Zewail City for science and technology
COMPUTER NETWORKS CIE447
SPRING 2021

Reliable Transport Protocol:

Implementation of GBN on top of UDP

DATE
Jun 09th, 2021

Fatma NourEl-Din
ID: 201700346

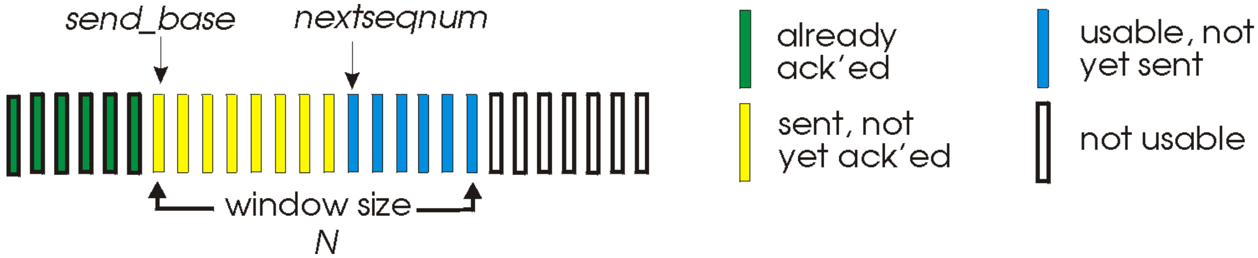
Ahmad Muhammad
ID: 201700259

Ahmed Abdelhafez
ID: 201700448

I. INTRODUCTION:

Reliable data transfer protocol is a collection of standard algorithms and methods to ensure the reliability and integrity of data transfer across the network. One of the protocols used for reliable data transfer is the Go-Back-N (GBN) protocol.

In GBN, the sender is allowed to transmit N number of unacknowledged packets. This N number is called the window size.



The window size controls the number of consecutive packets that are allowed to be transmitted without acknowledgement. An extra two pointers are used to complete the protocol. The first pointer is called “send_base” and the second is called “nextseqnum”. Send_base is defined as the smallest/oldest unacknowledged packet, whereas nextseqnum is defined as the sequence number of the next packet to be transmitted.

1. Screenshots verifying that the test files are received correctly.

II. Files reception verification: #1

Test file 1:

ReceivedText - Notepad

File Edit Format View Help

ALICE'S ADVENTURES IN WONDERLAND

Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

CHAPTER I

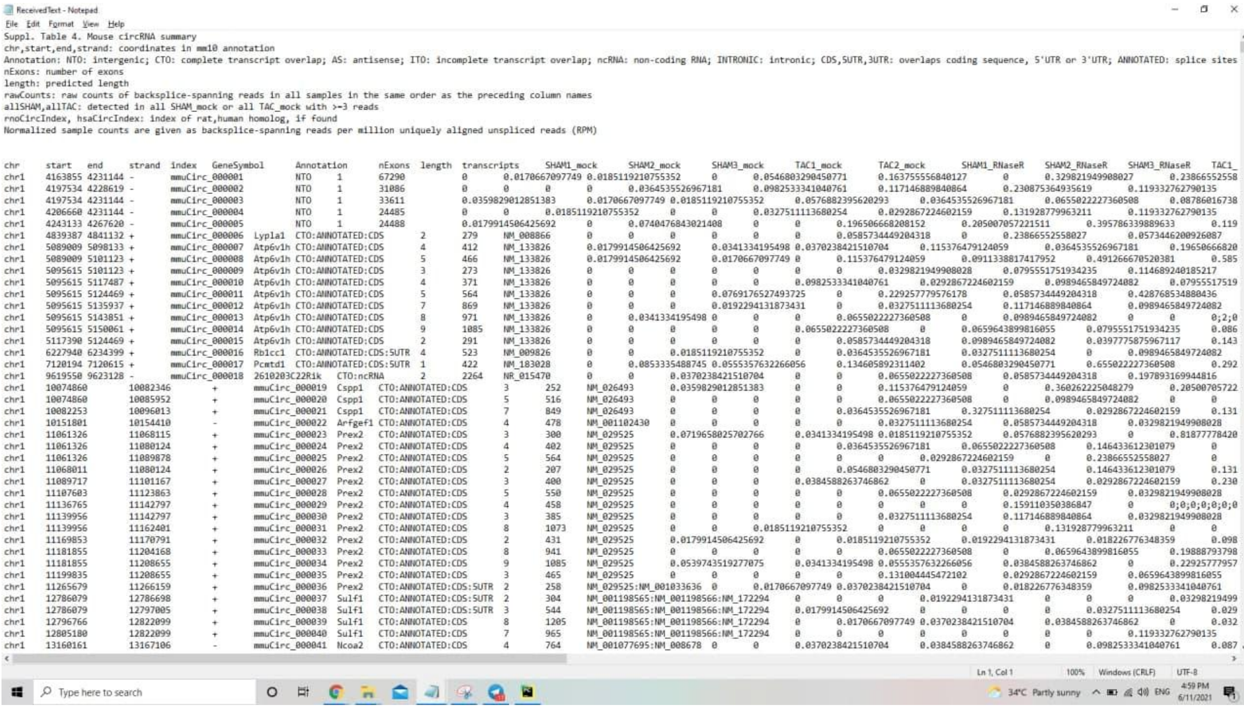
Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, `and what is the use of a book,' thought Alice `without pictures or conversation?'

