

# ADTH: Bounded Nodal Delay for Better Performance in Wireless Ad-hoc Networks

Lim, L. B., Spendlove, D., Guan, L. & Wang, X.

Author post-print (accepted) deposited by Coventry University's Repository

## Original citation & hyperlink:

Lim, LB, Spendlove, D, Guan, L & Wang, X 2019, 'ADTH: Bounded Nodal Delay for Better Performance in Wireless Ad-hoc Networks' *AD HOC Networks Journal*, vol. 83, pp. 25-40.

<https://dx.doi.org/10.1016/j.adhoc.2018.08.024>

DOI 10.1016/j.adhoc.2018.08.024

ISSN 1570-8705

Publisher: Elsevier

**NOTICE: this is the author's version of a work that was accepted for publication in *AD HOC Networks Journal*. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *AD HOC Networks Journal*, Vol. 83, DOI: 10.1016/j.adhoc.2018.08.024**

© 2018, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

# ADTH: Bounded Nodal Delay for Better Performance in Wireless Ad-hoc Networks

L.B. Lim<sup>a</sup>, D.J.G. Spendlove<sup>a</sup>, L. Guan<sup>a</sup>, X.G. Wang<sup>b</sup>

<sup>a</sup>Loughborough University, Epinal Way, Loughborough, United Kingdom

<sup>b</sup>Coventry University, Priory St, Coventry CV1 5FB, UK, United Kingdom

---

## Abstract

Delay is an unavoidable factor that occurs within networks and may be exacerbated by the nature of wireless ad-hoc networks. Maintaining a manageable level of delay may be required to provide satisfactory performance for each of the nodes that form the network. The variability of IoT devices, topologies and network conditions demand that a standalone and scalable scheme be used. ADTH is first shown to accomplish this through simulations with the NS-2 network simulator. The scheme was then used with testbed implementation with Gumstix devices and real-time traffic provided by an STC Traffic Generator. These demonstrated its effectiveness in managing flows of delay sensitive traffic, in addition to delivering superior bandwidth utilisation than standard policies.

*Keywords:* Adaptive Queue Management, Dynamic Queue Threshold, Delay Sensitive Traffic, Queuing Delay, Nodal Delay

---

## 1. Introduction

This paper presents an adaptive queue management scheme that can bound the nodal delay of wireless nodes to a required level, to provide better Quality of Service (QoS) for Internet of Things (IoT) devices and sensors. Nodal delay  
5 arises from numerous sources throughout a network, though the delay from certain sources will have a greater impact than others. For example, Media Access Control (MAC) layer contention and interference may cause QoS deterioration, which leads to a large variation in queuing delay and MAC layer delay. Both

need to be dealt with to guarantee a deterministic per-hop delay, whilst control-  
10 ling the per-hop delay would allow for the end-to-end delay for delay-sensitive  
traffic to be bound.

Thus, an adaptive queue management system that can constrain nodal delay  
under dynamic conditions is needed. The Adaptive Dynamic Threshold Queue  
Management (ADTH) scheme fulfils this need, by allowing the queuing thresh-  
15 old of wireless nodes to autonomously adapt to achieve a deterministic per-hop  
delay. ADTH achieves and maintains this delay, while being a generic and stan-  
dalone solution that allows it to easily be adopted by different hardware. The  
implications of the scheme on network delay, User Datagram Protocol (UDP)  
good-put, loss, and data yield were assessed through simulations with the NS-2  
20 software, then further validated through a testbed of Gumstix devices. The de-  
sign of the scheme will first be described, before the results of these simulation  
scenarios and testbed experiments will be discussed.

## 2. Related Work

This paper expands upon work previously undertaken to produce the Dy-  
25 namic THreshold (DTH) scheme [1]. Data analysis conducted during this study  
demonstrated that limiting queue length, or queuing delay, to a constant value  
was not sufficient to constrain network delay, particularly for delay-sensitive  
traffic in wireless ad-hoc networks. DTH also took an analytical approach that  
30 was poorly suited to wireless ad-hoc networks, due to the variability of interfer-  
ence and link quality that is inherent to such networks. A reliance on the results  
of priori queue analysis for queuing threshold estimations meant that DTH was  
unable to autonomously respond to changes in network dynamics. This issue,  
as well as a lack of consideration for MAC layer delay, was the primary impetus  
for the development of the adaptive, on-line approach of ADTH.

35 There are several examples of queue management schemes that have previ-  
ously been proposed for wireless ad-hoc networks [2, 3, 4, 5, 6, 7, 8, 9, 10, 11].  
However, most of these schemes focus on congestion issues with delay-tolerant

traffic, rather than delay-sensitive traffic. These schemes also aimed to alleviate traffic by probabilistically dropping packets, such that traffic sources could  
40 respond to packet loss events by adapting their sending rate. As delay-sensitive traffic is typically carried over UDP, which is incapable of responding to packet loss events, any scheme reliant on packet loss events would not be appropriate for use.

Other examples of similar work on improving performance for ad hoc net-  
45 works and audio transmission include the wireless traffic smoother (WTS) proposed in [12] and the Audio and Video Bridging (AVB) [13] technology. The WTS operates in a similar manner to the final ADTH implementation, in that it monitors queues and dynamically adjusts parameters to improve network performance. However, WTS distinguishes between real-time and non-real-time  
50 traffic to manage TCP packet streams, whereas ADTH is targeted for use with UDP streams of real-time traffic. AVB also aims to achieve QoS through the management of data streams, though lacks the simple design and implementation approach of ADTH.

### **3. Adaptive Dynamic Threshold Queue Management (ADTH)**

#### *3.1. Overview*

  
55

ADTH aims to control the maximum nodal delay of a wireless node to a required value and maintain this delay value to a tolerable amount. This is accomplished by dynamically adjusting the target queuing threshold via a feedback loop. This feedback loop results in a moveable queue threshold for the  
60 system queue, which maintains nodal delay around the specified values when the network becomes congested. Several system performance metrics are periodically measured to estimate the queuing threshold for the next sampling period.

There were several important findings and design considerations that influenced the design of ADTH. First, that the shared bandwidth available to each  
65 wireless node is unevenly distributed. Wireless nodes may have the same queue

length, or backlogs in queues, yet variation in queuing delay may be observed due to different system throughput in terms of how quickly packets are dequeued and transmitted by each node. Therefore, queue throughput became a key parameter for ADTH. Also, a higher network load causes network congestion, which leads to a higher packet loss rate and larger one-way delay (OWD). However, it is difficult to predict traffic load in a wireless ad-hoc network, so traffic load need not be considered in the ADTH design. Packet size may impact the magnitude of queuing delay and smaller packet sizes incur more overhead in channel access contention, hence packet size was used as a parameter.

Furthermore, maximum queue size was found to be a significant factor to be considered, as the magnitude of queuing delay is much larger than the magnitude of propagation delay, transmission delay and back-off delay. Yet, variations in queuing delay may also be caused by traffic load, system throughput and packet size. Determining an optimum queue size that can fulfil the delay requirement is critical for a highly dynamic network, which further motivates the need for an adaptive queue management approach. MAC layer delay is mainly caused by random back-off and retransmission in the network, thus should be considered a stochastic delay component. Due to the difficulty in predicting and controlling MAC layer delay, the delay should instead be offset by regulating the maximum queuing delay at node level.

## *3.2. Design*

### *3.2.1. Basic Controller Functionality*

The ADTH queue management scheme reacts to network changes by tuning the target queue threshold to bound nodal delay, by periodically assessing obtained measurements and statistics. This assessment depends on numerous variables that provide data concerning certain aspects of the network. The period between each assessment is referred to as the sampling time, which can be fine-tuned to balance the trade-off required between the accuracy of the controller and the system overhead. An estimate of the target queue threshold is made at each sampling interval. The maximum allowable queuing delay is

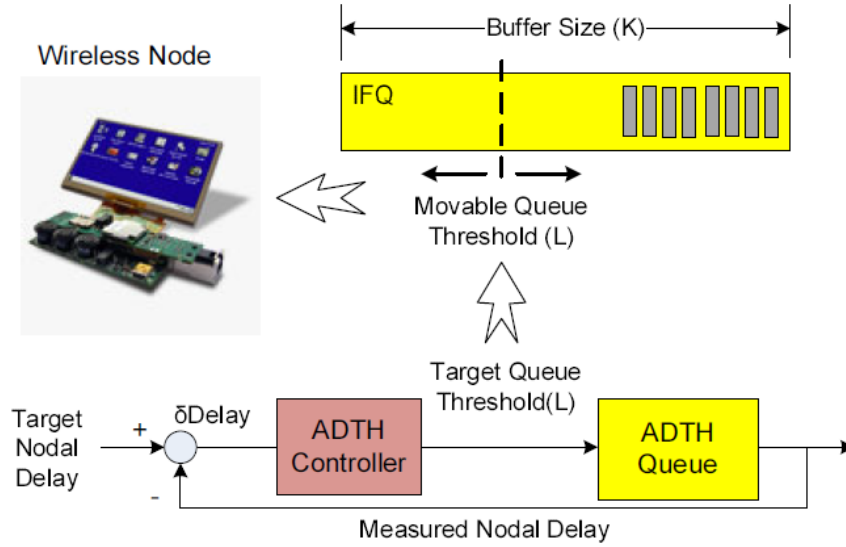


Figure 1: ADTH System Diagram

estimated by subtracting measured mean delay from target nodal delay.

An estimation of the maximum allowable queue size for the next sampling interval is made based on the queue threshold at the current sampling interval. Queue throughput is calculated from the number of bytes sent over the current  
 100 sampling interval. Mean packet size is the average obtained from the number of bytes sent and the number of packets sent, such that the estimation for the maximum allowable queue threshold can be expressed in number of packets. An overview of how the scheme operates is shown in Figure 1, while pseudo code  
 105 of the procedure is included as Algorithm 1.

### 3.2.2. Queue Length Factor

To increase the robustness of the scheme, a queue length factor is used to adapt the maximum allowable queue threshold. This is based on a feedback control loop and leads to a final target queue threshold, which absorbs the  
 110 impact of network changes that cause QoS deterioration. At each sampling interval, an error value is obtained by comparing the measured nodal delay against the target nodal delay. The normalised error is then used to update the

---

**Algorithm 1** ADTH Controller Pseudocode

---

```
if packets received  $\neq 0$  then
    meanPktSize := numSentBytes  $\div$  numSentPkts;
    queThrpt := numSentBytes  $\div$  samplingPeriod;
    estQueDelay := targetNodalDelay - meanMACDelay;
    estQueSize := estQueDelay  $\times$  queThrpt;
    estQueLength := estQueSize  $\div$  meanPktSize;
    error := ndlDlyTarget - ndlDlyMeasured;
    beta := error  $\div$  ndlDlyTarget;
    if ndlDlyTarget  $\leq$  ndlDlyMeasured then
        queLenFctr += beta;
        queLenFctr := MIN(minQueLenFctr, queLenFctr);
    else
        queLenFctr += beta  $\div$  2;
        queLenFctr := MAX(queLenFctr, maxQueLenFctr);
    end if
    queLenTrgt := queLenFctr  $\times$  queLenEst;
    if queLenTrgt < minQueLimit then
        queLenTrgt := minQueLimit;
    end if
    if queLenTrgt > maxQueLimit then
        queLenTrgt := maxQueLimit;
    end if
else
    queLenTrgt := maxQueLimit;
end if
if queLenTrgt < queLenCurrent then
    purge queue
    update enqueueIndex or dequeIndex
end if
Reset statistics and performance metrics
```

---

queue length factor. If measured nodal delay is larger than target nodal delay, then the queue length factor is reduced by the normalised error, which results  
115 in a smaller queue length frequency and hence a lower queue target threshold.

