## **IEEE 802.11 CSMA/CA study with ns3**

1. Download the ns3 code file lab05-wifi-2hidden-stns.cc. This file models 3 stations n0, n1 and n2, in a configuration such n1-n0 can hear each other, n2-n0 can hear each other, but n1 - n2 cannot. Thus n1, n2 are *hidden terminals w.r.t. each other.*

Study the code - it is well-commented, make sure you understand every line. The basic concept of an abstract helper object which gets installed on some concrete object representing an actual networking entity is the same as you studied in the previous lab. Move the file to scratch directory and run it as you learnt in the previous lab. Now answer the following questions (download this as odt, and answer by making space below the question). Wherever applicable answer all questions for with and without RTS/CTS.

* 1. What is the mechanism used in the code to achieve the topology of ‘hidden-ness’ vs reachability of the nodes? **First a default large loss of 200dB is set. This would disconnect all links. Then the loss for n0 -- n1 and n0 -- n2 link is set to 50dB. Thus, only these two links are live i.e n0 -- n1 can communicate, n0 -- n2 can communicate**
  2. Which particular one of the 802.11 family of protocols is selected in this simulation (for PHY layer)? **802.11a**
  3. What is the data rate of the channel? **54Mbps**

d.e. What is the flow configuration? (Who’s sending what to whom?) What are the flow parameters?

* + 1. **n1 → n0 CBR via UDP 2200Bpayload 10Mbps startTime 1.01s**
    2. **n2 → n0 CBR via UDP 2200Bpayload 10Mbps startTime 1.02s**

**At t=0.001s one packet of size 10B is sent as n1 → n0**

**At t=0.002s one packet of size 10B is sent as n2 → n0**

**The single packets have to be sent to workaround lack of perfect ARP**

For the above parameters, what’s the per node throughput? What’s the total channel throughput?

**Without RTS/CTS n1 → n0 : 9.4289**

**n2 → n0 : 8.91709**

**Total channel: 18.346**

**With RTS/CTS n1 → n0 : 10.1113**

**n2 → n0 : 10.096**

**Total channel: 20.2073**

Change the values so that the total data rate offered to the channel is about 10% of the channel data rate, equally divided among all sources. Keep increasing it to 20%, 30% … 90%. What’s the maximum throughput achieved for each value? What is the trend of the throughput vs offered load? Tabulate/plot the values and paste here.(For these experiments you may want to change the code to take values as input, not hardcoded, else recompiling will take time).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | input load per node (Mbps) | n1 tput | n2 tput | Total tput |
| w/o CTS/RTS | 2.7 | 2.72962 | 2.72453 | 5.45414 |
| with CTS/RTS | 2.72962 | 2.72453 | 5.45414 |
| w/o CTS/RTS | 5.4 | 5.37012 | 5.33956 | 10.7097 |
| with CTS/RTS | 5.45924 | 5.4516 | 10.9108 |
| w/o CTS/RTS | 8.1 | 8.18885 | 8.17867 | 16.3675 |
| with CTS/RTS | 8.18885 | 8.17867 | 16.3675 |
| w/o CTS/RTS | 10.8 | 8.78978 | 9.57913 | 18.3689 |
| with CTS/RTS | 10.5136 | 10.8854 | 21.399 |
| w/o CTS/RTS | 13.5 | 9.55112 | 8.94256 | 18.4937 |
| with CTS/RTS | 10.6027 | 11.4252 | 22.0279 |
| w/o CTS/RTS | 16.2 | 9.33723 | 9.28121 | 18.6184 |
| with CTS/RTS | 11.3157 | 10.7173 | 22.033 |
| w/o CTS/RTS | 18.9 | 9.05205 | 9.52056 | 18.5726 |
| with CTS/RTS | 10.9236 | 11.2291 | 22.1527 |
| w/o CTS/RTS | 21.6 | 8.94001 | 9.5995 | 18.5395 |
| with CTS/RTS | 13.4597 | 8.70066 | 22.1603 |
| w/o CTS/RTS | 24.3 | 9.4238 | 9.01385 | 18.4377 |
| with CTS/RTS | 7.84765 | 14.3483 | 22.196 |
| w/o CTS/RTS | 27 | 9.69626 | 8.76941 | 18.4657 |
| with CTS/RTS | 9.01131 | 13.1668 | 22.1781 |
| **MAX** | w/o CTS/RTS | **9.69626** | **9.5995** | **18.6184** |
| with CTS/RTS | **13.4597** | **14.3483** | **22.196** |

