Memory Allocation Tuning to Improve LAN Burst Handling Capabilities of the AMAZON Rapid

Author: Kevin Stocksdale

The following background information was derived from various E-mail exchanges between Reuters, Jade Communications, and Ericsson personnel.

Background:

Due to the very rapid increase in the peak data rates on the Reuters IDN network, particularly during the first hour of US Market Open, the AMAZON Rapids will drop frames. The configuration of the AMAZON Rapids is as follows:

- Running Tavarua Release 11.1.13.2
- 10M byte RAM
- 1 Dual WAN, each link operating at 1Mbps with compression enabled.
- 1 10baseT for bridging
- 1 10baseT for SNMP management.

As currently configured the standard Amazon with compression enabled can sustain a maximum input rate of 1.5Mbits/sec and the Rapid about twice that amount. In the real world however the data rates are not sustained, peaking at anything up to 8-10Mbits/sec which the Amazons appear to be able to handle for approximately 500-700Msecs, (Rapids almost twice that depending on the exact nature of the data) before dropping frames.

Reuters has requested a feasibility study into optimizing the Amazon Rapid memory allocation for improved peak LAN burst handling capabilities.

Purpose:

The subject of this report is to identify where and why the AMAZON Rapid drops frames during LAN bursts and to propose a solution to improved peak LAN burst handling capabilities.

Tools Used:

The SolCom RMON Utilities software (version 3.06) in conjunction with an ION Ethernet Probe (model LR10-E) were utilized to capture a series of Market Stream frames from the TESTTOOL application program. The Ethernet Probe can replay the captured frames either continuously, or by specifying a user selectable number of frames. The software also allows for a desired LAN "load" parameter expressed as a percentage of total LAN bandwidth.

The ION probe has a unique characteristic that is ideally suited for testing the LAN burst handling capabilities of the AMAZON. Before the probe stabilizes at the desired "load", one can observe (using a network general sniffer) a LAN burst of 10 to 20 percent more than the selected "load". If all the frames are the same size, the probe stabilizes at the desired “load” more quickly.

Analysis:

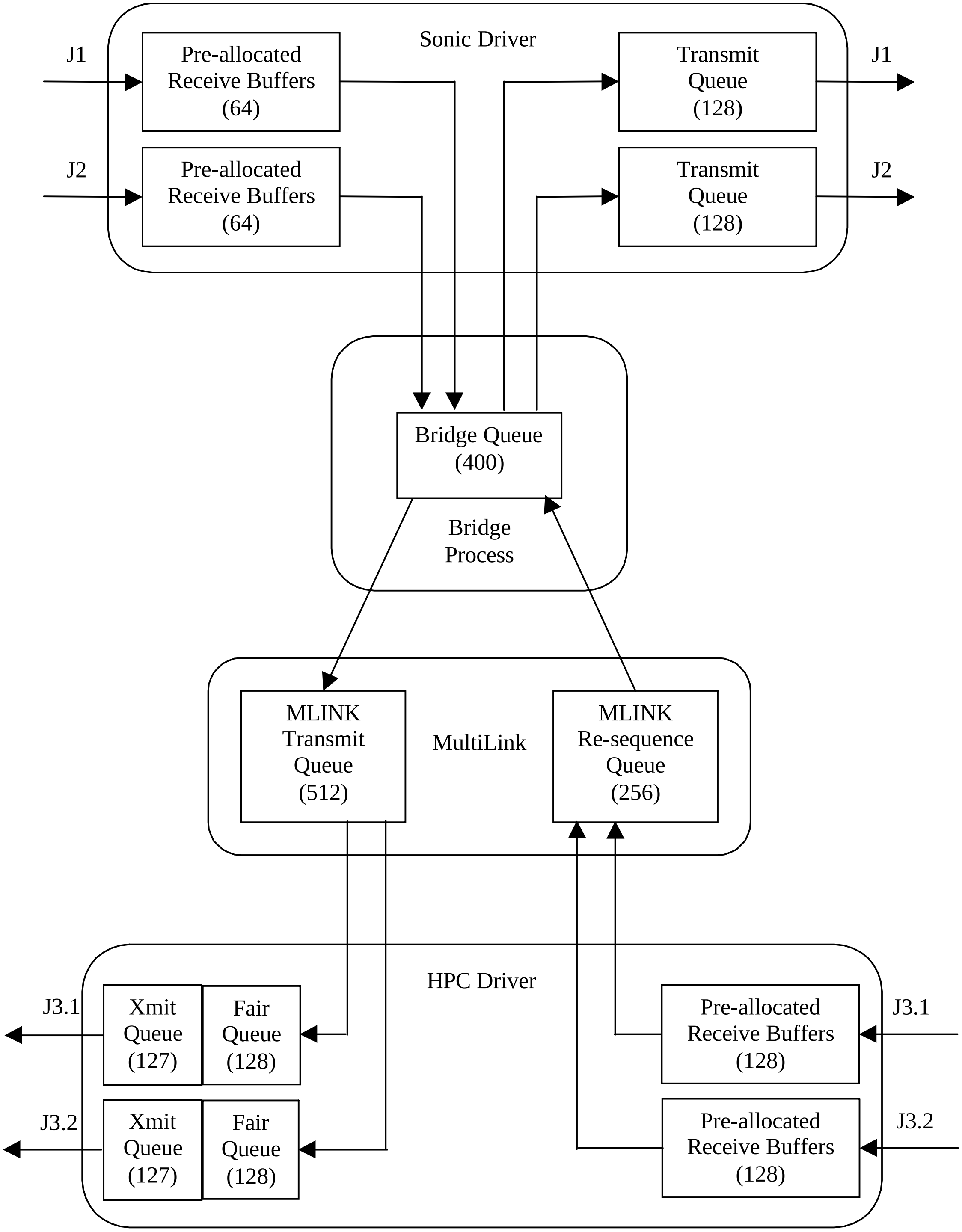
The first observation was the J1 InDisc counter incrementing. This can be caused by a number of reasons. Using a "debug" version of the Tavaura code that contains additional statistic collection and reporting capabilities I could determine the reason was due to the "missed packet tally counter" of the SONIC chip was rolling over. The SONIC chip will generate an interrupt when it misses a packet. The primary reasons are:

1. Lack of memory to buffer packet.
2. FIFO overrun
3. Receiver disabled.

The SONIC Driver handles input and output to the LAN. Buffers are pre-allocated from the 1600 byte buffer resource and inserted on the SONIC receiver ring. The SONIC chip generates an interrupt when a receive buffer contains a completed frame. The SONIC Driver will remove the buffer from the receive ring and check the frame for errors. After checking for various types of errors the SONIC driver will put the buffer on the Bridge Process (BP) Queue. If there is no room in the BP Queue then the frame is discarded, (InDisc is incremented) and the buffer is re-inserted into the SONIC receive ring. The SONIC driver will service all completed buffers during the interrupt cycle. After successfully putting the buffer on the BP Queue another buffer is allocated and inserted into the SONIC receive ring.

