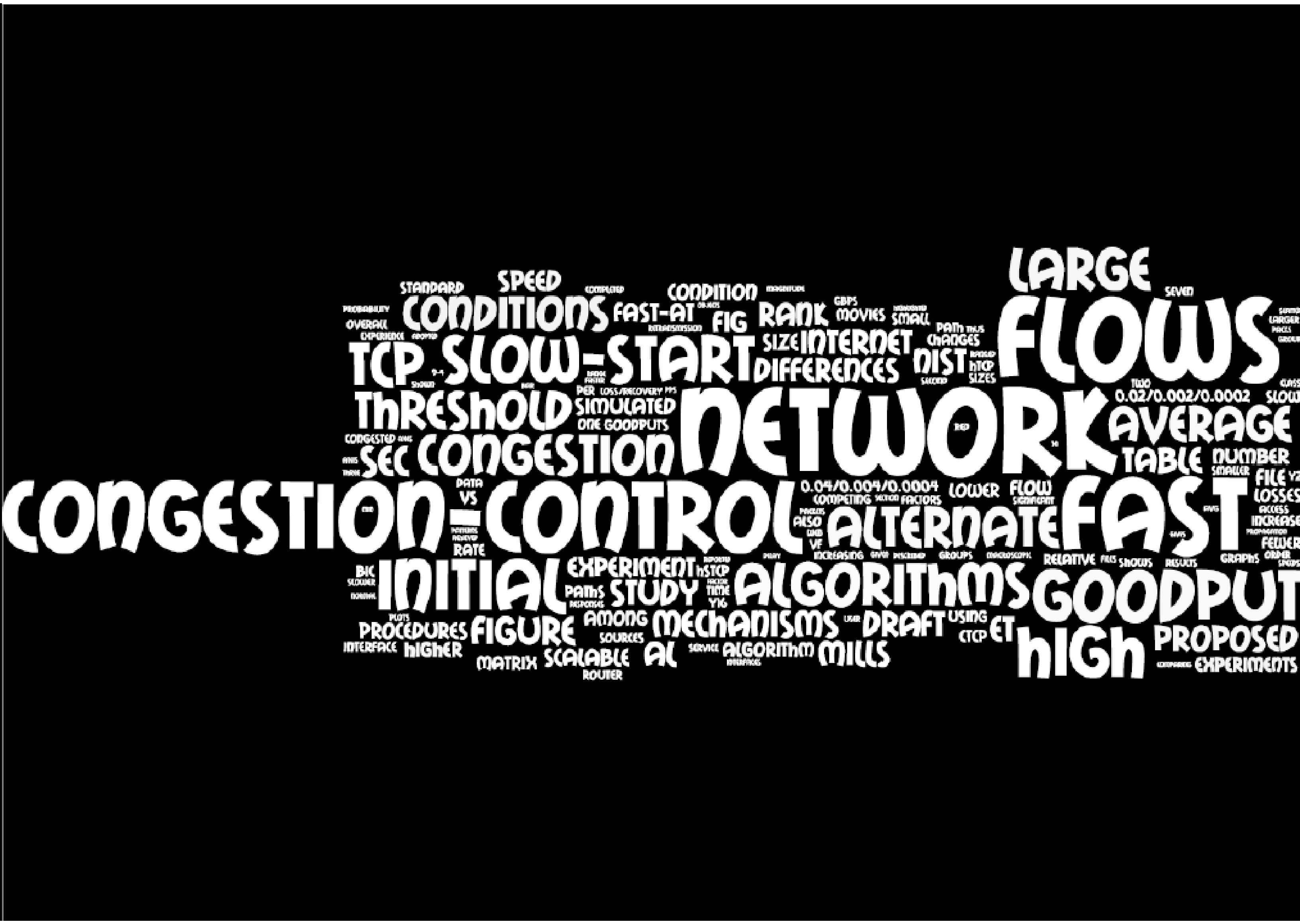
Chapter 9 – Comparing Congestion Control Regimes in a Large, Fast, Heterogeneous Network



9 Comparing Congestion Control Regimes in a Large, Fast, Heterogeneous Network

In this chapter, we repeat the fundamental experiment design and data analyses described in the previous chapter (Chapter 8), while increasing the scale (speed and size) of the simulated network by one order of magnitude. Because increasing network scale will also increase the computational resources needed for executing the experiments, we decided to limit the simulations to cases where the initial slow-start threshold is set high. We made this choice in order to focus on the loss/recovery aspects of the alternate congestion control algorithms.

Table 9-1. Comparison of Experiment with Congestion Control Algorithms in a Small Network (Chapter 8) vs. Experiment in a Large, Fast Network (Chapter 9)

Characteristic	Chapter 8 High SST	Chapter 9 High SST
Network Size (sources)	17.455×10^{-3} & 26.085×10^{-3}	174.6×10^{-3} & 261.792×10^{-3}
Backbone Speed (Gbps)	19.2 & 38.4	192 & 384
Packet Loss Rate	1×10^{-4} to 1×10^{-2}	2×10^{-9} to 2×10^{-2}
Initial Slow-Start Threshold	$2^{32}/2$ packets	$2^{32}/2$ packets
Alternate Congestion Control Algorithms & Associated Identifiers	1-BIC, 2-CTCP, 3-FAST, 4-FAST-AT, 5-HSTCP, 6-HTCP, 7-Scalable	1-BIC, 2-CTCP, 3-FAST, 4-FAST-AT, 5-HSTCP, 6- HTCP, 7-Scalable
Ratio (%) of Sources using Alternate Congestion-Control to Standard TCP Congestion-Control	30:70 & 70:30	30:70 & 70:30
Scenario	60 min. – 96-98% Web objects; 2-4% document transfers; smaller number of service-pack and movie downloads	60 min. – 96-98% Web objects; 2-4% document transfers; smaller number of service-pack and movie downloads

Table 9-1 highlights in red differences from the relevant experiment reported in Chapter 8. As indicated, we compared the seven congestion control algorithms under the same mix of sources with the same traffic patterns as used in Chapter 8. We also set the initial slow-start threshold to a high value and simulated network operation for one hour. As the table shows, we increased the number of sources and network speed tenfold. One ramification of increasing network speed is to extend the range of congestion conditions, as measured by packet-loss rate. Specifically, the experiment conditions in Chapter 9 led to five orders of magnitude lower congestion for the least congested case. Note, however, that the experiments in both Chapters 8 and 9 have the same order of losses under the condition with highest congestion. To the extent that faster network speeds permit lower congestion, and thus fewer losses, we expected the performance of the alternate algorithms to become closer to each other, and to the performance of standard TCP congestion control. This follows from the fact that in these experiments only loss/recovery procedures can distinguish the alternate algorithms from each other and from TCP. Fewer losses equate to fewer chances to distinguish among the various congestion control algorithms.