**All the throughputs roughly increase with input load upto a certain point and then start decreasing with further increase in load (decrease is due to increase in traffic causing interference and packet losses)**

Q) Find out the maximum throughput possible when there is no contention (only one source).

|  |  |  |
| --- | --- | --- |
|  | input load per node (Mbps) | n1 tput = total tput |
| w/o CTS/RTS | 32.4 | 25.6156 |
| with CTS/RTS | 22.611 |
| w/o CTS/RTS | 43.2 | 25.72 |
| with CTS/RTS | 22.5448 |
| w/o CTS/RTS | 54 | 25.7277 |
| with CTS/RTS | 22.5575 |
| w/o CTS/RTS | 64.8 | 25.6029 |
| with CTS/RTS | 22.5372 |
| w/o CTS/RTS | 75.6 | 25.6666 |
| with CTS/RTS | 22.5754 |
| w/o CTS/RTS | 86.4 | 25.6615 |
| with CTS/RTS | 22.555 |
| w/o CTS/RTS | 97.2 | 25.5876 |
| with CTS/RTS | 22.5474 |
| w/o CTS/RTS | 108 | 25.6538 |
| with CTS/RTS | 22.5703 |
| **MAX** | w/o CTS/RTS | **25.7277** |
| with CTS/RTS | **22.611** |

**Note that the values are greater than the tput values obtained with contention**

Q) Make any observations of the numbers with and without RTS/CTS

**When there is no contention, not using RTS/CTS gives higher tput**

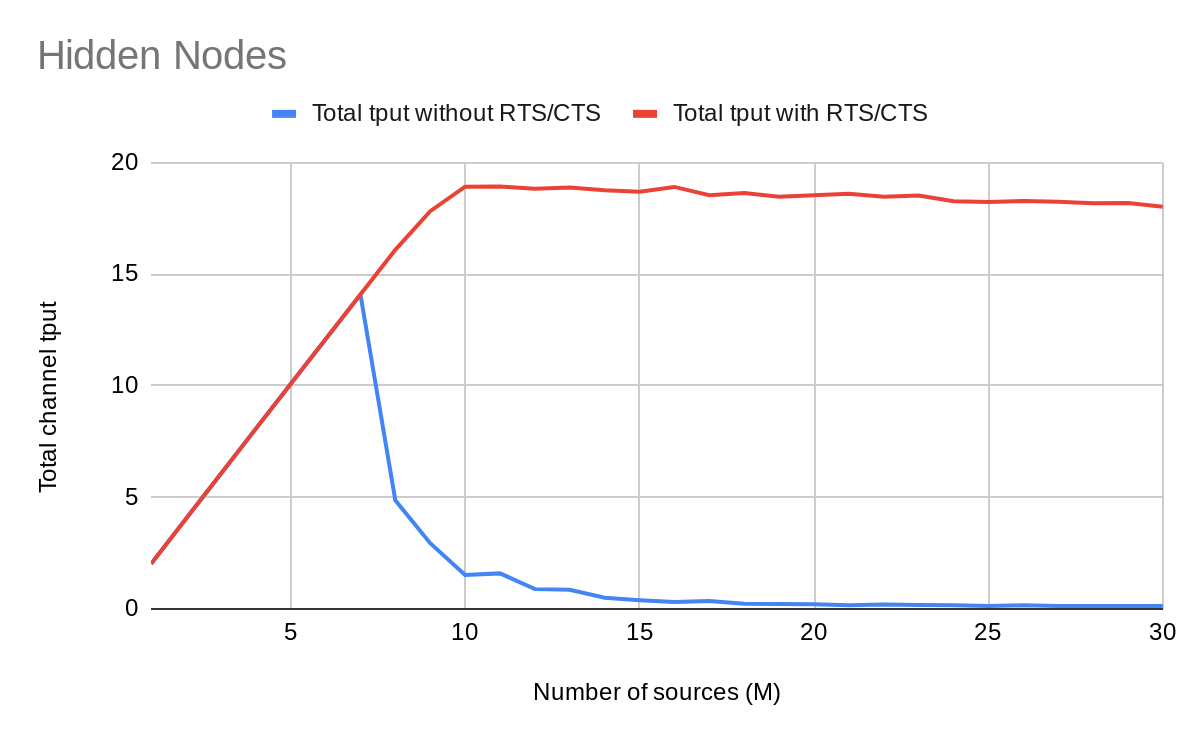
**WIth contention, using RTS/CTS (if nodes hidden) gives better tput values for almost all offered loads.**