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

There was nothing so VERY remarkable in that; nor did Alice

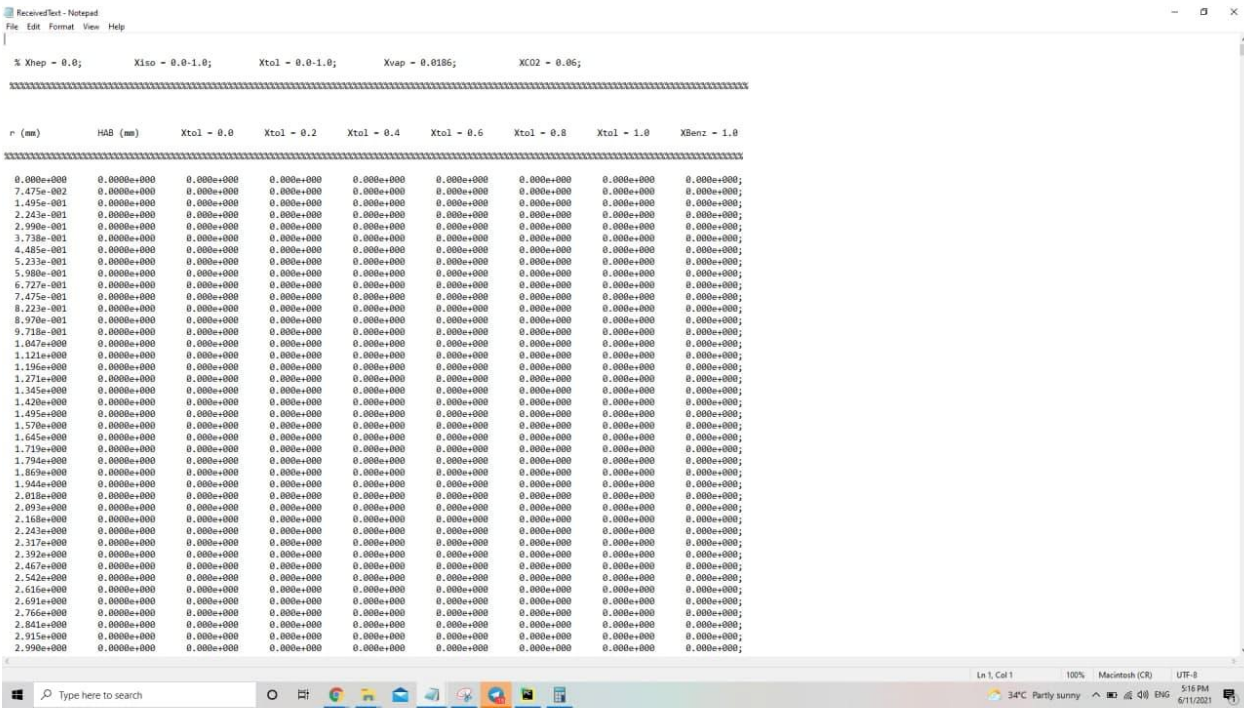
Test file 2:



```
Suppl. Table 4. Mouse circRNA summary
chr,start,end,strand: coordinates in mm10 annotation
Annotation: NTO: intergenic; CTO: complete transcript overlap; AS: antisense; ITO: incomplete transcript overlap; ncRNA: non-coding RNA; INTRONIC: intronic; CDS,5UTR,3UTR: overlaps coding sequence, 5'UTR or 3'UTR; ANNOTATED: splice sites
nExons: number of exons
length: predicted length
rawCounts: raw counts of backsplice-spanning reads in all samples in the same order as the preceding column names
allSHAM,allTAC: detected in all SHAM_mock or all TAC_mock with >=3 reads
rnoCircIndex, hsaCircIndex: index of rat,human homolog, if found
Normalized sample counts are given as backsplice-spanning reads per million uniquely aligned unspliced reads (RPM)


chr	start	end	strand	index	GeneSymbol	Annotation	nExons	length	transcripts	SHAM1_mock	SHAM2_mock	SHAM3_mock	TAC1_mock	TAC2_mock	SHAM1_RNaseR	SHAM2_RNaseR	SHAM3_RNaseR	TAC1_
chr1	4163855	4231144	-	mmuCirc_000001		NTO	1	67290		0	0.0170667097749	0.0185119210755352	0	0.0546803290450771	0.163755556840127	0	0.329821949908027	0.23866552558
chr1	4197534	4228619	-	mmuCirc_000002		NTO	1	31086		0	0	0	0	0.0364535526967181	0.0982533341040761	0.117146889840864	0.230875364935619	0.119332762790135
chr1	4197534	4231144	-	mmuCirc_000003		NTO	1	33611		0.0359829012851383	0.0170667097749	0.0185119210755352	0.0576882395620293	0.0364535526967181	0.0655022227360508	0.08786016738
chr1	4206660	4231144	-	mmuCirc_000004		NTO	1	24485		0	0	0.0185119210755352	0	0	0.0327511113680254	0.0292867224602159	0.131928779963211	0.119332762790135
chr1	4243133	4267620	-	mmuCirc_000005		NTO	1	24488		0.0179914506425692	0	0.0740476843021408	0	0	0.196506668208152	0.205007057221511	0.395786339889633	0.119
chr1	4839387	4841132	+	mmuCirc_000006	Lypla1	CTO:ANNOTATED:CDS	2	279	NM_008866	0	0	0	0	0	0	0.0585734449204318	0	0.23866552558027	0.0573446200926087
chr1	5089009	5098133	+	mmuCirc_000007	Atp6v1h	CTO:ANNOTATED:CDS	4	412	NM_133826	0.0179914506425692	0.0341334195498	0.0370238421510704	0.115376479124059	0.0364535526967181	0.19650666820
chr1	5089009	5101123	+	mmuCirc_000008	Atp6v1h	CTO:ANNOTATED:CDS	5	466	NM_133826	0.0179914506425692	0.0170667097749	0	0.115376479124059	0.0911338817417952	0.491266670520381	0.585
chr1	5095615	5101123	+	mmuCirc_000009	Atp6v1h	CTO:ANNOTATED:CDS	3	273	NM_133826	0	0	0	0	0	0	0.0329821949908028	0.0795551751934235	0.114689240185217
chr1	5095615	5117487	+	mmuCirc_000010	Atp6v1h	CTO:ANNOTATED:CDS	4	371	NM_133826	0	0	0	0	0	0.0982533341040761	0.0292867224602159	0.0989465849724082	0.07955517519
chr1	5095615	5124469	+	mmuCirc_000011	Atp6v1h	CTO:ANNOTATED:CDS	5	564	NM_133826	0	0	0	0.0769176527493725	0	0.229257779576178	0.0585734449204318	0.428768534880436
chr1	5095615	5135937	+	mmuCirc_000012	Atp6v1h	CTO:ANNOTATED:CDS	7	869	NM_133826	0	0	0	0.0192294131873431	0	0.0327511113680254	0.117146889840864	0.0989465849724082
chr1	5095615	5143851	+	mmuCirc_000013	Atp6v1h	CTO:ANNOTATED:CDS	8	971	NM_133826	0	0.0341334195498	0	0	0	0.0655022227360508	0	0.0989465849724082	0	0	0;2;0
chr1	5095615	5150061	+	mmuCirc_000014	Atp6v1h	CTO:ANNOTATED:CDS	9	1085	NM_133826	0	0	0	0	0	0.0655022227360508	0	0.0659643899816055	0.0795551751934235	0.086
chr1	5117390	5124469	+	mmuCirc_000015	Atp6v1h	CTO:ANNOTATED:CDS	2	291	NM_133826	0	0	0	0	0	0	0.0585734449204318	0.0989465849724082	0.0397775875967117	0.143
chr1	6227940	6234399	+	mmuCirc_000016	Rb1cc1	CTO:ANNOTATED:CDS:5UTR	4	523	NM_009826	0	0	0.0185119210755352	0	0.0364535526967181	0.0327511113680254	0	0.0989465849724082
chr1	7120194	7120615	+	mmuCirc_000017	Pcmtd1	CTO:ANNOTATED:CDS:5UTR	1	422	NM_183028	0	0.0853335488745	0.0555357632266056	0.134605892311402	0.0546803290450771	0.655022227360508	0.292
chr1	9619550	9623128	-	mmuCirc_000018	2610203C22Rik	CTO:ncRNA	2	2264	NR_015470	0	0	0.0370238421510704	0	0	0.0655022227360508	0.0585734449204318	0.197893169944816
chr1	10074860	10082346	+	mmuCirc_000019	Cspp1	CTO:ANNOTATED:CDS	3	252	NM_026493	0.0359829012851383	0	0	0.115376479124059	0	0.360262225048279	0.20500705722
chr1	10074860	10085952	+	mmuCirc_000020	Cspp1	CTO:ANNOTATED:CDS	5	516	NM_026493	0	0	0	0	0	0.0655022227360508	0	0.0989465849724082	0	0
chr1	10082253	10096013	+	mmuCirc_000021	Cspp1	CTO:ANNOTATED:CDS	7	849	NM_026493	0	0	0	0	0.0364535526967181	0.327511113680254	0.0292867224602159	0.131
chr1	10151801	10154410	-	mmuCirc_000022	Arfgef1	CTO:ANNOTATED:CDS	4	478	NM_001102430	0	0	0	0	0	0.0327511113680254	0.0585734449204318	0.0329821949908028
chr1	11061326	11068115	+	mmuCirc_000023	Prex2	CTO:ANNOTATED:CDS	3	300	NM_029525	0.0719658025702766	0.0341334195498	0.0185119210755352	0.0576882395620293	0	0.81877778420
chr1	11061326	11080124	+	mmuCirc_000024	Prex2	CTO:ANNOTATED:CDS	4	402	NM_029525	0	0	0	0	0.0364535526967181	0.0655022227360508	0.146433612301079	0
chr1	11061326	11089878	+	mmuCirc_000025	Prex2	CTO:ANNOTATED:CDS	5	564	NM_029525	0	0	0	0	0	0	0.0292867224602159	0	0.23866552558027	0
chr1	11068011	11080124	+	mmuCirc_000026	Prex2	CTO:ANNOTATED:CDS	2	207	NM_029525	0	0	0	0	0.0546803290450771	0.0327511113680254	0.146433612301079	0.131
chr1	11089717	11101167	+	mmuCirc_000027	Prex2	CTO:ANNOTATED:CDS	3	400	NM_029525	0	0	0	0.0384588263746862	0	0.0327511113680254	0.0292867224602159	0.230
chr1	11107603	11123863	+	mmuCirc_000028	Prex2	CTO:ANNOTATED:CDS	5	550	NM_029525	0	0	0	0	0	0.0655022227360508	0.0292867224602159	0.0329821949908028
chr1	11136765	11142797	+	mmuCirc_000029	Prex2	CTO:ANNOTATED:CDS	4	458	NM_029525	0	0	0	0	0	0	0	0	0.159110350386847	0	0;0;0;0;0;0
chr1	11139956	11142797	+	mmuCirc_000030	Prex2	CTO:ANNOTATED:CDS	3	385	NM_029525	0	0	0	0	0	0.0327511113680254	0.117146889840864	0.0329821949908028
chr1	11139956	11162401	+	mmuCirc_000031	Prex2	CTO:ANNOTATED:CDS	8	1073	NM_029525	0	0	0.0185119210755352	0	0	0	0	0.131928779963211	0	0
chr1	11169853	11170791	+	mmuCirc_000032	Prex2	CTO:ANNOTATED:CDS	2	431	NM_029525	0.0179914506425692	0	0.0185119210755352	0.0192294131873431	0.018226776348359	0.098
chr1	11181855	11204168	+	mmuCirc_000033	Prex2	CTO:ANNOTATED:CDS	8	941	NM_029525	0	0	0	0	0	0.0655022227360508	0	0.0659643899816055	0.19888793798
chr1	11181855	11208655	+	mmuCirc_000034	Prex2	CTO:ANNOTATED:CDS	9	1085	NM_029525	0.0539743519277075	0.0341334195498	0.0555357632266056	0.0384588263746862	0	0.22925777957
chr1	11199835	11208655	+	mmuCirc_000035	Prex2	CTO:ANNOTATED:CDS	3	465	NM_029525	0	0	0	0	0	0.131004445472102	0.0292867224602159	0.0659643899816055
chr1	11265679	11266159	+	mmuCirc_000036	Prex2	CTO:ANNOTATED:CDS:5UTR	2	258	NM_029525:NM_001033636	0	0.0170667097749	0.0370238421510704	0	0.018226776348359	0.0982533341040761
chr1	12786079	12786698	+	mmuCirc_000037	Sulf1	CTO:ANNOTATED:CDS:5UTR	2	304	NM_001198565:NM_001198566:NM_172294	0	0	0	0.0192294131873431	0	0	0	0.03298219499
chr1	12786079	12797005	+	mmuCirc_000038	Sulf1	CTO:ANNOTATED:CDS:5UTR	3	544	NM_001198565:NM_001198566:NM_172294	0.0179914506425692	0	0	0	0	0.0327511113680254	0.029
chr1	12796766	12822099	+	mmuCirc_000039	Sulf1	CTO:ANNOTATED:CDS	8	1205	NM_001198565:NM_001198566:NM_172294	0	0.0170667097749	0.0370238421510704	0.0384588263746862	0	0.032
chr1	12805180	12822099	+	mmuCirc_000040	Sulf1	CTO:ANNOTATED:CDS	7	965	NM_001198565:NM_001198566:NM_172294	0	0	0	0	0	0	0	0.119332762790135
chr1	13160161	13167106	-	mmuCirc_000041	Ncoa2	CTO:ANNOTATED:CDS	4	764	NM_001077695:NM_008678	0	0	0.0370238421510704	0.0384588263746862	0	0.0982533341040761	0.087
```