9.1 Changes in Experiment Design

Except as described in this section, we adopted the same parameter settings used for the experiment reported in Chapter 8. Below, we discuss the few changes we made in robustness factors and fixed factors and we report the resulting experiment conditions. We then describe how these few changes affected the domain view of the experiment conditions. We close with a recap of responses recorded.

9.1.1 Changes in Robustness Factors and Fixed Factors

Table 9-2 specifies the robustness factors and values we used for this experiment. Highlighted in red are the only changes from Chapter 8 – we multiplied the network speed settings by 10. Table 9-3 identifies (in red) the only change we made to the fixed factors used in Chapter 8. We multiplied the base number of sources by 10. These two changes led to the desired order of magnitude increase in network speed and size.

Table 9-2. Robustness Factors Adopted for Comparing Congestion Control Mechanisms (Changes from Chapter 8 highlighted in red)

Identifier	Definition	PLUS (+1) Value	Minus (-1) Value
x1	Network Speed	16000 p/ms	8000 p/ms
x2	Propagation Delay Multiplier	2	1
x3	Buffer Size Adjustment Factor	1	0.5
x4	Think Time	7500 ms	5000 ms
x5	Average File Size for Web Objects	150 packets	100 packets
x6	Distribution for Sizing Large Files	2	1
x7	Probability of Fast Source	.7	.3
x8	Probability of Alternate Congestion-Control Algorithm	.7	.3
x9	Multiplier on Base Number of Sources (ΔU)	3	2

Table 9-3. Key Fixed Factors Adopted for Comparing Congestion Control Mechanisms (Change from Chapter 8 highlighted in red)

Parameter	Definition	Value
Bsources	Basic unit for sources per access router	1000
P(Ns)	Probability source under normal access router	0.1
P(Nsf)	Probability source under fast access router	0.6
P(Nsd)	Probability source under directly connected access router	0.3
P(Nr)	Probability receiver under normal access router	0.6
P(Nrf)	Probability receiver under fast access router	0.2
P(Nrd)	Probability receiver under directly connected access router	0.2
sst_{INT}	Initial slow-start threshold (packets)	$2^{31}/2$

9.1.2 Changes in Orthogonal Fractional Factorial Design of Robustness Conditions

Increasing network speed caused the experiment conditions to change only with respect to a single factor (x1). The resulting 32 experiment conditions are shown in Table 9-4.

Table 9-4. Two-Level 2^{9-4} Orthogonal Fractional Factorial Design (Changes from Chapter 8 highlighted in red)

Factor->	x1	x2	x3	x4	x5	x6	x7	x8	x9
Condition	--	--	--	--	--	--	--	--	--
1	8000	1	0.5	5000	100	0.04/0.004/0.0004	0.7	0.7	3
2	16000	1	0.5	5000	100	0.04/0.004/0.0004	0.3	0.3	2
3	8000	2	0.5	5000	100	0.02/0.002/0.0002	0.7	0.3	2
4	16000	2	0.5	5000	100	0.02/0.002/0.0002	0.3	0.7	3
5	8000	1	1	5000	100	0.02/0.002/0.0002	0.3	0.7	2
6	16000	1	1	5000	100	0.02/0.002/0.0002	0.7	0.3	3
7	8000	2	1	5000	100	0.04/0.004/0.0004	0.3	0.3	3
8	16000	2	1	5000	100	0.04/0.004/0.0004	0.7	0.7	2
9	8000	1	0.5	7500	100	0.02/0.002/0.0002	0.3	0.3	3
10	16000	1	0.5	7500	100	0.02/0.002/0.0002	0.7	0.7	2
11	8000	2	0.5	7500	100	0.04/0.004/0.0004	0.3	0.7	2
12	16000	2	0.5	7500	100	0.04/0.004/0.0004	0.7	0.3	3
13	8000	1	1	7500	100	0.04/0.004/0.0004	0.7	0.3	2
14	16000	1	1	7500	100	0.04/0.004/0.0004	0.3	0.7	3
15	8000	2	1	7500	100	0.02/0.002/0.0002	0.7	0.7	3
16	16000	2	1	7500	100	0.02/0.002/0.0002	0.3	0.3	2
17	8000	1	0.5	5000	150	0.02/0.002/0.0002	0.3	0.3	2
18	16000	1	0.5	5000	150	0.02/0.002/0.0002	0.7	0.7	3
19	8000	2	0.5	5000	150	0.04/0.004/0.0004	0.3	0.7	3
20	16000	2	0.5	5000	150	0.04/0.004/0.0004	0.7	0.3	2
21	8000	1	1	5000	150	0.04/0.004/0.0004	0.7	0.3	3
22	16000	1	1	5000	150	0.04/0.004/0.0004	0.3	0.7	2
23	8000	2	1	5000	150	0.02/0.002/0.0002	0.7	0.7	2
24	16000	2	1	5000	150	0.02/0.002/0.0002	0.3	0.3	3
25	8000	1	0.5	7500	150	0.04/0.004/0.0004	0.7	1	2
26	16000	1	0.5	7500	150	0.04/0.004/0.0004	0.3	0.3	3
27	8000	2	0.5	7500	150	0.02/0.002/0.0002	0.7	0.3	3
28	16000	2	0.5	7500	150	0.02/0.002/0.0002	0.3	0.7	2
29	8000	1	1	7500	150	0.02/0.002/0.0002	0.3	0.7	3
30	16000	1	1	7500	150	0.02/0.002/0.0002	0.7	0.3	2
31	8000	2	1	7500	150	0.04/0.004/0.0004	0.3	0.3	2
32	16000	2	1	7500	150	0.04/0.004/0.0004	0.7	0.7	3

9.1.3 Changes in Domain View of Robustness Conditions

Changes in speed and size influence the domain view of our simulated network, as reported in Tables 9-5 through 9-7, where changes from the experiment in Chapter 8 are highlighted in red. Table 9-5 shows simulated router speeds for this experiment, which are comparable to speeds that might be seen in contemporary networks. Increasing **Bsources** (base number of sources) to 10^3 scales the number of potentially active flows to a level that matches the simulated network speeds. Table 9-6 shows the number of sources for each level of factor x9 (multiplier on **Bsources**). The number of receivers is four times the number of sources. We used the same topology, including propagation delays, as in previous experiments. Buffer sizing is influenced by three factors: network speed (x1), propagation delay (x2) and buffer-size adjustment factor (x3). Table 9-7 characterizes buffer sizes for each router level under both values for factor x3.

