CSE - 589

Analysis of Programming Assignment - 2

I have read and understood the course academic integrity policy.

Composed By :

SHAILESH ADHIKARI

#50246417

TIMEOUT SELECTION

The timeout selection for different protocols (Alternative Bit Transfer, Go Back N and Selective Repeat) works differently. For e.g. : In GBN a desired timeout will be more than the one in selective repeat. This is specially due to the retransmission and buffer management techniques used by these protocols.

Below is the analysis of different timeout in contrast to different loss probabilities, for ABT, GBN and SR protocols, implemented under Programming Assignment - 2.

(i) Alternative Bit Transfer Timeout Selection

LOSS	TIMEOUT = 11	TIMEOUT = 12	TIMEOUT = 15	TIMEOUT = 20
0.1	0.0198961	0.0198014	0.0200124	0.0199543
0.2	0.0200359	0.0200378	0.0201579	0.0200457
0.4	0.019836	0.0190653	0.016053	0.0128183
0.6	0.0092995	0.0085973	0.0070312	0.005439
0.8	0.0023707	0.0021863	0.0017446	0.0013576

Fig 1 : Average throughput observed using different timeouts in **ABT** against loss probabilities of 0.1, 0.2, 0.4, 0.6, 0.8

The timeout providing the best average throughput observed using the data from Fig. 1 and the representation of that data displayed in plot (Fig. 2, next page) is closer to the average round trip time of 10. Since, ABT is a stop and wait approach, the least amount of time waited after completion of expected RTT is desirable. Here, **timeout of 11 is used for generating further observations**.

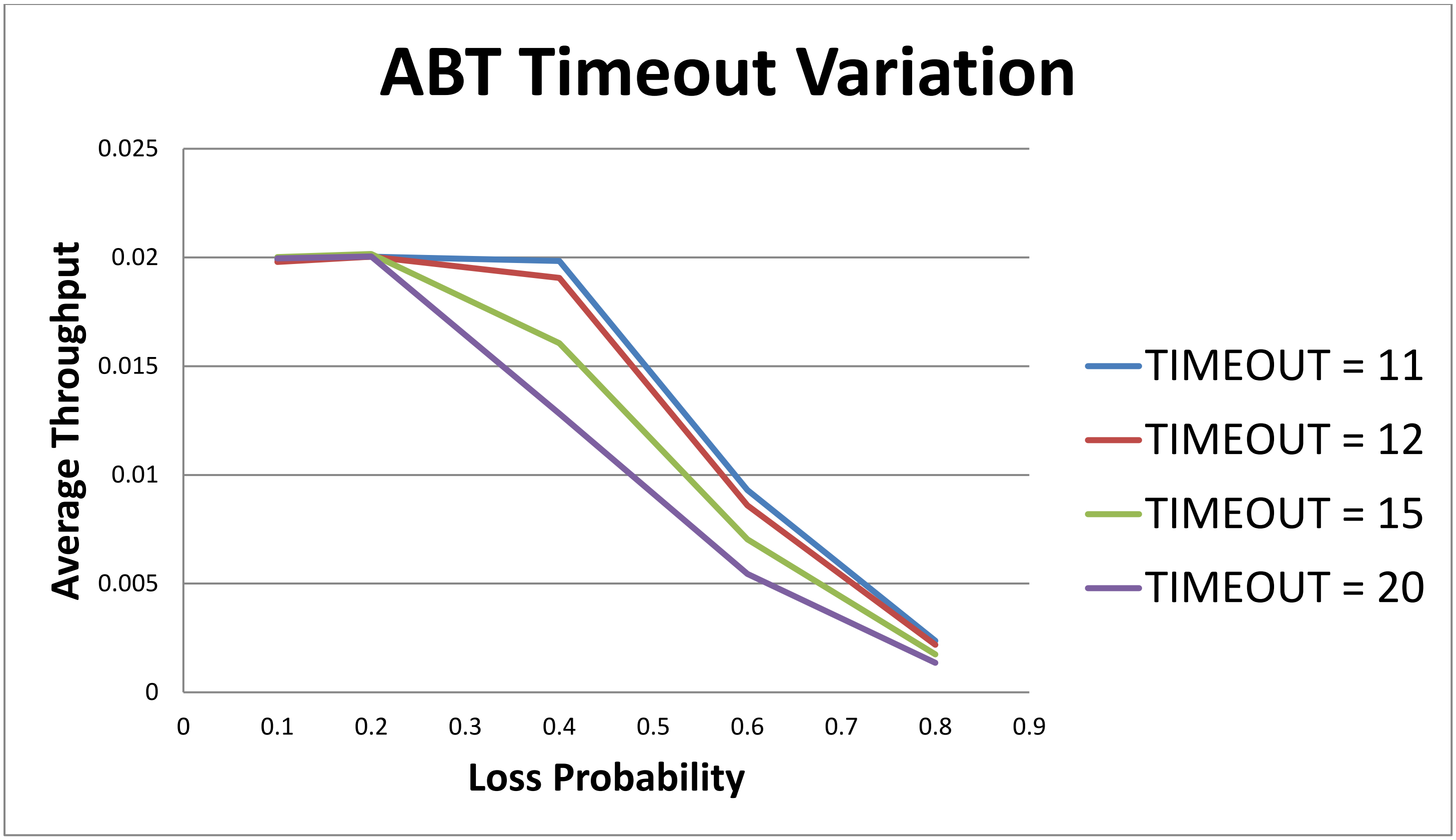

Fig 2 : The plot above shows that the **timeout value of 11** provides the **best average throughput** for **ABT**