The Bridge Process will service (forward, flood, discard) buffers found on the BP Queue. The BP Queue may contain buffers received by the LAN or WAN interfaces. The Bridge Process is invoked from the main processing loop (not during interrupt context) when triggers are set indicating there are buffers on the BP Queue. The Bridge Process examines the MAC address and protocol bits of each frame (one frame per buffer) and uses the FDB tables to determine how to forward the frame.

The following diagram depicts data flow through the bridge and identifies the number of buffers pre-allocated for receiving data, and queue depth thresholds.


Sonic Driver
J1
Pre-allocated Receive Buffers (64)
J2
Pre-allocated Receive Buffers (64)
Transmit Queue (128)
J1
Transmit Queue (128)
J2
Bridge Queue (400)
Bridge Process
MLINK Transmit Queue (512)
MultiLink
MLINK Re-sequence Queue (256)
HPC Driver
J3.1
Xmit Queue (127)
Fair Queue (128)
J3.2
Xmit Queue (127)
Fair Queue (128)
Pre-allocated Receive Buffers (128)
J3.1
Pre-allocated Receive Buffers (128)
J3.2

Current Memory Distribution:

Address	Description
0x00200000 0x00700000	7M bytes of Local memory
0x00900000 0x00c00000	3M bytes of Global memory
0x05407f00 0x05408000	32K non volatile memory

Local memory is accessible only by “local” components such as the M68040 CPU, SONIC chip, HPC chip, and hardware compressor. Local memory can be accessed directly or through a cache. To access local memory through cache the most significant address bit must be set. Global memory is accessible by local components and also by external components (for example the CPU on slave cards such as the Hex WAN card).

Local memory is further broken down as follows:

Address	Description	Size
0x00200000 0x00200400	Interrupt Vector Table	1024
0x00200400 0x00210400	Stack	65,535
0x00211400 0x0031cf28	Application Code, Constant Data, Text (acc4165.apl)	1,096,488
0x0031cf28 0x00900000	Common Pool (local)	6,172,888
0x00900000 0x00c00000	Common Pool (global)	3,145,728

The Common Pool is segmented into DBLKs CBLKs and MBLKs and the remaining portions are used for process specific context variables. Each of the many processes which are executing require a specific amount of memory for context switching, managing inter process communications, managing queues, lookup tables, and DMA rings. To determine the exact amount each process requires would exhaust the resources allocated to this task.

The DBLK segment is used for moving data through the bridge. The current DBLK memory distribution for the Reuters configuration specified above is:

Initial allocation	Buffer size (bytes)	Total Byte Count
257	128	32,896
512	768	393,216
517	1,600	827,200
10	1,792	17,920
		1,271,232

Proposed Changes:

There are 10Mbyte of RAM, but only 800K bytes (517 * 1600) is allocated to the 1600 DBLK segment. By using the "SET MEMORY INCREMENT DBLK 3" we can increase the allocation. But still, only 64 buffers are provided to each SONIC chip for receiving data. Initial conclusion were to:

1. Increase the Initial 1600 DBLK allocation.
2. Give the SONIC chip controlling J1 more buffers.
3. Remove the threshold imposed on the BP queue.

I made some changes to how memory is allocated. I increased the 1600 DBLK pool from 517 to 2112 and gave the SONIC chip controlling J1 1024 buffers. The SONIC chip controlling J2 still only has 64 buffers and this leaves another 1024 in reserve (1024 + 1024 + 64 = 2112). After these SONIC buffers are filled, they are put on the BP queue, and then there are still enough 1600 DBLK buffers in reserve to load the SONIC chip for the next burst. I also removed the 400 buffer limitation on the BP queue. Under servere bursts the SONIC chip can still miss packets. Moderate sustained bursts will cause MLINK to drop packets.

The new DBLK memory distribution for the Reuters configuration specified above is:

Initial allocation	Buffer size (bytes)	Total Byte Count
257	128	32,896
512	768	393,216
2112	1,600	3,379,200
10	1,792	17,920
		3,823,232

Conclusions:

After making the changes proposed above, there was a dramatic improvement in handling bursts of LAN frames. Due to problems encountered with the ION probe I cannot provide any numbers at this time. The effects on overall throughput, CPU utilization, and memory utilization were logged at various WAN speeds, WAN Delays, WAN Compression, and WAN frame packing types. These metrics have been provided in tabular and graph forms at the end of this document.

Testtool Throughput - No WAN Delays

	128K	256K	512K	625K	1.25M
Compression Off, Packing 0	236	458	919	1,121	2,244
Compression Off, Packing 1	236	473	948	1,161	2,323
Compression Off, Packing 2	236	473	948	1,161	2,322
Compression On, Packing 0	388	777	1,558	1,902	3,380
Compression On, Packing 1	415	839	1,692	2,071	3,867
Compression On, Packing 2	415	841	1,692	2,070	4,136

North CPU Utilization - No WAN Delays

	128K	256K	512K	625K	1.25M
Compression Off, Packing 0	7	8	11	12	19
Compression Off, Packing 1	3	5	9	10	19
Compression Off, Packing 2	7	8	11	12	19
Compression On, Packing 0	30	36	48	53	55
Compression On, Packing 1	15	20	31	36	54
Compression On, Packing 2	14	18	27	31	48

South CPU Utilization - No WAN Delays

	128K	256K	512K	625K	1.25M
Compression Off, Packing 0	2	3	7	8	16
Compression Off, Packing 1	2	4	8	10	19
Compression Off, Packing 2	2	4	8	10	19
Compression On, Packing 0	5	11	22	27	46
Compression On, Packing 1	5	11	21	25	49
Compression On, Packing 2	5	11	21	27	60

North Memory Utilization - No WAN Delays

	128K	256K	512K	625K	1.25M
Compression Off, Packing 0	11	11	11	11	11
Compression Off, Packing 1	4	4	4	4	4
Compression Off, Packing 2	11	11	11	11	11
Compression On, Packing 0	23	50	49	59	11
Compression On, Packing 1	22	23	22	21	10
Compression On, Packing 2	50	22	22	21	15

South Memory Utilization - No WAN Delays

	128K	256K	512K	625K	1.25M
Compression Off, Packing 0	3	2	1	2	2
Compression Off, Packing 1	2	5	2	5	4
Compression Off, Packing 2	0	2	4	6	7
Compression On, Packing 0	1	2	2	3	8
Compression On, Packing 1	5	5	7	8	13
Compression On, Packing 2	5	6	5	12	43