Table 9-5. Simulated Router Speeds

Router	PLUS (+1)	Minus (-1)
Backbone	384 Gbps	192 Gbps
POP	48 Gbps	24 Gbps
Normal Access	4.8 Gbps	2.4 Gbps
Fast Access	9.6 Gbps	7.2 Gbps
Directly Connected Access	48 Gbps	24 Gbps

Table 9-6. Number of Simulated Sources

PLUS (+1)	Minus (-1)
261.792×10^3	174.6×10^3

Table 9-7. Characterization of Simulated Buffer Sizes

	x3 = 1.0			x3 = 0.5		
Router	Min	Avg	Max	Min	Avg	Max
Backbone	651×10^3	1.464×10^6	2.604×10^6	326×10^3	732×10^3	1.302×10^6
POP	81×10^3	183×10^3	325×10^3	41×10^3	92×10^3	163×10^3
Access	13×10^3	29×10^3	52×10^3	6.5×10^3	14.6×10^3	25.9×10^3

Fig. 9-1 plots the retransmission rates for each of the 32 simulated conditions. The x axis is ordered by increasing retransmission rate. Using visual guidance, we divided congestion conditions into six categories moving from little congestion (**C1**) to relatively high congestion (**C6**). Except for the highest congestion category (**C6**), the simulated conditions exhibit several orders of magnitude reduction in congestion when compared with the experiments in Chapter 8 (recall Figs. 8-1 and 8-2).

To further explore the nature of congestion under the conditions simulated for this experiment, we examined six time series. We chose one condition from the middle of each congestion class. Fig. 9-2 plots related time series. We selected the following conditions, one from each congestion class C1 through C6: 4, 6, 31, 7, 29 and 19. The y axis indicates the number of flows in a particular state: connecting (gold) or active (red). Active flows may be operating in initial slow start (green), normal congestion avoidance (brown) or alternate congestion avoidance (blue). In these particular plots, CTCP flows were operating in the network along with flows using standard TCP congestion control procedures. The discussion considers only the relative distances between the curves on the graphs, so inability to read the axes will be immaterial. The number of active flows generally appears to be on the order of 10^4.

Under the least congested condition (4), nearly all active flows operate in initial slow-start, and few losses occur. In general, as congestion increases with condition, the relative number of active flows in initial slow-start decreases and the relative number under normal congestion avoidance procedures increases. That is, the green and brown lines come closer together.[1] The number of flows under alternate congestion avoidance procedures (blue) shifts up or down slightly depending on whether a particular condition has 30 % or 70 % of the sources equipped with an alternate congestion control algorithm.

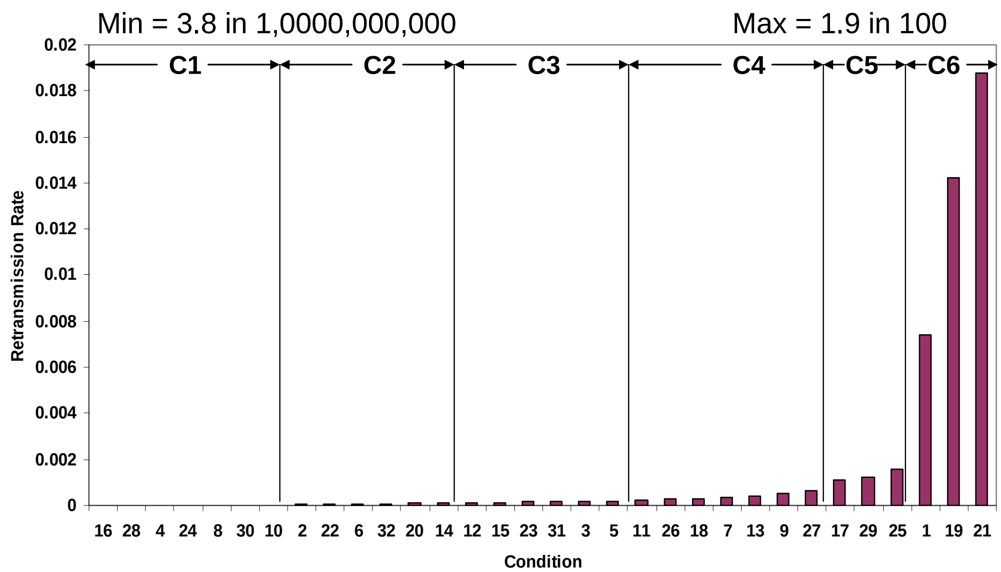


Figure 9-1. Conditions Ordered Least to Most Congested under High Initial Slow-Start Threshold

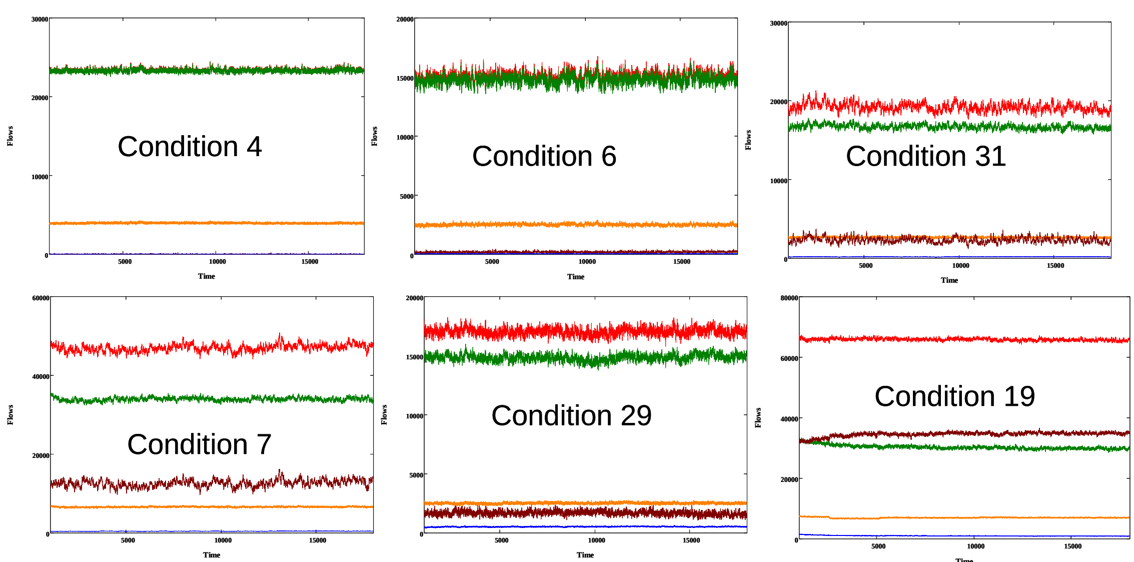


Figure 9-2. Distribution of Flow States for Six Conditions with Increasing Congestion (Flows are either connecting (gold) or sending (red) and sending flows may be in one of three congestion control states: initial slow start (green), normal congestion avoidance (brown) or alternate congestion avoidance (blue)

[1] Note that this trend is not monotonic – the green and brown lines move farther apart as condition advances from 7 to 29. We attribute this to the fact that condition 29 is the only condition among conditions 31, 7, 29 and 19 that has a lower probability of larger file sizes. This means more files can complete in initial slow start under condition 29, than under the other three conditions.

