

18 Wi-Fi Multimedia Extension (IEEE 802.11 EDCA)

18.1 Introduction

In this experiment, we will study an enhancement to WiFi that enables APs and STAs to handle various packet flows in such a way so as to provide differentiated quality of service (QoS) to these flows. In the original WiFi standard (IEEE 802.11 DCF), every device in the network has one output buffer (queue) for all packets to be transmitted onto the wireless channel. The consequence of this would be that a packet stream with strict delivery constraints and another with relatively loose delivery objectives are queued in the same output buffer at every device. Each such queue is scheduled by the DCF CSMA/CA (see the experiment “WiFi: UDP Download Throughput”), and when a queue gets its transmission opportunity the first (head-of-the-line (HOL)) packet is transmitted. This might result in packets with strict delivery constraints being kept waiting, while other, less urgent, packets get transmitted.

For example, an interactive voice call might have 200-byte packets being transmitted periodically, with a period of 20 ms. Ideally, for perfect voice playout at the receiver, this voice stream must arrive exactly as it was transmitted: every 200-byte packet must arrive, and with the gaps of 20 ms intact. If the voice packets are delayed excessively, or if the delay is highly variable, the playout is affected, and voice quality (and speaker interaction) is affected. On the other hand, TCP controlled file transfers can adapt to network delay and delay variability. Evidently, the solution is to create multiple output buffers in each device, of different transmit priorities, queue the more urgent packets in higher priority buffers, and create a mechanism for preferential transmission of the packets with tighter QoS requirements.

In this experiment we will study the EDCAF mechanism, an extension to DCF, which implements service differentiation in WiFi.

18.2 EDCAF: Access Categories

In the year 2005, the standard IEEE 802.11e (EDCAF-Enhanced Distributed Channel Access Function) was introduced, with the above issues in mind. In EDCAF, there are four Access Categories (AC0, AC1, AC2, and AC3), with AC3 being the highest priority and AC0 being the lowest. The assignment of application usage to these ACs was.

Access Category	Application	Example
AC0	Background	Background print job
AC1	Best Effort	TCP File Transfer
AC2	Video	Video Conference Video
AC3	Voice	Video Conference Voice

Table 18-1: EDCA 4 access categories for 802.11 both AP and STA

Now, if a device (an AP or a STA) is sending interactive voice packets then these packets can be queued in the AC3 buffer of the device, whereas packets of a simultaneous TCP file transfer can be queued in the AC1 buffer. The human brain is less sensitive to video packet losses, and delays in the rendering of video frames (than the human hearing system is of voice corruption), hence video is given a priority between voice and TCP. The lowest category, AC0, can be used for any application whose packets need to just be delivered, without any well-defined quality of service, for example, a low urgency bulk printing service.

Having created buffers into which the various priority packets are queued, a mechanism is needed to schedule transmissions from these buffers so that service differentiation is achieved. Ideally, strict priority service could be enforced, i.e., assuming that there is only AC3 and AC2 traffic in the network, if any device has a nonempty AC3 buffer, all packets from AC3 category should be served before any AC2 traffic is served. Further, ideally, the video and the TCP file transfers could have been assigned a guaranteed service rate, to meet their QoS requirements. Such strict priorities and guaranteed service would belong to the concept of Integrated Services. However, the IEEE 802.11 wireless access mechanism is distributed, and there is no central entity that has the instantaneous state of all the buffers in all the devices. Hence, strict priority or a guaranteed service rate is not possible to. Instead, the IEEE 802.11 series of standards adopted EDCAF (an extension to DCF) for scheduling the service of the access category queues at the contending devices. The EDCAF mechanism achieves Differentiated Services. How does the MAC layer in a device know which access category buffer to queue a packet in? This is achieved by the corresponding application using the DSCP (Differentiated Service Code Point) field in the IPV4 header to indicate the Differentiated Services class of the packet. The MAC layer of the device would have a table that maps the DSCP value to the access category.

