

به نام خدا



**Sharif University of Technology**

Department of Electrical Engineering

# Data Network

Instructor: Dr. Pakravan

MAC Sublayer Simulation

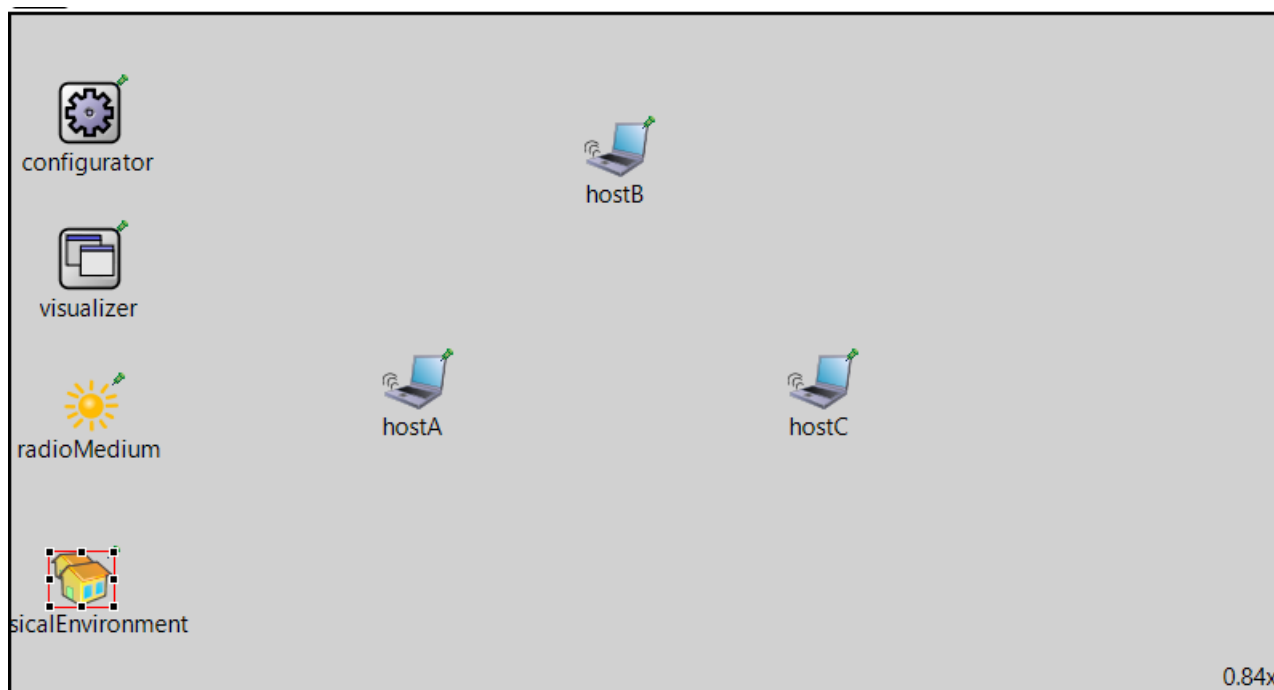
Mohammad Javad Amin

401211193

Fall 2023

# سوال ۱

با توجه به صورت سوال، شبکه را ترسیم می‌کنیم و فایل ned را می‌نویسیم.



حال با توجه به سناریوهایی که داریم فایل ini را می‌نویسیم.

توجه: برای نوشتن این قسمت از این LINK استفاده شده و مقادیر نرخ داده‌ها و نحوه تولید و ساینز بسته‌ها... مقداردهی می‌کنیم.

توجه: برای اجرای این فایل حتما باید از INET 4.5 استفاده شود.