9.1.4 Responses Measured

We measured the same responses for this experiment as we measured for the experiments discussed in Chapter 8. We measured 16 responses characterizing macroscopic behavior of the network and 28 responses representing user experience in each of 24 flow groups. Refer to back Sec. 8.1.4 for a definition of the responses.

9.2 Experiment Execution and Data Collection

Table 9-8 compares resource requirements for simulating the large, fast network against resource requirements for simulating the smaller network used in Chapter 8. Simulating the large, fast network over (7 algorithms x 32 conditions =) 224 runs, required about 11 processor years, compared with only 2/3 of a processor year for simulating the same number of runs given the smaller network. Scaling up the network by an order of magnitude led to increasing computation requirements by a factor of 16 or so. Table 9-9 shows that the increase in the number of packets sent and flows simulated was approximately linear (i.e., tenfold). The higher than linear increase in computation requirements can be attributed to extra processing time associated with managing larger event lists. Since we collected data in the same form as described in Sec. 8.2.2, increasing the scale of the simulation did not increase the amount of data collected.

9.3 Data Analysis Approach

We used the same data analysis approach described in Sec. 8.3. We focused mainly on user experience in each of 24 flow classes (recall Table 8-6), where we investigated both absolute and relative differences. We examined macroscopic data with detailed analyses for each of the 16 responses, applying a Grubbs' test to residuals about the mean associated with each of the 32 conditions.

Table 9-8. Comparing Resource Requirements for Simulating a Small Network (from Chapter 8) and a Large, Fast Network (from Chapter 9)

	Small, Slow Network with High Initial Slow-Start Threshold	Large, Fast Network with High Initial Slow-Start Threshold
CPU hours (224 Runs)	5.857×10^3	94.355×10^3
Avg. CPU hours (per run)	26.15	421.23
Min. CPU hours (one run)	12.58	203.04
Max. CPU hours (one run)	43.97	739.04
Avg. Memory Usage (Mbytes)	196.56	2.392×10^3

Table 9-9. Comparing Number of Simulated Flows and Packets for a Small Network (from Chapter 8) and a Large, Fast Network (from Chapter 9)

	Small, Slow Network with High Initial Slow-Start Threshold		Large, Fast Network with High Initial Slow-Start Threshold	
Statistic	Flows Completed	Data Packets Sent	Flows Completed	Data Packets Sent
Avg. Per Condition	11.466×10^6	3.414×10^9	116.317×10^6	33.351×10^9
Min. Per Condition	7.258×10^6	2.139×10^9	72.945×10^6	21.069×10^9
Max. Per Condition	17.391×10^6	$5.048, \times 10^9$	175.948×10^6	50.932×10^9
Total all Runs	2.568×10^9	764.740×10^9	26.055×10^9	7.471×10^{12}

9.4 Results

In this section, we present selected simulation results in three categories: (1) macroscopic network behavior, (2) absolute user experience and (3) relative user experience. We present only data that reveals behavioral similarities and differences of interest. In some cases, we compare results with results obtained from one of the experiments in Chapter 8. Specifically, we compare results under a high initial slow-start threshold.

9.4.1 Macroscopic Network Behavior

In general, as we found in the earlier experiment (Chapter 8), the data analyses reported in this section do not reveal much in the way of statistically significant changes in macroscopic network behavior. This appears due mainly to the general lack of congestion throughout the experiment conditions. As in the results from Chapter 8, we consider both FAST (algorithm 3) and FAST-AT (algorithm 4) together, which reduces the statistical significance of either algorithm considered alone because both algorithms share some traits (as described previously in Chapter 7). Despite the lack of statistical significance, we could discern patterns in macroscopic network behavior with respect to some responses. In most cases, the patterns detected here echo patterns seen in Chapter 8 under a high initial slow-start threshold. The patterns appeared less distinct in the current experiments because overall levels of congestion were much lower across most of the 32 simulated conditions. We report the patterns we found informative.

Fig. 9-3 shows the average number of flows attempting to connect. In the six conditions with highest congestion (17, 29, 25, 1, 19 and 21), FAST and FAST-AT had more flows pending in the connecting state than other algorithms. This was especially so for the three most congested conditions. This result is consistent with results from our other experiments, which showed that FAST and FAST-AT led flows to take longer to connect in the face of significant congestion. Most conditions in the current experiment did not lead to significant congestion, but where significant congestion existed FAST and FAST-AT induced more losses in SYN packets. In addition, as shown in Fig. 9-4, under highly congested conditions, FAST and FAST-AT induced higher retransmission rates. Fig. 9-4 also mirrors results in Fig. 8-36 – under conditions of lower congestion, Scalable TCP induced more losses and retransmissions than other algorithms. Comparing Fig. 9-4 with Fig. 8-36 shows that Scalable TCP induced more losses under more conditions in Fig. 9-4. This should be expected because the current experiment has significantly lower congestion under most conditions than was the case for the previous experiment (Chapter 8).