18.3 EDCAF: Service Differentiation Mechanisms

We begin by recalling how the basic EDCF works, since EDCAF is built as an extension of EDCF. In EDCF, after handling the HOL packet in its (single) buffer (which could result in the packet being transmitted successfully or discarded (due to exceeding the maximum number of reattempts)), a device waits for DIFS, samples an initial back-off for the next packet in the buffer, and begins to count down (at slot boundaries) until the back-off counter reaches zero,

at which instant the first attempt for the next packet is made. A collision leads to a new back-off being sampled from a distribution with a larger mean. All nodes behave in exactly the same manner, thus getting opportunities to transmit packets whenever their back-off counters reach zero. Thus, all devices (STAs and APs) have the same behavior (statistically), and there is no service differentiation.

Now consider an AP with AC3 packets to be transmitted (say, voice), and an STA with AC1 packets (say, TCP). After the AP transmits a voice packets, in EDCAF, the AP's MAC waits for AIFS3 (Arbitration Inter Frame Space for Category 3) which is 2 slots, and samples a back-off from a uniform distribution over 1 slot to 2^7 slots. On the other hand, at this point the STA waits for AIFS1, which is 3 slots. In addition, after a TCP packet has been transmitted (or dropped) the STA samples a back-off for the next packet from a uniform distribution over 1 slot to 2^{31} slots. Thus, the HOL packet waiting in the AC3 buffer has two advantages over the HOL packet waiting in the AC1 buffer:

- i. The back-off counter of the AC3 category starts counting down one slot earlier than the AC1 category.
- ii. The back-off counters are smaller for AC3 than for AC1.

These two mechanisms conspire to differentiate the wireless access service in favour of AC3. Note that we do not get strict priority. For example, if a voice packet has been transmitted by the AP, and after AIFS3, the back-off sampled is 3 slots, whereas the residual back-off of the TCP transfer (at the STA) was 2 slots, then the TCP packet will be transmitted next. However, the service differentiation is significant as the simulation results from NetSim will demonstrate later in this chapter. The following is the table of all EDCAF parameters as specified by the standard.

Access Category	CWmin	CWmax	AIFSN	Max TXOP (μs)
Background (AC_BK)	31	1023	7	3264
Best Effort (AC_BE)	31	1023	3	3264
Video (AC_VI)	15	31	2	6016
Voice (AC_VO)	7	15	2	3264

Table 18-2: EDCA access parameters for 802.11 b for both AP and STA

18.4 The Experimental Plan

- Voice over DCF: We first want to understand the limitation when carrying interactive voice over DCF.
 - With this in mind, we will set up several full-duplex voice calls between several STAs and the wired network, one such call for each STA. Each full-duplex voice call will be modelled by a periodic stream of 200-byte UDP packets (160B voice plus 40B of UDP/IP headers), generated at 20 ms intervals, from the STA to the wired network, and another

such, independent stream from the wired network to the STA. We will increase the number of STAs, thereby increasing the number full-duplex voice calls, and will determine the number of calls that can be handled.

- Then we will add on TCP controlled file transfer from the wired network to another STA. Due to reasons explained earlier in this chapter, the voice performance should degrade, leading to fewer calls being possible to handle along with the TCP transfer.
- Voice over EDCAF: Next, we repeat the above two experiments with the EDCAF mechanism enabled. We should find that it is possible to maintain a substantial number of voice conversations even while running the TCP file transfer. Next, we will study what happens if the number of TCP file transfers is increased, the question being whether the number of voice conversations that can be handled gets affected.

18.5 Simulation Experiments to Study IEEE 802.11 EDCAF

Open NetSim and click **Examples > Experiments > Wi-Fi-WME-802-11e-QoS-EDCA** as shown below **Figure 18-1**.