قسمت عمومی و مشترک سناریوها را به شکل زیر می‌نویسیم.

```
# ARP settings
*.host*.ipv4.arp.typeName = "GlobalArp"

# Network Interface Card settings
*.host*.wlan[*].mgmt.typeName = "Ieee80211MgmtAdhoc"
*.host*.wlan[*].agent.typeName = ""
*.host*.wlan[*].radio.typeName = "Ieee80211UnitDiskRadio"
*.host*.wlan[*].radio.transmitter.communicationRange = 300m
*.host*.wlan[*].bitrate = 6Mbps

# Medium Access Control settings
**.wlan[*].mac.dcf.channelAccess.pendingQueue.packetCapacity = 10
**.wlan[*].mac*.rateSelection.responseAckFrameBitrate = 6Mbps
**.wlan[*].mac*.rateSelection.dataFrameBitrate = 6Mbps
**.wlan[*].mac*.rateSelection.controlFrameBitrate = 6Mbps

# Visualizer settings
*.visualizer.mediumVisualizer.displaySignals = true
*.hostA.wlan[*].radio.displayCommunicationRange = true
*.hostB.wlan[*].radio.displayCommunicationRange = true
*.hostC.wlan[*].radio.displayCommunicationRange = true

*.hostA.wlan[*].radio.displayInterferenceRange = true
*.hostC.wlan[*].radio.displayInterferenceRange = true
```

```

*.visualizer.physicalLinkVisualizer.displayLinks = true
*.visualizer.dataLinkVisualizer.displayLinks = true
*.visualizer.dataLinkVisualizer.packetFilter = "UDP*"

# Traffic generation settings
*.hostA.numApps = 1
*.hostA.app[0].typename = "UdpBasicApp"
*.hostA.app[0].destAddresses = "hostB"
*.hostA.app[0].destPort = 5000
*.hostA.app[0].packetName = "UDPData"
*.hostA.app[0].messageLength = 1000byte
*.hostA.app[0].sendInterval = 5ms

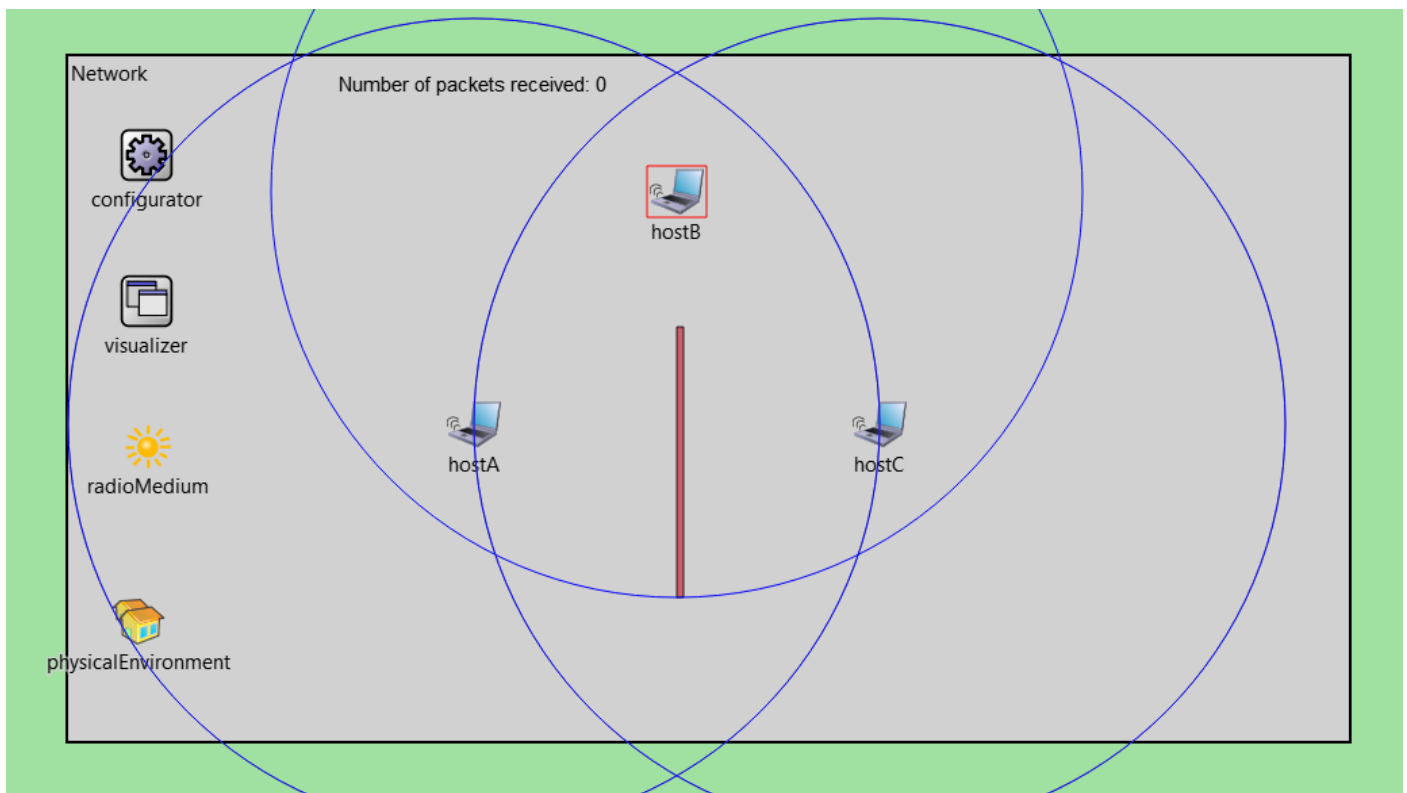
*.hostC.numApps = 1
*.hostC.app[0].typename = "UdpBasicApp"
*.hostC.app[0].destAddresses = "hostB"
*.hostC.app[0].destPort = 5000
*.hostC.app[0].packetName = "UDPData"
*.hostC.app[0].messageLength = 1000byte
*.hostC.app[0].sendInterval = exponential(5ms)

*.hostB.numApps = 1
*.hostB.app[0].typename = "UdpSink"
*.hostB.app[0].localPort = 5000

# Physical environment settings
*.radioMedium.obstacleLoss.typename = "IdealObstacleLoss"

```

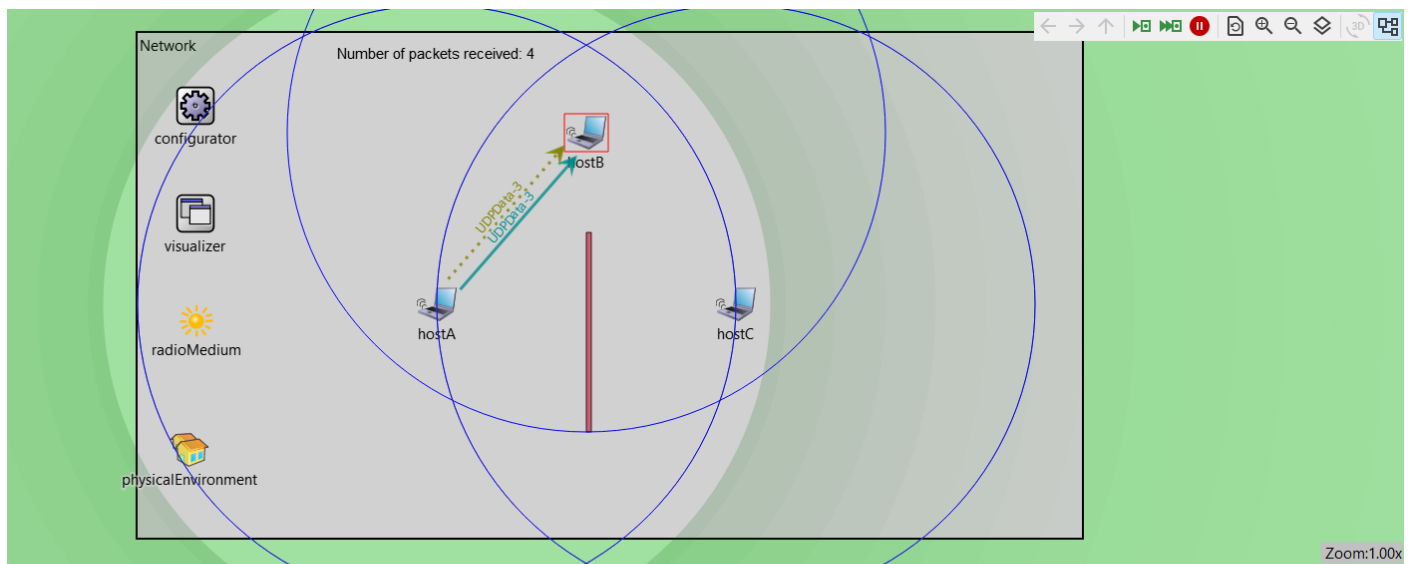
## سناریو اول: نود مخفی و بدون استفاده از RTS/CTS



```

# Configuration with wall and without RTS/CTS
[Config WallOnRtsOff]
description = "With wall, Without RTS/CTS"
*.physicalEnvironment.config = xmldoc("wall.xml")