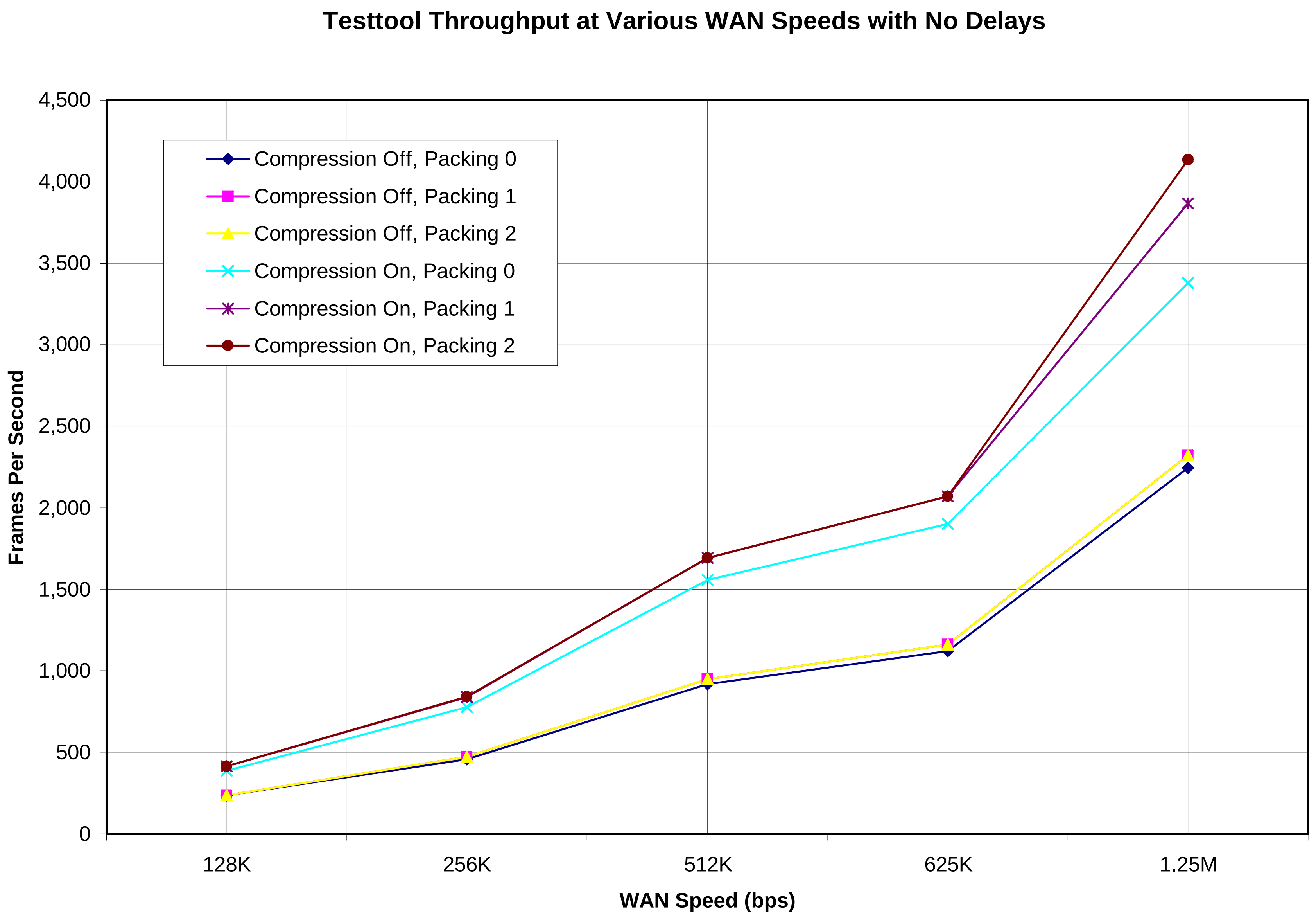
Testtool Throughput - WAN 512Kbps (each)					
	0	32	64	121	242
Compression Off, Packing 0		920	919	904	761
Compression Off, Packing 1		949	949	921	773
Compression Off, Packing 2		948	949	920	777
Compression On, Packing 0		1,558	1,558	1,374	1,070
Compression On, Packing 1		1,693	1,672	1,435	1,047
Compression On, Packing 2		1,693	1,671	1,439	1,069

North CPU Utilization - WAN 512K bps (each)					
	0	32	64	121	242
Compression Off, Packing 0		12	11	12	12
Compression Off, Packing 1		9	9	9	8
Compression Off, Packing 2		11	12	12	12
Compression On, Packing 0		53	53	52	43
Compression On, Packing 1		32	33	31	24
Compression On, Packing 2		29	29	27	42

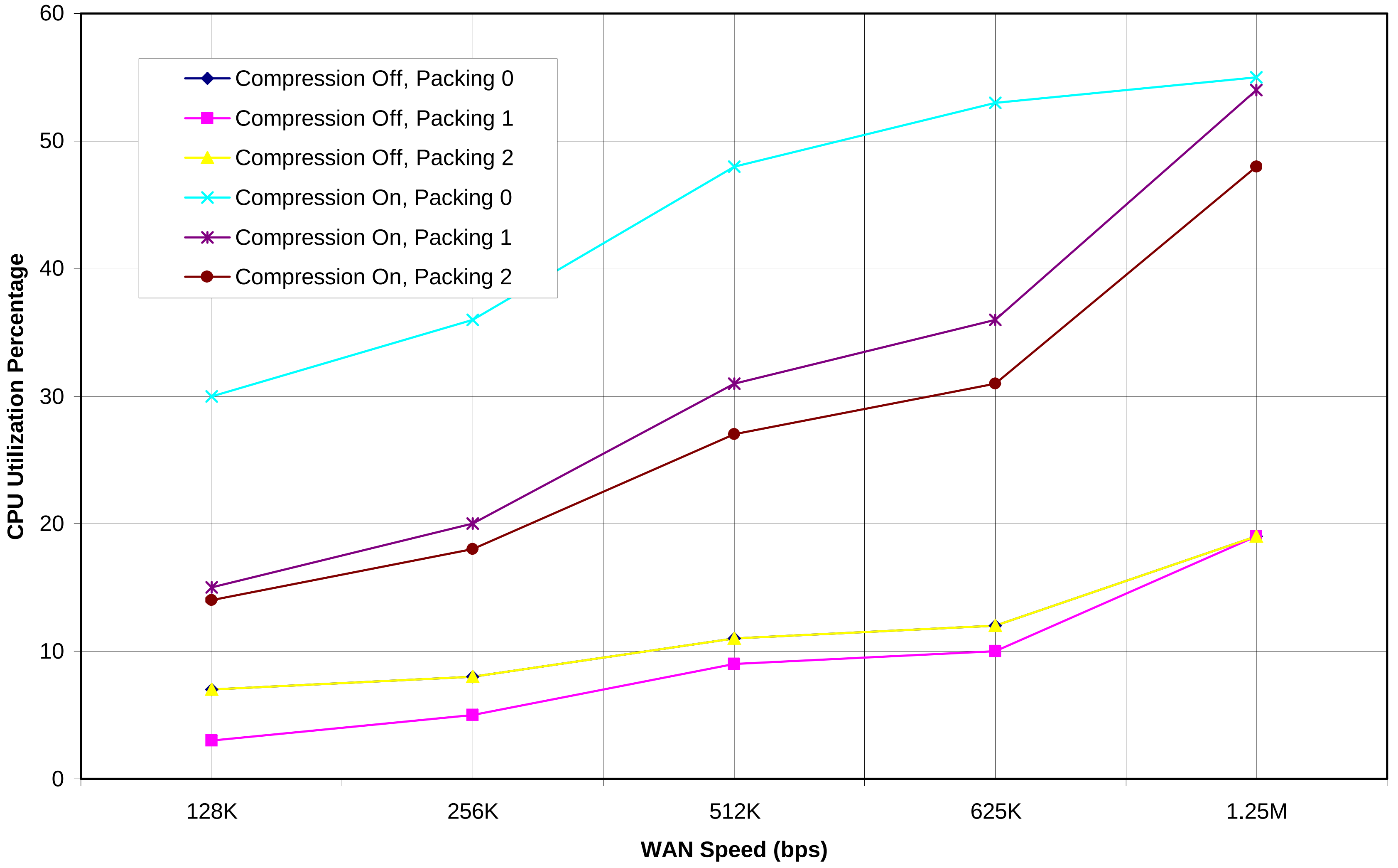
South CPU Utilization - WAN 512K bps (each)					
	0	32	64	121	242
Compression Off, Packing 0		7	7	10	7
Compression Off, Packing 1		8	8	9	6
Compression Off, Packing 2		8	8	9	6
Compression On, Packing 0		22	23	21	22
Compression On, Packing 1		21	21	18	16
Compression On, Packing 2		21	21	19	21