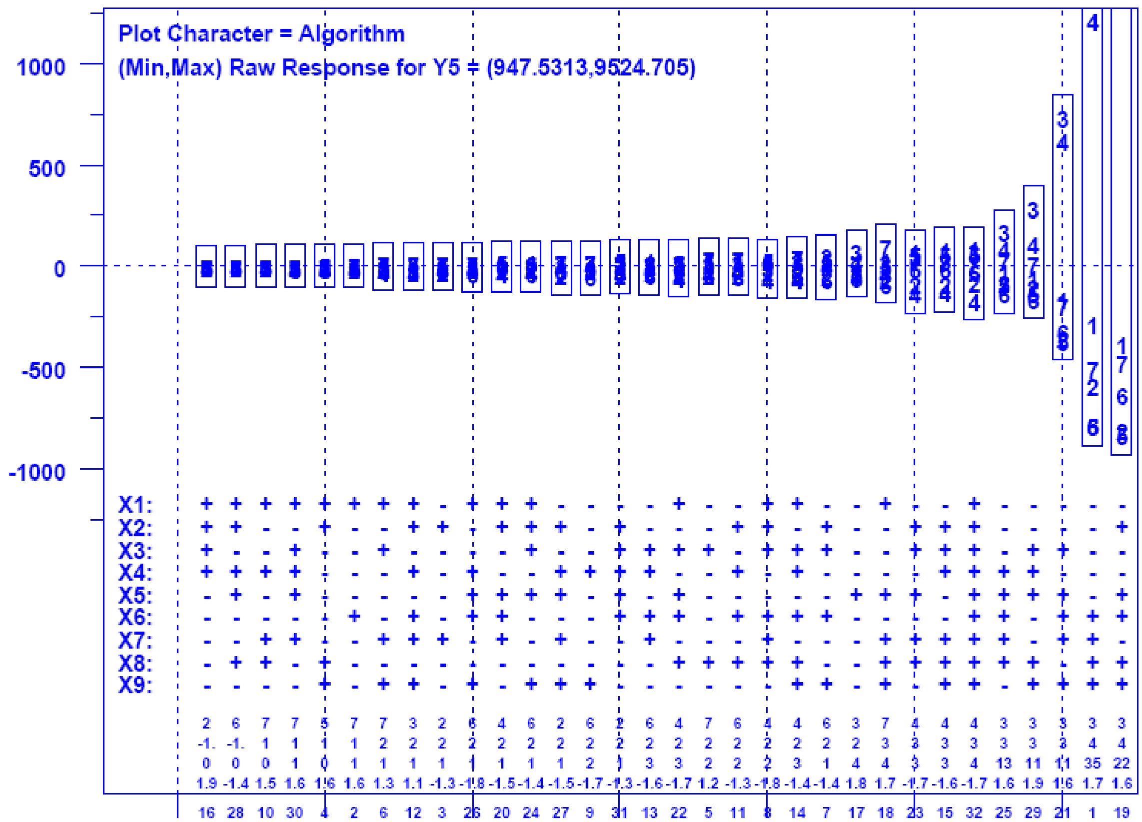


Figure 9-3. Average Number of Connecting Flows under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals.

Fig. 9-5 shows that FAST and FAST-AT completed substantially fewer flows per measurement interval under the three most congested conditions (1, 19 and 21). As shown in Fig. 9-6, the lower flow completion rate for FAST and FAST-AT under severe congestion (conditions 1, 19 and 21) resulted in millions fewer completed flows over the entire simulated hour.

Fig. 9-7 shows that Scalable TCP had a tendency to incur longer smoothed round-trip times, which resulted from larger network packet queues. This echoes results from the previous experiment (Chapter 8), where Scalable TCP round-trip times could be 2-10 ms higher on average than those of other algorithms. Fig. 9-8 shows that, under Scalable TCP, a higher proportion of completed flows were Web objects. Note, however, that the differences in proportion were quite small (most on the order of 10^{-4}). The case with respect to movie transfers is shown in Fig. 9-9. In more than half the simulated conditions, under all algorithms the same proportion of files transferred were movies (highlighted in black in Fig. 9-9). In the remaining conditions, differences were on the order of 10^{-6}. Overall, the differences in proportion of flows completed were very small. We attribute this to the fact that conditions generally exhibited little congestion.

Finally, as shown in Fig. 9-10, CTCP achieved a significant increase in average congestion window. This characteristic also appeared in pervious experiments. The higher network speed available in the current experiment enabled CTCP to achieve a more substantial advantage in average congestion window than reported for the slower network used in Chapter 8 (see Fig. 8-40).

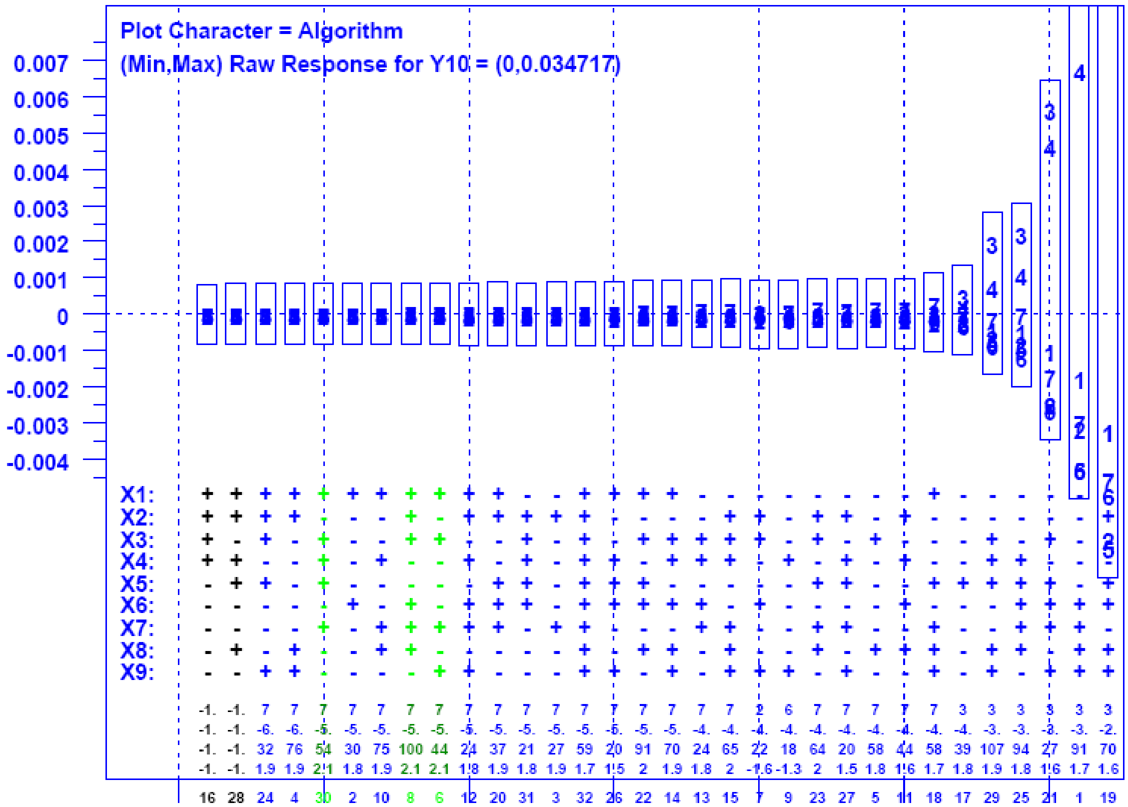


Figure 9-4. Average Retransmission Rate (proportion of packets resent) under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals. Columns highlighted in green indicate significant outliers on the high side, columns highlighted in black indicate no numeric difference measured among the congestion control algorithms and blue columns mean that differences among the congestion control algorithms were not statistically significant.