Test file 3:



ReceivedText - Notepad

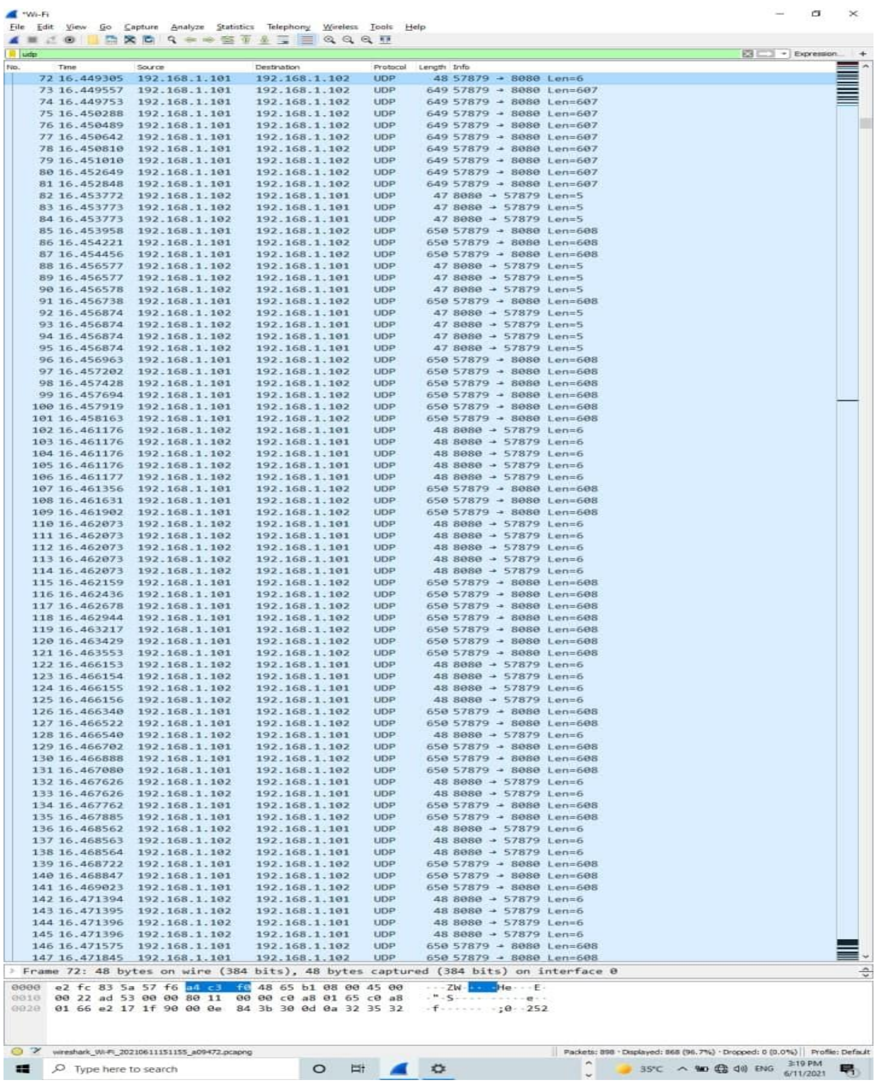
% Xhep = 0.0; Xiso = 0.0-1.0; Xtol = 0.0-1.0; Xvap = 0.0186; XCO2 = 0.06;

