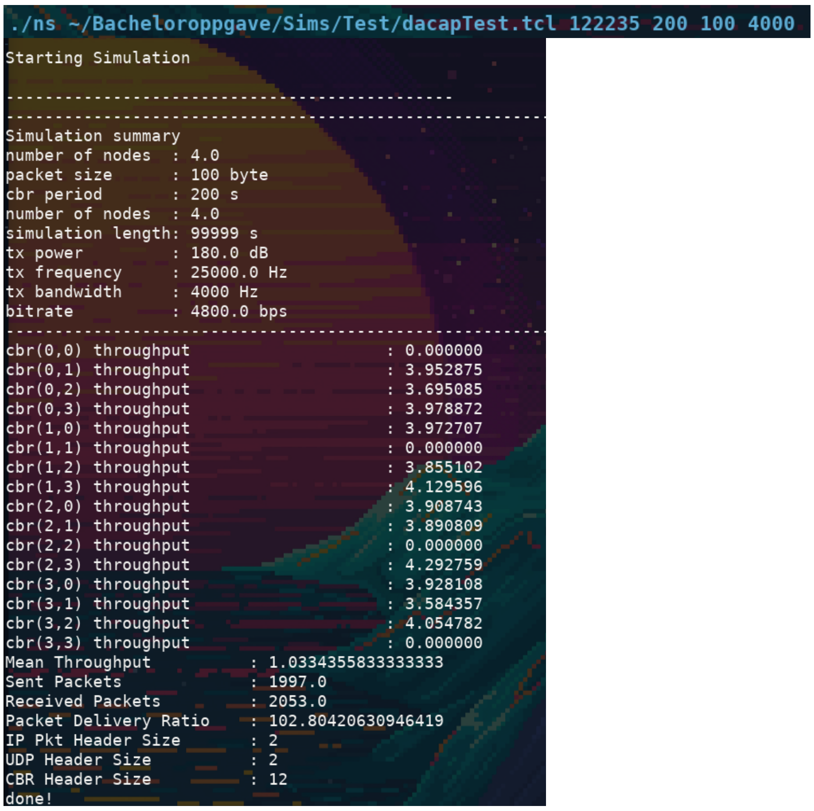


```
./ns ~/Bacheloroppgave/Sims/Test/dacapTest.tcl 122235 200 100 4000
Starting Simulation

---------------------------------------------
-------------------------------------------------------
Simulation summary
number of nodes  : 4.0
packet size      : 100 byte
cbr period       : 200 s
number of nodes  : 4.0
simulation length: 99999 s
tx power         : 180.0 dB
tx frequency     : 25000.0 Hz
tx bandwidth     : 4000 Hz
bitrate          : 4800.0 bps
-------------------------------------------------------
cbr(0,0) throughput                     : 0.000000
cbr(0,1) throughput                     : 3.952875
cbr(0,2) throughput                     : 3.695085
cbr(0,3) throughput                     : 3.978872
cbr(1,0) throughput                     : 3.972707
cbr(1,1) throughput                     : 0.000000
cbr(1,2) throughput                     : 3.855102
cbr(1,3) throughput                     : 4.129596
cbr(2,0) throughput                     : 3.908743
cbr(2,1) throughput                     : 3.890809
cbr(2,2) throughput                     : 0.000000
cbr(2,3) throughput                     : 4.292759
cbr(3,0) throughput                     : 3.928108
cbr(3,1) throughput                     : 3.584357
cbr(3,2) throughput                     : 4.054782
cbr(3,3) throughput                     : 0.000000
Mean Throughput          : 1.0334355833333333
Sent Packets             : 1997.0
Received Packets         : 2053.0
Packet Delivery Ratio    : 102.80420630946419
IP Pkt Header Size       : 2
UDP Header Size          : 2
CBR Header Size          : 12
done!
```