```

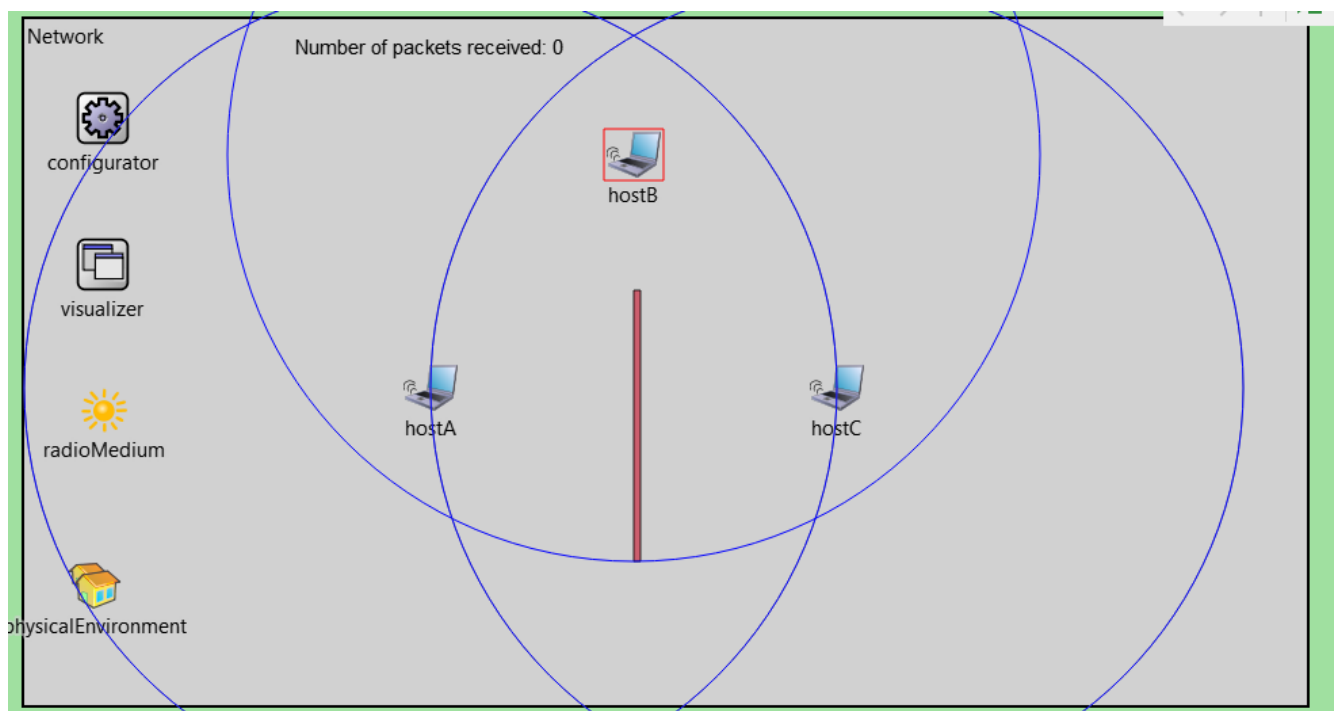


```
INFO (RX)Network.hostB.wlan[0].mac.px: Received frame from PHY: (inet::Packet)UDPData-3 (1064 B) (inet::SequenceChunk) length = 1064 B
INFO (Dcf)Network.hostB.wlan[0].mac.dcf: Processing lower frame: UDPData-3
INFO (Dcf)Network.hostB.wlan[0].mac.dcf: Processing received frame UDPData-3 as recipient.
** Event #288 t=0.031275328799 Network.hostB.wlan[0].llc (Ieee80211LlclPd, id=119) on UDPData-3 (inet::Packet, id=359)
INFO: Received (inet::Packet)UDPData-3 (1036 B) (inet::SequenceChunk) length = 1064 B from lower layer.
INFO (MessageDispatcher)Network.hostB.li: Dispatching packet to protocol, protocol = ipv4(50), servicePrimitive = 2, inGate = (omnetpp::cGate)in[2] ← wlan
INFO (MessageDispatcher)Network.hostB.bl: Dispatching packet to protocol, protocol = ipv4(50), servicePrimitive = 2, inGate = (omnetpp::cGate)in[1] ← li.
INFO (MessageDispatcher)Network.hostB.cb: Dispatching packet to protocol, protocol = ipv4(50), servicePrimitive = 2, inGate = (omnetpp::cGate)in[0] ← bl.
INFO (MessageDispatcher)Network.hostB.nl: Dispatching packet to protocol, protocol = ipv4(50), servicePrimitive = 2, inGate = (omnetpp::cGate)in[1] ← cb.
INFO (MessageDispatcher)Network.hostB.ipv4.lp: Dispatching packet to protocol, protocol = ipv4(50), servicePrimitive = 2, inGate = (omnetpp::cGate)in[2] ←
** Event #289 t=0.031275328799 Network.hostB.ipv4.ip (Ipv4, id=158) on UDPData-3 (inet::Packet, id=359)
```

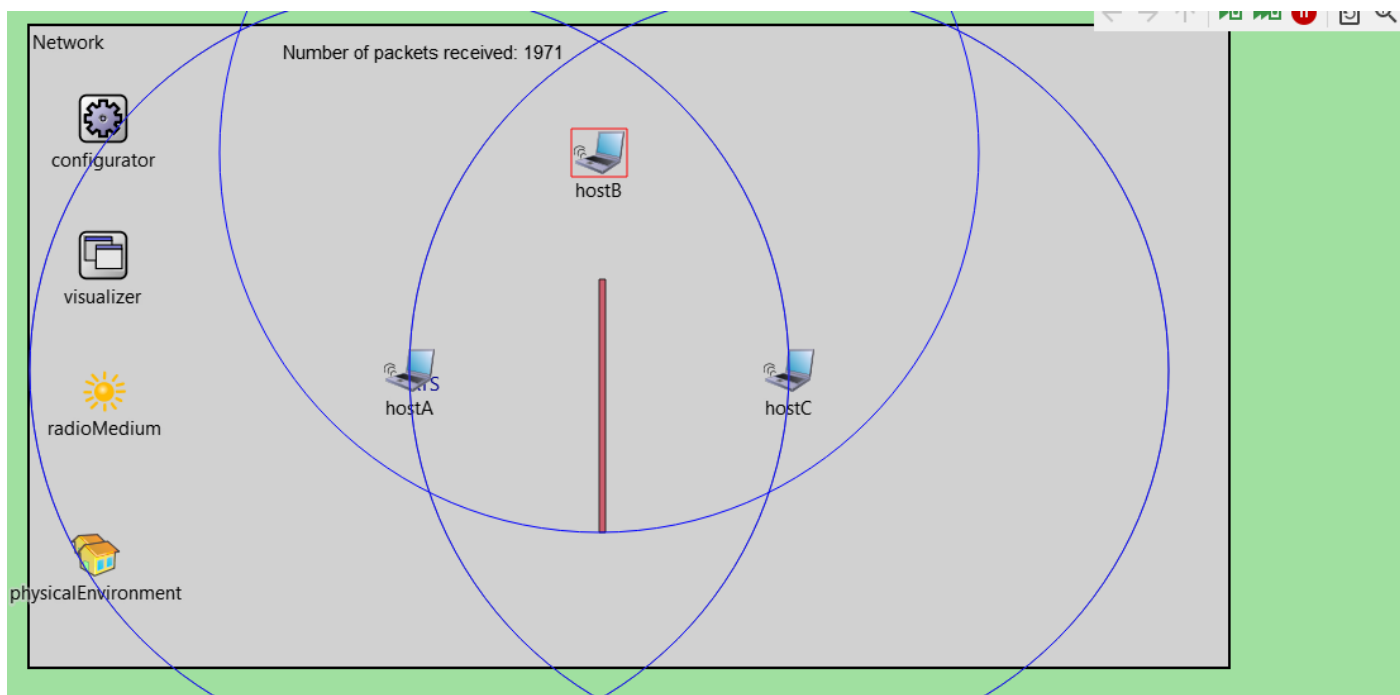
از اجرا، فیلم تهیه شده است.