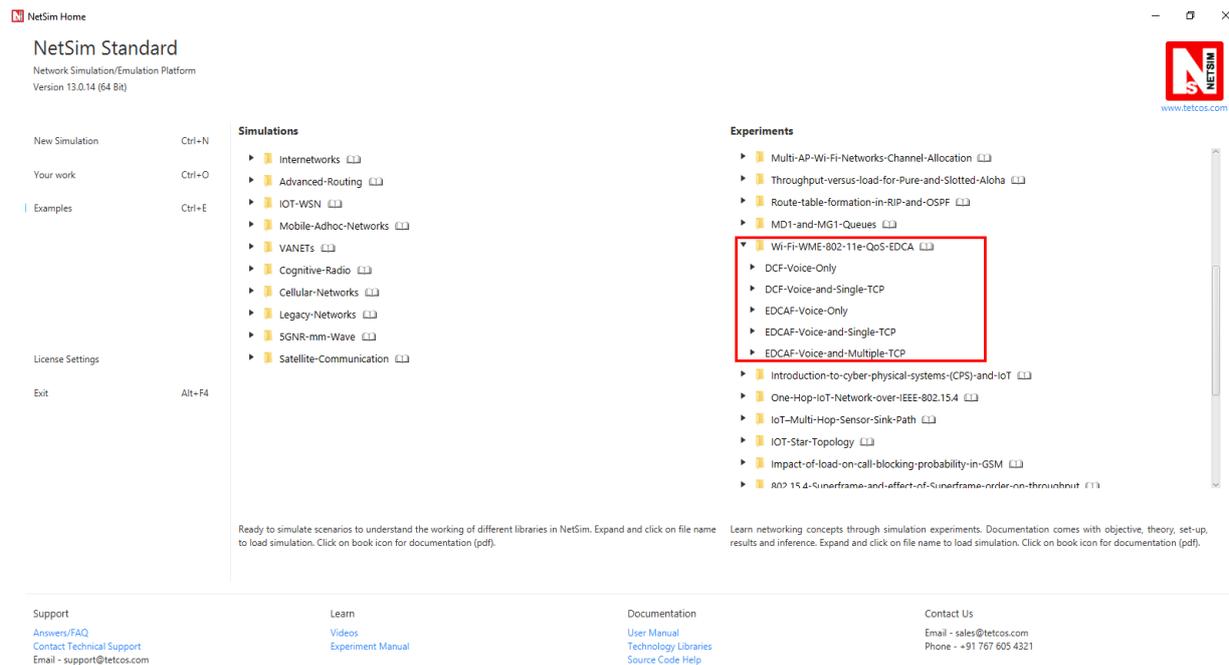


Figure 18-1: Experiments List

Case 1: DCF with full-duplex voice calls only

Network Scenario

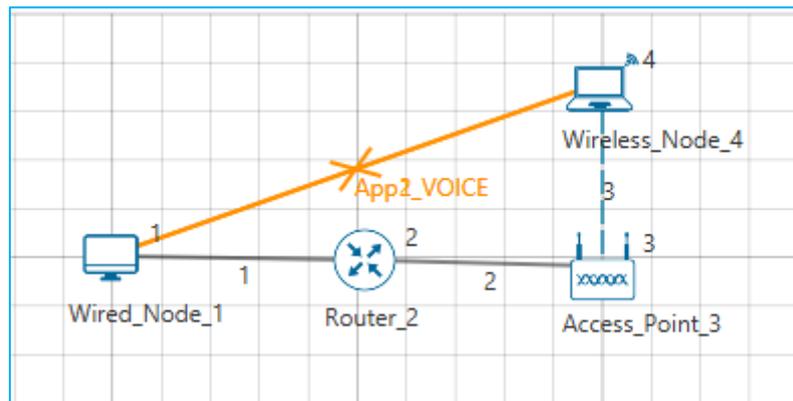


Figure 18-2: DCF with full-duplex voice calls only

Network Setup

- AP and STA operate in DCF.
- There is no error in all wired and wireless links.

Applications

- Two-way UDP voice calls from Node to Wireless_Node_i, with *i* being incremented. Each voice calls in NetSim is modelled as 2 one-way voice applications. The voice modelling option in NetSim UI currently allows transfer in one direction only. Hence, we model a two-way voice call as 2 one-way voice applications.
 - $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node *i*, while *N* represents the wired node or remote host.
 - Each voice call runs G.711 at 64 Kbps.