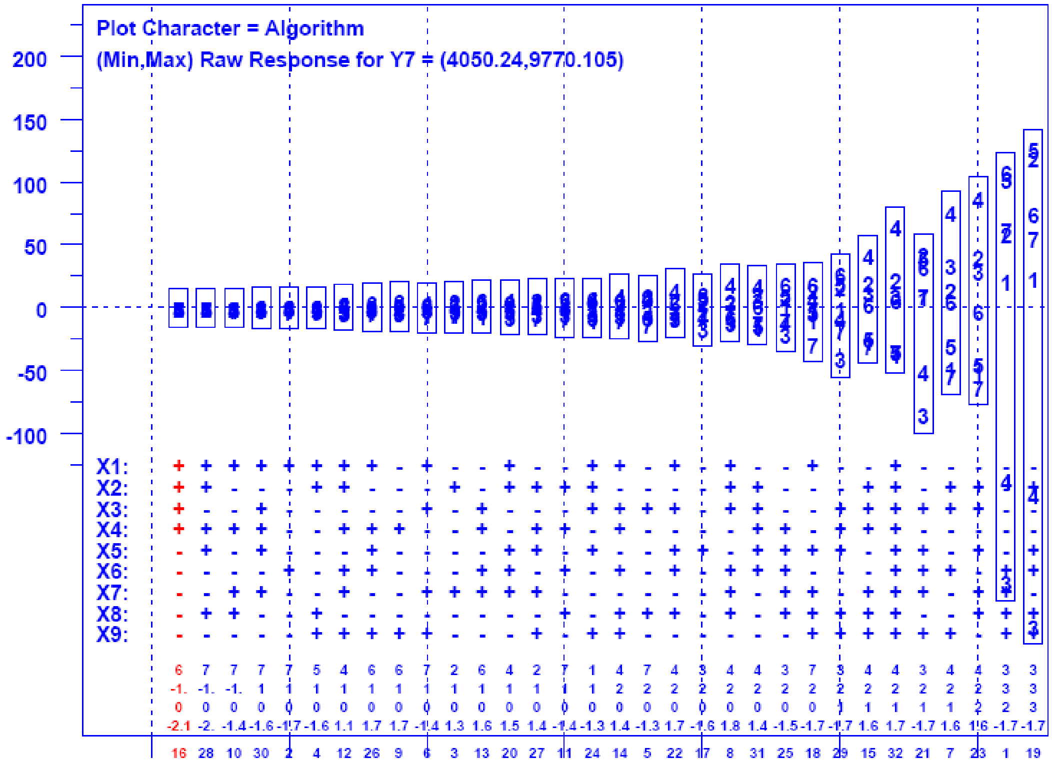


Figure 9-5. Average Flow Completion Rate (flows per 200 ms) under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals. Column highlighted in red denotes a statistically significant outliner on the low side.

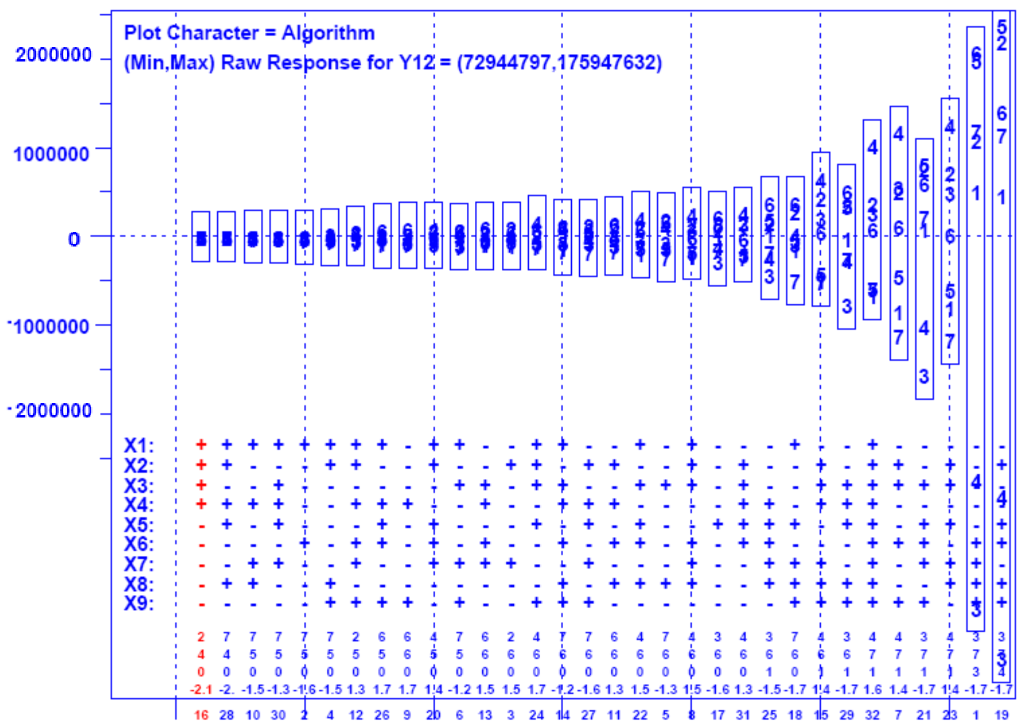


Figure 9-6. Aggregate Flows Completed under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals. Column highlighted in red denotes a statistically significant outlier on the low side.

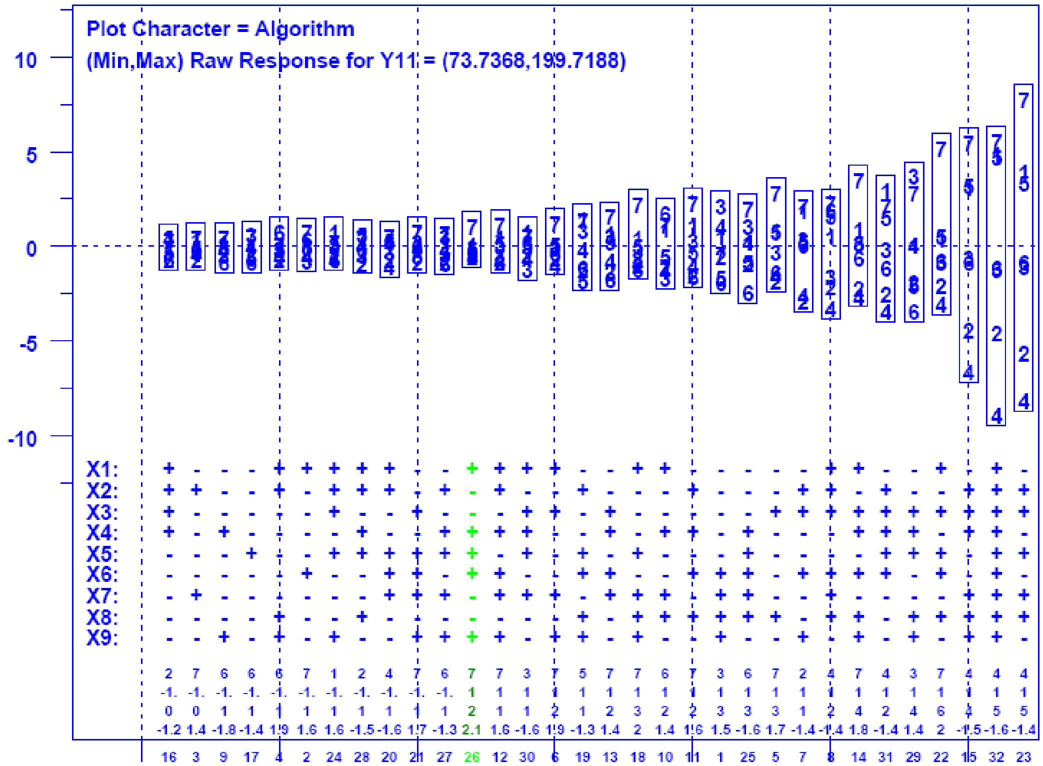


Figure 9-7. Average Smoothed Round-Trip Time (ms) under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals. Column highlighted in green denotes a statistically significant outlier on the high side.