خروجی: در  $t=5s$  تعداد بسته‌های دریافت شده: ۱۳۸۳

سناریو دوم: نود مخفی و با استفاده از RTS/CTS



```
# Configuration with wall and RTS/CTS enabled
[Config WallOnRtsOn]
description = "With wall, With RTS/CTS"
*.physicalEnvironment.config = xmldoc("wall.xml")
*.host*.wlan[*].mac.dcf.rtsPolicy.rtsThreshold = 100B
```



```

Medium)Network.radioMedium: Signal send count = 10010
Medium)Network.radioMedium: Reception computation count = 16016
Medium)Network.radioMedium: Interference computation count = 55752
Medium)Network.radioMedium: Reception decision computation count = 11852
Medium)Network.radioMedium: Listening decision computation count = 27972
Medium)Network.radioMedium: Reception cache hit = 83.9184 %
Medium)Network.radioMedium: Interference cache hit = 66.6667 %
Medium)Network.radioMedium: Noise cache hit = 0 %
Medium)Network.radioMedium: SNIR cache hit = 50 %
Medium)Network.radioMedium: Reception decision cache hit = 50 %
Medium)Network.radioMedium: Reception result cache hit = 0 %

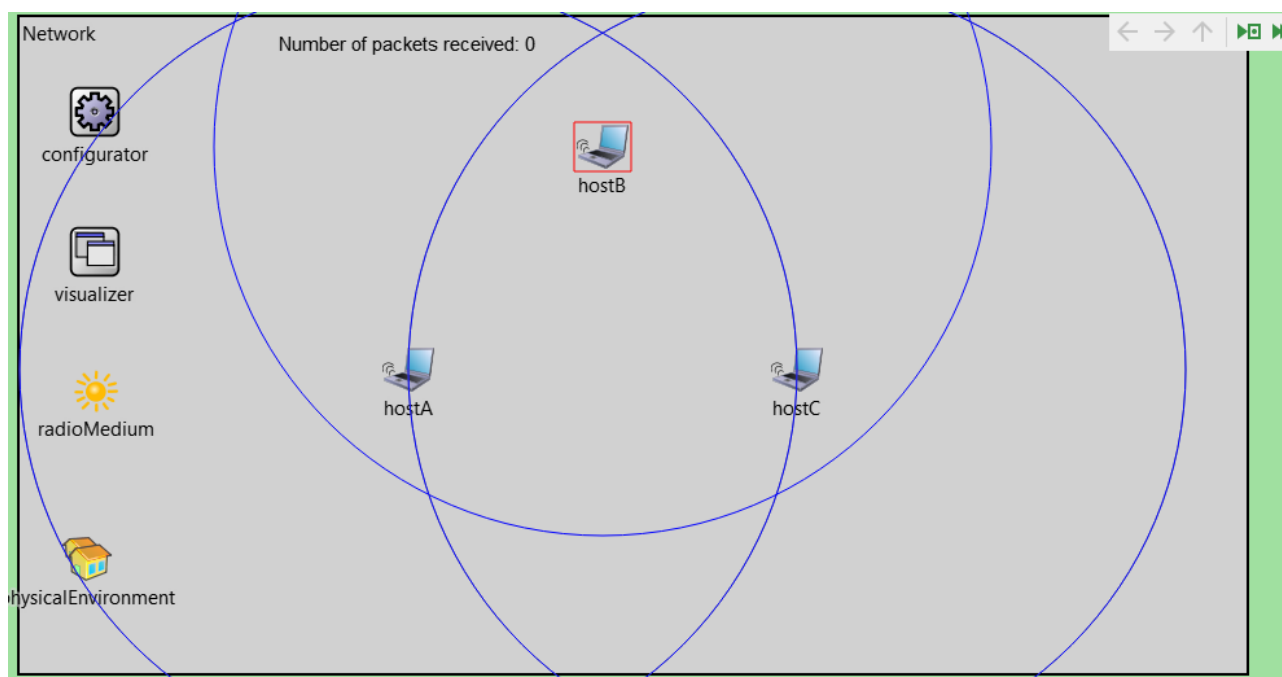
```

از اجرا، فیلم تهیه شده است.

خروجی: در  $t=5s$  تعداد بسته‌های دریافت شده: ۱۹۷۱

نتیجه: در مسئله نود مخفی استفاده از **RTS/CTS** باعث افزایش تعداد بسته‌های دریافتی شده و بهره کانال افزایش می‌یابد.