Case 2: DCF with full-duplex voice calls and a single TCP download

Network Scenario

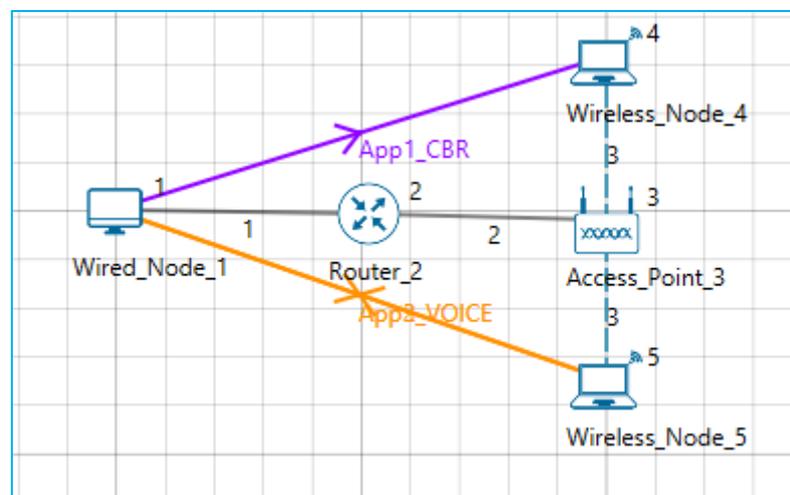


Figure 18-3: DCF with full-duplex voice calls and a single TCP download

Network Setup

- AP and STA operate in DCF.
- There is no error in all wired and wireless links.

Applications

- TCP full buffer (or saturated case) download from N to WN_1 . The saturation is generated by using a CBR application with Packet Size of 1460 B and Inter packet arrival time of 1000 μs . Since this generation rate is higher than the Wi-fi link it rate as a saturated (full buffer) TCP flow is achieved. The reason for emulating a TCP download using a CBR session, is that a TCP file download would take a longer time to simulate.
- Voice calls from N to WN_{i+1} with i being incremented. Two-way UDP voice calls from Node to Wireless_Node_ i . Each voice calls in NetSim is modelled as 2 one-way voice applications
 - $WN_{i+1} \rightarrow N$ and $N \rightarrow WN_{i+1}$. WN_i represent wireless node i , while N represents the wired node or remote host.
 - Each voice call runs G.711 at 64 Kbps.

Case 3: EDCAF with full-duplex voice calls only

Network Scenario

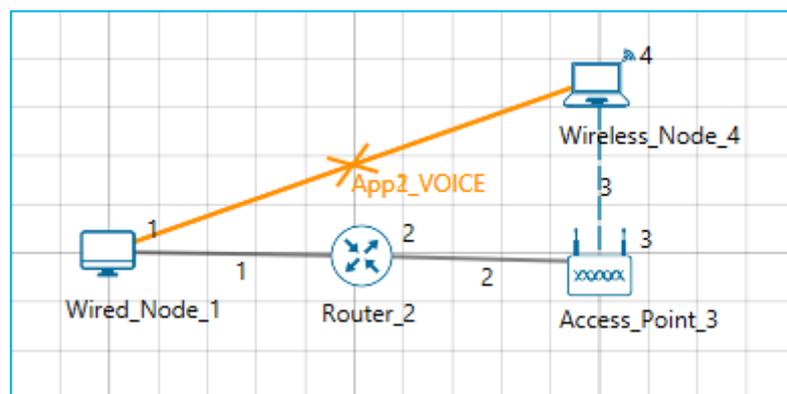


Figure 18-4: EDCAF with full-duplex voice calls only

Network Setup

- AP and STA operate in EDCAF, with EDCAF parameters set per reference paper.
- There is no error in all wired and wireless links.

Applications

- Two-way UDP voice calls from Node to Wireless_Node_ i , with i being incremented. Each voice calls in NetSim is modelled as 2 one-way voice applications. The voice modelling option in NetSim UI currently allows transfer in one direction only. Hence, we model a two-way voice call as 2 one-way voice applications.

- $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i , while N represents the wired node or remote host.
- Each voice call runs G.711 at 64 Kbps.