Alternatively, if the measured nodal delay is smaller than the target nodal delay, then the queue length factor is gradually increased by the magnitude of half the normalised error, such that network changes may be anticipated in a preventative manner. The target queue threshold is therefore not solely based  
120 on the performance metrics at the current sampling rate, but also from the previous sampling interval.

The target queue threshold is estimated at each interval and compared to the current queue length. Should the target queue threshold be smaller than the current queue length, then all packets beyond the target queue threshold are  
125 discarded. Packets are discarded from either the tail of the queue or the head of the queue, depending on whether a drop-tail or drop-front policy is being deployed, respectively. This early discarding of packets can alleviate congestion indirectly, while also reducing the wastage of bandwidth on the transmission of packets that will eventually be discarded at their destination.

### 130 3.2.3. *Queuing Delay and MAC Delay*

In the context of the ADTH scheme, queuing delay refers to the amount of time that elapses while a packet is held inside of a queue. A circular buffer is used to record the entry time of each packet to the queue, while producer and consumer indexes are used to track packets entering or leaving the queue,  
135 respectively. These indexes wrap over when the top of the ring buffer has been reached and are used to obtain per packet queuing delay.

The interval between two consecutive packets being de-queued from a queue is referred to as the MAC delay. For the purposes of ADTH, MAC delay is the aggregation of MAC contention delay, transmission delay, processing delay and  
140 propagation delay. Rather than using a single delay value, a mean MAC delay is calculated for the ADTH target queue threshold estimation. The previous queue state must be tracked, otherwise the queue idle time will be included in



the MAC delay and an incorrect value will be measured.

#### 3.2.4. Feedback Control Loop

145 The ADTH controller features a feedback control loop that aims to adjust the queue length factor, based on the normalised error measured from the control variable. The starting point for the target queue threshold estimation is the initial queue length factor, which begins at 1 to ensure that the first target threshold estimation is solely based on the current system performance.

150 The queue length factor is updated after the initial sampling interval, based on the normalised error measured at each sampling interval. The queue length factor is bounded by a lower bound to prevent the factor value from becoming negative. An upper bound is used to prevent the factor value from being increased too aggressively, due to a false positive normalised error. Such an error  
155 may occur in the presence of low traffic load, or a complete lack of congestion.

#### 3.2.5. Sampling Interval

A smaller sampling interval will cause the controller to run at a higher frequency, thus incurring a higher overhead in terms of computational power and execution time. Although the overhead of ADTH is low, with a run-time complexity of  $O(1)$ , the sampling frequency cannot be set at too high a value. How-  
160 ever, if the sampling frequency is too low, then the accuracy of the controller will be comprised, as the controller will be slow to respond to network changes.

If the magnitude of target nodal delay is small, then the sampling rate needs to be high. This is because a fast response to the output error is needed,  
165 otherwise the system will be unable to cope with the changes and nodal delay may exceed its required target value. The ADTH controller will be able to react to network changes with larger target nodal delay, which may then allow for the sampling frequency to be lowered.

Preliminary simulations were performed to assess different sampling intervals  
170 for target nodal delay. Nodal delay could be maintained at a much lower level than the target when the sampling time was substantially less than the target

nodal delay. Nodal delay regulated around the target when the sampling time was half the target nodal delay, however nodal time would exceed the target when the sampling time was greater than half the target nodal delay.

### 175 3.2.6. Dropping Policy

The First In First Out Drop Tail (FIFO-DT) and First In First Out Drop Front (FIFO-DF) queue disciplines can both be coupled with the ADTH scheme. With ADTH enabled, each discipline will have a moveable queue threshold, based on the current system performance and network dynamics, instead of a  
180 constant maximum queue length. The chosen dropping policy will have a direct impact on the maximum nodal delay experienced by a packet.

Simulation results have shown drop front policy to have lower nodal delay than drop tail policy, although both FIFO-DT and FIDO-DF have a similar packet loss rate and UDP delivery rate, and that nodal delay can be bounded  
185 by applying the ADTH queue management scheme to both FIFO policies. These results also suggested that drop front policy should be used for delay-sensitive traffic. The combination of FIFO-DT with either ADTH-DT or ADTH-DF cannot efficiently control nodal delay and one-way delay within the required range.

The deadline miss ratio for FIFO-DT with ADTH-DT was higher than the  
190 coupling of FIFO-DT with ADTH-DF. This was because if the previous packet has exceeded the target nodal delay, then there is a high probability that the following packets in the queue will also exceed the target nodal delay. All packets received at the destination met the deadline when FIFO-DF was coupled with  
195 the ADTH scheme, however FIFO-DF with ADTH-DF would be recommended, as this combination gave better results in terms of bounded nodal delay and bounded one-way delay.

### 3.3. Limitations and Assumptions

In addition to the capabilities of ADTH and the advantages that the scheme  
200 should provide, it was known that there would be limitations of the design and

certain assumptions would need to be made. Target nodal delay requirements could not be too stringent, otherwise the MAC layer delay may exceed the target value when there is heavy congestion in the network. ADTH is incapable of applying a constraint to nodal delay if the MAC layer delay exceeds the delay requirement. There is also no upper bound for the target nodal delay requirement, although the maximum limit for queue size acts as an upper bound for the estimated target queue threshold. ADTH constrains nodal delay through nodal delay requirements from users, or network administrators, with higher packet loss as a trade-off. Should there be no network congestion, then ADTH will behave similarly to standard Drop Tail or Drop Front schemes.

ADTH bounds nodal delay to enable bounded end-to-end delay for delay-sensitive traffic flows, but only if routing paths satisfying the end-to-end delay requirement exist between the source and destination nodes. Routing agents should be responsible for discovering paths that can meet the end-to-end delay requirements for real-time traffic flows. These agents must also handle any changes made to these paths that are caused by the mobility of nodes. Therefore, only nodes capable of running ADTH should be considered for route selection, so that a deterministic end-to-end delay can be achieved. Processing delay and queuing delay from above the network layer are not considered by the ADTH controller when estimating the target queue threshold, as ADTH considers them to be application layer delay.

ADTH was not designed to maintain per flow QoS and so fairness amongst flows was not considered in the design. This increases the scalability of ADTH, as no effort is needed to track per flow performance. It is also not the aim of ADTH to alleviate congestion in a network, rather it is the responsibility of applications to adapt their sending rates. ADTH can constrain nodal delay regardless of packet type, as no effort is spent classifying, or differentiating, packet types. However, it is impractical to constrain nodal delay, or end-to-end delay, of TCP packets by dropping packets, since no deadline is associated with TCP packets. Finally, ADTH is a generic approach that is transparent to MAC variants, hence can be adopted by any node that supports timer interrupt and

Table 1: NS-2 Simulation Configurations

Parameters	Configurations
Radio Propagation Model	TwoRayGround
Wireless Mode	IEEE 802.11b
Interface Queue	DropTail/PriQueue
Routing Protocol	AODV
Virtual Carrier Sensing	OFF
Transmit Power	15dBm
Transmission Range	30m
Carrier Sensing Range	$\gg 2x$ transmission range
Transmission Data Rate	11Mbps (no auto-fallback)

timestamps.

## 4. Simulation Results

### 4.1. Overview

235 Validation of the ADTH scheme was to be achieved through the simulation of a small wireless ad-hoc network, which would be comprised of three stationary nodes, named Node 0, Node 1 and Node 2. The open-source network simulation tool NS-2 was chosen as the simulator and the configurations used are shown in Table 1. A drop-front policy had been shown to give the best performance, 240 hence ADTH-DF was to be applied to each node and would be compared with performance of the standard FIFO-DF queue discipline. All nodes would have a default queue size of 1000 packets.

Several scenarios were tested to determine the effectiveness of the scheme, though a common approach was used in each. The simulation would begin with 245 Node 0 sending Constant Bit Rate (CBR) traffic at a set rate ( $R$ ) to Node 2.

Node 2 would then start its own CBR transmission to Node 0 after 25 seconds. Node 0 reduces its transmission rate by half at 50 seconds and Node 2 reduces its transmission rate by half at 75 seconds. Both nodes maintain their reduced rate until the simulation is ended at 100 seconds. This approach aimed to replicate  
250 the effect of sudden changes in traffic flows and the subsequent increases, or decreases, of queuing delay, nodal delay, throughput and other attributes.

Several QoS metrics were to be obtained from the simulations to analyse the effectiveness and efficiency of the ADTH scheme. Metrics such as packet loss delay, one-way delay (OWP) and UDP goodput have previously been mentioned.  
255 Additional metrics from the simulation were to be the miss target nodal delay ratio, which expresses the percentage of packets that experience a nodal delay greater than  $D_{Nr}$ , the deadline miss ratio, which denotes the percentage of packets that arrive at their destination past the deadline time, and the data yield, which indicates the percentage of packets successfully sent by the source.

#### 260 4.2. Scenario 1: Different Target Nodal Delay

##### 4.2.1. Scenario 1 (a): Same $D_{Nr}$ for All Wireless Nodes

This simulation scenario aimed to show that ADTH-DF could constrain nodal delay to different target values, which were specified by the delay requirement.  $R$  was configured to  $1.8Mbps$  to simulate a highly congested network and  
265 each node was assigned the same value from a range of 0.2s to 1.0s, with stepping of 0.2s. As the traffic sources were two hops away from their destinations, the maximum allowable one-way delay ( $OWD_R$ ) was set as  $2 \times D_{Nr}$ .

Results obtained from the simulation show that a deadline miss ratio of 0% was achieved with the ADTH-DF scheme, thus no packets arrived at their  
270 destination after the  $OWD_R$  had been exceeded. These packets will have been dropped, either at the source or an intermediary node. This should reduce the number of collisions within the network, yet also increases the packet loss ratio. Although ADTH-DF being enabled caused the packet loss ratio to be higher by 2% to 4%, depending on the  $D_{Nr}$  value, the performance gains were very  
275 convincing from a network delay perspective.

$OWD$  was observed to increase and decrease with corresponding changes in traffic load with FIFO-DF, with a peak  $OWD$  of greater than 7s being recorded. In contrast, ADTH-DF could constrain the  $OWD$  below the specified  $OWD_R$  for all values of  $D_{Nr}$ . Overall  $OWD$  trends are shown Figure 2.