(ii) Go Back N Timeout Selection

LOSS	TIMEOUT -15	TIMEOUT -20	TIMEOUT -25	TIMEOUT - 30	TIMEOUT - 35	TIMEOUT - 50	TIMEOUT - 100
0.1	0.0125234	0.0189344	0.0200547	0.0199538	0.0198639	0.0200108	0.0196463
0.2	0.0105886	0.0149646	0.0199728	0.018249	0.0191338	0.0199383	0.0159578
0.4	0.0052297	0.0134122	0.0187832	0.0186099	0.0146258	0.0109582	0.0065638
0.6	0.0017906	0.0029689	0.0019183	0.0025317	0.0041363	0.0022315	0.0027513
0.8	0.0005255	0.000473	0.0005855	0.0004376	0.0006819	0.0005975	0.0007607

Fig 3 : Average throughput observed using different timeouts in **Go Back N** for **window = 10** against loss probabilities of 0.1, 0.2, 0.4, 0.6, 0.8

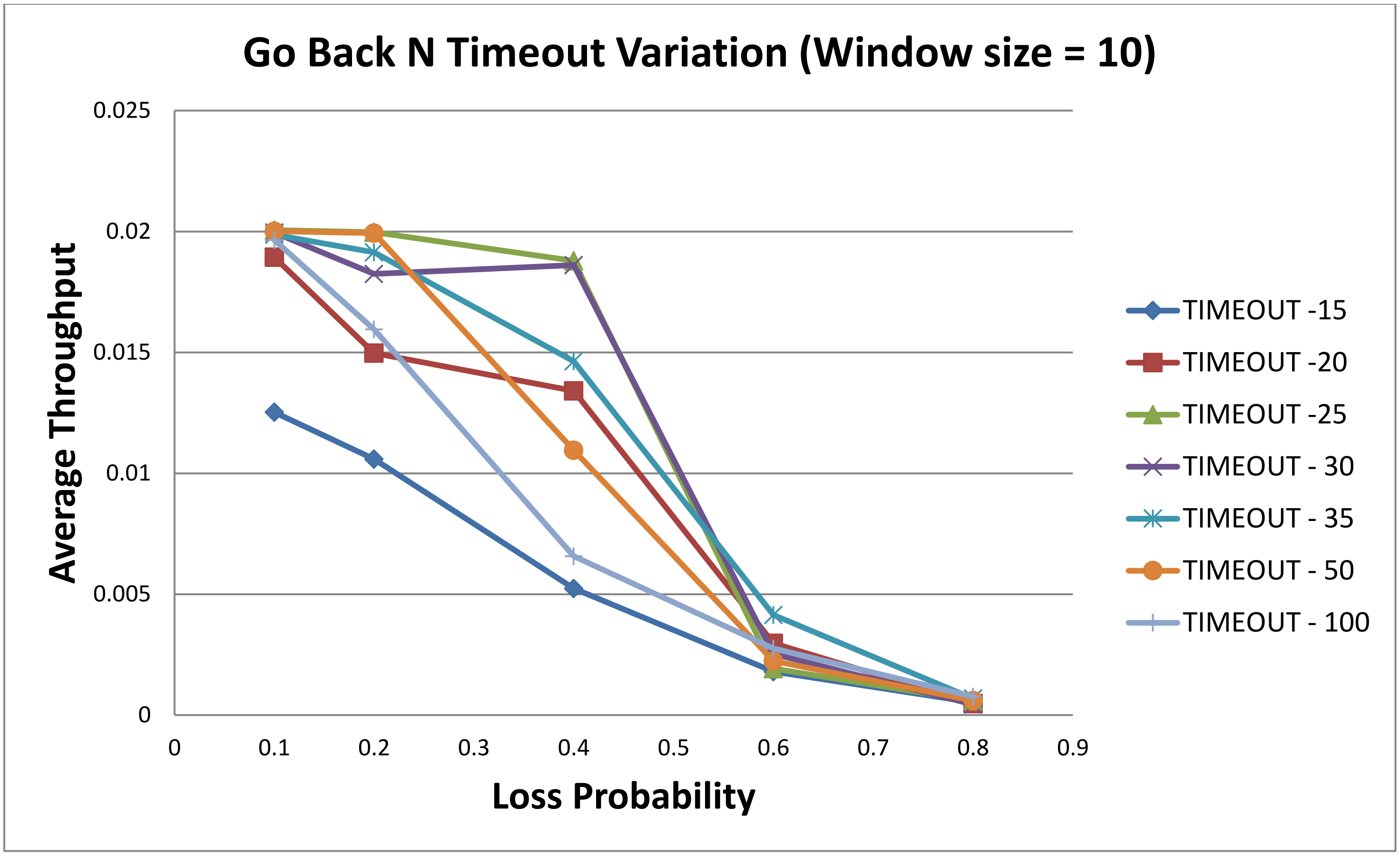

Fig 4 : The plot above displays the **Go Back N** throughput versus loss indicators for different timeouts. The **window size** used for above observations is **10**.

LOSS	TIMEOUT -20	TIMEOUT -25	TIMEOUT - 30	TIMEOUT -35	TIMEOUT - 50	TIMEOUT - 100
0.1	0.0190613	0.0200547	0.0199701	0.0198665	0.020017	0.0123349
0.2	0.0153951	0.0182409	0.0183091	0.0162205	0.0193682	0.0054848
0.4	0.0124171	0.0142846	0.0090189	0.0051639	0.0048438	0.004808
0.6	0.0034464	0.0032946	0.0031917	0.0029441	0.0032392	0.0043783
0.8	0.0023737	0.002552	0.0028788	0.00301	0.0033105	0.0018737

Fig 5 : Average throughput observed using different timeouts in Go Back N for **window = 50** against loss probabilities of 0.1, 0.2, 0.4, 0.6, 0.8