%%

r (mm)	HAB (mm)	Xtol = 0.0	Xtol = 0.2	Xtol = 0.4	Xtol = 0.6	Xtol = 0.8	Xtol = 1.0	XBenz = 1.0
0.000e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
7.475e-002	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.495e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.243e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.990e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
3.738e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
4.485e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
5.233e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
5.980e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
6.727e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
7.475e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
8.223e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
8.970e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
9.718e-001	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.047e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.121e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.196e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.271e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.345e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.420e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.495e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.570e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.645e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.719e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.794e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.869e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
1.944e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.018e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.093e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.168e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.243e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.317e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.392e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.467e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.542e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.616e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.691e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.766e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.841e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.915e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;
2.990e+000	0.0000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000	0.000e+000;

III. Wireshark: #2

1- a screenshot of the wireshark showing the transmission window


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Destination Protocol Length Info
Frame 72: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface 0
Type here to search

2- packet 0

```
72 16.449305  192.168.1.101   192.168.1.102   UDP   48 57879 → 8080 Len=6
> Frame 72: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface 0
0000  e2 fc 83 5a 57 f6 a4 c3  f0 48 65 b1 08 00 45 00   ···ZW··· ·He···E·
0010  00 22 ad 53 00 00 80 11  00 00 c0 a8 01 65 c0 a8   ·"·S···· ·····e··
0020  01 66 e2 17 1f 90 00 0e  84 3b 30 0d 0a 32 35 32   ·f······ ·;0··252
```

3- random packet (12)

```
87 16.454456  192.168.1.101   192.168.1.102   UDP   650 57879 → 8080 Len=608
> Frame 87: 650 bytes on wire (5200 bits), 650 bytes captured (5200 bits) on interface 0
0000  e2 fc 83 5a 57 f6 a4 c3  f0 48 65 b1 08 00 45 00   ···ZW··· ·He···E·
0010  02 7c ad 5f 00 00 80 11  00 00 c0 a8 01 65 c0 a8   ·|·_···· ·····e··
0020  01 66 e2 17 1f 90 02 68  86 95 31 32 0d 0a 62 65   ·f·····h ··12··be
0030  66 6f 72 65 2c 20 61 6e  64 20 62 65 68 69 6e 64   fore, an d behind
0040  20 69 74 20 77 61 73 20  61 20 6c 69 74 74 6c 65    it was  a little
0050  0d 0a 64 6f 6f 72 20 61  62 6f 75 74 20 66 69 66   ··door a bout fif
0060  74 65 65 6e 20 69 6e 63  68 65 73 20 68 69 67 68   teen inc hes high
0070  3a 20 20 73 68 65 20 74  72 69 65 64 20 74 68 65   :  she t ried the
0080  20 6c 69 74 74 6c 65 20  67 6f 6c 64 65 6e 20 6b    little  golden k
0090  65 79 0d 0a 69 6e 20 74  68 65 20 6c 6f 63 6b 2c   ey··in t he lock,
00a0  20 61 6e 64 20 74 6f 20  68 65 72 20 67 72 65 61    and to  her grea
00b0  74 20 64 65 6c 69 67 68  74 20 69 74 20 66 69 74   t deligh t it fit
00c0  74 65 64 21 0d 0a 0d 0a  20 20 41 6c 69 63 65 20   ted!····   Alice 
00d0  6f 70 65 6e 65 64 20 74  68 65 20 64 6f 6f 72 20   opened t he door 
00e0  61 6e 64 20 66 6f 75 6e  64 20 74 68 61 74 20 69   and foun d that i
00f0  74 20 6c 65 64 20 69 6e  74 6f 20 61 20 73 6d 61   t led in to a sma
0100  6c 6c 0d 0a 70 61 73 73  61 67 65 2c 20 6e 6f 74   ll··pass age, not
0110  20 6d 75 63 68 20 6c 61  72 67 65 72 20 74 68 61    much la rger tha
0120  6e 20 61 20 72 61 74 2d  68 6f 6c 65 3a 20 20 73   n a rat- hole:  s
0130  68 65 20 6b 6e 65 6c 74  20 64 6f 77 6e 20 61 6e   he knelt  down an
0140  64 0d 0a 6c 6f 6f 6b 65  64 20 61 6c 6f 6e 67 20   d··looke d along 
0150  74 68 65 20 70 61 73 73  61 67 65 20 69 6e 74 6f   the pass age into
0160  20 74 68 65 20 6c 6f 76  65 6c 69 65 73 74 20 67    the lov eliest g
0170  61 72 64 65 6e 20 79 6f  75 20 65 76 65 72 20 73   arden yo u ever s
0180  61 77 2e 0d 0a 48 6f 77  20 73 68 65 20 6c 6f 6e   aw.··How  she lon
0190  67 65 64 20 74 6f 20 67  65 74 20 6f 75 74 20 6f   ged to g et out o
01a0  66 20 74 68 61 74 20 64  61 72 6b 20 68 61 6c 6c   f that d ark hall
01b0  2c 20 61 6e 64 20 77 61  6e 64 65 72 20 61 62 6f   , and wa nder abo
01c0  75 74 0d 0a 61 6d 6f 6e  67 20 74 68 6f 73 65 20   ut··amon g those 
01d0  62 65 64 73 20 6f 66 20  62 72 69 67 68 74 20 66   beds of  bright f
01e0  6c 6f 77 65 72 73 20 61  6e 64 20 74 68 6f 73 65   lowers a nd those
01f0  20 63 6f 6f 6c 20 66 6f  75 6e 74 61 69 6e 73 2c    cool fo untains,
0200  20 62 75 74 0d 0a 73 68  65 20 63 6f 75 6c 64 20    but··sh e could 
0210  6e 6f 74 20 65 76 65 6e  20 67 65 74 20 68 65 72   not even  get her
0220  20 68 65 61 64 20 74 68  6f 75 67 68 20 74 68 65    head th ough the
0230  20 64 6f 6f 72 77 61 79  3b 20 60 61 6e 64 20 65    doorway ; `and e
0240  76 65 6e 20 69 66 0d 0a  6d 79 20 68 65 61 64 20   ven if··  my head 
0250  77 6f 75 6c 64 20 67 6f  20 74 68 72 6f 75 67 68   would go  through
0260  2c 27 20 74 68 6f 75 67  68 74 20 70 6f 6f 72 20   ,' thoug ht poor 
0270  41 6c 69 63 65 2c 20 60  69 74 20 77 6f 75 6c 64   Alice, ` it would
0280  20 62 65 20 6f 66 0d 0a  76 65                      be of··  ve
```