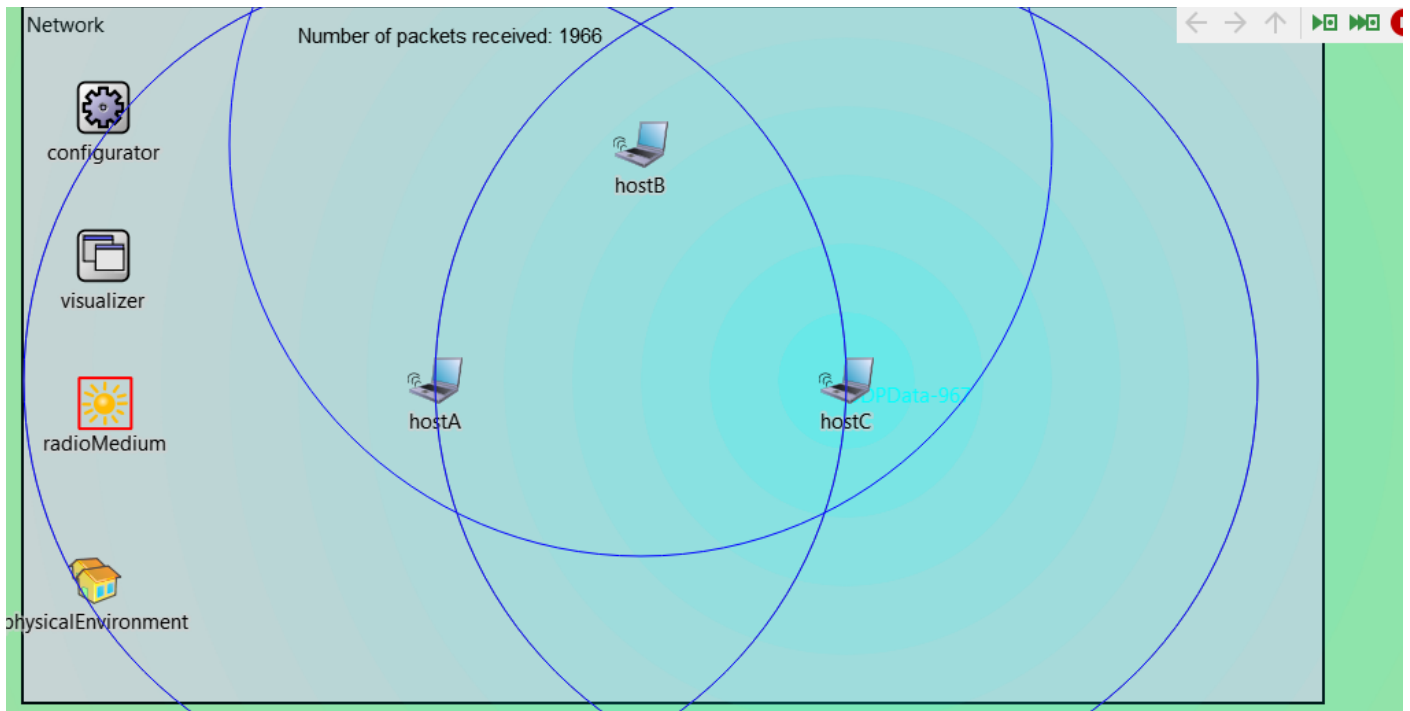
سناریو سوم: شبکه بدون نود مخفی و بدون استفاده از **RTS/CTS**



```
# Configuration without wall and without RTS/CTS
```

```
[Config WallOffRtsOff]
```

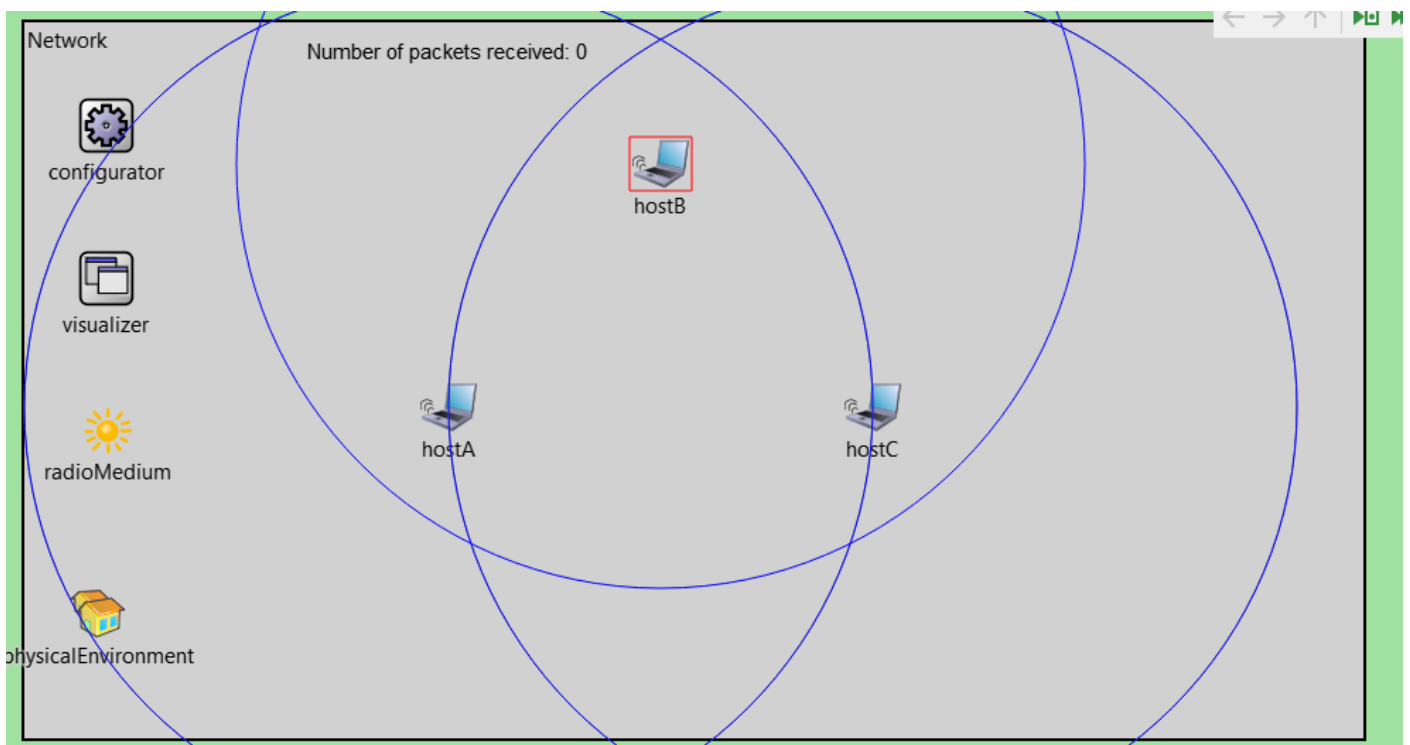
```
description = "No wall, No RTS/CTS"
```



از اجرا، فیلم تهیه شده است.