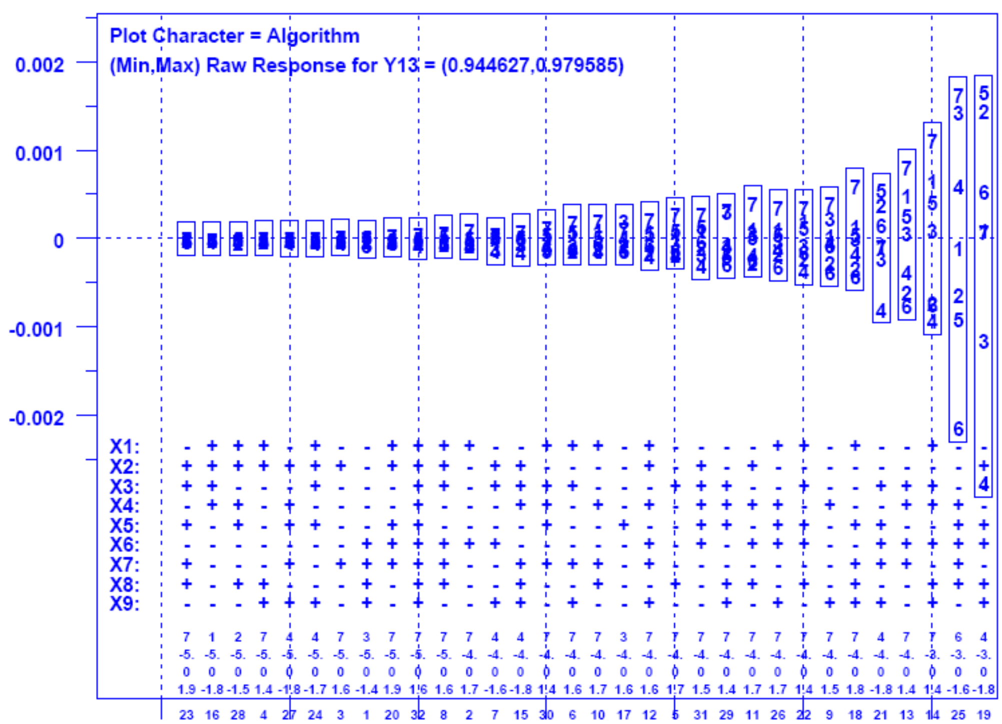


Figure 9-8. Web Objects as Proportion of Flows Completed under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals.

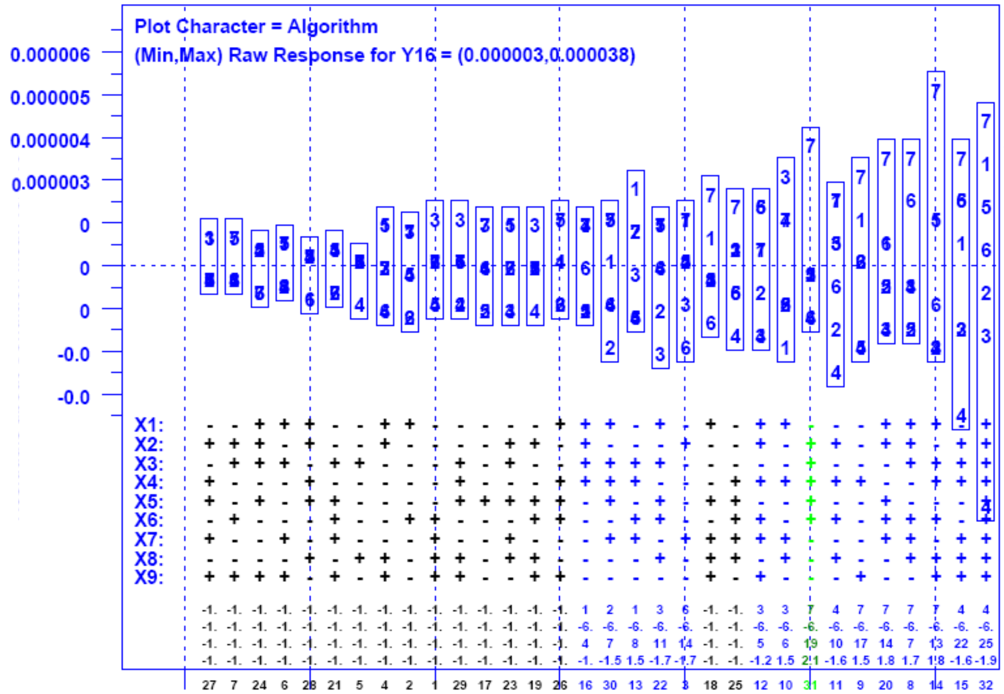


Figure 9-9. Movies as Proportion of Flows Completed under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals. Column highlighted in green indicates significant outlier on the high side, columns highlighted in black indicate no numeric difference measured among the congestion control algorithms.

