTT can be quite large compared with that for algo 1. So, for algo 2, during the early slow start phase of the TCP flow when the congestion window size is still small lot of slots may get wasted, as the flow is waiting for TCP acknowledgement to be received. This non-utilization of data slots reduces the available throughput per flow thereby further affecting performance of algo 2 for the centralized protocol.

Thus our results of this section particularly highlight the need to minimize the RTT to improve the performance. In the next section we carry out detailed performance comparison of the different protocols.

V. Performance Comparison

We carried out measurements to study the behaviour of the protocols for different aspects such as performance under varying load, under packet loss, etc, the results of which are discussed below.

A. Performance under varying load

We carried out measurements to study the behaviour of the different protocols for three different traffic loads. We observed at low load there was not much difference in the performance of the different protocols but the difference was more apparent at moderate and high load. Accordingly, Fig. 2 shows the CDF of response time for the different protocols at moderate and high load. At moderate load, as can be seen from Fig. 2(a), the performance of the other protocols except centralized protocol is similar. The performance of the centralized protocol is slightly poor with the 90th percentile response time being about 2 secs more than the other protocols.

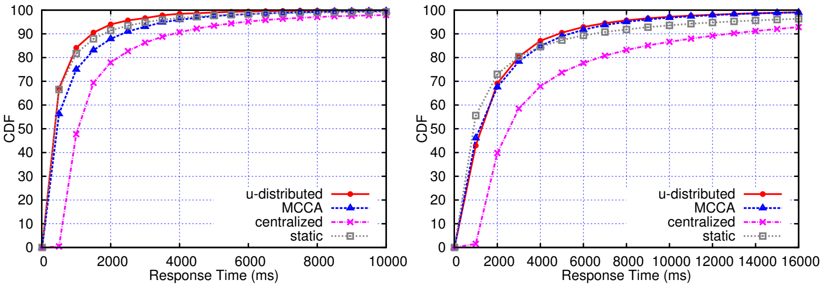
At high load, as can be seen from Fig. 2(b), the performance of u-distributed and MCCA protocol is quite similar and better than other protocols especially for higher percentile values. Though the response time values for the static protocol till about 85th percentile are similar to that for u-distributed and MCCA protocol, the higher percentile response time values are much more. For e.g. 98th percentile response time difference between the static and u-distributed/MCCA protocol is as much as 6 secs. While the performance of the centralized protocol is much worse with the 90th percentile response time being as much as about 8 secs more than that for u-distributed/MCCA and the difference further increases at higher percentile values. Recall that a difference of even 1 sec can affect the interactivity while browsing [12], so a difference of 6-8 secs is quite significant.

The reason for better performance of the u-distributed and MCCA protocol protocol is that for these protocols the throughput among the flows gets divided based on their TCP congestion window size. This is since in these protocols a node reserves slots for all the backlogged packets in its queue. So if there are more number of packets of flow A in the queue than flow B, then more packets for flow A will get transmitted in the frame, implying that flow A gets higher bandwidth. The result of this is that since for large sized flows the TCP congestion window size has stabilized to a higher value, these typically get higher share of the bandwidth which in-turn decreases their response time, hence improving the performance.

As against this for the static protocol the slots are alloted to all the links in the tree connecting the gateway node to the other nodes in the network. So if a particular link does not have any flow passing through it, then slot alloted for that link gets wasted decreasing the available throughput per flow. Under increasing load, the available throughput per flow further decreases. This in turn increases the response time of the large sized flows affecting the performance. Fig. 3 shows the response time vs file size distribution for the MCCA, centralized and static protocol at high load. As can be seen from Fig. 3, for static protocol, quite a few large sized flows of 200KB or more have response time much more than that for MCCA protocol.