280 Variations in the target queue threshold were observed at each point in time where traffic flows were altered. The threshold equalled the maximum limit while no congestion was detected in the first 25s, then a drastic change of the target queue threshold occurred after the traffic load was doubled at 25s. While network congestion was eased at 50s, due to total traffic load being reduced to  
285  $3R/2$ , the target queue threshold was not allowed to return to the maximum default limit. The threshold instead maintained the same value as the maximum estimated queue threshold.

Furthermore, the network congestion that occurs at 25s also causes a backlog to quickly build up in the queue of each node. The instantaneous queue  
290 length quickly increases, as does the queuing delay, but also quickly eases once congestion is reduced at 50s. The time taken for queues to be drained depends on the number of backlogs in the queue and system throughput. A lower target nodal delay will produce a lower target queue threshold, thus less packets will be backlogged in the queue and less time will be needed to drain the queue at  
295 lower values of  $D_{Nr}$ .

$OWD$  is constrained indirectly by the bounded nodal delay at each hop. Although no packets missed the  $OWD_R$  deadline,  $OWD$  being an aggregation of nodal delays from source to destination caused some packets to overshoot the target nodal delay. This overshooting effect may be avoided by lower nodal  
300 delay being experienced at intermediate nodes. Nevertheless, the missed target nodal delay ratio was as low as 0.0059% for Node 0 when  $D_{Nr}$  was set to 0.2s.

ADTH-DF and FIFO-DF both showed similar trends for UDP goodput, though ADTH was evidently the better performer. As the network became less congested, the UDP goodput of ADTH quickly caught up to the new network  
305 conditions, while FIFO-DF took longer due to a longer congestion state. The additional amount of time spent with the FIFO-DF scheme caused the network

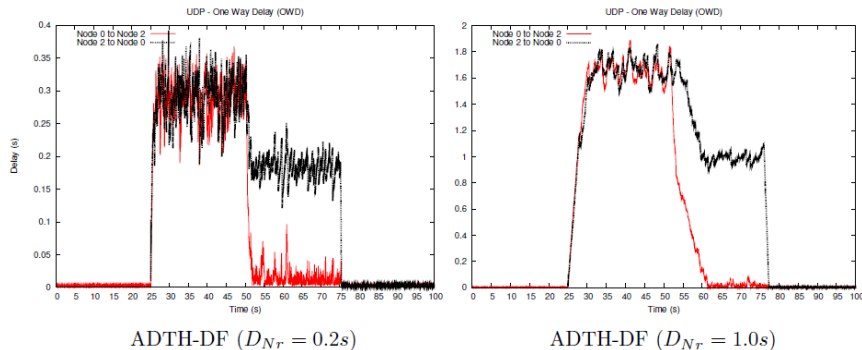


Figure 2: Scenario 1 (a): Overall *OWD* Trend for ADTH-DF ( $D_{Nr} = 0.2s$  and  $D_{Nr} = 1.0s$ )

load to not be reduced until the queue was drained, again showing how the ADTH scheme benefits the overall state of the network.

#### 4.2.2. Scenario 1 (b): Different $D_{Nr}$ for Each Wireless Node

310 Rather than implementing a homogeneous  $D_{Nr}$  for all nodes, a heterogeneous  $D_{Nr}$  throughout the network could potentially cause variations in how the flow of traffic between nodes was managed. Despite these variations, the maximum *OWD* was recorded as being less than or equal to the required value and the deadline miss ratio was 0% for both CBR flows. In sharp contrast, the deadline  
 315 miss ratios for the two flows when the FIFO-DF scheme was in use were 32.83% and 78.82%. Furthermore, despite this drastic difference in deadline miss ratio, the packet loss ratio for ADTH-DF was merely 3% higher than that of FIFO-DF. As each node was given a unique  $D_{Nr}$  value and the delay was still well managed throughout the network, as shown by the moving target queue threshold in  
 320 Figure 3 then the standalone nature of ADTH has been confirmed.

#### 4.2.3. Scenario 1 (c): Stringent $D_{Nr}$

This scenario was designed to demonstrate the effects of a very low value being set for the  $D_{Nr}$ , as prior analysis of the ADTH scheme had determined that an overly stringent  $D_{Nr}$  value could lead to issues. The MAC layer delay  
 325 could be of greater value than  $D_{Nr}$ , which would then cause the scheme to fail.

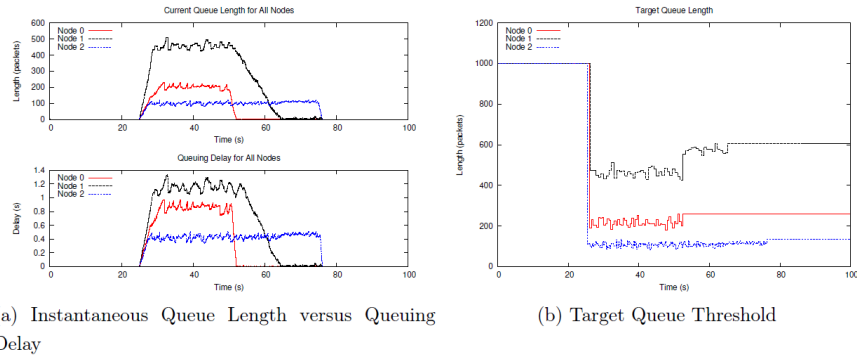


Figure 3: Scenario 1 (b): Queuing Delay Regulation via a Movable Target Queue Threshold Estimated by ADTH

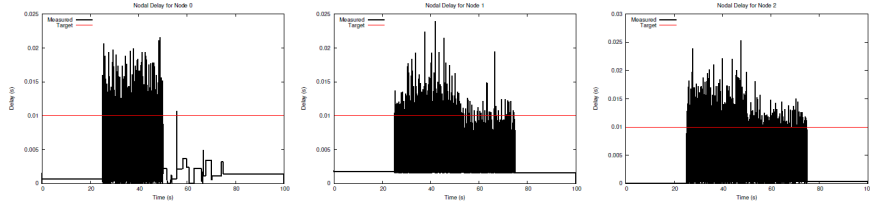


Figure 4: Scenario 1 (c): Nodal Delay for Wireless Nodes with ADTH Enabled and a Stringent  $D_{Nr}$

Figure 4 shows the delay failing to be kept below the set limit. The packet loss ratio was found to be 14.05% for the CBR flow from Node 0 to Node 2 and 21.37% from Node 2 to Node 0, which were higher by 4% and 6%, respectively, when compared with FIFO-DF.

330 The miss target nodal delay ratio was 0.95% for Node 0, 1.99% for Node 1 and 5.43% for Node 2, while the miss deadline ratio was 0.95% for the CBR flow from Node 0 to Node 2 and 1.60% from Node 2 to Node 0. These values would likely be substantially higher in a more congested network with a lower value set to  $D_{Nr}$ , though it may also be recognised that some of the overshoot values that comprised the miss deadline ratio would have been cancelled out by  
 335 the aggregation of nodal delay from other nodes.



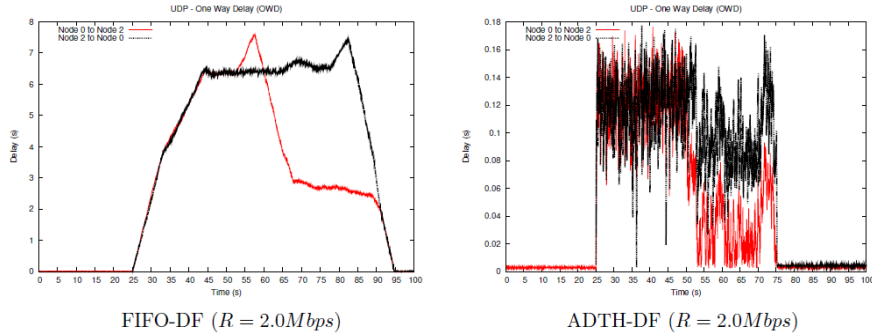


Figure 5: Scenario 2: Overall OWD Trend (ADTH-DF versus FIFO-DF)

#### 4.3. Scenario 2: Different Traffic Load

With a fixed  $D_{Nr}$ , varying levels of network traffic should show how ADTH-DF can adapt to network changes caused by load. The  $R$  value ranged from 1.0Mbps to 2.0Mbps with stepping of 0.25Mbps, while  $D_{Nr}$  remained fixed at 0.1s. With low amounts of traffic load and congestion, minimal  $OWD$  was recorded for both schemes and no packet loss was observed. As traffic rate increased, so too did the overall packet loss ratio for both schemes. However, the deadline miss ratio for ADTH remained at 0%, while the ratio for FIFO-DF was very high at certain points. For example, the deadline miss ratio for FIFO-DF ranged from 36% to 57% for Flow 0 and from 55% to an intolerable 94% for Flow 1 when  $R$  increased from 1.5Mbps to 2.0Mbps.

In contrast, ADTH performed well under the different network loads and successfully bounded the measured  $OWD$  to be less than or equal to  $OWD_R$  for all cases, as shown in Figure 5, despite the target nodal delay being only 100ms. The failure of FIFO-DF to maintain the  $OWD$  to the maximum allowable value is another clear indication that a scheme such as ADTH is needed. As before, these significant improvements were accompanied by only a minor increase of 3% to 4% to the packet loss ratio.

As with the previous scenario, queue backlogs and UDP goodput values from the simulation results were also examined. Network congestion became an issue as  $R$  increased from 1.5Mbps, which caused the expected increase in queue

backlogs and queuing delay. ADTH-DF was once again successful in its efforts to constrain the target queue threshold, which led to the bounding of nodal delay and *OWD* for the packets. FIFO-DF provided higher UDP goodput when  $R = 1.5Mbps$ , with an upwards spike being observed from the period between 50s and 60s, as buffered packets were being drained from the full IFQ during this period.

Despite the higher level of goodput, it is important to note that many packets would go on to encounter a severe amount of congestion and reach their destination with substantially large *OWD* values. ADTH-DF controlled the interface queue at a lower limit, therefore it takes less time to drain the interface queue and packets are discarded earlier, which enables more packets to be forward to their destinations within the delay requirement. This approach mitigates the network congestion implicitly, as the UDP goodput for ADTH-DF catches up at a faster rate than FIFO-DF when the traffic load reduces.

#### 4.4. Scenario 3: Different Packet Size

ADTH should also be able to react to network dynamics caused by variations in packet size. Smaller sized packets being transported under the same packet load should lead to greater congestion, thereby increasing the packet loss ratio. For this scenario, packet size varied from 300B to 1500B with stepping of 300B, all nodes had a  $D_{Nr}$  of 1.5s,  $OWD_R$  for both directions was 3.0s,  $R$  was set to 1.6Mbps and sampling interval was  $1/2 D_{Nr}$ .

Packet loss ratio was found to increase by 2% to 3% in certain cases with ADTH-DF enabled, while the measured *OWD* also slightly exceeded the  $OWD_R$  when packet size was set to 600B, with a deadline miss ratio of 0.58%. Nevertheless, these performance drops were a negligible detriment to performance when compared to the performance gains in other aspects. Furthermore, it is believed that a smaller value being used for the sampling interval would have resolved the  $OWD_R$  issue. A relatively high value of 0.75s was used for the sampling interval during the simulation. Should a lower value have been used, then ADTH-DF would have been able to more aggressively react to the changes

in network dynamics.

