Dayananda Sagar College of Engineering
Department of Electronics and Communication Engineering
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru– 560 078.
(An Autonomous Institute affiliated to VTU, Approved by AICTE & ISO 9001:2008 Certified)
Accredited by National Assessment and Accreditation Council (NAAC) with 'A' grade

CCN AAT

Course: Computer Communication Network   Semester: 6A

Course Code: 21EC63   Date: 25/06/2024

A Report on

PERFORMANCE ANALYSIS OF CSMA/CA

Submitted By

A. Himanshu	**1DS21EC001**
Abhijeet J Nayak	**1DS21EC004**
Abhinav Sundriyal	**1DS21EC006**
Aditya Kiran Pawaskar	**1DS21EC016**

Faculty In-charge: Prof N. Chandrashekhara

Signature of Faculty In-charge

INTRODUCTION

In the realm of computer networking, ensuring efficient and fair access to shared communication channels is crucial. One pivotal protocol designed to achieve this balance is Carrier Sense Multiple Access with Collision Avoidance, or CSMA/CA. Unlike its more widely known counterpart, CSMA/CD (Collision Detection), which is used in Ethernet networks, CSMA/CA is specifically tailored for wireless networks where collisions are more problematic due to the nature of radio signals.

CSMA/CA operates on the principle of nodes sensing the wireless medium before transmitting data. Rather than detecting collisions after they occur (as CSMA/CD does), CSMA/CA aims to prevent collisions proactively. This proactive approach involves a careful exchange of control packets between communicating nodes to reserve the channel for transmission, thereby reducing the likelihood of interference and collisions.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a fundamental medium access control protocol used in wireless networks to regulate how nodes access a shared communication channel. Unlike its wired counterpart, CSMA/CD (Collision Detection), which is used in Ethernet networks, CSMA/CA is specifically designed to mitigate the challenges posed by the unreliable nature of wireless transmissions, where collisions can occur due to factors such as hidden terminals, signal propagation delays, and varying signal strengths.

Principles of CSMA/CA:

1. Carrier Sensing:

Before transmitting data, a node using CSMA/CA first listens to the wireless channel to determine if it is idle (no other transmissions detected). This step is crucial because multiple nodes within range may attempt to transmit simultaneously, leading to collisions and wasted bandwidth.

2.Backoff Mechanism:

If the channel is busy when a node wants to transmit (after sensing it as idle), it waits for a randomly chosen backoff period before attempting to transmit again. This randomized backoff helps to further reduce the likelihood of collisions when multiple nodes contend for the channel simultaneously.

Operation of CSMA/CA:

1**. Channel Sensing**: A device listens to the wireless channel to determine if it is idle. If the channel is busy, the device waits for a random duration before sensing the channel again.

2. **Contention Window Selection**: If the channel is idle, the device selects a random value from its contention window. The size of the contention window typically increases after each unsuccessful transmission attempt to reduce the probability of collisions.

3. **Transmission:** After waiting for the specified duration, the device initiates transmission if the channel is still idle. It sends the data frame and waits for an acknowledgment from the receiving device.

4. **Acknowledgment and Retransmission**: If the acknowledgment is received within the specified time frame, the transmission is successful. If not, the device assumes a collision or transmission failure and enters a backoff phase, randomly selecting a new contention window and attempting to transmit again.

TCL Script

```
set ns [new Simulator]
#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
#Open the Trace files
set file1 [open csma_ca.tr w]
set winfile [open WinFile w]
$ns trace-all $file1
#Open the NAM trace file
set file2 [open csma_ca.nam w]
$ns namtrace-all $file2
```

```
#Define a 'finish' procedure
proc finish {} {
global ns file1 file2
$ns flush-trace
close $file1
close $file2
exec nam csma_ca.nam &
exit 0
}
#Create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color red
$n1 shape box
#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Ca Channel]
#Setup a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
```

```
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 0.01mb
$cbr set random_ false
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 124.5 "$cbr stop"
# next procedure gets two arguments: the name of the
# tcp source node, will be called here "tcp",
# and the name of output file.
```

```
proc plotWindow {tcpSource file} {
global ns
set time 0.1
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
set wnd [$tcpSource set window_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file" }
$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 5 "$ns trace-annotate \"packet drop\""
$ns at 125.0 "finish"
$ns run
```