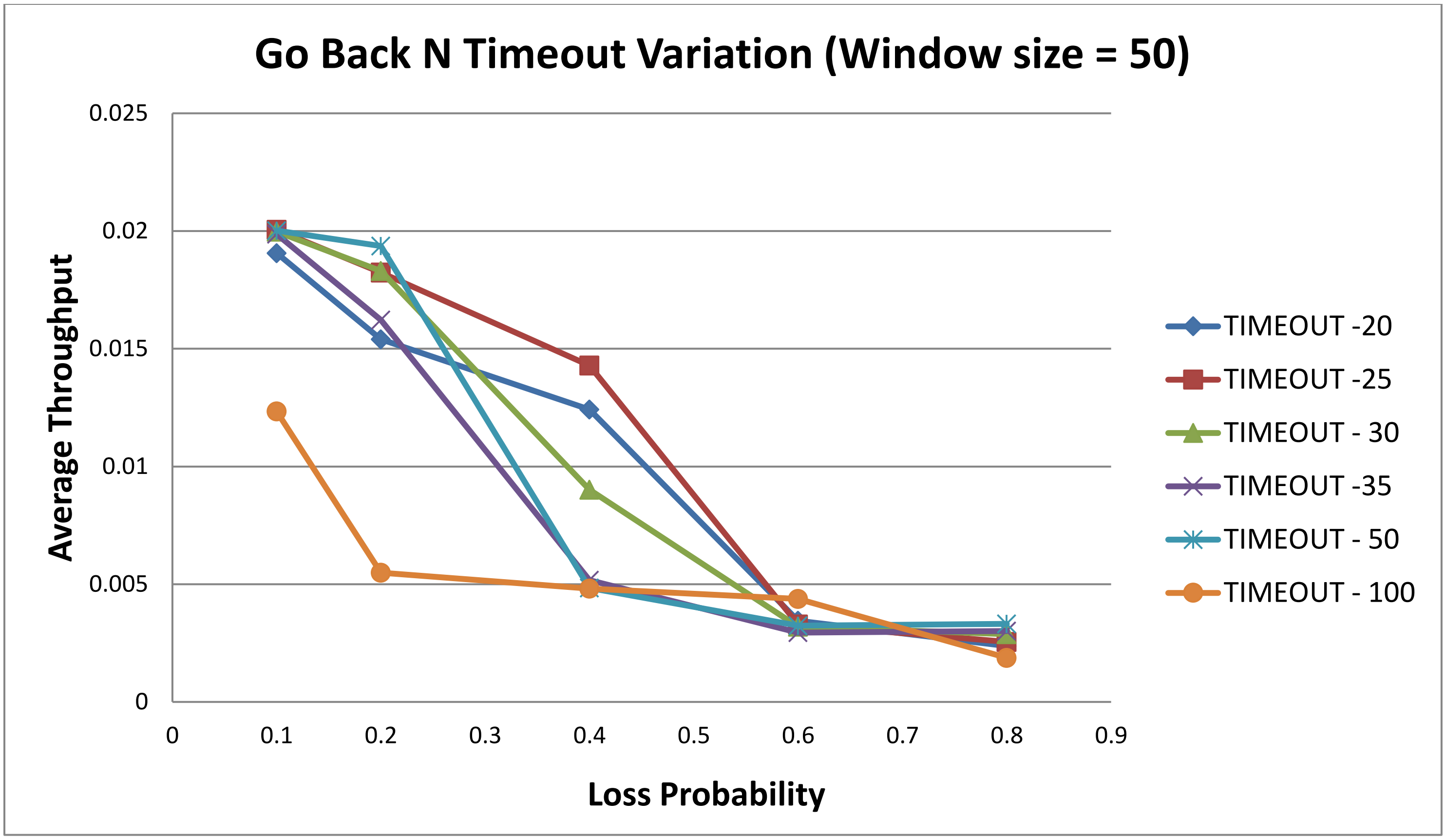

Fig 6 : The plot above displays the throughput versus loss indicators for different timeouts. The **window size** used for above observations is **50**.

By the observations of the **Figures 3-6**, it is fairly visible that **Go Back N doesn't handle higher corruption well**. This is majorly due to the main idea of going back to the start of the window and dropping the successful packets at the receiver. For larger values of window size this trend intensifies making GBN even more ineffective.

For this assignment the **timeout selection value is picked as 25** since it is giving **relatively fair performance** (observed in data in figures above) and also since it **gives fair amount of time to the delayed packets to be received**. This helps in **avoiding the unnecessary transmissions probable during larger window size**.

(iii) Selective Repeat Timer Selection

LOSS	TIMEOUT -11	TIMEOUT -15	TIMEOUT -20	TIMEOUT -25
0.1	0.017749667	0.020003	0.0200995	0.0201335
0.2	0.018663	0.0200312	0.0200089	0.0200188
0.4	0.0200025	0.0200361	0.0168146	0.0199952
0.6	0.0109786	0.0187292	0.0043196	0.0118805
0.8	0.0044761	0.0047931	0.0035146	0.0028849

Fig 7 : Average throughput observed using different timeouts in **Selective Repeat** for **window = 10** against loss probabilities of 0.1, 0.2, 0.4, 0.6, 0.8