North Memory Utilization - WAN 512K bps (each)					
	0	32	64	121	242
Compression Off, Packing 0		14	14	15	18
Compression Off, Packing 1		5	4	4	5
Compression Off, Packing 2		15	16	16	19
Compression On, Packing 0		66	67	76	82
Compression On, Packing 1		26	28	32	31
Compression On, Packing 2		26	27	31	80

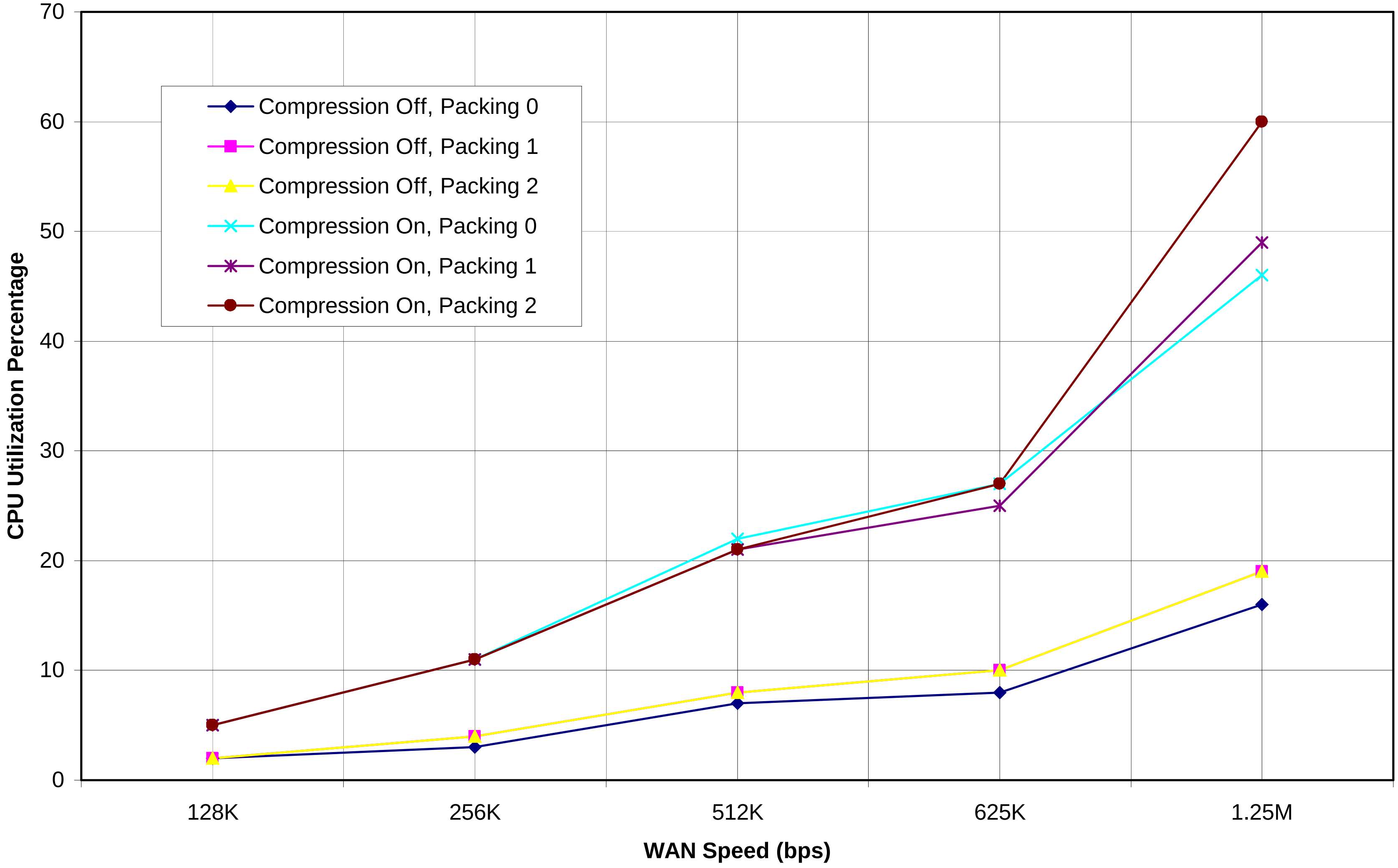
South Memory Utilization - WAN 512K bps (each)					
	0	32	64	121	242
Compression Off, Packing 0		3	6	14	10
Compression Off, Packing 1		5	10	10	6
Compression Off, Packing 2		6	9	9	8
Compression On, Packing 0		3	9	12	29
Compression On, Packing 1		6	12	8	17
Compression On, Packing 2		8	9	11	32