There are two reasons for poor performance of centralized protocol. First, it incurs high flow set-up time. The higher flow set-up time increases the number of simultaneous flows in the network. This in turn decreases the available throughput per flow thereby affecting performance. Fig. 4 shows the probability distribution of number of flows in the network for the different protocols at high load.[3] As can be seen the probability distribution is similar for other protocols, except centralized protocol, for which probability of having more

[3] The number of flows in the network is sampled at the instant the TCP data/ack packet reaches end destination



(a) moderate load (b) high load

Fig. 2: CDF of response time under moderate and high load

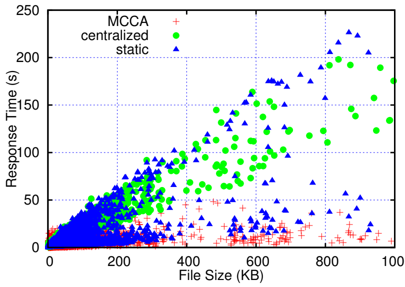


Fig. 3: Response time vs filesize at high load

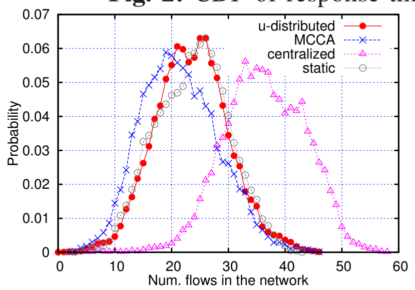
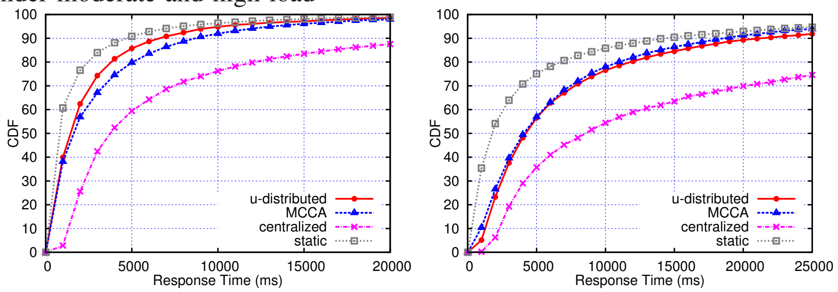


Fig. 4: PDF of number of simultaneous flows at high load



(a) moderate load (b) high load

Fig. 5: CDF of response time under 10% packet loss

number of simultaneous flows in network is also much more.

Another reason for poor performance of the centralized protocol is that it follows flow based reservation i.e. in a data schedule, one slot is reserved for a flow at the link through which the flow passes. The result of this is that the available throughput gets equally divided among the active flows. So as the number of simultaneous flows in the network increases the throughput available for a particular flow decreases. The disadvantage of this is that a large sized flow may get smaller throughput which in-turn increases its response time thereby affecting the performance. As can be seen from Fig. 3 the response time for the centralized protocol increases as the file size increases and for the large sized flows the response time is much more than that for the MCCA protocol.

Thus these results show that the distributed protocols perform better under increasing load, as these protocols allocate higher bandwidth to large sized flows. Given this comparative performance under no wireless losses, how does it change under wireless losses? We study this next.

B. Performance under wireless packet loss

Since packet loss is characteristically present in wireless networks, we carried out measurements to study the effect of the same on the performance.

Accordingly Fig. 5 shows the CDF of response time for the different protocols under 10% per link uniform packet loss at both moderate and high load.[4] As can be seen from Fig. 5 the performance of the static protocol is much better than the other protocols especially at high load. The performance of the centralized protocol is significantly worse with the 90th percentile difference in the response time between the centralized and the static protocol being over 15 secs at moderate load. At high load the performance of the centralized protocol further degrades. The performance of the u-distributed is slightly better than the MCCA protocol, at moderate load, with the 90th percentile response time of u-distributed protocol being about 2 secs less than the MCCA protocol. However, at high load the performance of the u-distributed and MCCA protocol is similar with the 90th percentile difference in the response time, at high load, between the static and the u-distributed/MCCA protocol being about 4 secs. We observed that as we increased the per link packet loss to more than 10%, the performance of the two distributed protocols further degraded in comparison with the static protocol, while the performance of the centralized protocol was even more worse.

We analyze the reasons for the results observed as follows. The effect of packet loss is two fold, firstly it decreases the throughput as the lost packets have to be retransmitted. Secondly it can result in data packet collisions. This is since the data schedule is propagated via the control packets, if the control packet is lost the neighbouring nodes can have different data schedules which can result in data packet collisions further increasing the packet loss.

	u-distributed	MCCA	Centralized	Static
moderate load	13.1%	12%	10.9%	10.0%
high load	13.8%	12.3%	10.8%	10.0%

TABLE III: Packet loss statistics for the different protocols

Table III shows the average per link data packet loss statistics for the different protocols. Note that these statistics are for the data packets lost due to corruption or packet collisions. As we have kept maximum number of retransmission high (i.e. 15), we observed that the number of packets exceeding maximum retransmissions was small for all the protocols.