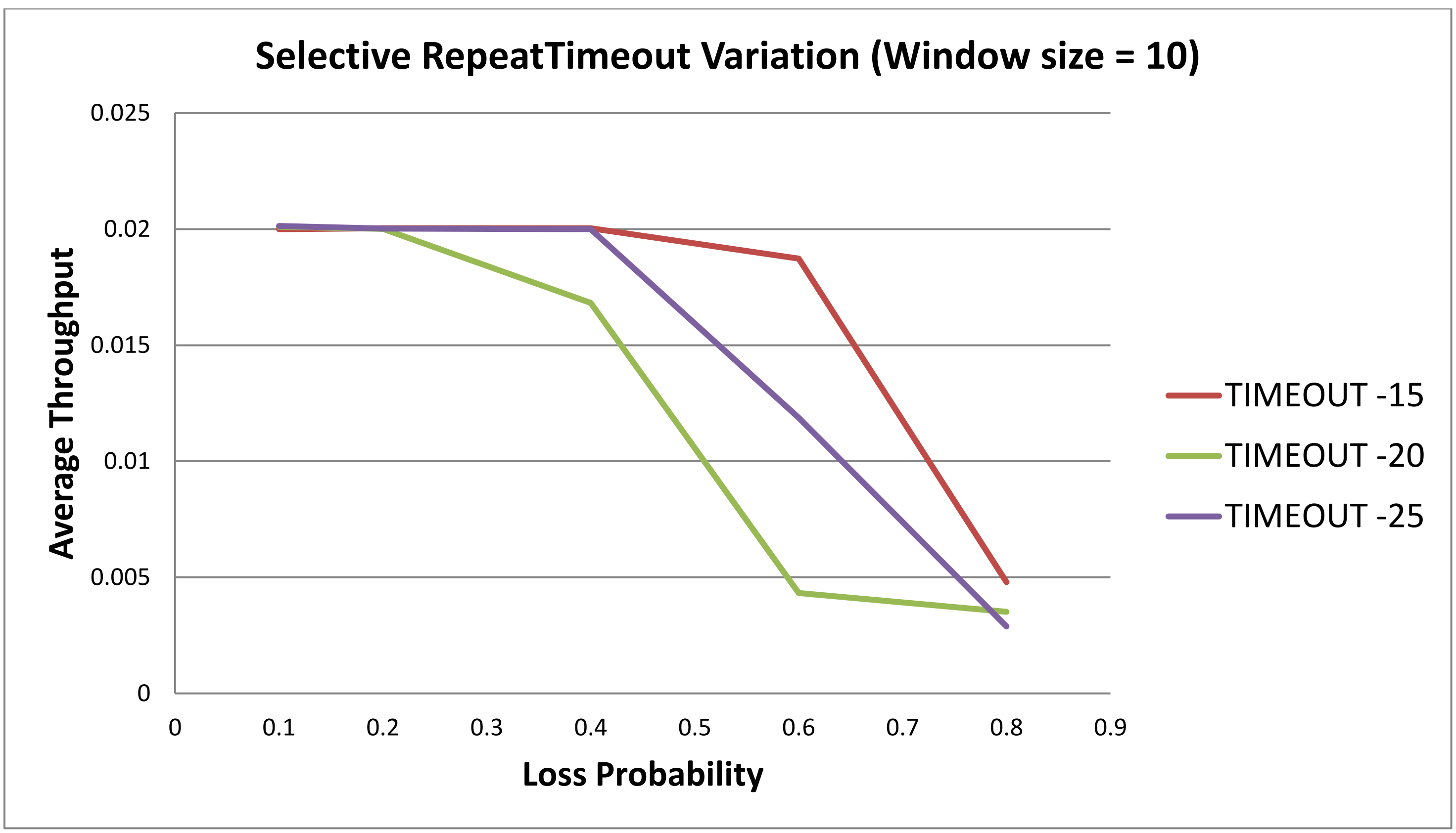

Fig 8 : The plot above displays the **Selective Repeat** throughput versus loss indicators for different timeouts. The **window size** used for above observations is **10**.

LOSS	TIMEOUT -15	TIMEOUT -20	TIMEOUT -25
0.1	0.0200151	0.0200949	0.020112
0.2	0.0200516	0.0199946	0.0200007
0.4	0.0200348	0.020034	0.0200744
0.6	0.0198499	0.0200055	0.0199802
0.8	0.0137584	0.0111204	0.0091293

Fig 9 : Average throughput observed using different timeouts in **Selective Repeat** for **window = 50** against loss probabilities of 0.1, 0.2, 0.4, 0.6, 0.8