The recorded  $OWD$  for FIFO-DF was almost double that of the delay recorded with ADTH-DF enabled, while the deadline miss ratio was also very high in most cases. Had a smaller  $D_{Nr}$  value been used, then it is likely that the deadline miss ratio for FIFO-DF would have been even higher.  $OWD$  was bounded to less than or equal to  $OWD_R$  with ADTH-DF enabled and the miss target nodal delay ratio was negligible.

In general, the trends of  $OWD$  for both schemes were clearly impacted by the different packet sizes, particularly the congestion levels and queue fullness. This causes variation in MAC layer delay and queuing delay, while a severely congested network with a traffic load of small packets leads to a high  $OWD$ . These impacts may be somewhat mitigated by the ADTH-DF scheme, as the queuing delay will be constrained by various levels of queue backlog in each IFQ. For example, Node 1 was observed to have the highest throughput and queue threshold, yet its queuing delay was constrained to around the same target as others.

Finally, unlike other scenarios, ADTH-DF behaved similarly to FIFO-DF in terms of UDP goodput, with the results for 1500B packets shown in Figure 6 being almost identical. This indicates that the self-adaptation of queue size does not have a significant impact on UDP goodput, but also that the lenient  $D_{Nr}$  value prevented severe congestion from occurring in the simulation, hence the queue backlog did not reach the estimated target queue threshold.

#### 4.5. Scenario 4: Multiple CBR Flows with Mixed Packet Size

The ADTH-DF scheme should be a generic and scalable approach to queue management, which does not require per-flow state or statistics. This environment was simulated as multiple CBR traffic flows being carried over UDP transport agents, with varying transfer rates and packet sizes. Five CBR flows were set to  $0.3Mbps$  and sent from Node 0 to Node 2. Two flows had packets of 200B in size, two flows had packets of 600B in size, and one flow had packets of 1500B in size. Eight CBR flows set to  $0.2Mbps$  were sent from Node 2 towards

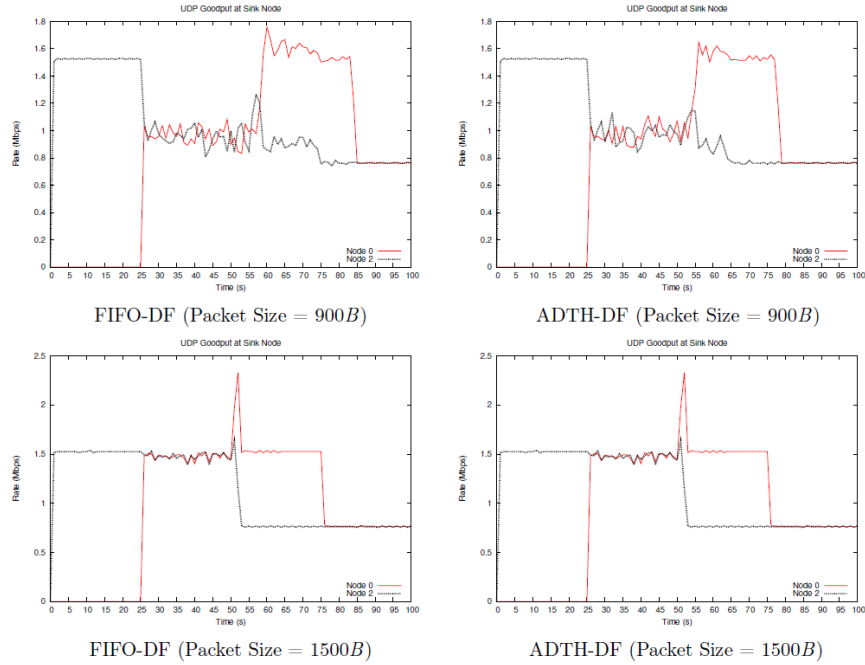


Figure 6: Scenario 3: UDP Goodput for ADTH-DF versus FIFO-DF

Node 0. Two flows had packets of 400B in size, two flows had packets of 600B in size, two flows had packets of 800B in size, and two flows had packets of 1000B in size.  $D_{N_r}$  was set to 0.5s and maximum queue limit was set to 1000 packets, while FIFO-DF was used with maximum queue limits of 50 and 1000 packets.

Performance of FIFO-DF-50 was superior to FIFO-DF-1000 in terms of data yield and maximum  $OWD$ . Although the packet loss ratio was lower with FIFO-DF-1000, data yield was extremely low and most packets failed to reach their destination by the required deadline. ADTH-DF outperforms both FIFO schemes and demonstrates its ability to bound nodal delay with an appropriate sampling frequency, even with multiple CBR flows traversing the network. With a sampling time of 0.25s, few packets reach the destination later than the deadline. The deadline miss ratio was only 0.13% for the flow from Node 2 to Node 0, while only 1 out of 25,507 packets was found to have exceeded the  $OWD_R$ . This ratio became 0% when the sampling time was set to less than half

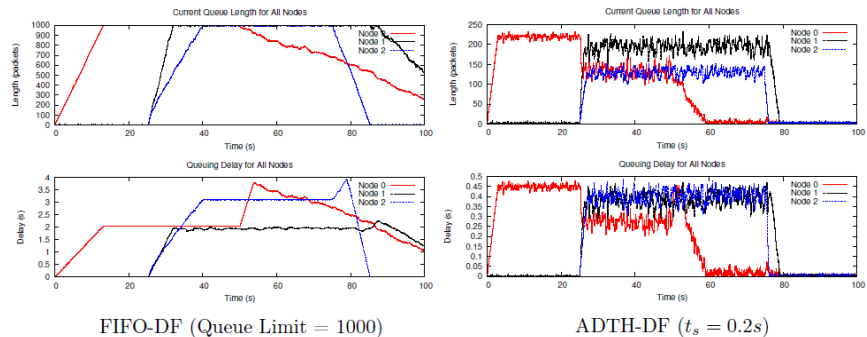


Figure 7: Scenario 4: Instantaneous Queue Length versus Queuing Delay for ADTH-DF and FIFO-DF

the  $D_{Nr}$ , due to a faster reaction to network dynamics.

The maximum *OWD* recorded for ADTH-DF was higher than FIFO-DF-50, though within the required bound. It is likely coincidental that the queue limit of 50 worked well for the scenario, as such a low queue limit will not be optimum after changes in traffic load, number of neighbours, or required delay. This once again reinforces the advantage of an adaptive queue threshold, as ADTH-DF showed better overall performance than FIFO-DF-50, but would also, for example, allow for more buffer spaces for packet buffering should  $D_{Nr}$  be set to a higher value.

Target queue thresholds were dynamically adjusted by all nodes to bound nodal delay through the queuing delay constraint, with some of the results being shown in Figure 7. These thresholds ranged from 125 to 225 packets with  $D_{Nr}$  set to 0.5s. This demonstrates the reason for the higher data yield of ADTH-DF when compared with FIFO-DF-50, as more packets can be buffered.

#### 4.6. Scenario 5: Different Number of Hops

A global view of the network should not be required by ADTH, as it was designed as a standalone queue management scheme that binds nodal delay based on the information gathered by each node. Because of this, the number of hops from source node to destination node should be irrelevant to its performance. This scenario used a different hop count ( $H$ ) to test the scalability of the ADTH

scheme, where  $H$  ranged from 3 to 15 with stepping of 3 hops.  $R$  was fixed at 0.8Mbps with  $OWD_R$  set to 1.5s and  $D_{Nr}$  set to  $OWD_R/H$ . ADTH-DF with a queue limit of 1000 packets was simulated with FIFO-DF with a queue limit of 50 packets and FIFO-DF with a queue limit of 1000 packets.

It should be expected that an increase in  $H$  will lead to an increase in the packet loss ratio and a decrease in the amount of available bandwidth, as more nodes become involved in the forwarding of packets and compete for channel access. Interference and congestion is then exacerbated, while elevated levels of MAC delay may be encountered due to the amount of MAC layer retransmissions. During the simulations, ADTH-DF produced the highest loss ratio of all three schemes, while the difference between the packet loss ratio of ADTH-DF and FIFO-DF-1000 was only significant for 12 and 15 hops.

This is not to say that ADTH-DF provided worse performance, as Figure 8 shows that network performance was clearly improved in terms of  $OWD$ .  $OWD$  was lowest with ADTH-DF and highest with FIFO-DF-1000, with the worst recorded  $OWD$  value being more than 50s, which is unacceptable for delay-sensitive traffic. FIFO-DF-50 had similar  $OWD$  values when compared to ADTH-DF below 9 hops, however ADTH-DF is superior at distances beyond this  $H$ . As shown in the previous scenario, although the low queue limit provides acceptable levels of delay during certain situations, the limit will not be optimal as network dynamics change.

Deadline miss ratio was shown to be 0% with ADTH-DF, while both FIFO-DF schemes produced very high ratios that increased dramatically when  $H$  was increased. ADTH-DF had a higher packet loss ratio, yet the data yield from ADTH-DF was the best of all schemes. However, it is notable that the data yield decreased with each increase in  $H$ . Increasing  $H$  and causing more congestion also leads to a larger backlog of packets being stored in buffers. Higher queue limits will clearly allow more packets to be queued, thus increasing the queuing delay as packets wait to be forwarded. ADTH-DF was able to limit the amount of queuing delay by reducing the  $D_{Nr}$  value, as  $H$  was increased. Although this caused a higher number of packets to exceed the  $D_{Nr}$  limit and be dropped, such

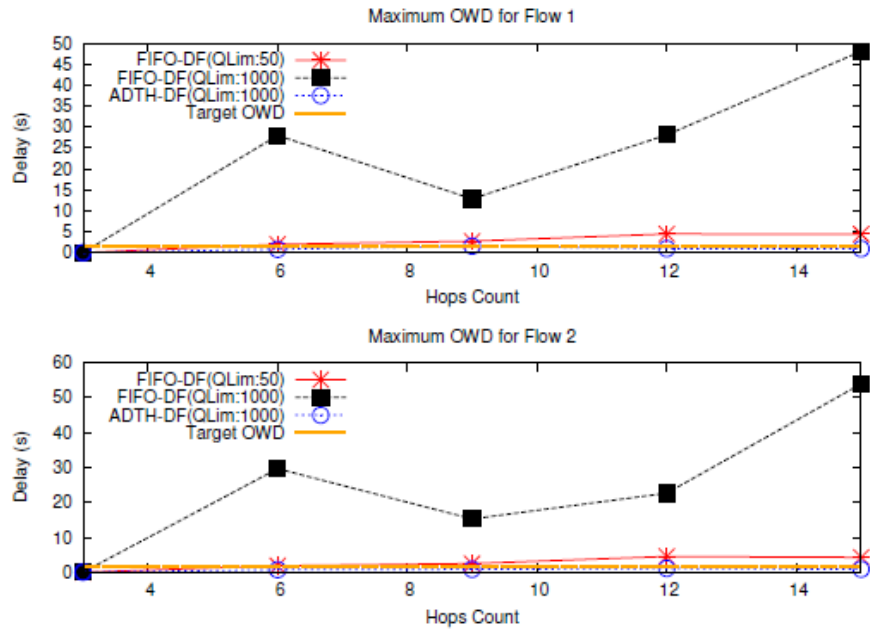


Figure 8: Scenario 5: Overall Maximum OWD for ADTH-DF and FIDO-DF

an approach maintained a lower backlog of packets in the queue. Less packets  
 being needlessly forwarded allows for more bandwidth to be available, therefore  
 network performance is not as severely impacted by congestion.