As can be seen from Table III the data packet loss for the static protocol is about 10% which is same as the wireless packet loss that we have kept for the measurements. This is since in the static protocol, as the data schedule is fixed there is no data packet collisions due to nodes following incorrect

[4] Such high packet loss rates are not uncommon [7]

schedule. As the data packet loss for the static protocol is less than the other protocols, its performance is also better.

The data packet loss for both the u-distributed and MCCA protocol is more than the simulated packet loss of 10%. This, as explained above, is due to data packet collisions resulting from loss of control packets. The data packet loss of about 13-14%, for the u-distributed is about 1-2% more than the packet loss of 12% for the MCCA protocol. The data packet loss is more for u-distributed protocol as for it the data packets can collide with the slot reservation request packets thereby increasing the effective data packet loss.

Even though the data packet loss for u-distributed protocol is more than MCCA protocol, its performance is better at moderate load. This is since the u-distributed protocol is more efficient in utilizing the available throughput than the MCCA protocol as we elaborate later in Sec. V-E.

For the centralized protocol, as can be seen from Table III, the packet loss is about 11% which is only about 1% more than the simulated packet loss of 10%. The packet loss is less for the centralized protocol, since for it the data schedule is broadcast via control packets. The same schedule is broadcast by all the nodes in the network. Since a node hears the data schedule from all its neighbours, the instances of a node following incorrect schedule are going to be less [2].

Even though the packet loss for the centralized protocol is less its performance is significantly worse. This is for the same reason as explained above in Sec. V-A for poor performance of the centralized protocol under high load. As centralized protocol follows flow based reservation it cannot allocate higher share of bandwidth to large sized flows. Also as the flow set-up time is more for the centralized protocol, the number of simultaneous flows in the network also increases which decreases the available bandwidth per flow. In addition, in case of packet loss, the lost packet is transmitted only in the next data schedule frame. This causes the RTT to increase which in turn increases the response time of small sized flows. As explained before, as the response time of small size flows increases the number of simultaneous flows in the network also increases which further reduces the available bandwidth per flow. Thus the aggregate effect of these three factors results in the centralized protocol performing much more poorly.

Another thing to note is that even though the effective packet loss for the static protocol is about 10%, the degradation in its performance is more than 10%. For e.g. as can be seen from Fig. 2(a), the 90th percentile response time for the static protocol, at moderate load and under no packet loss, is about 2 sec while the response time under packet loss is about 4 sec (Fig. 5(a)) which is about 100% degradation in performance.

The reason for this higher degradation in the performance is that due to packet loss the throughput decreases due to packet retransmissions. As the throughput decreases the response time of the flows increases which in-turn increases the number of simultaneous flows in the network. The increase in the number of simultaneous flows further decreases the available throughput per flow which particularly increases the response time of the large sized flows thereby affecting the performance.

Thus our results show that the static protocol is most resilient to wireless packet loss as it does not involve data packet collisions due to loss of control packets.

C. *Performance under single channel operation*

Till now we have carried out measurements for multi-channel operation making use of 3 non overlapping channels. But for WiFi commodity hardware the channel switching latency can be about 4-5ms [27]. Given that for TDMA the slot duration can be of order of 1-2ms, such high channel switching latency can incur high overhead for multichannel operation. So it is also essential to study the performance of the different protocols under single channel operation. So we carried out the measurements for the same.

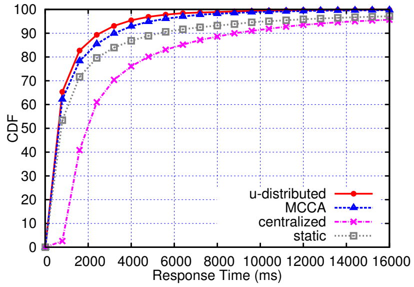


Fig. 6: CDF of response time with single channel operation

Fig. 6 shows the CDF of response time for the different protocols under single channel operation at moderate load. As can be seen from Fig. 6, the performance of the u-distributed protocol is better than the other protocols. The performance of the MCCA protocol is similar to the u-distributed protocol except for small difference. The performance of the static protocol is much worse than the u-distributed protocol with the 95th percentile difference in the response time being about 6 secs. The performance of the centralized protocol is even more worse than the static protocol. We observed similar results at high load also.