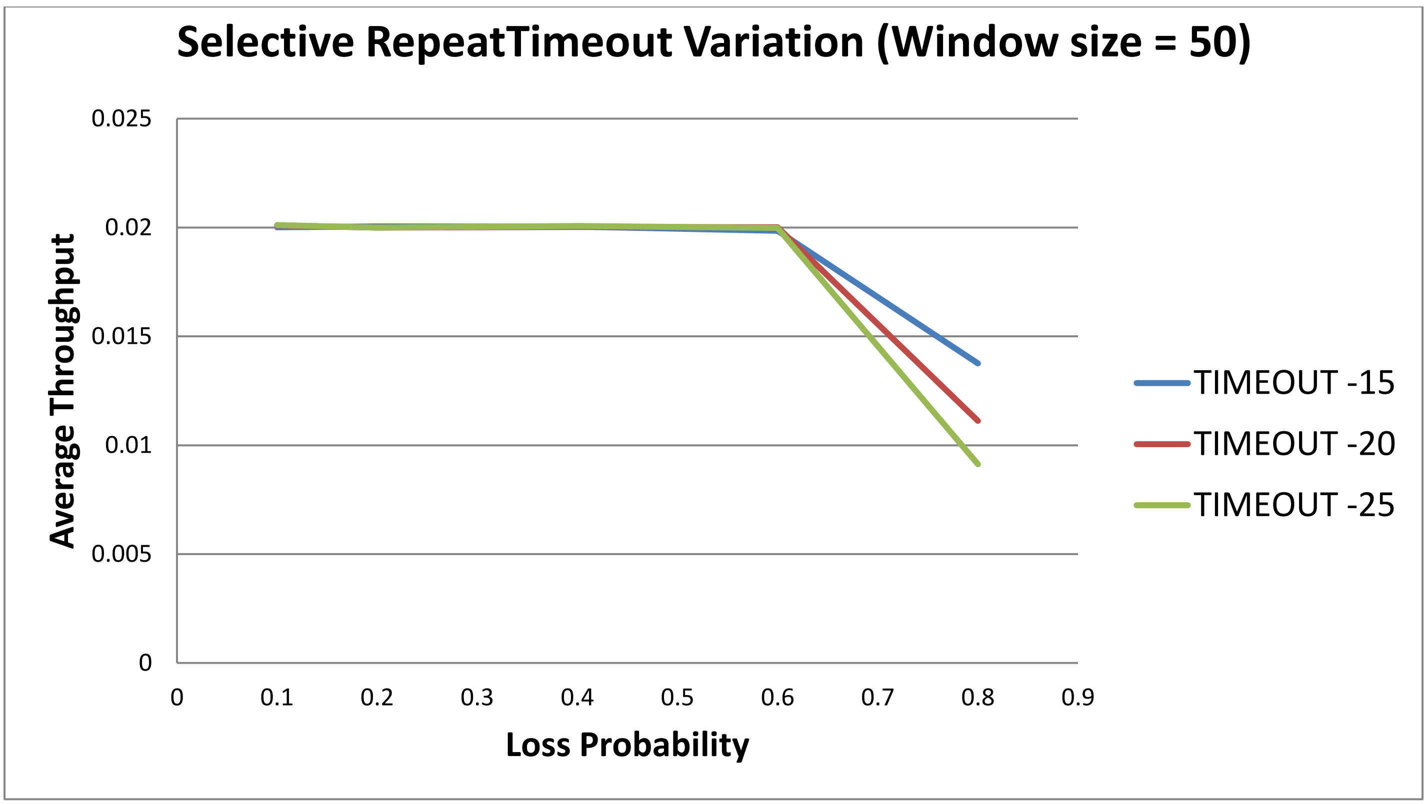

Fig 10 : The plot above displays the **Selective Repeat** throughput versus loss indicators for different timeouts. The **window size** used for above observations is **50**.

From the above observations (Fig 7 - Fig 10) , it is visible that timeout 15 works well with both windows of 10 and 50. Since, **selective repeat buffers the data at receiver** side, it is a **better** approach **to use timeout closer to expected round trip time**. For further observations **15 is** used **as timeout value**.

EXPERIMENTS

Experiment - 1

Experiment 1 is conducted on **constant values** of **1000 messages to be sent by entity A**, a **mean time of 50 between message arrivals (from A's layer5)** and a **corruption probability of 0.2.**

The **experiments** are done on the **basis** of different **loss probabilities of 0.1, 0.2, 0.4, 0.6 and 0.8** and **window size of 10 and 50 (for GBN and SR), comparing** the 3 protocols **(ABT, GBN and SR).**

PART - A

(Window size: 10)