خروجی: در  $t=5s$  تعداد بسته های دریافت شده: ۱۹۶۶

سناریو چهارم: شبکه بدون نود مخفی و با استفاده از RTS/CTS

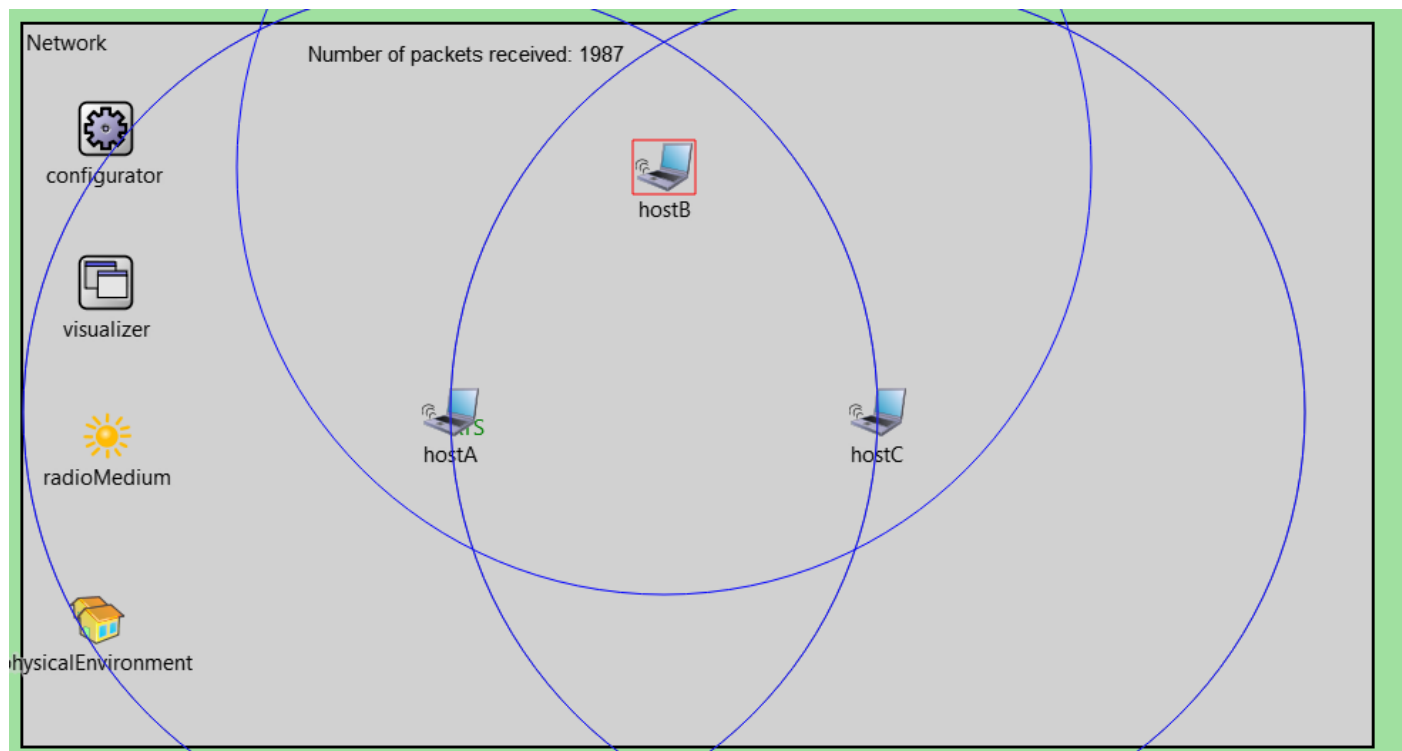


```
# Configuration without wall With and RTS/CTS
```

```
[Config WallOffRtsOn]
```

```
description = "No wall, With RTS/CTS"
```

```
*.host*.wlan[*].mac.dcf.rtsPolicy.rtsThreshold = 100B
```



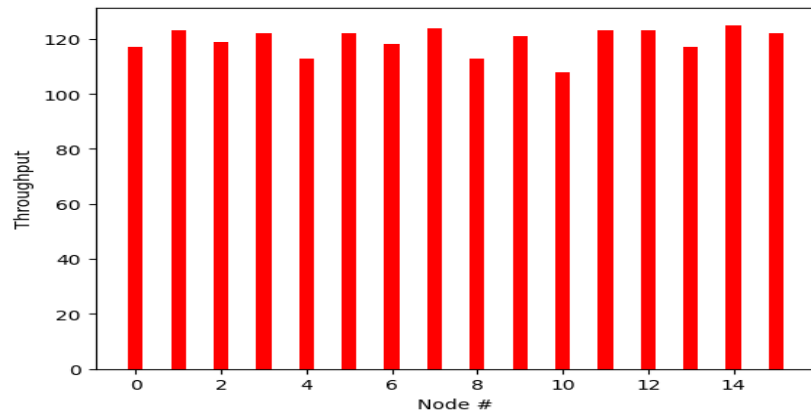
از اجرا، فیلم تهیه شده است.

خروجی: در  $t=5s$  تعداد بسته‌های دریافت شده: ۱۹۸۷

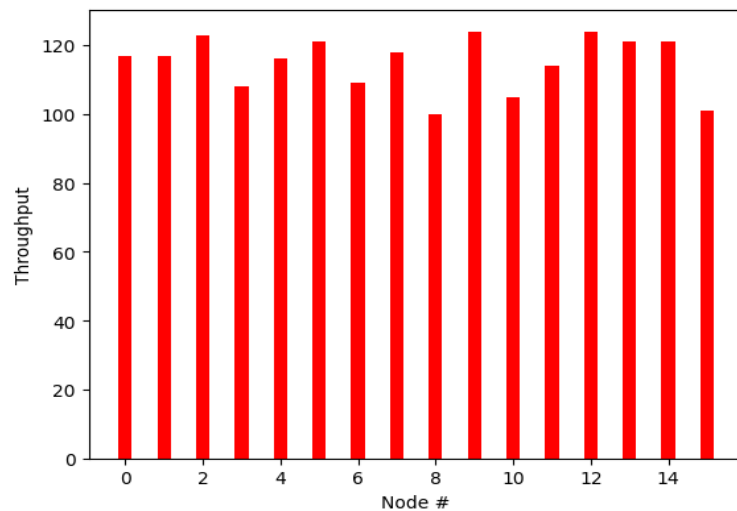
نتیجه: در مسئله بدون نود مخفی استفاده از **RTS/CTS** باعث افزایش جزئی تعداد بسته‌های دریافتی شده و بهره کانال افزایش جزئی می‌یابد.

نتیجه کلی: استفاده از **RTS/CTS** باعث افزایش بهره کانال شده و در سناریو نود مخفی باعث افزایش ۴۲ درصدی بهره کانال شده و در سناریو بدون نود مخفی باعث افزایش ۱ درصدی بهره کانال می‌شود.