Case 4: EDCAF with full-duplex voice calls and a single TCP download

Network Scenario

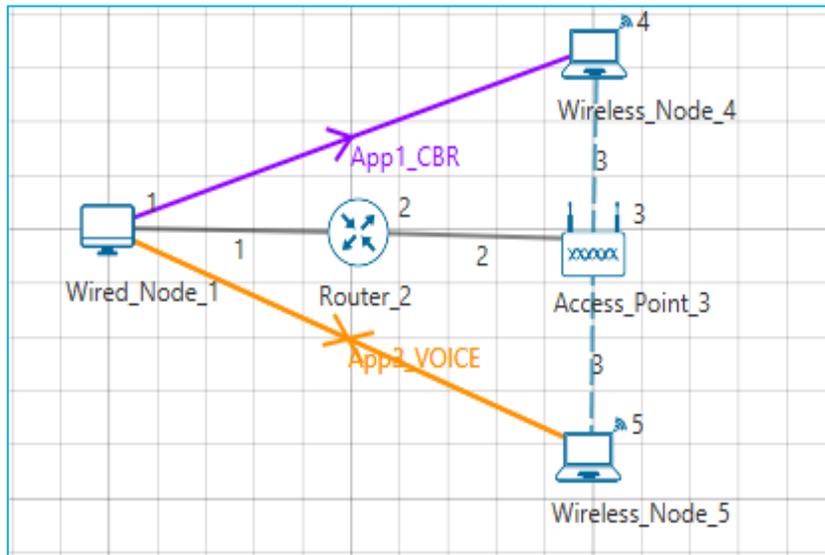


Figure 18-5: EDCAF with full-duplex voice calls and a single TCP download

Network Setup

- AP and STA operate in EDCAF, with EDCAF parameters set per reference paper. In the AP, TCP would be queued in AC_BE while Voice packets would be queued in AC_VO.
- There is no error in all wired and wireless links.

Applications

- TCP full buffer (or saturated case) download from N to WN_1 . The saturation is generated by using a CBR application with Packet Size of 1460 B and Inter packet arrival time of 1000 μs . Since this generation rate is higher than the Wi-fi link it rate as a saturated (full buffer) TCP flow is achieved.
- Voice calls from N to WN_{i+1} with i being incremented. Two-way UDP voice calls from Node to $Wireless_Node_i$. Each voice calls in NetSim is modelled as 2 one-way voice applications.
 - $WN_{i+1} \rightarrow N$ and $N \rightarrow WN_{i+1}$. WN_i represent wireless node i , while N represents the wired node or remote host.
 - Each voice call runs G.711 at 64 Kbps.

Case 5: EDCAF with full-duplex voice calls and multiple TCP downloads

Network Scenario

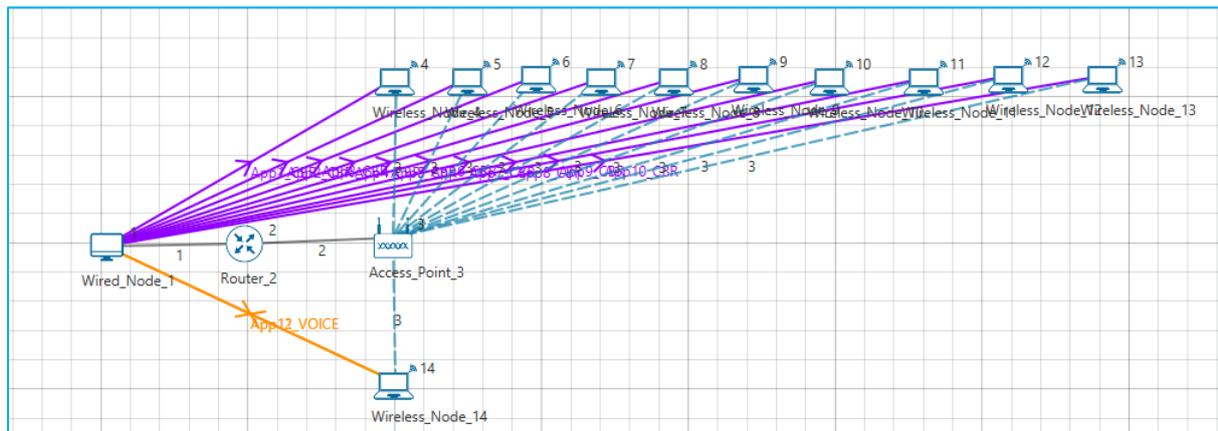


Figure 18-6: EDCAF with full-duplex voice calls and multiple TCP downloads

Network Setup

- AP and STA operate in EDCAF mode.
- There is no error in all wired and wireless links.