Due in part to the packet loss ratio being higher with ADTH-DF, UDP  
 goodput is lower for ADTH-DF than the two FIFO-DF schemes. However,  
 the results obtained from two FIFO-DF schemes do not show a similar trend.  
 FIFO-DF-50 gives higher UDP goodput, yet Figure 9 shows how one traffic  
 flow monopolizes the bandwidth. In contrast, FIFO-DF-1000 and ADTH-DF  
 do show similar trends and there is no obvious sign of bias to any one data  
 flow. FIFO-DF-1000 does occasionally provide higher throughput of FIFO-  
 DF-1000, although this is at the expense of exceptionally large *OWD* values.  
 Considering the other beneficial effects to QoS metrics that ADTH-DF provides,  
 then ADTH-DF is still a better choice for real-time traffic.

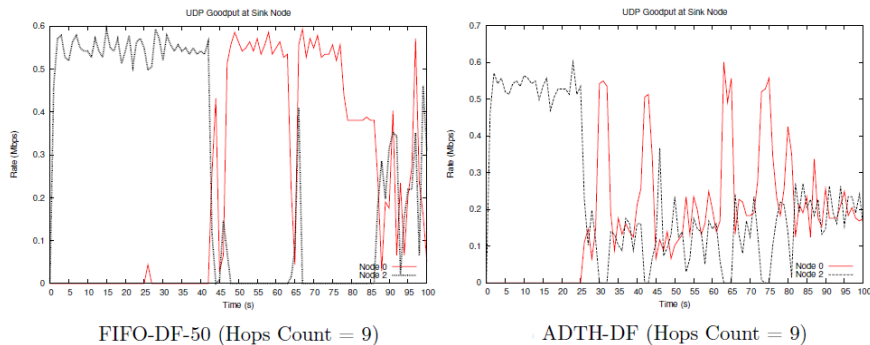


Figure 9: Scenario 5: UDP Goodput for ADTH-DF versus FIFO-DF

#### 4.7. Summary of Findings

The various simulation scenarios demonstrated that the design principles behind the ADTH scheme were valid and that, under controlled circumstances, positive results would be achieved. ADTH indirectly contributes to congestion mitigation, as it reduces bandwidth wastage from the transmission of packets that will eventually be discarded at the destination. There is no overhead in terms of messaging or signalling to gather information for the estimation process. ADTH does not maintain per flow information, which enables it to have a lightweight design and be highly scalable.

The simulations also show the simplicity of the design of ADTH. Only two parameters need to be configured to adopt the ADTH scheme, namely target nodal delay and sampling interval. The trade-off between packet loss and latency is worthwhile if network delay is more critical than packet loss, as nodal delay at each hop is significantly reduced. While the simulation results also indicated the scalability of ADTH, it must be acknowledged that simulations cannot capture network dynamics, particularly those that caused by the typical characteristics of wireless networks and environmental interference.



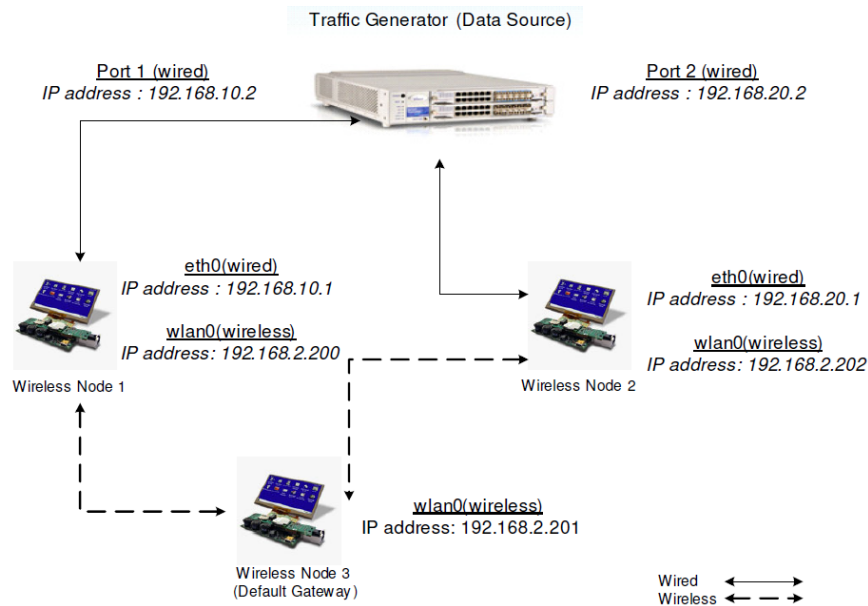


Figure 10: ADTH Testbed Topology Used for Experiments 1, 2, 3, 4 and 5

## 5. Testbed Implementation

### 5.1. Overview

515 The ADTH scheme has been shown to be effective and efficient when nodal delay is constrained in NS-2 simulations. Therefore, testing the scheme with physical devices was the next logical phase for its development. The obtained simulation results were deterministic and the various simulation scenarios only captured neighbour nodes interference, rather than environmental interference.

520 It is also difficult to adequately model characteristics such as signal fading and path loss with NS-2. Furthermore, randomness in the network will be caused by variations in processing speed, traffic inter-arrival time and hardware timing.

The simplest approach to introducing these factors was to implement the scheme on a small number of devices, which would form a testbed for the design.

525 As the scalability of ADTH was proven during simulation testing, it was deemed adequate to use a small testbed setup to show the feasibility of the ADTH implementation. Figure 10 shows a diagram of the topology that was used for

experiments 1, 2, 3, 4 and 5, while three additional nodes were used for the topology of experiment 6.

## 530 5.2. Hardware Implementation

### 5.2.1. Devices Used

Gumstix devices [14] were used for each node of the testbed. These are Linux-based miniature computers capable of embedded processing and network communication. Each device was comprised of a Verdex Pro XL6P motherboard, an XScale PXA-270 processor with a clock speed of  $624\text{Mhz}$ ,  $128\text{MB}$  of RAM and  $32\text{MB}$  of flash memory. A Netpro-vx expansion board was used to provide a Fast Ethernet port and IEEE 802.11 b/g wireless communication, via a FullMac Marvell 88W8385 WiFi module that uses the proprietary libertas driver [15]. The module used the non-free firmware automatically loaded from the Linux kernel and the standard Linux rate-controller.

Traffic to be forwarded by the Gumstix devices was provided by a Spirent Test Center (STC) traffic generator [16]. This piece of hardware is capable of sending real-time traffic at a required rate and capturing traffic with several performance metrics. The STC is highly accurate when measuring timing of latency through the usage of timestamps inserted into generated packets. The default packet sizes for the traffic generated by the STC are  $128\text{B}$ ,  $256\text{B}$ ,  $1024\text{B}$ ,  $1280\text{B}$  and  $1518\text{B}$ . To provide complementary results to the simulation experiments, the settings of the transceiver were configured to be the same as NS-2 simulation. Hence, the maximum transfer data rate was set to  $11\text{Mbps}$ .

### 550 5.2.2. Considerations and Assumptions

The processing delay for receiving packets from the Ethernet port and forwarding packets to the wireless interface are ignored, due to the assumption that processing delay is minimal and far smaller than queuing delay and MAC layer delay. Also, the queuing delay contributed by the Ethernet port can be ignored, as the interface queue will also be drained quickly enough for there to be no backlog in the queue. A detailed analysis for internal variables is not

recommended for ADTH, as these operations are intrusive and may alter the timings of the system and overall system performance. Instead, performance results are collected at end points by the STC traffic analyser.

### 560 5.2.3. *Linux Kernel*

The queuing operations of ADTH are similar to the default packet FIFO queue discipline in the Linux kernel, except that the queue threshold is regulated by the ADTH controller. The ADTH controller is invoked in a timer interrupt context when a timeout occurs. A kernel timer is used to schedule a periodic  
565 timeout based on the specified sampling interval. The ADTH implementation in the Linux kernel deviates from the initial design in several ways. Queuing delay measurement in the Linux kernel makes use of the timestamp field in the data structure of the Linux socket buffer, to store the timestamp instead of the ring buffer implementation. The queuing delay is then calculated based on this  
570 timestamp when the packet is de-queued.

### 5.2.4. *Locking Mechanism*

A locking mechanism protects critical sections of code to provide mutual exclusive access and to prevent race conditions. The queue lock must be acquired whenever packets are queued or de-queued. The ADTH controller acquires  
575 the queue lock when the size of the ADTH queue needs to be reduced. The lock is then released after packets that would potentially miss the deadline are discarded. The execution time of the timer interrupt cannot be too long, otherwise important interrupts may be blocked. Therefore, the operation to shrink the ADTH controller queue is moved from the timer interrupt function  
580 to the de-queue function, as the queue lock will have already been obtained when the de-queue operation is invoked.

### 5.2.5. *Traffic Control Utility*

The user space utility *tc* is used to associate queues with output devices for packet transmission and is installed via the *iproute2* package. This utility is  
585 needed to associate the ADTH queue with the wireless interface of each Gumstix

device. Modifications were made to enable the ADTH queue to be used as an interface queue.

Communication between the *tc* utility and an ADTH queue is via messages being passed through rtnetlink sockets. These sockets are used to exchange traffic control parameters between the user space and the kernel space, thus  
590 allowing the user to configure the target nodal delay, sampling interval and maximum queue limit parameters of the ADTH controller.

#### 5.2.6. Processing Overhead

The elapsed time for the ADTH estimator is measured by inserting times-  
595 tamps at the beginning and the end of the ADTH estimator function. Although such an approach inevitably introduces overhead and alters the recorded timings, the overhead incurred to each packet was considered insignificant when compared to end-to-end delay. Additionally, all processing of these timestamps was delegated to the STC traffic analyser, rather than the Gumstix devices. If  
600 ADTH was used in a situation where the overhead was deemed to be significant, then the overhead could be further reduced by sampling the nodal delay at a lower frequency.

### 5.3. Experiment Results

#### 5.3.1. Experiment 1: ADTH validation

This experiment was to validate the claim of ADTH being able to constrain  
605 nodal delay regardless of packet sizes and traffic load, provided that the network was not too saturated. ADTH was validated and compared against pfifo with packet sizes ranging from 128B to 1518B and incremental traffic loads from 200 *Kbps* to 2000 *Kbps*, with stepping of 200 *Kbps*.  $D_{Nr}$  was set to 200ms and  
610  $OWD_r$  was 400ms. The experiment was conducted with  $t_s$  set to 100ms and then set to 50ms.