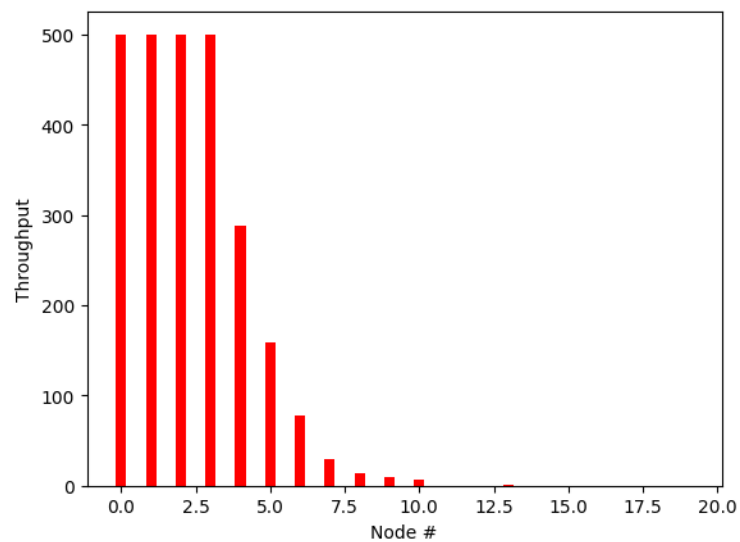
set nodes = 16, packet size = 1 slots, simtime = 2000