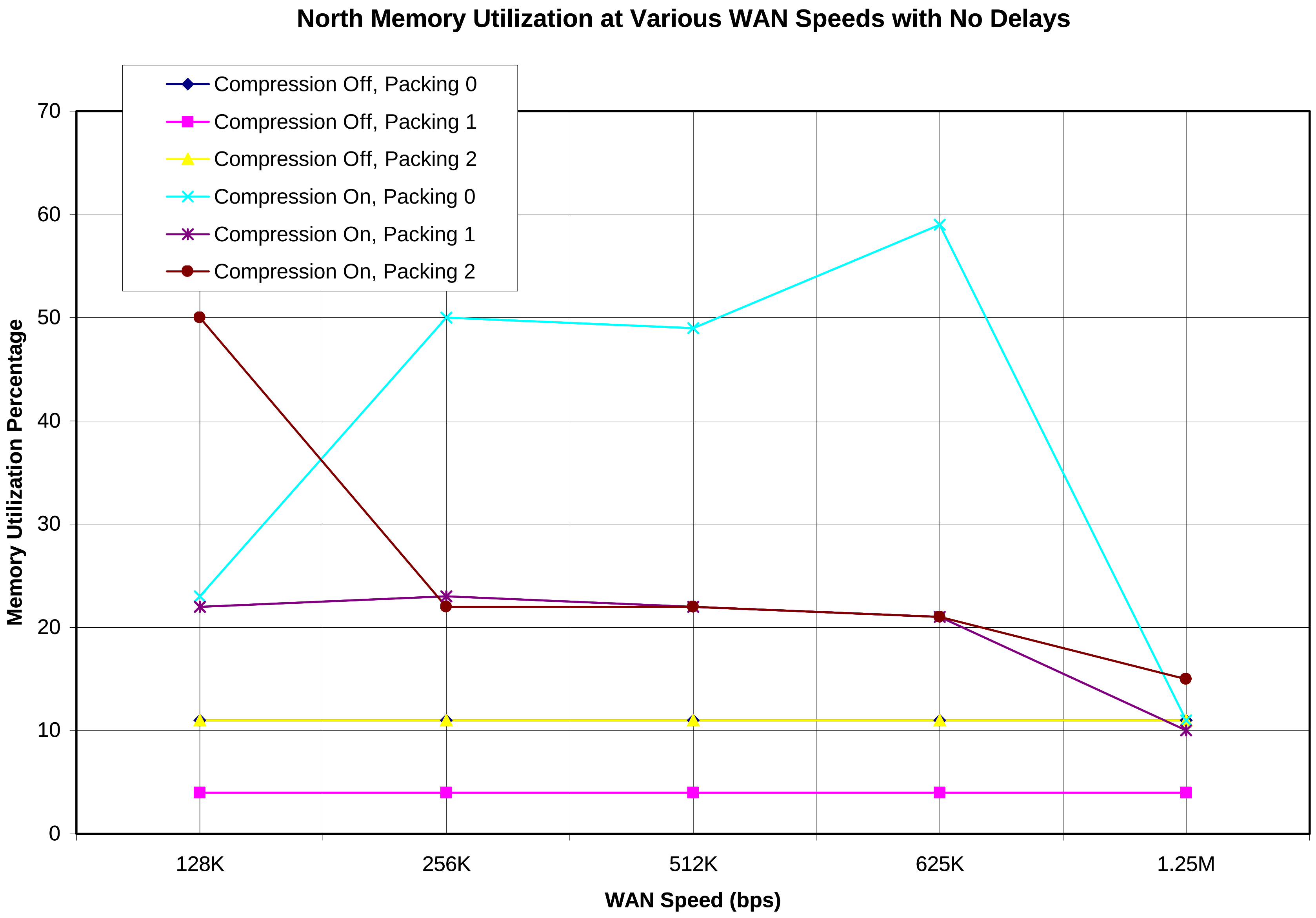

Testtool Throughput at Various WAN Speeds with No Delays
Compression Off, Packing 0
Compression Off, Packing 1
Compression Off, Packing 2
Compression On, Packing 0
Compression On, Packing 1
Compression On, Packing 2
Frames Per Second
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
128K
256K
512K
625K
1.25M
WAN Speed (bps)

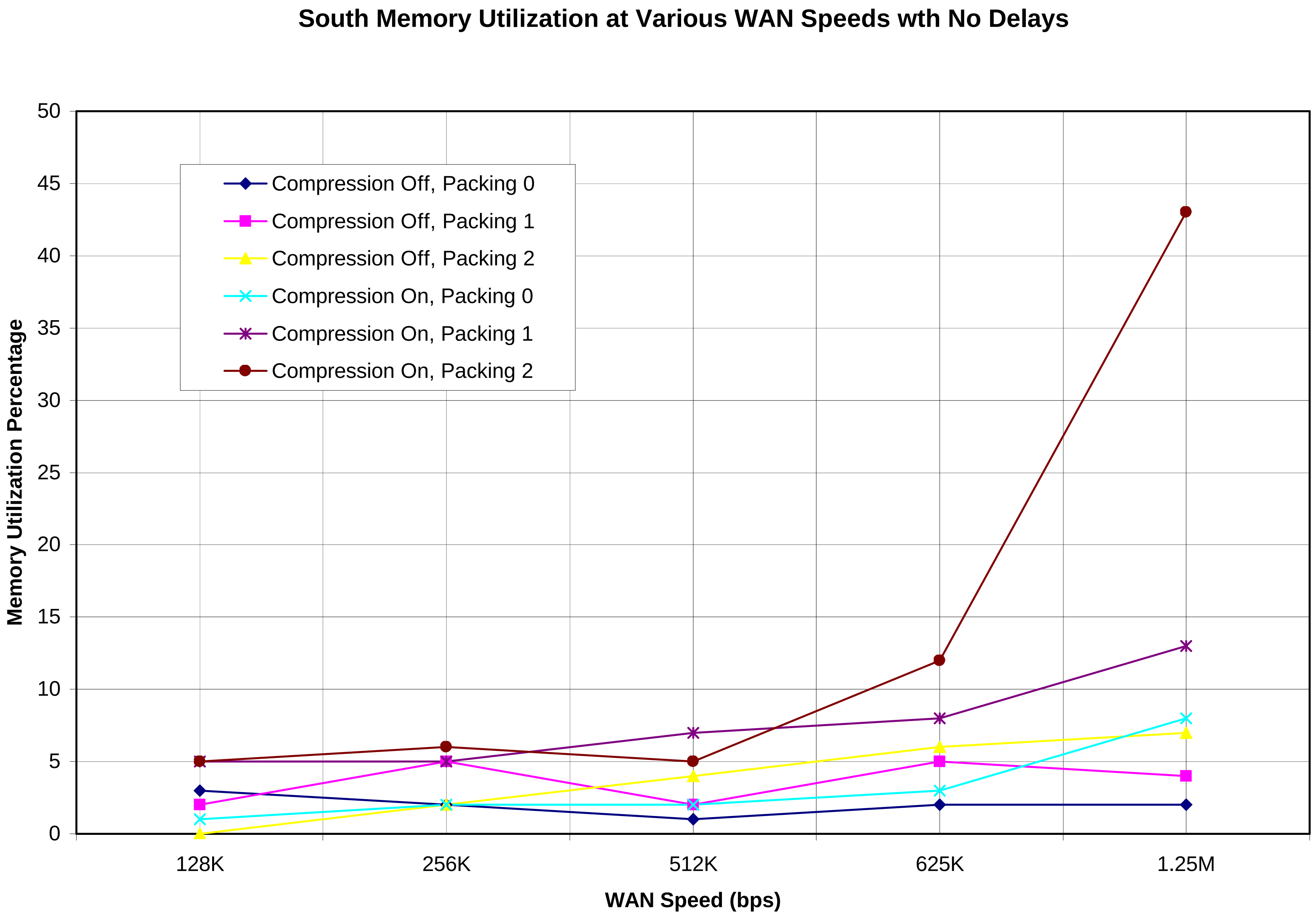
North CPU Utilization at Various WAN Speeds with No Delays