Decreasing the number of channels from 3 to 1 decreases the available throughput. So in that case the performance of the u-distributed and the MCCA protocol is better under single channel operation, since these two protocols are efficiently able to utilize the available throughput by allocating higher share of bandwidth to the large sized flows.

The performance of the static protocol is much worse than the u-distributed protocol. The reason, as explained earlier, is due to inefficiency of the static protocol wherein slots alloted for links that do not have flows to carry are unutilized. This in effect reduces the throughput available for a particular flow thereby affecting the performance. The effect is much more pronounced under single channel operation wherein the available throughput is also less than that for multi-channel operation. Likewise the performance of the centralized protocol is quite poor for the same reason explained above in Sec. V-A i.e. higher flow set-up time and inability to allocate higher throughput to large sized flows.

Thus our results show that protocols employing distributed scheduling perform much better under single channel operation wherein available throughput is much less compared with that for multi-channel operation. Thus the benefit of employing dynamic scheduling vis-a-vis static scheduling for TDMA is particularly apparent in case of single channel operation.

D. Performance with co-existing large HTTP file downloads

We have carried out measurements for the web browsing traffic wherein flow sizes are much smaller. How do the protocols behave in presence of co-existing large sized HTTP file downloads? We carried out measurements to study the same. We made three clients initiate large sized HTTP file downloads that lasted throughout the duration of the simulation, and measured the response time. Accordingly Fig. 7 shows the CDF of the response time for the different protocols in presence of three large sized HTTP file downloads at moderate load.

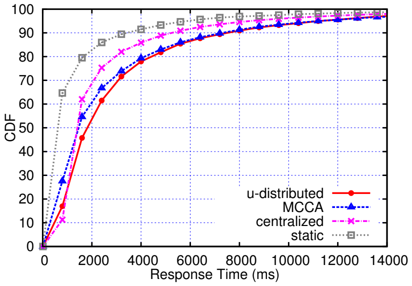


Fig. 7: CDF of response time with co-existing large HTTP file downloads

As can be seen from Fig. 7, the performance of the static protocol is better than the other protocols. Different from the results of Fig. 2(a) the performance of the centralized protocol is better than both the distributed protocols with 90th percentile response time being about 2 secs less than the two distributed protocols. The the performance of the u-distributed and the MCCA protocol is similar and much worse than the static protocol with the 95th percentile response time being about 4 secs more than that for static protocol. We observed that the performance of the u-distributed and MCCA protocol degrades further at high load.

The reason for poor performance of the u-distributed and the MCCA protocol is same as the reason for their better performance under high load as explained earlier in Sec. V-A. Both the u-distributed and the MCCA protocol allocate throughput to the flows based on their TCP congestion window size. Since for longer duration flows the TCP congestion window size would have stabilized to a large value they will end up taking higher share of the bandwidth thereby minimizing the throughput available to web browsing traffic and hence affecting its performance. As since both the centralized and the static protocol allocate equal throughput to the all the flows, their performance does not degrade as much.

E. Efficiency in utilizing available throughput

We also carried out measurements to study as to how efficiently the different protocols can utilize the available throughput. For this, in each data slot, we measured the maximum number of active (i.e. links that have data packet to send) links that can be scheduled in that slot and the number of links that are actually scheduled. We define efficiency as, $Efficiency = \frac{Num.\ slots\ in\ which\ max\ possible\ active\ links\ are\ scheduled}{Num.\ slots\ in\ which\ atleast\ one\ active\ link\ is\ scheduled}$

Accordingly Table IV lists the efficiency percentage for the different protocols at moderate and high load.

	u-distributed	MCCA	Centralized	Static
moderate load	40.6%	32.1%	13.6%	31.6%
high load	7.4%	4.6%	2.0%	15.4%

TABLE IV: Efficiency of the different protocols

As can be seen from Table IV efficiency of all the protocols is quite less, especially at high load. At high load, the efficiency is less than 10% for all protocols, except for static protocol for which the efficiency is slightly more about 15%.