Applications

- 10 TCP downloads from N to WN_1 through WN_{11}
 - TCP full buffer (or saturated case) download from N to WN . The saturation is generated by using a CBR application with Packet Size of 1460 B and Inter *packet* arrival time of $1000 \mu s$. Since this generation rate is higher than the Wi-fi link it rate as a saturated (full buffer) TCP flow is achieved.
- UDP Voice calls from N to WN_{11+i} with i being incremented.
- Each voice calls in NetSim is modelled as 2 one-way voice applications.
 - $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i , while N represents the wired node or remote host.
 - Each G.711 at 64 Kbps.

18.6 Simulation Results

Case 1: DCF with full-duplex voice calls only

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i , while N represents the wired node or remote host.
- A mean delay of $20,000 \mu s$ is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 11 calls
- For the case $WN_i \rightarrow N$ this threshold is crossed at 20 calls

No of voice calls	Mean Delay Voice Avg $N \rightarrow WN_i \forall i$ (AP to STAs)	Mean Delay Voice $WN_i \rightarrow N \forall i$ (STAs to AP)	Avg
1	1181.64	1119.24	
2	2417.94	1666.23	
3	3737.10	2117.73	
4	5089.30	2558.37	
5	6433.89	3000.71	
6	7808.15	3426.20	
7	9159.53	3929.14	
8	10544.23	4383.35	
9	12026.21	4851.28	
10	13879.48	5459.24	
11	4571867.59	6308.80	
12	12197132.38	6889.13	
13	18832275.60	7408.73	
14	24246265.61	8200.14	
15	29106781.12	9032.72	
16	33825893.09	10071.78	
17	37129900.58	11548.87	
18	40666670.30	14032.98	
19	43206845.02	16694.24	
20	45890755.08	28593.24	
21	45778456.09	74205.13	
22	45570258.85	1517955.44	

Table 18-3: DCF with full-duplex voice calls only. $N \rightarrow WN_i$ and $WN_i \rightarrow N$, where N represents the wired node and WN_i represents the i -th wireless node. Therefore $N \rightarrow WN_i$ represents the flows from AP to STAs while $WN_i \rightarrow N$ represents the flows from STAs to AP.

* Mean Delay Voice $WN_i \rightarrow N$ = Average of the delay of all the applications flowing $WN_i \rightarrow N$

Case 2: DCF with full-duplex voice calls and single TCP download

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i , while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 1 call itself

No of TCP Connections	No of voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean Delay Voice $WN_i \rightarrow N$
1	1	95704.39	2977.85
	2	140275.69	3320.34

Table 18-4: DCF with full-duplex voice calls and single TCP download $N \rightarrow WN_i$ and $WN_i \rightarrow N$

Case 3: EDCAF with full-duplex voice calls

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i , while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 13 calls
- For the case $WN_i \rightarrow N$ this threshold is crossed at 21 calls

No of voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean Delay Voice $WN_i \rightarrow N$
1	1000.70	704.67
2	1720.33	1575.46
3	2460.83	2464.80
4	3231.18	3356.51
5	4084.78	4274.84
6	5092.33	5068.09
7	5985.39	5936.53
8	7083.65	6648.27
9	8193.41	7373.58
10	9226.28	8115.11
11	10508.87	8671.12
12	11697.42	9477.25
13	460068.57	12297.35
14	495197.84	12778.37
15	498731.77	13294.44
16	499873.05	13932.10
17	498605.48	14600.35
18	497835.90	15401.32
19	494160.35	16329.19
20	488004.13	17798.12
21	492066.02	20362.62

Table 18-5: EDCAF with full-duplex voice calls $N \rightarrow WN_i$ and $WN_i \rightarrow N$

Case 4: EDCAF with full-duplex voice calls and single TCP download

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i , while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 13 calls

No of TCP Connections	No of Voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean Delay Voice $WN_i \rightarrow N$
1	1	1643.79	1846.09
	2	2556.46	2877.60
	3	3384.93	3735.75
	4	4245.55	4614.23
	5	5240.15	5456.84
	6	6392.43	6221.13
	7	7495.18	6921.69
	8	8587.71	7605.80
	9	9706.95	8334.68
	10	10905.77	9004.00
	11	12372.39	9759.22
	12	18327.87	11032.25
	13	485918.40	12289.42
	14	495651.53	12800.99