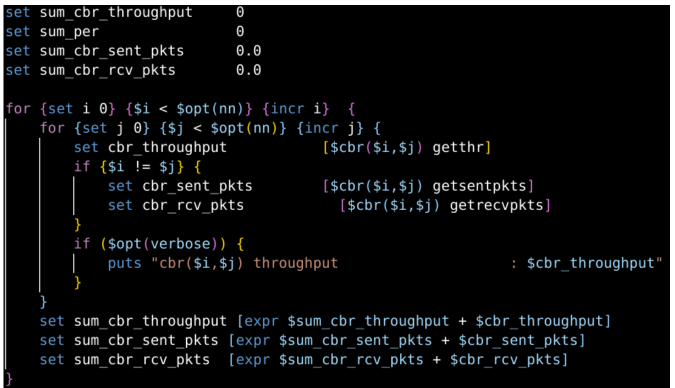
When running the sample DACAP simulation test it is possible to get a packet delivery ratio of over 100%
After looking thorugh the TCL-code it seems plausible that the reason for this is the way the number of sent and recieved packets are counted.
As we can see from the code below only [$cbr($i , opt(nn) getsentpkts] are added to the sum of sent packets.
(edge case when $i is equal to opt(nn) where [$cbr(opt(nn),opt(nn)-1] is added to the sum)
In other words: In a scenario of 4 nodes where all share the same channel, the sum of sent packets are calcuted as (packets sent from node0 to node3) + (packets sent from node1 to node3) + (packets sent from node2 to node3) + (packets sent from node3 to node2)

Number of recieved packets are calculated the same way.

This can be fixed by incrementing sum_cbr_sent_pkts for each iteration of the innermost for-loop, right after the "set cbr_sent_pkts [$cbr($i,$j) getsentpkts]".
In other words: Count the sum of sent packets for each node's connection to another node.

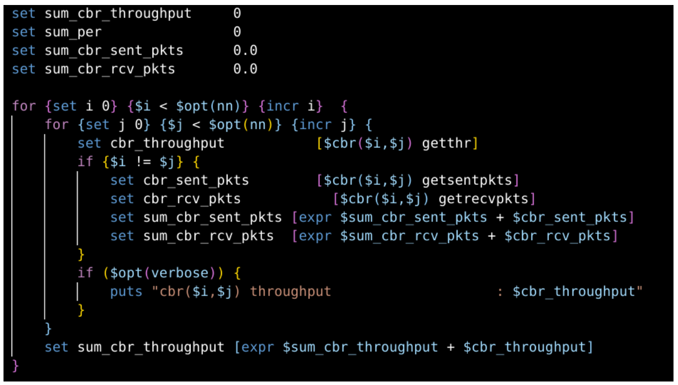
The sum of recieved packets can be fixed in the same way.

It is worth noting that this changes the number of sent and recieved packets to be n-1 times larger (n = number of nodes) as it counts the sum of sent and recieved packets for each node's connection to another node.



```
set sum_cbr_throughput     0
set sum_per                0
set sum_cbr_sent_pkts      0.0
set sum_cbr_rcv_pkts       0.0

for {set i 0} {$i < $opt(nn)} {incr i}  {
    for {set j 0} {$j < $opt(nn)} {incr j} {
        set cbr_throughput            [$cbr($i,$j) getthr]
        if {$i != $j} {
            set cbr_sent_pkts         [$cbr($i,$j) getsentpkts]
            set cbr_rcv_pkts            [$cbr($i,$j) getrecvpkts]
        }
        if ($opt(verbose)) {
            puts "cbr($i,$j) throughput                    : $cbr_throughput"
        }
    }
    set sum_cbr_throughput [expr $sum_cbr_throughput + $cbr_throughput]
    set sum_cbr_sent_pkts [expr $sum_cbr_sent_pkts + $cbr_sent_pkts]
    set sum_cbr_rcv_pkts  [expr $sum_cbr_rcv_pkts + $cbr_rcv_pkts]
}
```



```
set sum_cbr_throughput     0
set sum_per                0
set sum_cbr_sent_pkts      0.0
set sum_cbr_rcv_pkts       0.0

for {set i 0} {$i < $opt(nn)} {incr i}  {
    for {set j 0} {$j < $opt(nn)} {incr j} {
        set cbr_throughput            [$cbr($i,$j) getthr]
        if {$i != $j} {
            set cbr_sent_pkts         [$cbr($i,$j) getsentpkts]
            set cbr_rcv_pkts            [$cbr($i,$j) getrecvpkts]
            set sum_cbr_sent_pkts [expr $sum_cbr_sent_pkts + $cbr_sent_pkts]
            set sum_cbr_rcv_pkts  [expr $sum_cbr_rcv_pkts + $cbr_rcv_pkts]
        }
        if ($opt(verbose)) {
            puts "cbr($i,$j) throughput                    : $cbr_throughput"
        }
    }
    set sum_cbr_throughput [expr $sum_cbr_throughput + $cbr_throughput]
}
```