The poor efficiency of the u-distributed and MCCA protocol is due to per hop slot reservation i.e. at each node data packets have to wait for slots to be reserved. This waiting period leads to inefficiency. Since in MCCA data slots are reserved in contention slots, the data packets received in a particular frame have to wait till next frame for reserving the data slots. As against this in the u-distributed protocol since slot reservation requests are transmitted in data slots itself, the slots for the data packets received can be reserved as soon as the transmission opportunity is available. So the efficiency of the u-distributed protocol is slightly higher than the MCCA protocol.

For the centralized protocol the packets for a flow cannot be transmitted till its slot reservation request reaches the root node and new schedule is propagated through the network. As this flow set-up time is higher than other protocols, the efficiency for the centralized protocol is also less.

The poor efficiency of the static protocol is due to static slot allocation. For e.g. consider links 1 and 2 are scheduled in first slot and links 3 and 4 in second slot. Lets say at a particular time only links 1 and 3 are active and both are non-interfering links. The static allocation in this case leads to inefficiency as links 1 and 3 could have been allocated the same slot leading to 50% throughput improvement.

F. Effect of topology

The different measurement results that we have given above are for one of the topology consisting of 30 mesh nodes, of the five topologies that we have used for our simulations. We observed that measurement results for other topologies were largely similar except a few.

For one of the topologies of 30 mesh nodes we observed that, in contrast to the results of Sec. V-B, the performance of the static protocol, in presence of wireless packet loss at high load, was worse than both the distributed protocols. For this topology we observed that the number of flows sharing a common link was more than the other topologies. As for this topology more number of flows share a link, the available bandwidth per flow decreases which in turn affects the performance of the static protocol more as explained in Sec. V-A.

Likewise for another topology consisting of 30 mesh nodes we observed that the performance of both the distributed protocols was slightly better and similar to the static protocol in presence of wireless packet loss at moderate load. For this topology we observed that the size of two hop neighbourhood of a node is smaller than the other topologies. As the size of two hop neighbourhood of a node is small, the number of data packet collision due to loss of control packets is also less and so the performance of the two protocols does not degrade as much at moderate load.

Thus these results illustrate that topology can also have an effect on the performance of the protocols. Specifically it highlights that the performance of the static protocol can further degrade for the topologies wherein large number of flows share a common link. Likewise performance of the distributed protocols can be better, under packet loss, for the topologies for which the size of two-hop neighbourhood of a node is small.

In the next section we comment on how the performance of different protocols can be improved.

VI. Discussion

The reason for poor performance of u-distributed and the MCCA protocol in presence of wireless packet loss, especially at moderate load, is data packet collisions resulting from nodes having incorrect schedule. The instances of nodes having incorrect schedule can be minimized by preventing loss of control packets by making use of FEC or by making the nodes broadcast the data schedule periodically. Also both the distributed protocols perform poorly in presence of co-existing large sized HTTP file downloads. For improving the performance in presence of large sized HTTP file downloads, such flows can be rate limited.

The reason for poor performance of the centralized protocol under high load, as mentioned in Sec. V-A is the higher flow set-up time and the use of flow based slot reservation scheduling mechanism. An alternative to flow based slot reservation, would be to use a scheduling scheme that allocates data slots to the nodes based on their queue size. This will ensure that the flows that have larger TCP congestion window size get higher share of the bandwidth. However, given that the queue information at the nodes changes quite frequently, conveying this information frequently to the root node can incur considerable overhead. So given the requirement of allocating higher share of bandwidth to the large sized flows, what should be the scheduling policy that incurs small flow set-up time is a interesting question that needs to be investigated.

The static protocol performs poorly under high load and single channel operation. The performance under both high load and single channel operation can be improved if large sized flows are given higher share of the bandwidth. For this instead of using round robin scheduling policy that we have used, some other priority based scheduling policy that allocates higher priority to packets of the large sized flows can be used.

VII. Conclusion

We have carried out detailed performance comparison of four different TDMA MAC protocols for HTTP web browsing traffic. We observed that none of the protocols perform well under all the dimensions that we have considered for our evaluation; not even u-distributed and MCCA protocol that are specified by recent standards. Infact the performance of these two standard protocols is poor even in comparison with naive static approach under commonly arising condition of wireless packet loss and presence of large sized HTTP file downloads. This implies there is need for improvement of the different protocols, as outlined above.