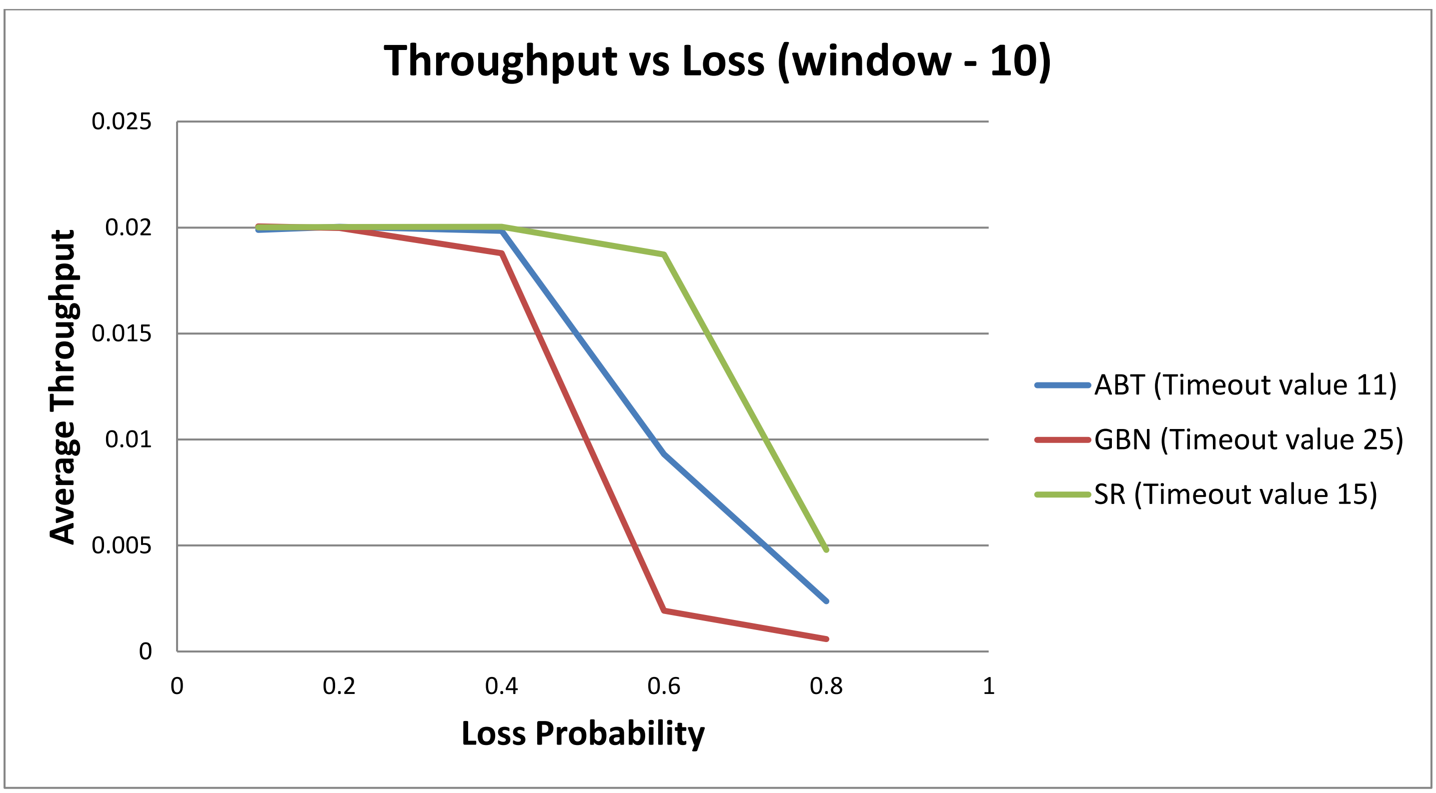

Fig 11 : The Selective repeat is a clear winner out of the 3 approaches with **window size = 10**.

PART - B

(Window size: 50)

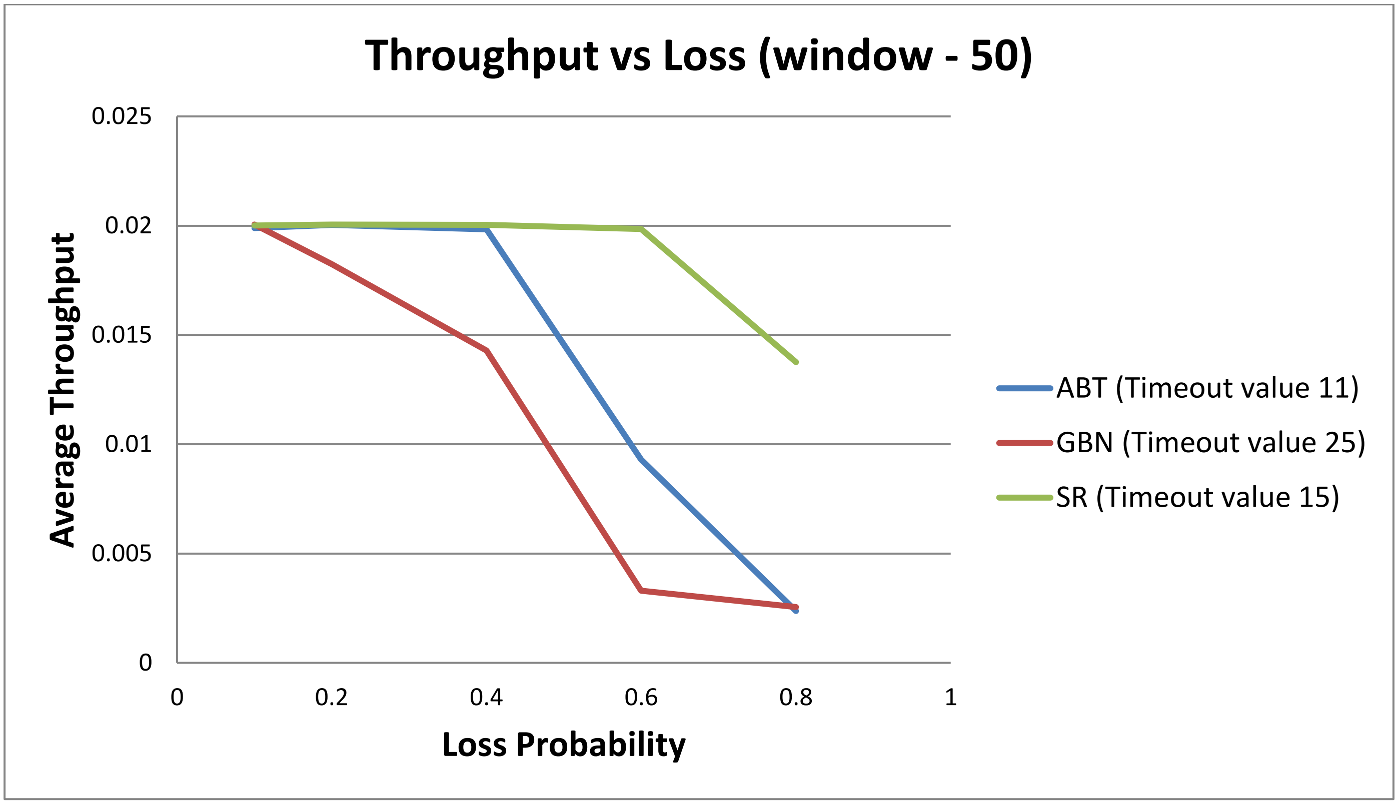

Fig 12 : Window size 50 used for GBN and SR. SR provides the best average throughput out of the three.

The performance of Selective Repeat is best for both the window sizes. GBN's performance suffers with the increase in window size and the increase in loss probability.

ABT also performs poorly with the increase in loss probability as the transmissions constantly timeout with the loss of packets.

Experiment - 2

Experiment 2 also is conducted on **constant values** of **1000 messages to be sent by entity A**, a **mean time of 50 between message arrivals (from A's layer5)** and a **corruption probability of 0.2.**

The **experiments** are done on the **basis** of different **window sizes: {10, 50, 100, 200, 500} for GBN and SR** and **the 3 loss probabilities: {0.2, 0.5, 0.8}, comparing** the 3 protocols **(ABT, GBN and SR).**

PART - A

(Loss Probability - 0.2)