The average *OWD* recorded for pfifo was shown to be greater than 6s, up to a maximum of 14s. Network congestion occurred with small packets when R was only 0.2*Mbps*, while larger packets caused congestion when R was greater

615 than or equal to  $0.6Mbps$ . These findings are consistent with results observed during the NS-2 simulations. The packet delivery ratio of pfifo decreased when the network becomes congested. The ratio became extremely low when the network was saturated with heavy traffic load where packets were of size 128B and 256B. Packet delivery ratio was 100% with an uncongested network and  
620 remained greater than or equal to 50% for most cases when network contention was high. However, the deadline miss ratio could be as high as 100%, which would be unacceptable for delay-sensitive applications.

Results of the experiment for ADTH showed a significant improvement over network delay. Average  $OWD$  was less than or equal to  $OWD_r$  for both sampling  
625 intervals, although some packets arrived after having exceeded the required  $OWD$ . Maximum recorded  $OWD$  was greater than 600ms when  $t_s$  was set to  $1/2 D_{Nr}$  and greater than 400ms when  $t_s$  was set to  $1/4 D_{Nr}$  when the network was severely congested, yet the deadline miss ratio remained low. Most packets arrived at their destination within the bounds of  $OWD_r$ , except for when the  
630 network was highly saturated.

Both schemes had similar packet delivery ratios, although the deadline miss ratio was noticeably improved with ADTH. The deadline miss ratio of ADTH when  $t_s$  was set to  $1/2 D_{Nr}$  was higher than when  $t_s$  was set to  $1/4 D_{Nr}$ , as the ADTH controller must be invoked at a higher frequency to manage the higher  
635 level of interference in a laboratory environment. A higher response rate to measured errors mitigates the overshooting of nodal delays. Most cases showed the deadline miss ratio for ADTH with  $t_s$  set to  $1/4 D_{Nr}$  to be either 0% or less than or equal to 3%, though this ratio did reach 6% for packet sizes of 1024B and 1280B at traffic loads greater than or equal to  $1.8Mbps$ . Deadline miss ratio  
640 for ADTH with  $t_s$  set to  $1/4 D_{Nr}$  was 0.99% with standard deviation (SD) of 2.67%, compared to 4.01% and 7.47% with  $t_s$  set to  $1/2 D_{Nr}$ . However, these ratios are still superior to those of pfifo, which had a 47.51% deadline miss ratio with SD of 49.51%.

Packet loss ratio trends were similar for pfifo and ADTH, though the early  
645 discard nature of ADTH caused the ratio for ADTH was generally slightly higher

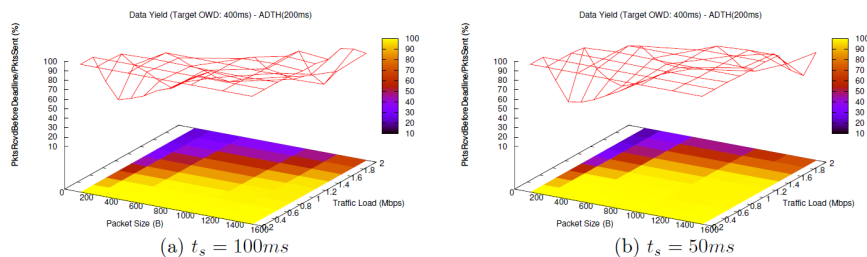


Figure 11: Overall Data Yield Ratio Trend for ADTH

than pfifo. The packet loss ratio becomes higher when traffic load increases and packet size decreases. Packet loss ratios are similar in uncongested networks with light traffic, yet the ratio will be higher for ADTH, as the target nodal delay was more stringent. Packet loss ratios were lower for  $t_s$  set  $1/4 D_{Nr}$  than  
650 when  $t_s$  set to  $1/2 D_{Nr}$ , as ADTH performs better with a finer sampling interval.

The ADTH scheme provided significant improvements to the control over the network delay, as was shown by the data yield being almost the same as the packet delivery ratio. The deadline miss ratio was close to 0% when the ADTH scheme was used, even with a congested network, as is shown in Figure  
655 11. Data yield for pfifo dropped significantly once the network became severely congested and could be as low as 0%. The average data yield for ADTH at  $t_s$  set to  $1/4 D_{Nr}$  was 76.19%, while pfifo managed only 51.94%.

This experiment showed that the ADTH controller provides better control over network delay with a finer sampling interval, as a smaller interval causes  
660 the ADTH controller to be invoked to control the queue size more frequently. This allows the ADTH controller more quickly respond to network dynamics and prevent overshooting of nodal delay for packets in the queue. A finer sampling period and faster response has been shown to be desirable in a testbed environment subject to high levels of interference. Because of this, the sam-  
665 pling interval of less than or equal to  $1/4 D_{Nr}$  was used for each subsequent experiment.

### 5.3.2. Experiment 2: Different traffic loads with random packet sizes

This experiment used varying packet sizes to evaluate what effect this would have on the performance of the ADTH scheme. Packet sizes ranged from 128B to 1518B and bi-directional traffic load (R) ranged from 1.2 to 1.8Mbps with stepping of 150Kbps. The  $D_{Nr}$  value was set to 200ms for each node, thus  $OWD_r$  was 400ms, and the sampling interval was to set to 50ms, or  $1/4 D_{Nr}$ . Each configuration used in the experiment of the pfifo and ADTH schemes was executed 50 times.

Results for pfifo showed that the network was not highly congested when R was less than or equal to 1.35Mbps. Average  $OWD$  recorded for pfifo was less than or equal to  $OWD_r$ , but the average deadline miss ratio was the non-negligible 3.43%, with SD 9.99%, when R was 1.2Mbps and 5.05%, with SD 16.86%, when R was 1.35Mbps. The deadline miss ratio increased drastically to an average of over 97% once R was greater than or equal to 1.5Mbps. Furthermore, the average  $OWD$  for pfifo increased from less than 1s to almost 4s.

The success of ADTH can be seen in the average  $OWDs$  recorded, which Figure 12 illustrates were almost all less than or equal to  $OWD_r$ . This is except for the very low deadline miss ratios of 0.55%, with SD of 1.01%, and 1.64%, with SD 1.12%, when R was 1.65Mbps and 1.8Mbps, respectively. These exceptions may be explained as packets transmitted at the next sampling interval can exceed the queuing delay, due to lower queue throughput or MAC layer delay exceeding the estimate. Therefore, some packets may overshoot the target nodal delay before the ADTH controller was invoked.

MAC layer delay increases as the network becomes more congested, which requires the queuing delay to be smaller to keep the bounded nodal delay achievable. Therefore, a higher packet drop rate was expected when the ADTH scheme was in use. The packet loss ratio of ADTH observed during the experiment was only 2% to 3% higher than pfifo, yet the average  $OWD$  recorded showed a ten-fold gain in network delay control. Packet loss ratios for both schemes were

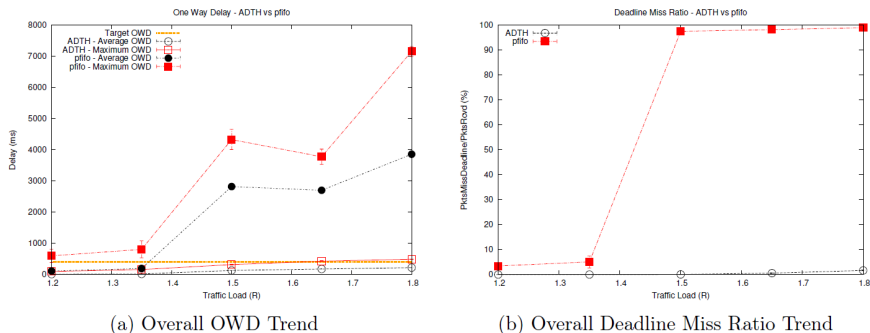


Figure 12: Overall OWD and Deadline Miss Ratio Trend (ADTH versus pfifo)

similar with an uncongested or lightly congested network. The largest gap in packet loss ratio can be seen when R was 1.5Mbps, where network contention was high, but the network had not yet become saturated with packets. The packet loss ratios were 6.53% for ADTH and 15.98% for pfifo, due to the early discard attribute of ADTH easing network contention by wasting less bandwidth on transmitting overdue packets.

Average throughput for both schemes was close to the expected throughput when traffic load was light, while system performance deteriorated significantly once R reached 1.5Mbps. Average throughput decreased with increasing traffic load, while higher packet loss ratio and larger latency were also observed. Average throughput for ADTH was slightly lower than pfifo when R was greater than or equal to 1.65Mbps, but higher than pfifo when R was 1.5Mbps, as the ADTH regulation reduces the contention of the network. Average throughput was much lower than expected when contention was high, due to bandwidth being wasted on medium contention and packet retransmission at the MAC layer.

Some spikes in recorded *OWD* may be observed when viewing the trends over the 50 runs conducted when R was 1.5Mbps and 1.8Mbps, presumably due to interference in the network varying over time. Deadline miss ratios with pfifo for all of these runs were greater than 97% and the average OWDs for all runs were significantly higher than those for ADTH. The use of ADTH saw the



average *OWD* reducing from over 2s to less than 200ms when R was 1.5Mbps and from around 4s to around 200ms when R was 1.8Mbps.

720 Overall, this experiment demonstrated that ADTH can constrain network delay with only slightly higher packet loss for certain cases. Data yield with ADTH was very close to the data delivery ratio, while large nodal delay and *OWD* with pfifo caused the data yield to be extremely low.

### 5.3.3. Experiment 3: Different target nodal delays with random packet sizes and 725 random traffic loads

This experiment was to emulate the conditions of time-varying traffic load and medium contention, then evaluate ADTH with different target nodal delay requirements. Experiment 2 had demonstrated how packet loss and *OWD* was low for ADTH when R was below 1.35Mbps, with high levels of contention and network saturation being observed once R increase beyond 1.5Mbps. This  
730 experiment aimed to show that ADTH could adapt to dynamic changes in the network efficiently and autonomously. It also intended to show that ADTH could be configured to bound nodal delay at a different required value in a dynamic environment.

735 Packet sizes ranged from 128B to 1518B, while traffic load varied between 1Mbps and 1.8Mbps.  $D_{Nr}$  values ranged from 100ms to 500ms with stepping of 100ms and the sampling interval was set to  $1/4 D_{Nr}$ . The experiment was run 50 times for each  $D_{Nr}$  value with the ADTH scheme, but only 50 times with pfifo, as the performance result for each run would be identical.