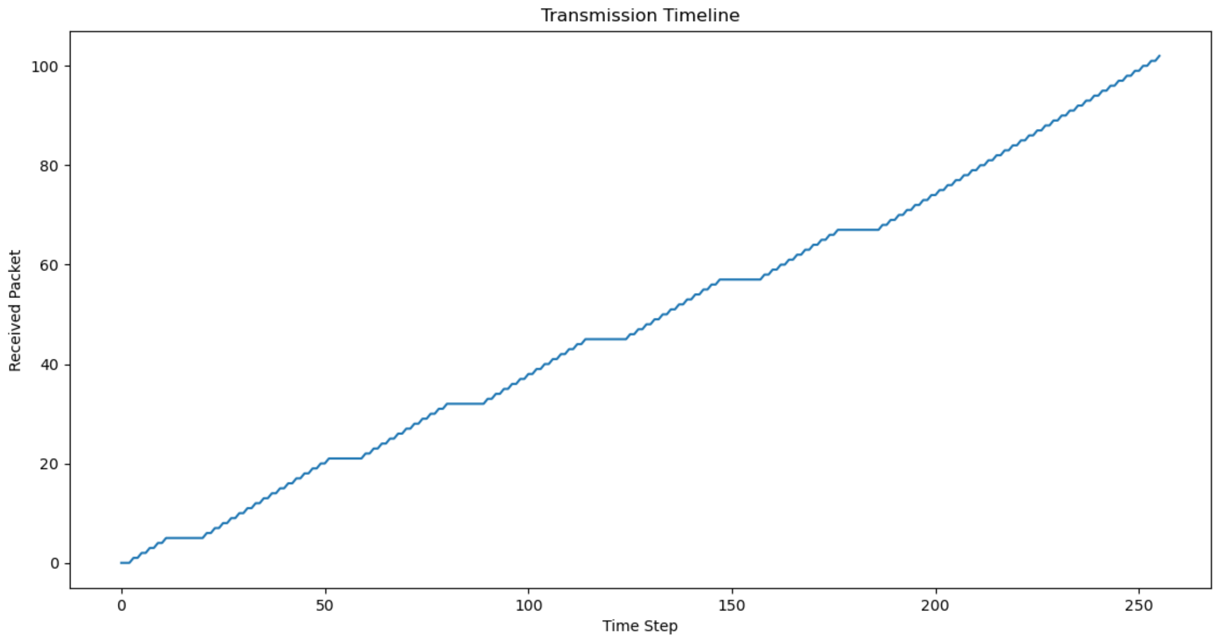
4- ack of packet 12

```
104 16.461176  192.168.1.102   192.168.1.101   UDP   48 8080 → 57879 Len=6
> Frame 104: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface 0
0000  a4 c3 f0 48 65 b1 e2 fc  83 5a 57 f6 08 00 45 00   ···He··· ·ZW···E·
0010  00 22 4d 6b 40 00 40 11  69 44 c0 a8 01 66 c0 a8   ·"Mk@·@· iD···f··
0020  01 65 1f 90 e2 17 00 0e  9c 58 41 63 6b 20 31 32   ·e······ ·XAck 12
```

IV. Test_file_1.txt :

A. MSS = 604, N = 10, timeout = 0.1

1. **Received packet plot:**



2. **Time elapsed:**

0.93790 second
1.78366 second
1.76229 second

Average = 1.494616667 second

3. **Transmission rate estimation:**

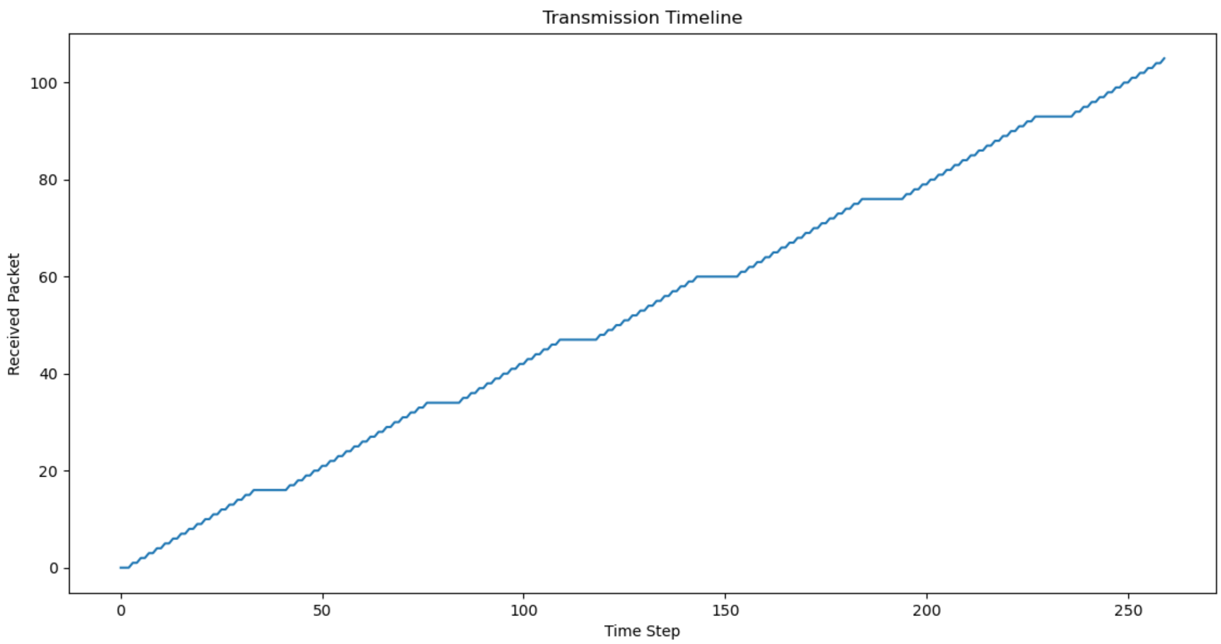
158.3740 KB/s
83.27801 KB/s

84.28773 KB/s

Average = 108.64658 KB/s

B. MSS = 1460, N = 10, timeout = 0.1

1. **Received packet plot:**



2. **Time elapsed:**

   0.74687 second
   0.77561 second
   0.68785 second

   Average = 0.736776667 second

3. **Transmission rate estimation:**

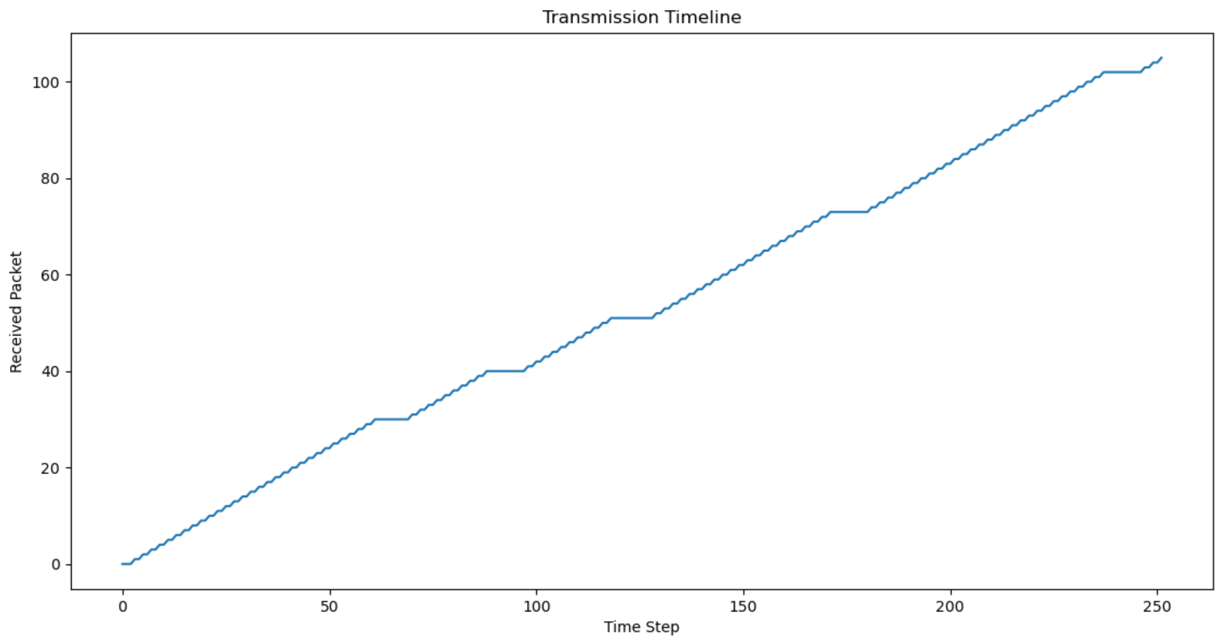
198.88072 KB/s
191.51148 KB/s
215.94577 KB/s