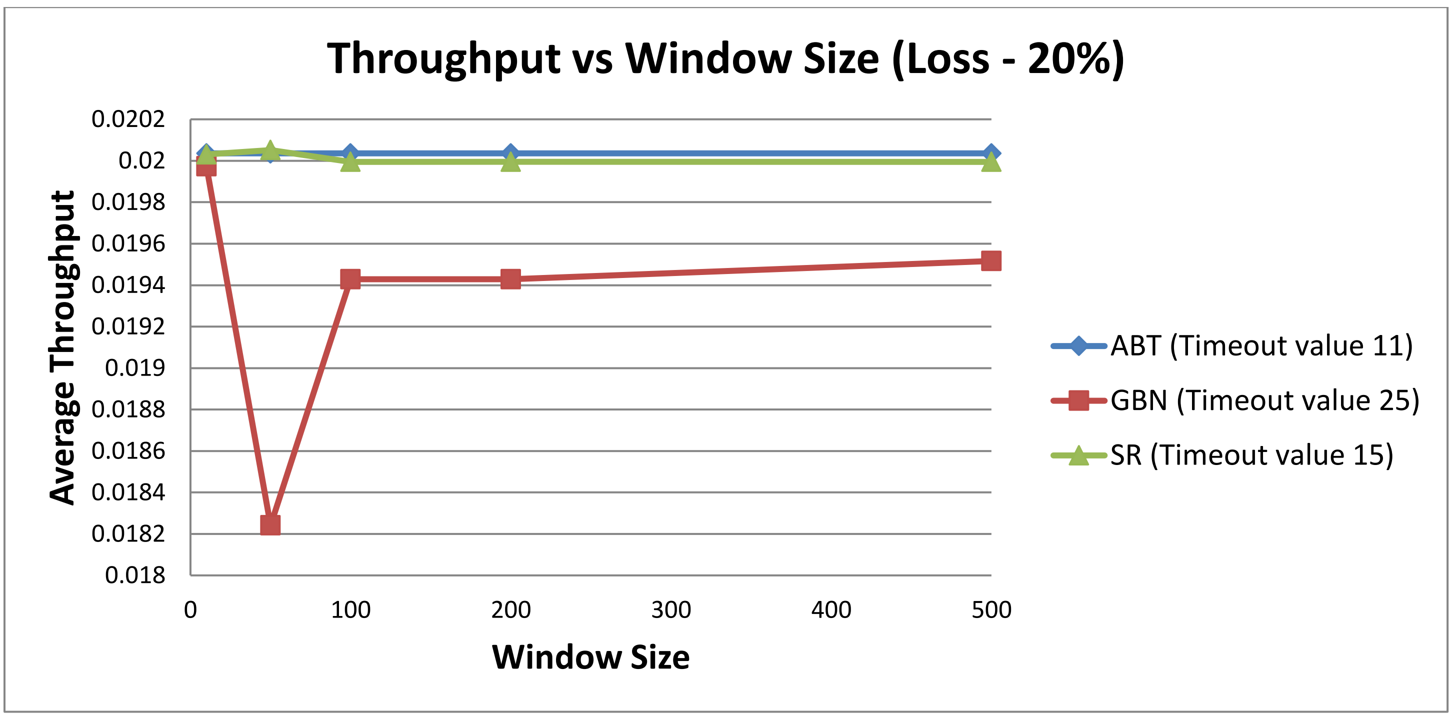

Fig 13 : For loss probability of 0.2 all the three approaches give similar throughput.

PART - B

(Loss Probability - 0.5s)