740 The *OWD* delay results show that pfifo was incapable of satisfying any delay requirements, with average *OWD* of almost 3s and a maximum *OWD* of nearly 7s. Latency was very high for a destination only two hops away from the source node and the deadline miss ratio was too high to be acceptable for delay-sensitive traffic. The average deadline miss ratio when  $D_{Nr}$  equalled 100ms was  
745 88.37%, with SE 2.36, and was 77.32%, with SE 2.63, when  $D_{Nr}$  equalled 500ms. Dynamic network conditions had a noticeable effect on the results obtained for pfifo, as sudden peaks and valleys may be observed when viewing the values of

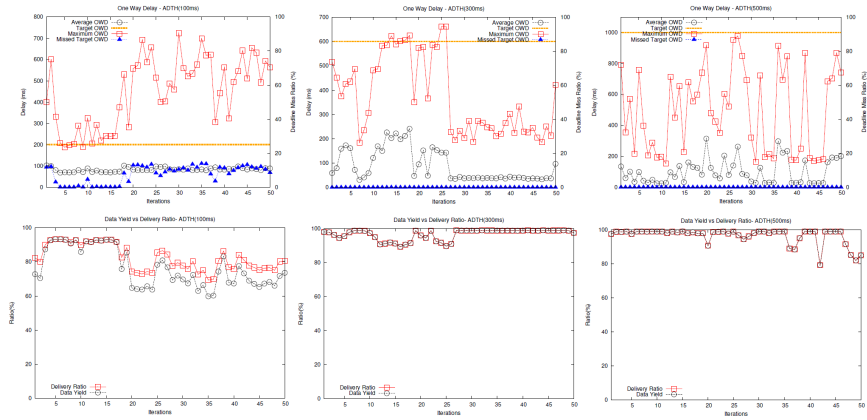


Figure 13: OWD with Deadline Miss Ratio Trend and Data Yield for ADTH with  $D_{Nr}$  set to 100ms, 300ms and 500ms

each iteration.

Average  $OWD$  for ADTH was less than  $OWD_r$  for all values of  $D_{Nr}$ , yet not all packets reached the destination without exceeding the required  $OWD_r$ . However, the deadline miss ratio was not high and could be negligible. For example, the average ratio when  $D_{Nr}$  was set to 100ms was 7.57% with SE of 0.73, while the average ratio when  $D_{Nr}$  was set to 200ms was 0.29% with SE of 0.04. The deadline miss ratio become 0% once target nodal delay requirement was greater than 200ms. It may be observed that a lower delay requirement leads to a higher deadline miss ratio, likely due to the MAC layer delay being higher than the required value with high levels of contention.

The packet loss ratio for ADTH was  $\pm 8\%$  when compared to pfifo and higher in most cases. The ratio should decrease as the delay requirement was relaxed, although other factors will impact the contention level of the network and thus packet loss. Therefore, the packet loss ratio for ADTH was lower than pfifo when  $D_{Nr}$  was set to 300ms and 500ms and does not abide to the factor caused by relaxation of the delay requirement.

The first few runs of experiments where  $D_{Nr}$  was 100ms showed that the maximum recorded  $OWD$  was lower for the repetitions from 3 to 18. These

repetitions also showed low deadline miss ratios. Variation in the levels of network interference will contribute to variation in performance, yet ADTH actively regulating the nodal delay of each node means that the average *OWD* does not fluctuate by any large amount. Higher contention and interference results in a higher deadline miss ratio, which leads to data yield to be lower than the packet delivery ratio. Experiment runs when  $D_{Nr}$  was set to 300ms show how packet loss decreased after 27 runs, as contention and interference levels were lower. Packet delivery ratio also become higher when  $D_{Nr}$  was higher, as can be seen in Figure 13.

Despite a high packet delivery ratio, data yield for pfifo was extremely low and ranged from 12% to 23% as  $D_{Nr}$  increased from 100ms to 500ms. Data yield for ADTH was quite close to the packet delivery ratio for all cases, with packet delivery ratios when  $D_{Nr}$  was 300ms and 500ms being higher than those for pfifo. ADTH has indirectly reduced the contention level by adapting the target queue threshold dynamically, which enables the early dropping of packets and reduces bandwidth wastage.

#### 5.3.4. Experiment 4: Minimum target nodal delay

It has previously been established that a limitation of the ADTH scheme was that the target nodal delay cannot be too stringent. Very high network contention means that nodes compete for channel access and many packet collisions occur. This leads to nodes needing to randomly back-off frequently, which increases the MAC layer delay. Should the MAC layer delay become larger than the set  $D_{Nr}$  value, then ADTH will fail to constrain the nodal delay within the required range. This experiment set the lowest bound to 20ms and so made  $OWD_r$  40ms, with the sampling interval set to 10ms. Bi-directional traffic was generated from the STC at a rate of 1.8Mbps with random packet sizes, which was a high traffic load that would saturate the network.

With an average deadline miss ratio as high as 99.13%, the results of the experiment clearly show that ADTH was unable to bound nodal delay when MAC layer delay exceeds  $D_{Nr}$ . The waiting time before packet retransmission

ranged from 31 to 1023. This time increased exponentially after each collision was detected and was reset once successful transmission occurs, though Gumstix hardware limits the number of retransmissions to eight.

Nodal delay was still constrained to a lower value, though higher packet loss was a trade-off for achieving this. ADTH aggressively shrinks queue size in a best-effort attempt to meet the requirement. The average recorded *OWD* was 133ms with SD 22ms and maximum *OWD* was 373ms. The packet loss ratios of this experiment may be compared to those obtained from Experiment 2, as the same configurations were used. The ratio of this experiment was 49.99% with SD 8.22%, while the ratio from Experiment 2 was 38.11% with SD 1.08%. This shows that the more stringent delay requirement contributed to a higher packet loss ratio.

Packet delivery rate was low, an average of 50.01% , and data yield was less than 1%, due to the high packet loss and deadline miss ratios. Therefore, the experiment demonstrated that ADTH can only function correctly if the delay requirement is not set to be too stringent and unrealistic.

### 5.3.5. Experiment 5: End-to-End Delay for Voice Data

Maintaining QoS for the purposes of audio and video transmission was an important design consideration for ADTH, hence it was appropriate to test the performance of the scheme with such technology. ITU-T G.114 [17] states that the acceptable delay for conversational voice applications is preferably below 150ms, with a maximum limit of less than 400ms. This experiment used CBR traffic to represent voice traffic and was carried out with incremental numbers of concurrent voice sessions, which ranged from 11s to 25s. Each voice traffic flow consisted of 20 packets per second, based on the G.711 codec rate, and each packet was 218B in size. As with previous experiments, results with pfifo would be compared to those where the ADTH scheme was used.  $D_{Nr}$  was set to 100ms and  $t_s$  was set to  $1/4 D_{Nr}$ . The test setup was simplified to disregard any potential Application layer QoS factors.

The experiment results show that all voice flows suffered high end-to-end

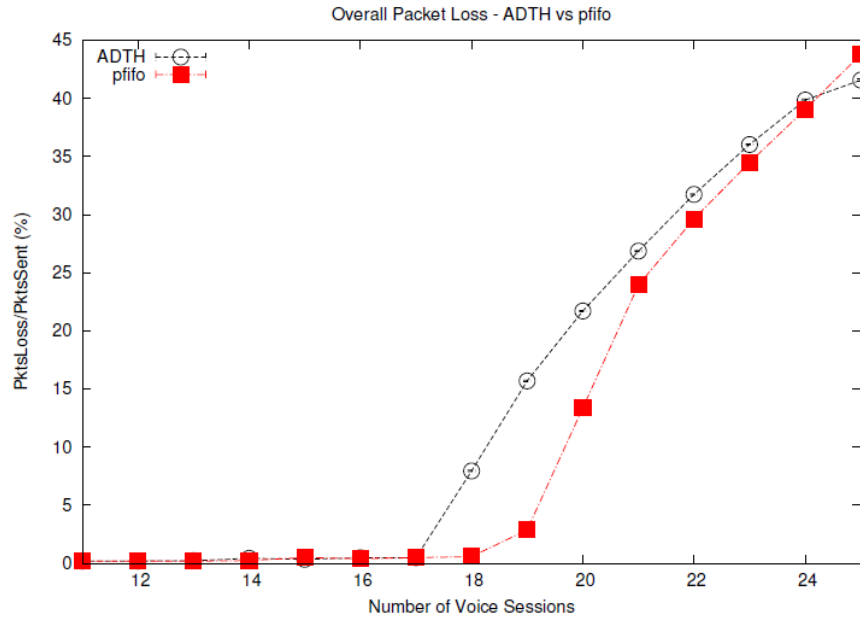


Figure 14: Overall Packet Loss Ratio Trend (ADTH versus pfifo)

delay for pfifo when the number of concurrent voice calls surpassed the network capacity. The *OWD* trend also changes drastically when the network becomes congested, particularly when the number of voice sessions increased from 18 to 19. Average *OWD* was greater than 1s, rising to over 4.5s when the number of voice sessions was 25, which was clearly unacceptable for voice traffic.

Adopting the ADTH queue discipline successfully controlled the *OWD* within the acceptable range for conversational voice traffic, with an overall average of less than 150ms for all cases and a maximum *OWD* that never exceeded 300ms. Network performance improved significantly in terms of network latency, though this came at the expense of higher packet loss ratio.

Figure 14 shows that the packet loss ratio for ADTH sharply increased with 17 sessions, compared to a small increase at 18 sessions for pfifo. This was because packets were being aggressively dropped by ADTH when congestion was detected, to regulate the queue size and meet the per-hop delay requirement. ITU-T G.1010 [18] states that conversational voice traffic is sensitive to packet

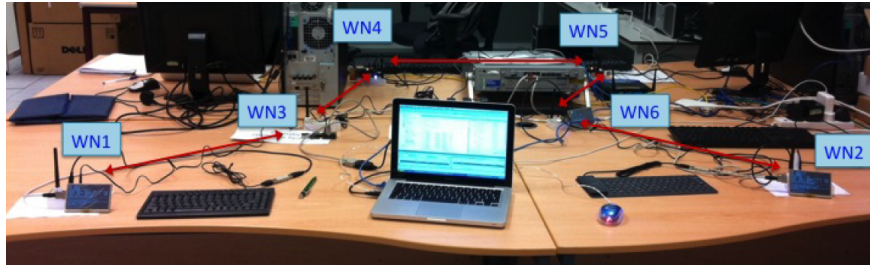


Figure 15: ADTH Testbed Topology Used for Experiment 6

loss, hence it was deemed preferable to have a loss ratio of less than 3%. An alternative admission control would be needed to deal with this issue, as a limitation of ADTH is that it constrains network delay and increases packet loss. However, it should be noted that the data yield of pfifo was far worse than  
845 the data yield with ADTH.

The experiment results show that ADTH may be inappropriate for use with applications that are sensitive to packet loss. However, ADTH can provide several useful indicators for the current congestion level of the network, such as queue fullness and the packet loss ratio. Using ADTH in conjunction with  
850 another application could potentially lead to early warnings and better response. The pfifo queue discipline cannot provide these indicators, therefore any queue overflows will occur before admission control or applications have time to react.

### 5.3.6. Experiment 6: Scalability

The addition of three more wireless nodes to the testbed allowed for the scalability and standalone nature of the ADTH scheme to be demonstrated.  
855 All nodes were placed within the transmission range of each other, but each node could not directly contact all other nodes. This created a high contention, multi-hop network that is shown in Figure 15. There were five hops between WN1 and WN2, as WN3, WN4, WN5 and WN6 acted as intermediate devices.  
860 Bi-directional traffic to and from WN 1 and WN2 consisted of randomly sized packets, ranging from 128B to 1518B, with a random traffic load, ranging from 0.2Mbps to 0.8Mbps.