set nodes = 16, packet size = 7 slots, simtime = 14000

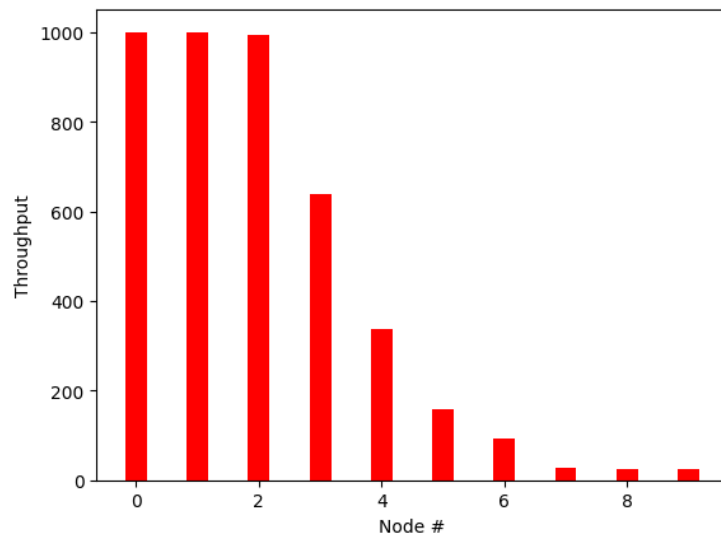


set nodes = 20, skewed load, packet size = 1 slots, simtime = 10000





set nodes = 10, skewed load, packet size = 1 slots, simtime = 10000

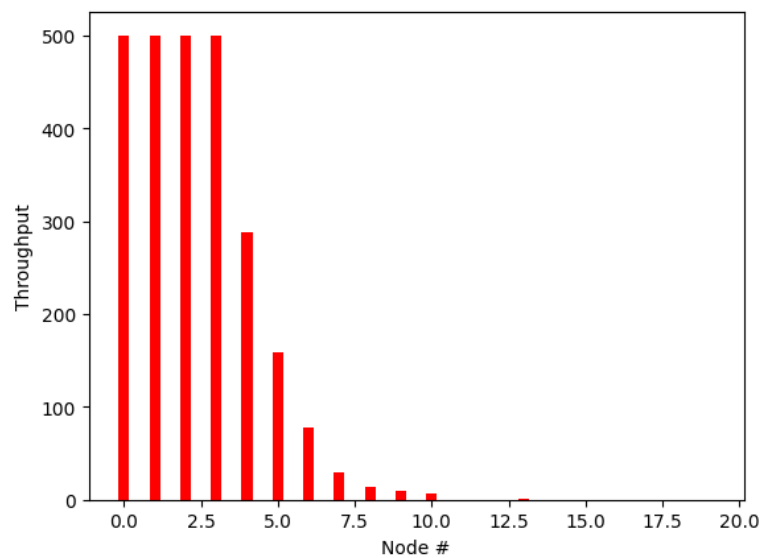


Time 10000 attempts 4294 success 4294 util 0.43

Inter-node fairness: 0.52

Inter-node weighted fairness: 0.87

set nodes = 20, skewed load, packet size = 1 slots, simtime = 10000

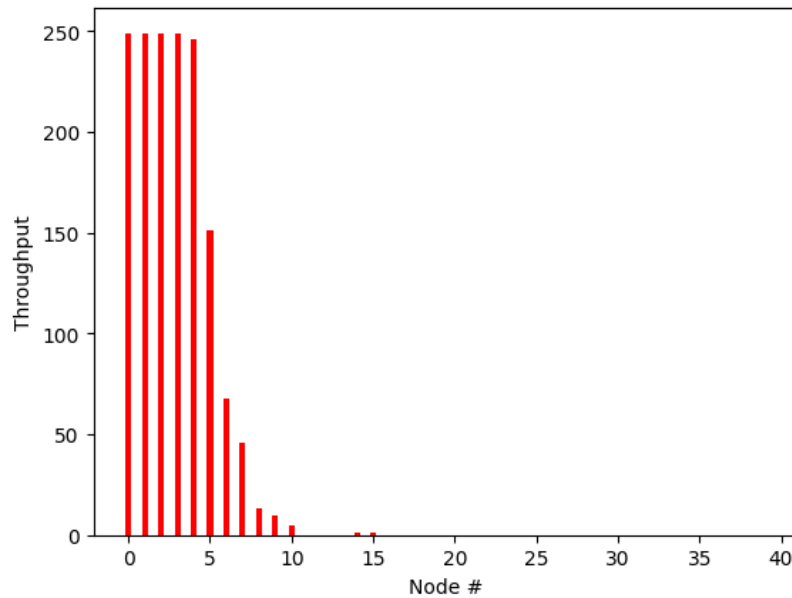


Time 10000 attempts 2582 success 2582 util 0.26

Inter-node fairness: 0.30

Inter-node weighted fairness: 0.48

set nodes = 40, skewed load, packet size = 1 slots, simtime = 10000

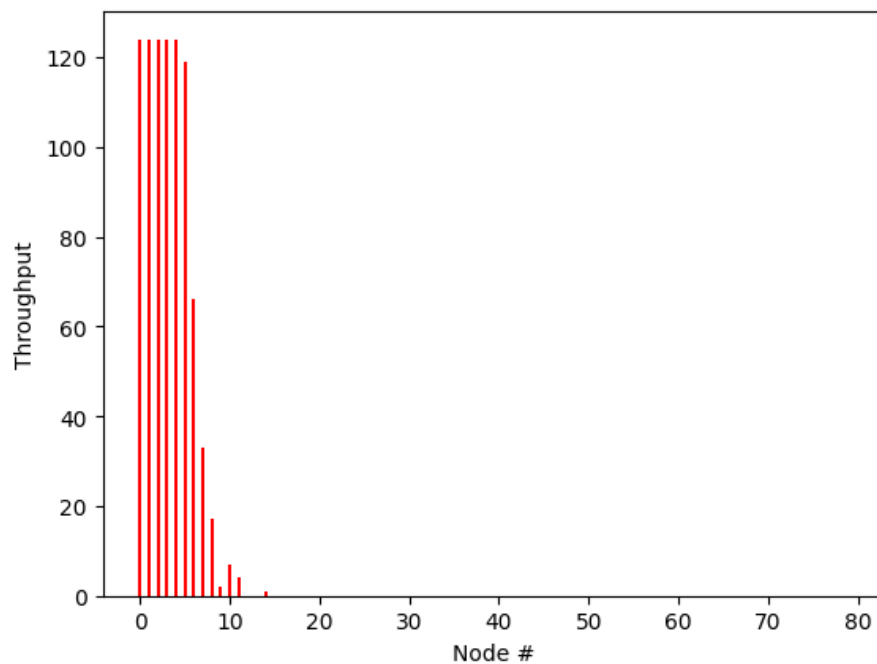


Time 10000 attempts 1537 success 1537 util 0.15

Inter-node fairness: 0.17

Inter-node weighted fairness: 0.12

set nodes = 80, skewed load, packet size = 1 slots, simtime = 10000



Time 10000 attempts 869 success 869 util 0.09

Inter-node fairness: 0.10

Inter-node weighted fairness: 0.08

در حالتی که skewed load داشته باشیم؛ با افزایش تعداد نودهای، در نودهای شماره بالا احتمال این در slot خود باری برای ارسال داشته باشند کمتر بوده و در آن slotها کانال خالی می ماند و هرچه نودها زیادتر باشند احتمال خالی ماندن کانال بیشتر شده و نوبت به نودهای شماره پایین که احتمال لود آنها بیشتر است دیرتر می رسد به همین دلیل بهره کانال با افزایش تعداد نود کاهش می یابد.

set nodes = 21, skewed load, packet size = 1 slots, simtime = 10000

Time 10000 attempts 2511 success 2510 util 0.25

set nodes = 22, skewed load, packet size = 1 slots, simtime = 10000

Time 10000 attempts 2423 success 2423 util 0.24

## Stabilizing Aloha

برای الگوریتم مقداردهی  $p$  برای این که اگر تعداد نودها بین ۸ تا ۱۶ باشد و بایستی fairness کمتر از ۰.۹ بوده و هرچه نزدیک تر به ۱ باشد بهتر است از به روش زیر عمل شده است.

```
def on_collision(self, packet):
    # for plots of collisions
    self.coll.append(self.network.time)

    self.p = max(self.p*0.9, self.pmin)

def on_xmit_success(self, packet):
    # for plots of successful transmissions
    self.sent.append(self.network.time)

    self.p = self.pmax
```

nodes= 8, pmin= 0 pmax= 1

Time 10000 attempts 13443 success 6151 util 0.62

Inter-node fairness: 0.72

Inter-node weighted fairness: 0.72

با مقادیر داده شده  $pmin=0$  یعنی بعد از collision مجدد سریعاً ارسال می کند و  $pmax=1$  یعنی احتمال باید زمان بیشتری صبر کند. این مقدار دهی باعث کاهش fairness می شود چون احتمال capture effect بیشتر شده (بعضی از نودها غالب شوند و نوبت به بقیه کمتر برسد)

با انتخاب  $pmax=0.35$  و packet size = 1

Time 10000 attempts 15484 success 3512 util 0.35

Inter-node fairness: 0.98

Inter-node weighted fairness: 0.98

با انتخاب  $p_{max}=0.017$  و  $packet\ size = 10$

Time 70000 attempts 2011 success 1232 util 0.18

Inter-node fairness: 0.91

Inter-node weighted fairness: 0.91

در هر اجرا اعداد متفاوت می باشند و در حالتی که سایز بسته ها ۱۰ باشد هر  $p_{max}$  کمتر باشد احتمال غالب شدن یک نود کمتر می شود ( این ها بیشتر مقدار fairness بود که گرفتیم.)

## CSMA

$p_{min} = 0$  and  $p_{max} = 1$  , size = 10, simtime= 100000 nodes= 8

Time 100000 attempts 11219 success 7596 util 0.76

Inter-node fairness: 0.86

Inter-node weighted fairness: 0.86

$p_{min} = 0$  and  $p_{max} = 0.79$  , size = 10, simtime= 100000 nodes= 8

Time 100000 attempts 11558 success 7563 util 0.76

Inter-node fairness: 0.95

Inter-node weighted fairness: 0.95

Time 100000 attempts 11360 success 7652 util 0.77

Inter-node fairness: 0.95

Inter-node weighted fairness: 0.95

## CSMA with contention windows

retry enabled, packet size = 10, simtime= 100000 nodes= 16, cwmin= 1, cwmax= 256

Time 100000 attempts 20072 success 3568 util 0.36

Inter-node fairness: 0.93

Inter-node weighted fairness: 0.93

fairness خوب است ولی بهره کانال بد است.

در اجرای اولیه بدون تنظیم‌های شمارنده انتظار، هر نود پشت‌سرهم یک فاصله زمانی تصادفی داخل پنجره مسابقه انتخاب می‌کرد و بررسی می‌کرد که آیا کانال در آن فاصله زمانی خالی است یا نه. اگر کانال خالی بود، نود سعی می‌کرد که اطلاعات را انتقال دهد.

با این حال، این رویکرد دارای یک نقص بود: چند نود ممکن بودند که همزمان یک واحد زمانی را انتخاب کنند، که منجر به collision می‌شد. زمانی که بیش از یک نود به انتظار می‌افتادند و منتظر می‌ماندند تا یک انتقال جاری به اتمام برسد، آن‌ها ممکن است که پس از اتمام انتقال جاری یک واحد زمانی را برای انتقال انتخاب کنند. این انتخاب همزمان منجر به افزایش collision و کاهش کلی بهره‌وری کانال می‌شد.

بعد از اصلاح کند و اجرای روش گفته شده

Time 100000 attempts 13857 success 6299 util 0.63

Inter-node fairness: 0.96

Inter-node weighted fairness: 0.96