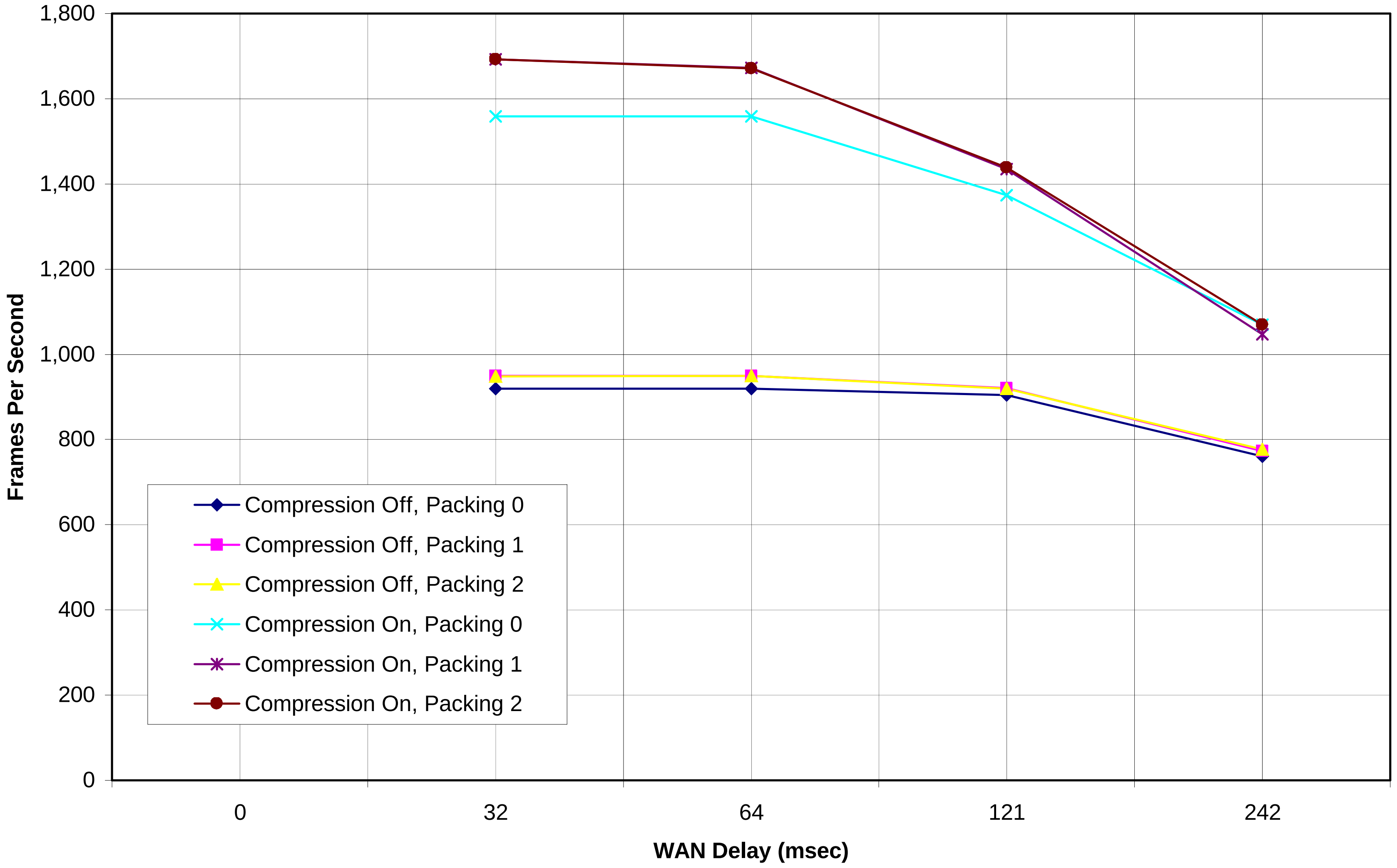
South CPU Utilization at Various WAN Speeds with No Delays