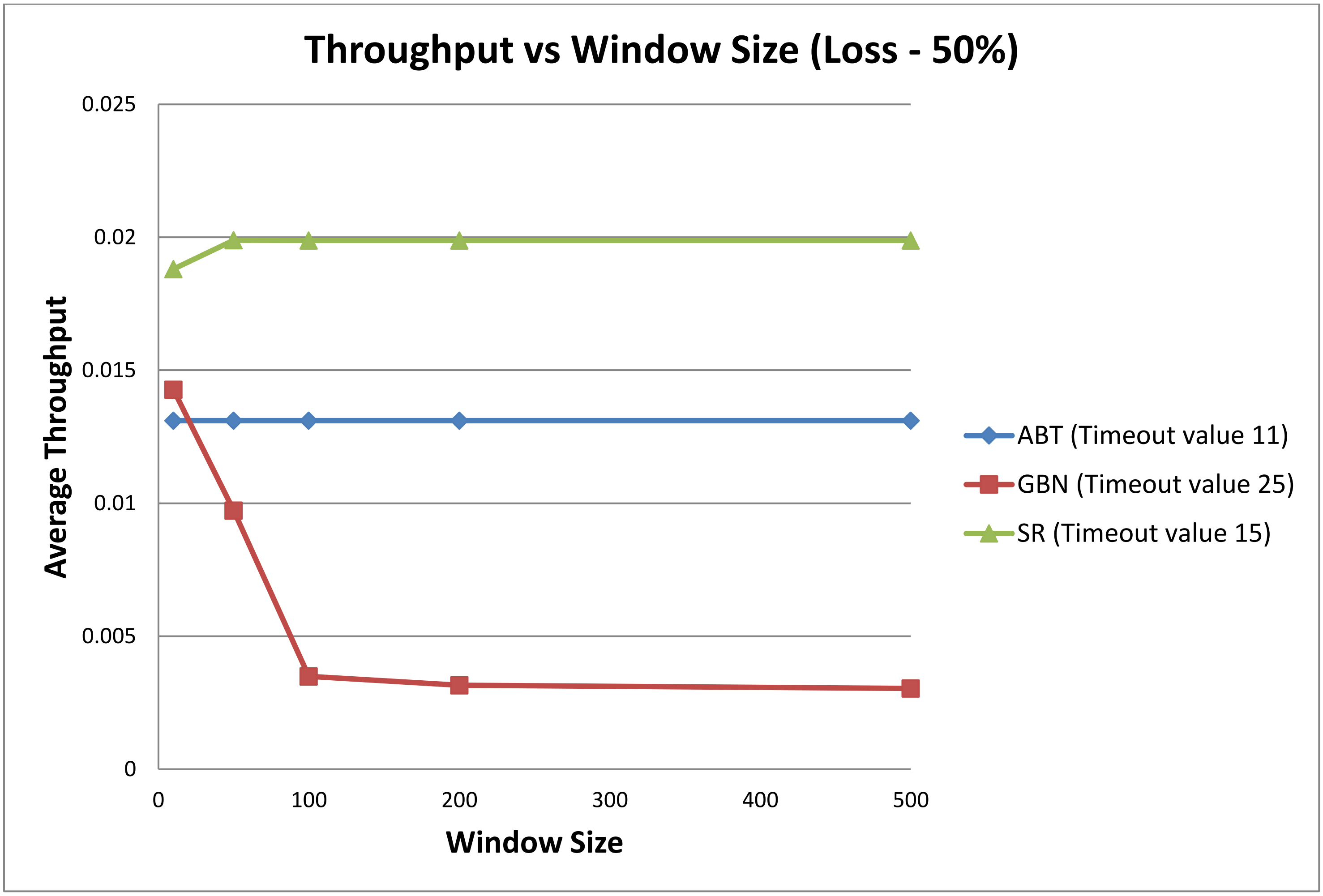

Fig 13 : For loss probability of 0.5, SR performs far better than GBN and ABT

From Fig 13 we can observe that the GBN performs good at start but drops with the increase of window size.

PART - C

(Loss Probability - 0.8)

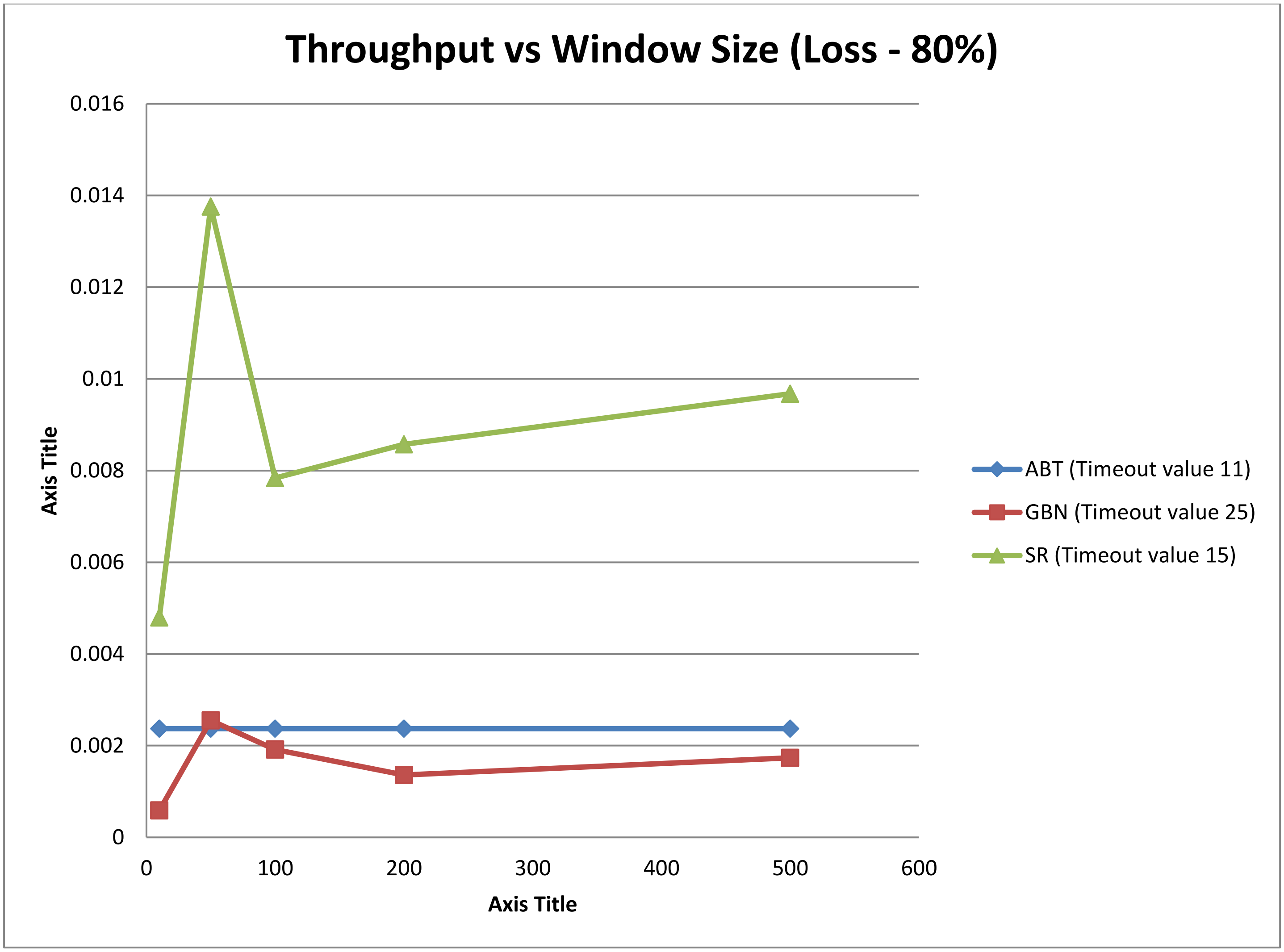

Fig 13 : For loss probability of 0.8, SR still performs far better than GBN and ABT

From the above experiments, it is quite clear that **Selective Repeat is the most evolved approach**. **GBN** seems to work fine with smaller windows and less loss rates but **performs very poorly** as both the said **factors (window size and loss rate) increase**. The main reason for this trend of GBN is the retransmission of all the packets over and over again. As far as **ABT** is concerned, the performance provided gets **very poor**.