References

[1] S. Xu and T. Saadawi, “Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks,” *Computer Networks*, 2002.

[2] V. Gabale, B. Raman, K. Chebrolu, and P. Kulkarni, “Lit mac: Addressing the challenges of effective voice communication in a low cost, low power wireless mesh network,” in *ACM DEV*, 2010.

[3] A. Rao and I. Stoica, “An overlay mac layer for 802.11 networks,” in *Mobisys*, 2005.

[4] http://www.ieee802.org/16/tge/.

[5] M. Islam, M. Alam, C. Hong, S. Lee *et al.*, “emcca: An enhanced mesh coordinated channel access mechanism for ieee 802.11 s wireless mesh networks,” *Communications and Networks*, vol. 13, no. 6, 2011.

[6] H. Shetiya and V. Sharma, “Algorithms for routing and centralized scheduling to provide qos in ieee 802.16 mesh networks,” in *Wireless multimedia networking and performance modeling*, 2005.

[7] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level measurements from an 802.11 b mesh network,” *ACM SIGCOMM*, 2004.

[8] A. Dhekne, N. Uchat, and B. Raman, “Implementation and evaluation of a tdma mac for wifi-based rural mesh networks,” *NSDR09*, 2009.

[9] S. Gebertl, R. Priesl, D. Schlosserl, and K. Heckg, “Internet access traffic measurement and analysis,” in *Traffic Monitoring and Analysis*, 2012.

[10] F. Nah, “A study on tolerable waiting time: how long are web users willing to wait?” *Behaviour & Information Technology*, 2004.

[11] A. Sears, J. Jacko, and M. Borella, “Internet delay effects: how users perceive quality, organization, and ease of use of information,” in *ACM CHI*, 1997.

[12] “The economic impacts of unacceptable web site download speeds.” white paper, Zona Research, 1999. [Online]. Available: http://www.zonaresearch.com/deliverables/white_papers/wp17/index.htm

[13] B. Mah, “An empirical model of http network traffic,” in *INFOCOM’97*.

[14] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu, “Modelling and performance analysis of the distributed scheduler in ieee 802.16 mesh mode,” in *ACM Mobihoc*, 2005.

[15] L. Bao and J. Garcia-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” in *ACM Mobicom*, 2001.

[16] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” *Wireless Networks*, vol. 12, no. 1, 2006.

[17] D. Ghosh, A. Gupta, and P. Mohapatra, “Scheduling in multihop wimax networks,” *ACM MC2R*, vol. 12, no. 2, 2008.

[18] B. Kim, D. Kwak, H. Song, H. Lee, and J. Ma, “An adaptive holdoff algorithm based on node state for ieee 802.16 mesh mode with coordinated distributed scheduling,” in *IEEE PIMRC*, 2008.

[19] C. Zhu and M. Corson, “A five-phase reservation protocol (fprp) for mobile ad hoc networks,” in *IEEE INFOCOM*, 1998.

[20] I. Rhee, A. Warrier, J. Min, and L. Xu, “Drand: distributed randomized tdma scheduling for wireless ad-hoc networks,” in *ACM Mobihoc*, 2006.

[21] L. van Hoesel and P. Havinga, “A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches,” in *IPSN*, 2004.

[22] Y. Zhang, J. Zheng, and W. Li, “A simple and effective qos differentiation scheme in ieee 802.16 wimax mesh networking,” in *WCNC*, 2007.

[23] R. Laufer, T. Salonidis, H. Lundgren, and P. Le Guyadec, “Xpress: a cross-layer backpressure architecture for wireless multi-hop networks,” in *ACM Mobicom*, 2011.

[24] S. Jansen and A. McGregor, “Simulation with real world network stacks,” in *IEEE Simulation Conference*, 2005.

[25] T. Callahan, M. Allman, and V. Paxson, “A longitudinal view of http traffic,” in *PAM*, 2010.

[26] http://www.cse.iitb.ac.in/∼vsevani/phd/tech_report.pdf.

[27] A. Sharma and E. Belding, “Freemac: framework for multi-channel mac development on 802.11 hardware,” in *PRESTO*, 2008.