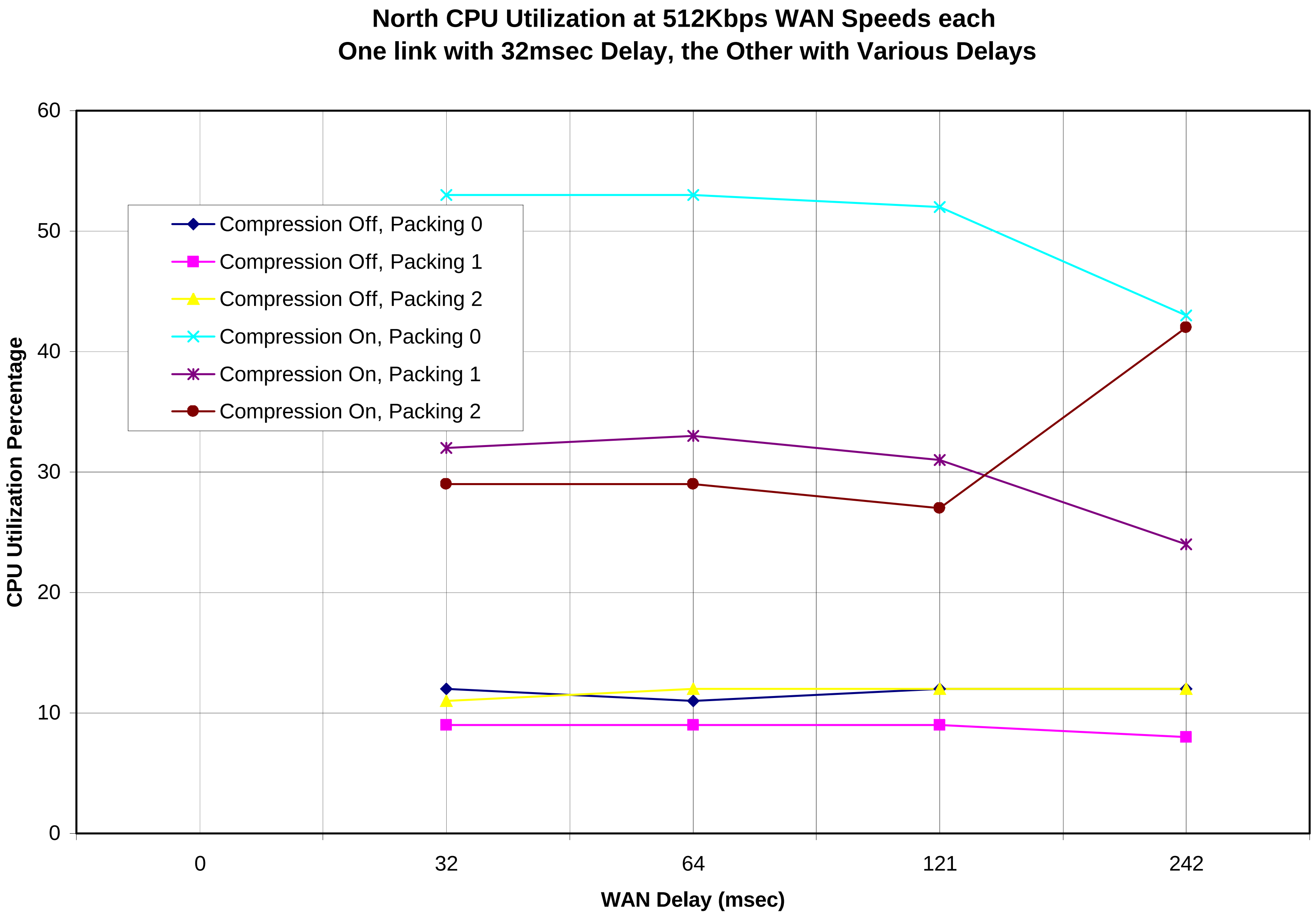


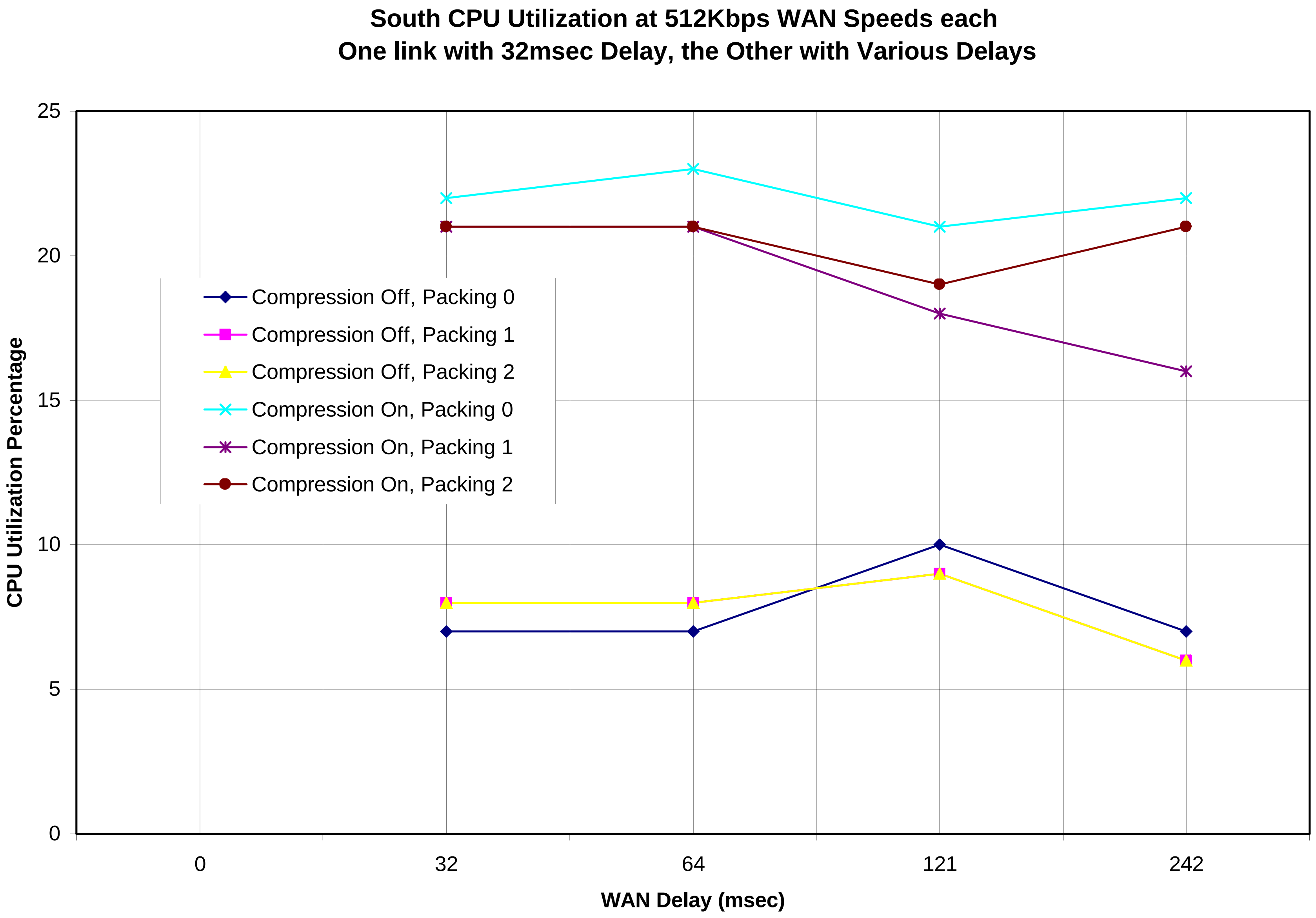
North Memory Utilization at Various WAN Speeds with No Delays
Compression Off, Packing 0
Compression Off, Packing 1
Compression Off, Packing 2
Compression On, Packing 0
Compression On, Packing 1
Compression On, Packing 2
70
60
50
40
30
20
10
0
Memory Utilization Percentage
128K
256K
512K
625K
1.25M
WAN Speed (bps)

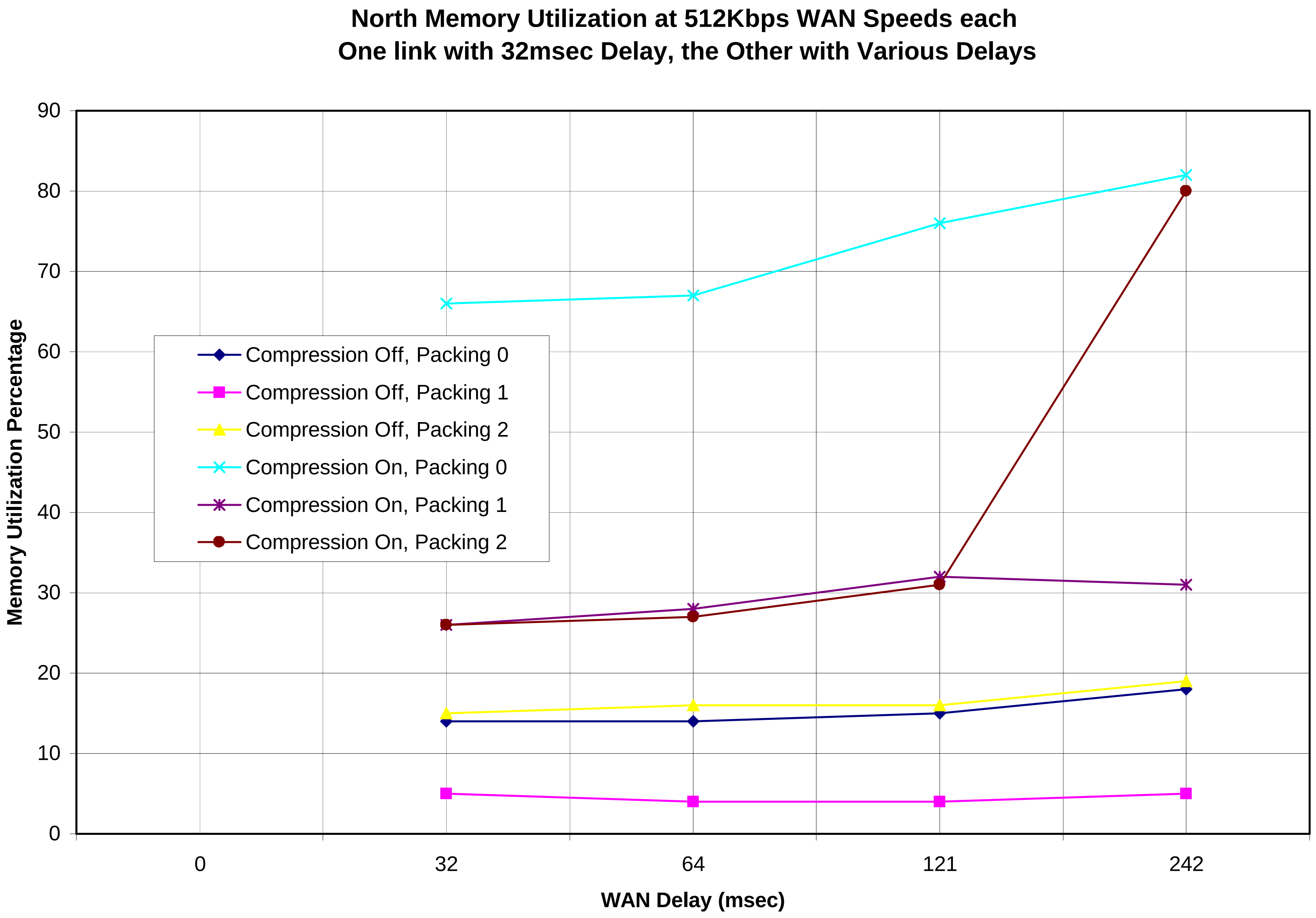
South Memory Utilization at Various WAN Speeds wth No Delays
Memory Utilization Percentage
0
5
10
15
20
25
30
35
40
45
50
128K
256K
512K
625K
1.25M
WAN Speed (bps)
Compression Off, Packing 0
Compression Off, Packing 1
Compression Off, Packing 2
Compression On, Packing 0
Compression On, Packing 1
Compression On, Packing 2