Table 18-6: EDCAF with full-duplex voice calls and single TCP download $N \rightarrow WN_i$ and $WN_i \rightarrow N$

Case 5: EDCAF with full-duplex voice calls and multiple TCP downloads

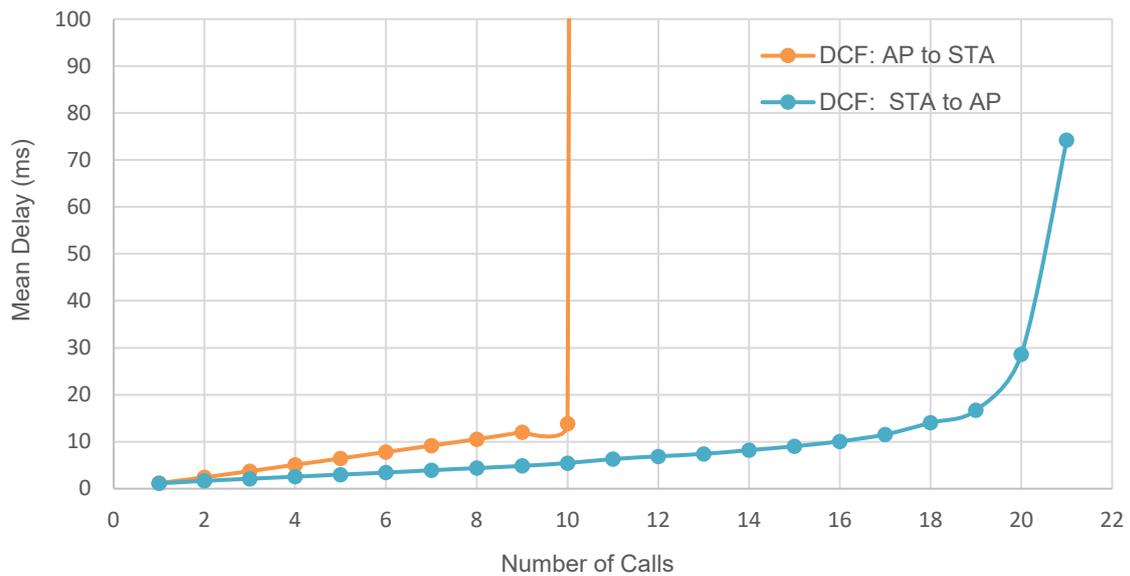
- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i , while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.

- For the case $N \rightarrow WN_i$ this threshold is crossed at 13 calls

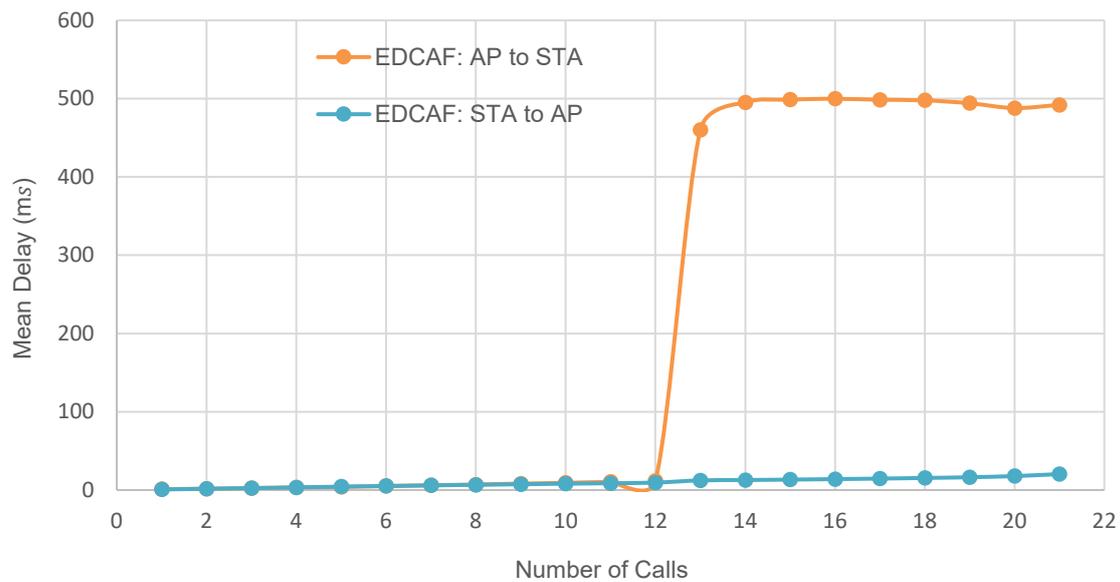
No of TCP Connections	No of voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean delay Voice $WN_i \rightarrow N$
10	1	1675.83	1841.49
	2	2607.92	2837.06
	3	3410.97	3749.06
	4	4232.94	4625.14
	5	5278.15	5461.30
	6	6424.54	6197.22
	7	7573.05	6966.70
	8	8720.54	7630.33
	9	9876.57	8308.63
	10	11070.64	9003.63
	11	12753.59	9850.23
	12	18795.31	11159.34
	13	475632.93	12308.13
	14	496716.61	12765.60