Q) RTS/CTS was designed mainly to solve the hidden terminal problem. So perhaps it is not very useful when there aren’t hidden terminals? Modify the code and design and run experiments to validate/invalidate this hypothesis. Paste all results and conclusions below:

**Same table when n1 n2 can see each other**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | input load per node (Mbps) | n1 tput | n2 tput | Total tput |
| w/o CTS/RTS | 2.7 | 2.72962 | 2.72453 | 5.45414 |
| with CTS/RTS | 2.72962 | 2.72453 | 5.45414 |
| w/o CTS/RTS | 5.4 | 5.45924 | 5.4516 | 10.9108 |
| with CTS/RTS | 5.45924 | 5.4516 | 10.9108 |
| w/o CTS/RTS | 8.1 | 8.18885 | 8.17867 | 16.3675 |
| with CTS/RTS | 8.18885 | 8.17867 | 16.3675 |
| w/o CTS/RTS | 10.8 | 10.921 | 10.9032 | 21.8242 |
| with CTS/RTS | 10.9185 | 10.9032 | 21.8217 |
| w/o CTS/RTS | 13.5 | 13.5488 | 13.3655 | 26.9142 |
| with CTS/RTS | 12.6092 | 12.4997 | 25.1089 |
| w/o CTS/RTS | 16.2 | 12.9453 | 14.165 | 27.1103 |
| with CTS/RTS | 12.5277 | 12.7824 | 25.3101 |
| w/o CTS/RTS | 18.9 | 13.5361 | 13.7499 | 27.286 |
| with CTS/RTS | 12.5455 | 12.7009 | 25.2464 |
| w/o CTS/RTS | 21.6 | 13.8238 | 13.2407 | 27.0645 |
| with CTS/RTS | 12.6449 | 12.5175 | 25.1624 |
| w/o CTS/RTS | 24.3 | 13.4062 | 13.4367 | 26.8429 |
| with CTS/RTS | 12.6958 | 12.5506 | 25.2464 |
| w/o CTS/RTS | 27 | 13.6634 | 13.755 | 27.4184 |
| with CTS/RTS | 12.8486 | 12.4412 | 25.2897 |
| **MAX** | w/o CTS/RTS | **13.8238** | **14.165** | **27.4184** |
| with CTS/RTS | **12.8486** | **12.7824** | **25.3101** |

**It can be seen that for low offered tput, using CTS/RTS does not affect observed tputs. However, as offered load increases, not using CTS/RTS is beneficial. Thus, the above values validate the hypothesis. CTS/RTS is effectively unnecessary when the “hidden node problem” does not arise.**

1. Modify the code to model a given *M* number of source nodes in the configuration that n1,....nM are all sending data to n0. Let all of n1...nM be hidden from each other. Each one can be just like the source nodes n1,n2 in the given file. Study the following - again, with and without RTS/CTS, and in fact wherever applicable study what effect RTS/CTS has on the throughput.
   1. Total channel throughput as a function of increasing number of nodes. Start with each node bring a low offered rate (e.g. only 2Mbps).Paste the table and plot here.

* 1. Now make all n1 to nM such that they are no longer hidden - all are audible to each other. Now do the same analysis, and compare with when hidden and RTS/CTS effect on the throughput in both case.