Average = 202.112656667 KB/s

So many duplicate acks happened, so we will decrease the timeout to 0.01

C. MSS = 1460, N = 10, timeout = 0.01

1. **Received packet plot:**



2. **Time elapsed:**

0.21873 seconds
0.17916 seconds
0.24434 seconds

Average = 0.214076667 second

3. **Transmission rate estimation:**
   679.0844 KB/s
   829.0753 KB/s
   607.9122 KB/s

   Average = 705.3573 KB/s

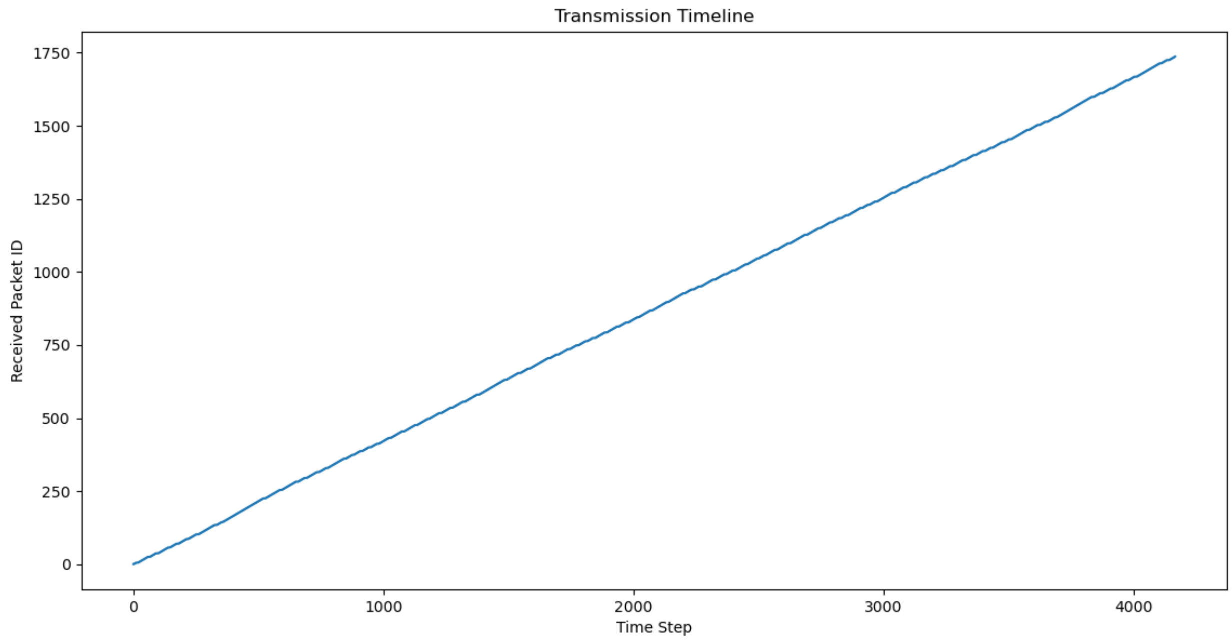
Comment:

First, we randomly initialized MSS, N and timeout to 4, 4 and 2. However, we noticed that MSS = 4 bytes is too small for that file so by trial and error we choose MSS = 604 bytes and N = 10 packets. Then, we noticed that too many duplicate acks were received before the timeout happened, so we decreased the timeout to 0.1 sec. After that, we choose MSS = 1460 bytes since the WiFi standard uses a maximum MSS of 1500 bytes. So, again too many duplicate acks were received so we decreased the timeout to 0.01 sec.