Testtool Throughput at 512Kbps WAN Speeds each One link with 32msec Delay, the Other with Various Delays

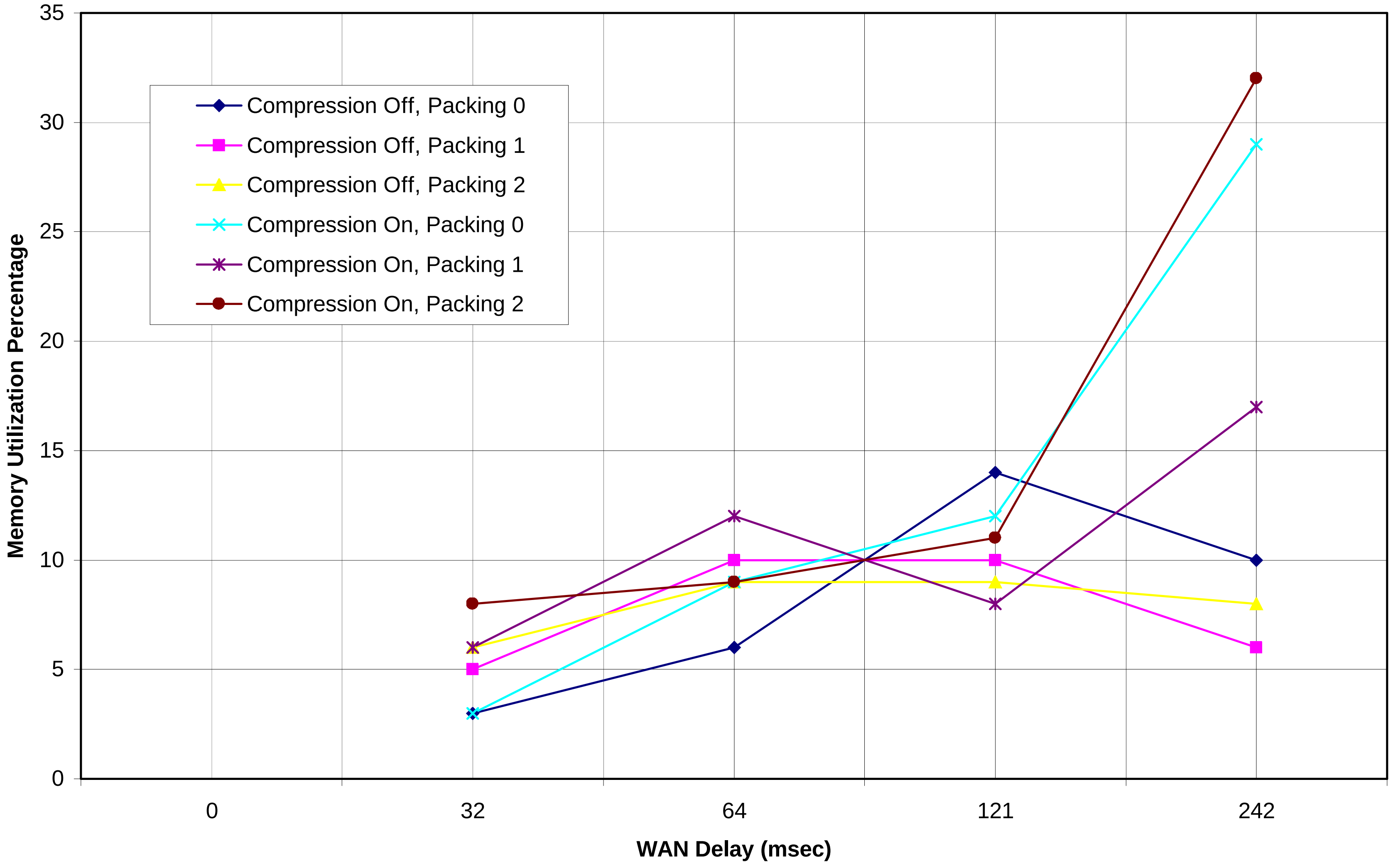



North CPU Utilization at 512Kbps WAN Speeds each
One link with 32msec Delay, the Other with Various Delays
CPU Utilization Percentage
60
50
40
30
20
10
0
0
32
64
121
242
WAN Delay (msec)
Compression Off, Packing 0
Compression Off, Packing 1
Compression Off, Packing 2
Compression On, Packing 0
Compression On, Packing 1
Compression On, Packing 2


South CPU Utilization at 512Kbps WAN Speeds each
One link with 32msec Delay, the Other with Various Delays
CPU Utilization Percentage
25
20
15
10
5
0
0
32
64
121
242
WAN Delay (msec)
Compression Off, Packing 0
Compression Off, Packing 1
Compression Off, Packing 2
Compression On, Packing 0
Compression On, Packing 1
Compression On, Packing 2


North Memory Utilization at 512Kbps WAN Speeds each
One link with 32msec Delay, the Other with Various Delays
Memory Utilization Percentage
90
80
70
60
50
40
30
20
10
0
0
32
64
121
242
WAN Delay (msec)
Compression Off, Packing 0
Compression Off, Packing 1
Compression Off, Packing 2
Compression On, Packing 0
Compression On, Packing 1
Compression On, Packing 2

South Memory Utilization at 512Kbps WAN Speeds each One link with 32msec Delay, the Other with Various Delays


Memory Utilization Percentage
WAN Delay (msec)
Compression Off, Packing 0
Compression Off, Packing 1
Compression Off, Packing 2
Compression On, Packing 0
Compression On, Packing 1
Compression On, Packing 2