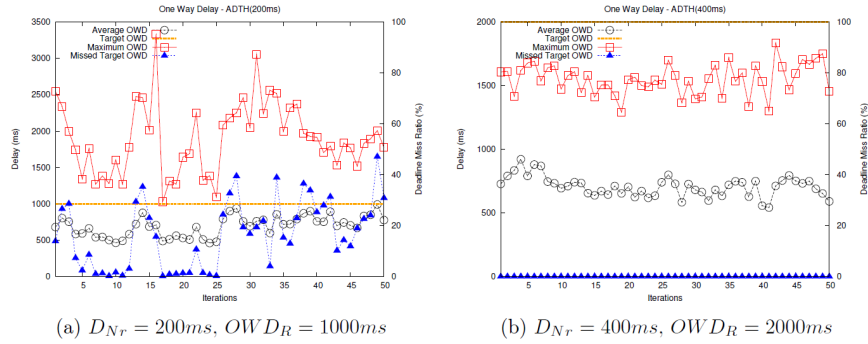


Figure 16: OWD Trend for ADTH (5 hops)

The amount of interference and competition between nodes led to a very high level of network contention. Average and Maximum *OWD* values recorded for pfifo were almost all unacceptably large. The mean of average *OWD* was 10.56s, with SD 2.79s, and the mean of maximum *OWD* was 19.13s, with SD 5.08s. The maximum *OWD* could be greater than 20s, up to a peak of 37s. If the required nodal delay was set to 200ms, then the deadline miss ratio would be as high as 97.73% and the data yield extremely low, although a mean packet delivery ratio of 57.89% was achieved.

The experiment was repeated for the ADTH scheme, where  $D_{Nr}$  was set to 200ms and 400ms with a sampling interval of  $1/4 D_{Nr}$ . Figure 16 shows that ADTH was able to bound network delay when  $D_{Nr}$  was 200ms, yet not efficiently. The lack of efficiency was due to network contention being too high and MAC delay at each node being large. The average *OWD* was less than  $OWD_r$  and the deadline miss ratio was much lower than pfifo, although it was still an average of 16.37% with SD 13.24% on average. ADTH was able to constrain nodal delay effectively when  $D_{Nr}$  was set to 400ms, with maximum *OWD* being under  $OWD_r$  for all 50 runs and the deadline miss ratio being 0%.

System performance fluctuated greatly when  $D_{Nr}$  was set to 200ms, as ADTH aggressively dropped packets for smaller delay requirements. The suppression of a queue in one node may lead to a lower queuing delay of any subsequent hops, as the packets are dropped instead of being forwarded. Fur-

thermore, fluctuations are highly dependent on interference and contention level  
885 in the network. ADTH is less capable of regulating the queue if the MAC layer  
delay is large, as queuing delay must be small to compensate. Thus, only a few  
packets can be buffered in this case. A 0% deadline miss ratio means that data  
yield was the same as the packet delivery ratio when  $D_{Nr}$  was set to 400ms,  
while the yield was only slightly lower than the delivery ratio when  $D_{Nr}$  was set  
890 to 200ms.

Packet loss ratio for ADTH was noticeably higher than pfifo when  $D_{Nr}$  was  
200ms, but  $D_{Nr}$  being set to 400ms saw ADTH achieving an average ratio of  
41.65% compared to 42.11% for pfifo. However, the average *OWD* for ADTH  
was 707ms compared to 10.56s for pfifo. The efficiency of ADTH was highlighted  
895 by comparing the data yield of pfifo and ADTH. Data yield for pfifo was less  
than 2%. The average data yield  $D_{Nr}$  was set to 200ms for ADTH when was  
25.71%, which improved to 58.35% when  $D_{Nr}$  was set to 400ms.

In conclusion, ADTH has been shown to constrain network delay with mul-  
tiple hops between the source and destination. As with previous experiments,  
900 it was vital that the target nodal delay required was within a reasonable range.

#### 5.4. Summary and Discussion

The testbed results have clearly demonstrated the feasibility of the ADTH  
scheme for use with real wireless nodes. There were also positive results with  
regards to how the scheme is scalable to varying sizes of networks. This success  
905 strongly implies that the scheme could be widely adopted as a generic queuing  
discipline for network nodes to constrain per-hop nodal delay. The processing  
overhead incurred by the ADTH controller was low and had minimal impact  
to overall system performance, which makes it particularly appropriate for use  
with network devices used for IoT purposes.

910 ADTH being a generic and standalone queuing discipline was also confirmed  
by the testbed results, as only changes at the network layer were required to  
add ADTH as a new queue discipline. Changes at the user space utilities could  
then be used to configure ADTH, with the main parameters being the target



nodal delay and sampling interval. It is recommended that the sampling interval  
915 abides by the rule of being less than or equal to a quarter of the target nodal  
delay, as the interference is assumed to be greater in the real world than in  
the simulation. ADTH has also been shown to adapt to network dynamics  
autonomously in testbed environment, where network contention levels and link  
quality varied over time.

920 The experiment results show that ADTH is able to constrain network delay  
effectively, via the per-hop queue regulation mechanism. There will be a trade-  
off between a higher packet ratio and lower network delay when the network  
is either severely congested, or the target nodal delay is too stringent. Despite  
this, the loss ratio was found to be only higher by an average of 3% to 5%  
925 while the network was lower by a factor of X. This was dependent on the target  
nodal delay requirement and the network contention level. There will also be  
a reduction in the amount of wasted bandwidth and other resources, as more  
unnecessary packets are dropped earlier. Under certain circumstances, ADTH  
may improve the packet loss ratio through more bandwidth being available  
930 for the transmission of useful packets, as well as reducing the overall network  
contention level.

Per-hop nodal delay was guaranteed, though only with the assumption that  
the target nodal delay is not overly stringent. A guarantee on end-to-end delay  
requires the assistance of QoS-aware routing, which can discover appropriate  
935 routes that satisfy the specified end-to-end delay requirement. However, de-  
terministic per-hop latency characteristics enable a deterministic end-to-end  
latency to be achieved, which may facilitate the QoS-aware routing protocols  
to more easily discover routes without the need for latency estimations from  
intermediate nodes.

## 940 **6. Conclusions and Future Work**

This paper has described the design of an adaptive queue management  
scheme for bounding nodal delay to improve network performance, while justi-

fyng the design choices made and explaining the known limitations. The results of the various simulation scenarios validate many of these design choices and demonstrate that, under controlled conditions, the scheme is functional and beneficial to network performance. Significant improvements were observed in the values obtained for one-way delay and overall queue management, which should lead to smoother performance for the network as a whole.

ADTH is a generic scheme that was invented to provide bounded delay transmission for a generic wireless ad hoc network. The proof of concept and its effectiveness has been proved in the simulation and real testbed experiments. ADTH is transferable to latest IEEE 802.11ac standard and future 802.11 ax standard. Different access categories and priority queues can be applied with a different dynamic queueing threshold to bound its required delays.

The implementation of ADTH with current standards will be explored in future work. Other future work could involve testing the performance of the scheme in more complex network environments, such as field testing sensors that form a mesh-multi-hop network with dynamic routing protocols, and determining whether the standalone nature of the scheme makes it capable of being integrated to other QoS-driven standards and mechanisms, such as the OpenFWWF firmware for certain Broadcom wireless network cards [19].

## 7. Acknowledgement

This work was sponsored by the Engineering and Physical Science Research Council, United Kingdom, under grant CASE/CNA/07/82 and the Royal Society, United Kingdom.

## References

## References

- [1] L.B. Lim, L. Guan, A. Grigg, I.W. Phillips, X.G. Wang, I.U. Awan, Controlling mean queueing delay under multi-class bursty and correlated traffic, *Journal of Computer and System Sciences* 77 (2011) 898–916.

- [2] Y. Chen, L. Li, A random early expiration detection based buffer management algorithm for real-time traffic over wireless networks, in: *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference, 2005*, pp. 507–511.
- 975 [3] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, M. Gerla, The impact of multihop wireless channel on tcp performance, *Mobile Computing, IEEE Transactions on* 4(2) (2005) 209–221.
- [4] PSRED: A Queue Management Algorithm with Priority Self-adaptive Random Early Detection for Ad Hoc Network.
- 980 [5] H. Xu, Q. Xue, A. Ganz, Adaptive congestion control in infrastructure wireless lans with bounded medium access delay, in: *Mobility and Wireless Access Workshop, 2002 (MobiWac 2002), 2002*, pp. 44–49.
- [6] Queue management in mobile ad-hoc networks (MANETS).
- [7] C. Lochert, B. Scheuermann, M. Mauve, A survey on congestion control for  
985 mobile ad hoc networks: Research articles, *Wirel. Commun. Mob. Comput.* 7(5) (2007) 655–676.
- [8] P. Marbach, Distributed scheduling and active queue management in wireless networks, in: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications, IEEE, 2007*, pp. 2321–2325.
- 990 [9] P. Marbach, Y. Lu, Active queue management and scheduling for wireless networks: The single-cell case, in: *Information Sciences and Systems, 2006 40th Annual Conference, 2006*, pp. 1560–1565.
- [10] E. Natsheh, A.B. Jantan, S. Khatun, S. Subramaniam, Fuzzy active queue management for congestion control in wireless ad-hoc, *The International Arab Journal of Information Technology* 4(1).  
995
- [11] C.W. Yuen, W.C. Lau, O.C. Yue, Capel: A packet discard policy for real-time traffic over wireless networks, in: *Communications, 2007. ICC '07. IEEE International Conference, 2007*, pp. 158–163.

- [12] L.L. Bello, G.A. Kaczynski, F. Sgró, O. Mirabella, A wireless traffic  
1000 smoother for soft real-time communications over ieee 802.11 industrial net-  
works, in: IEEE Conference on Emerging Technologies and Factory Au-  
tomation, 2006. ETFA'06., 2006, pp. 1073–1079.
- [13] O. Kleineberg, P. Fröhlich, D. Heffernan, Fault-tolerant audio and video  
bridging (avb) ethernet: A novel method for redundant stream registration  
1005 configuration (2012) 1–8.
- [14] Gumstix Inc., gumstix — dream, design, deliver.  
URL <https://www.gumstix.com/>
- [15] Linux Wireless, libertas (2015).  
URL [http://linuxwireless.sipsolutions.net/en/users/Drivers/](http://linuxwireless.sipsolutions.net/en/users/Drivers/libertas/)  
1010 [libertas/](http://linuxwireless.sipsolutions.net/en/users/Drivers/libertas/)
- [16] Spirent Communications plc, Spirent testcenter - verifying network and  
cloud evolution (2018).  
URL <https://www.spirent.com/Products/TestCenter>
- [17] ITU-T, G.114 one way transmission time.
- 1015 [18] ITU-T, G.1010 end user multimedia qos categories.
- [19] F. Gringoli, L. Nava, Open firmware for wifi networks: a unibs ntw group  
project (2016).  
URL <http://netweb.ing.unibs.it/~openfwf/>