V. Test_file_2.txt :

A. MSS = 1460, N = 10, timeout = 0.01

1. **Received packet plot:**



2. **Time elapsed:**

   3.88226 seconds
   4.53675 seconds
   3.82122 seconds

   Average = 4.08 seconds

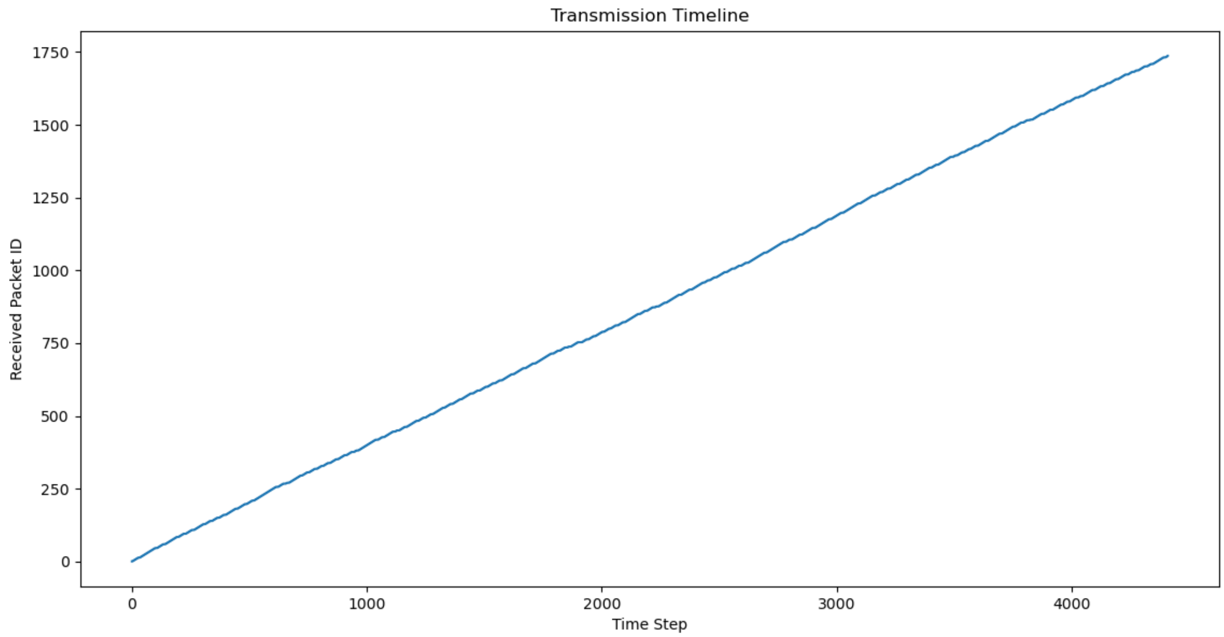
3. **Transmission rate estimation:**

   652.92003 KB/s
   558.72825 KB/s
   663.35025 KB/s

   Average = 624.99951 KB/s

B. MSS = 1460, N = 10, timeout = 0.001

1. **Received packet plot:**



2. **Time elapsed:**

3.186680316925049 seconds
3.7689402103424072 seconds
3.0713553428649902 seconds

Average = 3.3423252900441488 seconds

3. **Transmission rate estimation:**

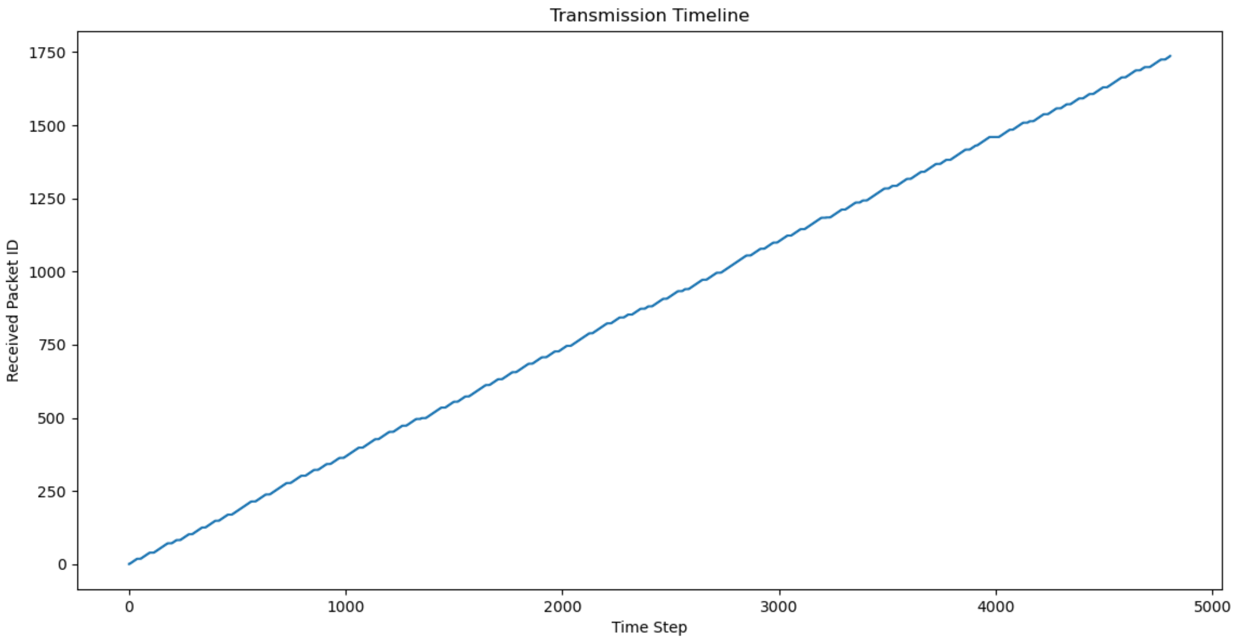
795.4393751193521 KB/s,

672.5527226577344 KB/s,
825.3069791773112 KB/s

Average = 764.4330256514659 KB/s

C. MSS = 1460, N = 20, timeout = 0.001

1. **Received packet plot:**



2. **Time elapsed:**

3.10949 seconds
2.61004 seconds
2.41928 seconds

Average = 2.71294 seconds

3. **Transmission rate estimation:**

   815.18372 KB/s
   971.17558 KB/s
   1047.7509 KB/s

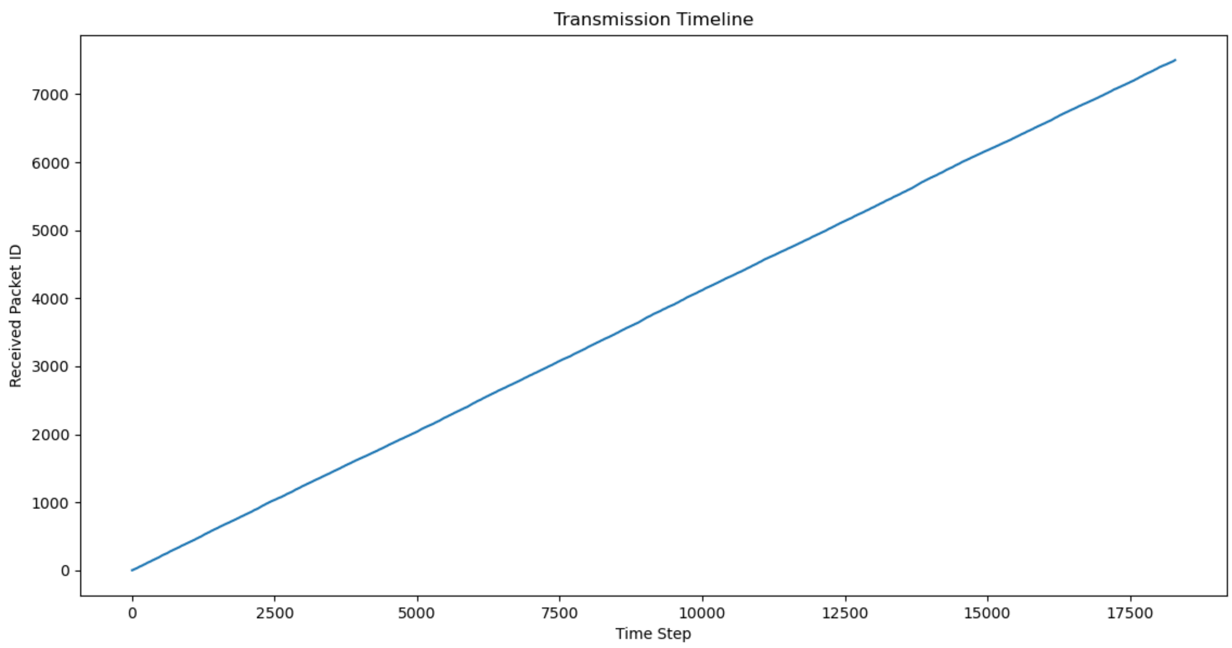
   Average = 944.7034 KB/sec

comment:
As we can see, increasing the window size and decreasing the timeout interval will yield a higher data rate. But this isn't optimal as increasing the window size could result in problems if the number of lost packets increased. This will end up filling the channel with a lot of packets wich will decrease the performance.