AWK Script

```
BEGIN{
tcp=0;
cbr=0;
count=0
}
{
if($1=="-" && $5=="tcp")
tcp++
if($1=="-" && $5=="cbr")
cbr++
if($1=="d")
count++;
}
END{
printf("\n no of tcp packet sent =%d\n",tcp);
printf("\n no of cbr packet sent =%d\n",cbr);
printf("\n no of  packet dropped =%d\n",count);
}
```

Results

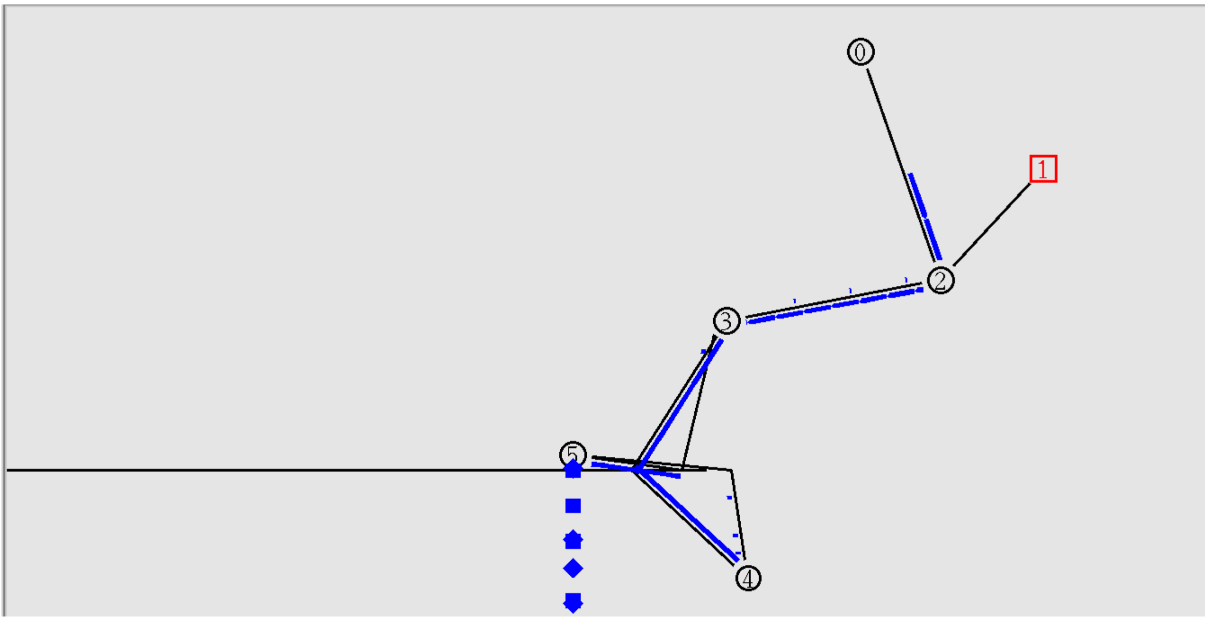

Fig1: Simulation of CSMA/CA

```
no of tcp packet sent =16424
no of cbr packet sent =468
no of  packet dropped =8567
```

Fig2: results when random time interval.

```
no of tcp packet sent =21992
no of cbr packet sent =468
no of  packet dropped =11351
```

Fig 3:results when some time interval

Advantages of CSMA/CA:

- Collision Avoidance: By incorporating a contention window and randomized backoff mechanism, CSMA/CA helps in avoiding collisions more effectively than CSMA/CD, especially in wireless networks where collision detection is challenging.
- Fairness: The protocol promotes fairness by ensuring that all nodes have an equal opportunity to access the channel, thus preventing any single node from monopolizing bandwidth.
- Efficiency: Despite the overhead introduced by RTS/CTS frames, CSMA/CA improves overall channel efficiency by reducing the number of retransmissions caused by collisions.

Applications:

- CSMA/CA is widely used in Wi-Fi networks, Bluetooth, and other wireless technologies where multiple devices share a common communication medium.
- Its effectiveness in managing channel access makes it suitable for environments requiring reliable and efficient wireless communication.

Conclusion

In conclusion, CSMA/CA plays a critical role in ensuring efficient and fair access to the wireless medium by proactively avoiding collisions. Understanding its mechanisms is essential for designing and managing robust wireless networks that can handle the complexities and challenges inherent in wireless communications.