Table 18-7: EDCAF with full-duplex voice calls and multiple TCP downloads $N \rightarrow WN_i$ and $WN_i \rightarrow N$

Comparison Charts



(a)



(b)

Figure 18-7: The plot of Mean Delay Vs. Number of Calls for DCF and EDCAF (a & b)

18.7 Discussion of the Simulation Results

18.7.1 Results for DCF

1. Observations

- Only voice calls: With one voice conversation, the mean packet delay for the wired-to-wireless (downlink) direction (i.e., these packets queue in the AP) is 1.18161 ms, whereas in the wireless-to-wired (uplink) direction (i.e., these packets queue in the STA) the mean packet delay is 1.11924 ms (we will, henceforth, report these delays in milliseconds and round-off to two decimal places). These mean delays increase as the number of voice conversations increase. We notice that with 10 conversations the downlink mean packet delay is 13.69 ms, whereas the uplink packet delay is 5.47 ms. An increase of one more call result in the downlink mean packet delay becoming 4027.85 ms and the uplink mean packet delay being 6.28 ms.
- One TCP download to one STA from a wired node and increasing number of full-duplex voice calls: With just one voice call the mean delay is 116.00 ms in the downlink and 3.18 ms in the uplink. These values should be compared with 1.18 ms and 1.12 ms, respectively. Thus, with just one TCP connection, a single voice call experiences substantially larger mean delay.

2. Discussion

- With increasing number of voice calls (without any simultaneous TCP download) the dramatic change in the downlink delay, when the number of voice calls is increased from 10 to 11 is due to the downlink queue becoming unstable, i.e., the arrival rate of bits exceeds the rate at which the DCF wireless access mechanism can service the bits queued at the AP. The sudden transition from low delays to very high delays is typical of a queue going a stable regime to an unstable regime.
- It is interesting to note that at this transition point, the uplink delays have not increased as dramatically. In fact, in the uplink direction the transition from stability to instability appears in going from 22 calls to 23 calls. This difference in the downlink and uplink directions is because all the downlink voice packet streams are handled at one queue (the AP's WiFi buffer), with one contention process, whereas each uplink voice packet stream has its own buffer with its own contention process. If all the uplink voice streams had also been from one STA then the same phenomenon would have been observed in both directions.
- Next, we saw that with a single downlink TCP transfer the downlink mean delay of a single voice call is almost 100 times that without TCP. This occurs because the TCP transfer over the local area network builds up a large window, most of which resides in the AP buffer. The TCP file transfer packets are large (about 1500 bytes). A single voice stream generates 200-byte packets at 20 ms intervals. The downlink voice packets see a very large buffer, due to the TCP packets queued in the AP buffer. It may be noted here, that with this kind of delay, even a single interactive voice call will not be supported.

18.7.2 Results for EDCAF

1. Observations

- With voice calls alone the transition in downlink delay occurs in going from 12 to 13 calls.
- With TCP downloads (1 or 10 downloads) the transition in downlink voice packet delay does not change as compared to without TCP

2. Discussion

- EDCAF creates different buffers for voice and for TCP file transfers (AC3 and AC1, respectively). The service differentiation mechanism between these buffers is described earlier in this chapter. The experimental results show that voices call performance is not seriously affected by the TCP controlled file transfers.
- As before, and for the same reasons, the voice capacity is limited by the service rendered to the AP buffers.