VI. Test_file_3.txt :

A. MSS = 1460, N = 10, timeout = 0.01

1. **Received packet plot:**



2. **Time elapsed:**

   21.94919 seconds
   22.06693 seconds
   22.13259 seconds

   Average = 22.04957

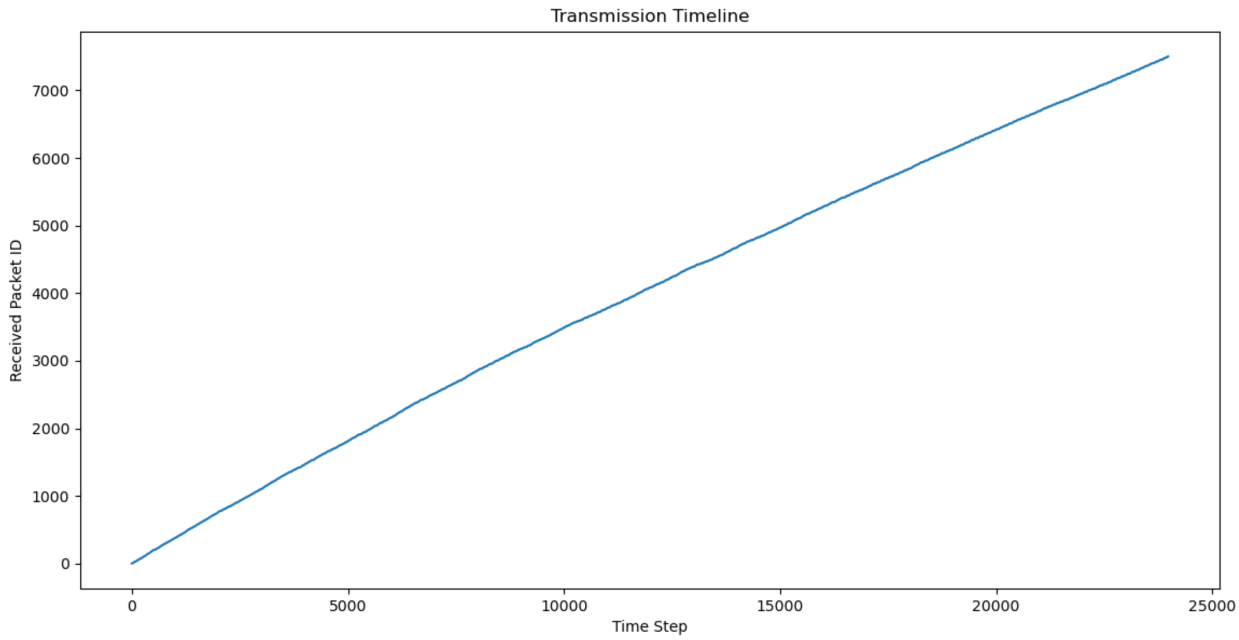
3. **Transmission rate estimation:**

   499.03311 KB/s
   496.37045 KB/s
   494.89802 KB/s

   Average = 496.76719 KB/s

B. MSS = 1460, N = 20, timeout = 0.001

1. **Received packet plot:**



2. **Time elapsed:**

20.00398 seconds
17.24471 seconds
17.22197 seconds

Average = 18.15688 seconds

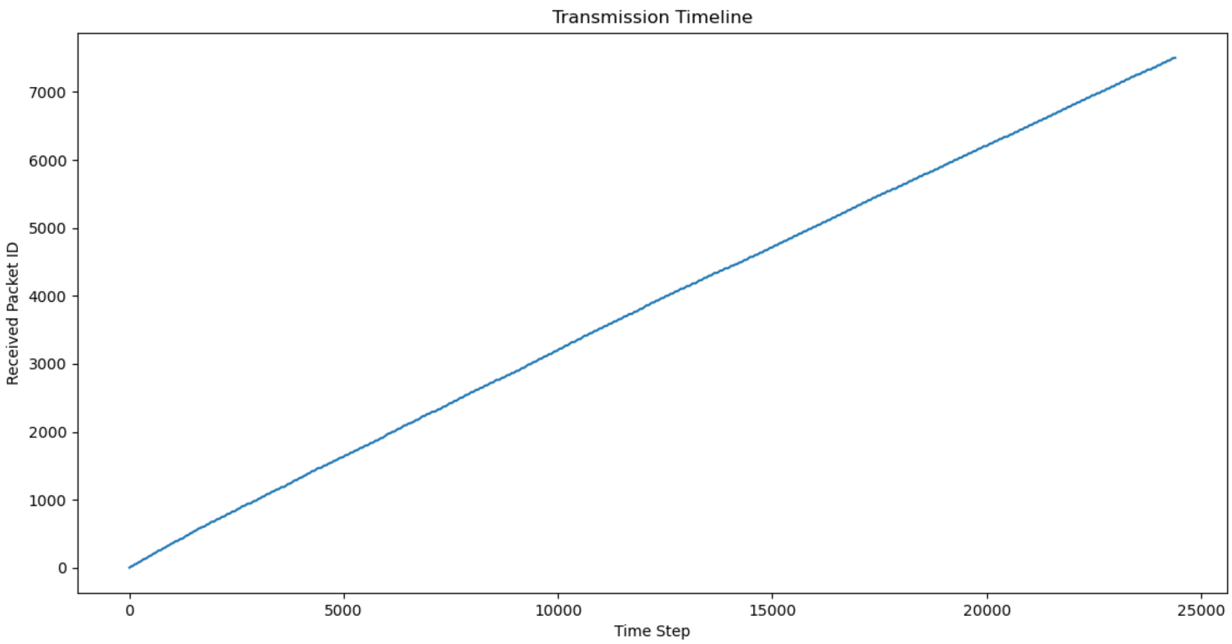
3. **Transmission rate estimation:**

547.55953 KB/s
635.17293 KB/s
636.01156 KB/s

Average = 606.24800 KB/s

C. MSS = 1460, N = 30, timeout = 0.0001

1. **Received packet plot:**



2. **Time elapsed:**

   15.93969 seconds
   18.21239 seconds
   16.82952 seconds

   Average = 16.99386 seconds

3. **Transmission rate estimation:**

   687.17592 KB/s
   601.42406 KB/s

650.84267 KB/s

Average = 646.48088 KB/s

comment:

The same observation as for the test file 2. However, a large window size is more adequate for larger files. It will speed up the duration of transmission, which is favorable for end users.

VII. Parameters optimization :

VIII. The rationale behind fixing MSS with 1460 bytes is that the Wifi protocol allows for 1500 bytes but we excluded the headers size and with small calculations we agreed on 1460.

IX. The window size and timeout interval are tuned using trial and error. We were trying to slightly increase the N to send more packets and slightly increase the timeout to wait for the acknowledgements of the whole window.

A. Test_file_1.txt:

MSS = 1460
N = 10
timeout = 0.01

Transmission rate = 705.3573 KB/s

B. Test_file_2.txt:

MSS = 1460,
N = 20,
timeout = 0.001

Transmission rate = 944.7034 KB/sec

C. Test_file_3.txt:

MSS = 1460,

N = 30,
timeout = 0.0001

Transmission rate = 646.48088 KB/s

X. Packet flooding attack tool :

https://www.imperva.com/learn/ddos/udp-flood/

UDP flooding attack is one of common (DDOS) attacks which floods a random port of the host with an overwhelming number of packets making it unresponsive to other clients. Also, spoofing the sender address is a way to make the attack anonymous. The reason behind the dangers of this attack is the unreliability of the UDP protocol. Unlike TCP, UDP does not three-way handshake or any other extra step to verify the sender. This increases the vulnerability of the UDP protocol.

Our implementation of GBN can be turned into a flooding attack tool if we increase the window size to be a large number as 10,000 or more. With the spoofing and more than one machine, this attack could be classified as DDOS threats which endangers the network infrastructure making the server more prone to fail.

The laws and regulations concerning DDOS attacks vary from country to another. In the US in 2019, a man was sentenced 27 months in prison and $95, 000 by a federal court for conducting multiple DDOS attacks on a video gaming company.[1][2]

Depending on the size of the attack and the size of the business the loss vaires. The attack could put the servers down and if the company has no counter measures for such incidents, they will probably lose a lot of money.
An incident occured in 2000, when multiple DDOS attacks were conducted on internet giants Yahoo, Amazon, eBay, E-trade and other websites in several days. The Yankee group estimated the loss to be around $1.2 B dollars. They later found out that the attack was conducted by a 15 year old canadian boy under the nickname of “mafiaboy”. [3]

XI. Protocol improvement :

It is clearly seen that the main problem of GBN is bandwidth utilization. In case of lost packets or a timeout even. The sender retransmits the whole window again, and with large window size, this fills up the bandwidth of the channel. One way to improve this behaviour is buffering the received out-of-order packets at the receiver and the sender should only retransmit the unacked packet only instead of retransmitting the whole window. Then, when the receiver receives the lost packet it re-orders the packets again and sends them to the above layer. This protocol is called the “Selective Repeat” protocol and it is an enhancement of the GBN protocol to ensure efficient bandwidth utilization.

Another improvement, incase of packet loss and the sender is block in the timer state, if the sender received multiple acks of previous packets, this is an indication that the packet was lost so we can end the timer and resend the packet without waiting for the timer to finish. This little tweak saves time and increases the performance of protocol.

Resources:

[1] https://www.justice.gov/usao-sdca/pr/utah-man-sentenced-computer-hacking-crime

[2]https://www.theregister.com/2019/07/04/gamebusting_ddos_wielder_derptrolling_sentenced_to_two_years_in_the_clink/

[3]https://nsfocusglobal.com/wp-content/uploads/2017/01/Distributed_Denial_of_Service_Attacks__An_Economic_Perspective__Whitepaper.pdf