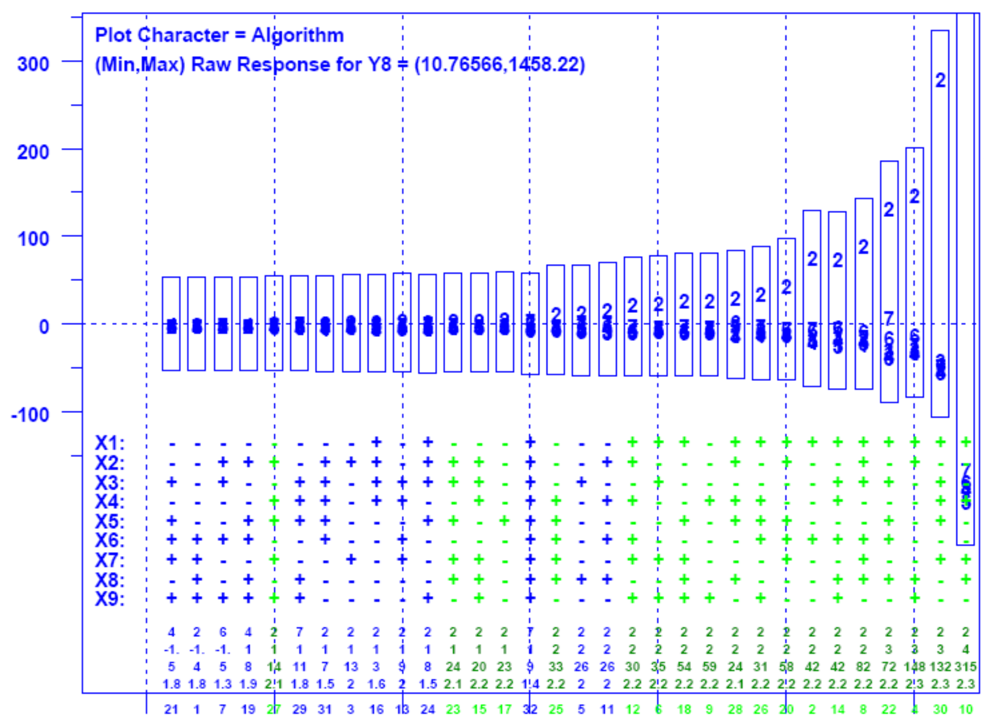


Figure 9-10. Average Flow Congestion Window Size (packets) under High Initial Slow-Start Threshold – y axis gives residuals around the mean value for each condition and x axis gives conditions ordered by increasing range of residuals. Columns highlighted in green indicate significant outliers on the high side.

9.4.2 Absolute User Experience

Table 9-10 summarizes the average goodput – response y2(u) – experienced by users in each of the 24 flow groups (dimensioned by file size, path class and interface speed) under each of the seven alternate congestion control algorithms. Table 9-11 provides a similar summary of the average goodput – response y16(u) – experienced by TCP users in each of the 24 flow groups when competing with flows in each of the seven alternate congestion control algorithms.

Table 9-10. Average Goodput (pps) per Flow Group under Each Alternate Congestion Control Algorithm for a Large, Fast Network with High Initial Slow-Start Threshold – file sizes include movies (M), service packs (SP), documents (D) and Web objects (WO); path classes include very fast (VF), fast (F) and typical (T); interface speeds include fast (F) and normal (N)

			ALTERNATE CONGESTION-CONTROL ALGORITHM						
File	Path	Interface	BIC	CTCP	FAST	FAST-AT	HSTCP	HTCP	STCP
M	VF	F	60728	60595	60778	60745	60574	60665	60319
		N	7920	7923	7933	7919	7929	7928	7930
	F	F	32495	29893	29981	29094	31177	29745	34822
		N	6793	6236	6865	6603	6798	6609	7091
	T	F	30155	26632	27177	26672	29328	26451	32629
		N	6669	6258	6674	6478	6643	6266	6954
SP	VF	F	30551	30876	30568	30628	30719	30594	30646
		N	7440	7429	7434	7431	7430	7431	7427
	F	F	16381	16273	16426	16675	16327	16250	16920
		N	6174	6029	6302	6338	6111	6012	6222
	T	F	17772	17262	17692	17794	17607	17151	18160
		N	6226	6112	6393	6424	6174	6047	6295
D	VF	F	2377	2375	2368	2372	2360	2360	2377
		N	1942	1939	1941	1946	1944	1949	1937
	F	F	1434	1438	1427	1438	1440	1449	1436
		N	1257	1264	1252	1262	1260	1268	1257
	T	F	1774	1786	1770	1786	1782	1795	1781
		N	1513	1527	1514	1526	1519	1532	1516
WO	VF	F	448	450	451	457	449	451	451
		N	424	424	425	424	426	425	426
	F	F	278	279	276	279	279	280	278
		N	267	268	266	268	268	269	267
	T	F	348	351	347	349	350	352	349
		N	332	335	331	333	334	336	333

Since the tables are somewhat dense with numbers, we present this information in the form of bar graphs (Fig. 9-11 through 9-14) – one figure per file size: movie (M), service pack (SP), document (D) and Web object (WO). (The legend for the bar graphs is shown in Fig. 8-27.) The top row of graphs in each figure displays the average goodput in packets per second (pps) achieved in a large, fast network with a high initial slow-start threshold, while, for comparison, the bottom row of graphs displays average goodput achieved in a smaller, slower network with high initial slow-start threshold (as reported previously in Sec. 8.4.2.1). When examined vertically, the first two columns of graphs consider flows transiting very fast (VF) paths, the second two columns consider flows transiting fast (F) paths and the final two columns consider flows transiting typical (T) paths. Within a given path class, the first vertical sub-column reports goodput for flows with fast (F) interface speeds (80 x 10^3 pps), while the second vertical sub-column reports goodput for flows with normal (N) interface speeds (8 x 10^3 pps). Each column of graphs is labeled with the relevant path class and interface speed (e.g., VF-F).

Table 9-11. Average Goodput (pps) per Flow Group on TCP Flows Competing with Each Alternate Algorithm for a Large, Fast Network with High Initial Slow-Start Threshold – file sizes include movies (M), service packs (SP), documents (D) and Web objects (WO); path classes include very fast (VF), fast (F) and typical (T); interface speeds include fast (F) and normal (N)

			ALTERNATE CONGESTION-CONTROL ALGORITHM						
File	Path	Interface	BIC	CTCP	FAST	FAST-AT	HSTCP	HTCP	STCP
M	VF	F	60576	60463	60718	60754	60813	60454	60627
		N	7926	7930	7924	7930	7931	7935	7932
	F	F	27158	27312	27828	28014	27842	28280	27318
		N	5962	6005	5975	6017	6109	5961	5847
	T	F	24246	25378	24303	24974	24830	25232	24859
		N	5809	6017	5841	5899	5864	5887	5770
SP	VF	F	30852	30692	30657	30714	30819	30767	30944
		N	7439	7440	7443	7448	7438	7446	7440
	F	F	15578	16188	15827	16046	15741	15986	15758
		N	5787	5922	5779	5834	5828	5893	5728
	T	F	16620	16966	16812	16936	16801	16973	16717
		N	5795	5975	5847	5895	5852	5955	5743
D	VF	F	2383	2383	2386	2367	2391	2380	2378
		N	1950	1947	1943	1947	1946	1947	1960
	F	F	1433	1439	1421	1430	1440	1443	1433
		N	1254	1264	1242	1256	1260	1268	1253
	T	F	1766	1784	1754	1771	1779	1794	1773
		N	1507	1527	1498	1512	1518	1532	1509
WO	VF	F	456	451	451	452	451	455	453
		N	426	428	427	427	427	427	427
	F	F	278	280	276	279	279	281	279
		N	267	269	265	268	268	270	268
	T	F	348	351	346	349	350	353	349
		N	332	335	330	333	334	336	333

Figs. 9-11 through 9-14 reveal two main points. First, in a larger, faster network, flows for large files (movies and service packs) over fast interfaces (80 x 10^3 pps) achieve significantly higher average goodputs than similar flows in a smaller, slower network. Second, average goodputs achieved by competing TCP flows in a larger, faster network appear closer to average goodputs achieved by competing TCP flows in a smaller, slower network. These two points appear due to generally reduced congestion in the larger, faster network. Recall that under a high initial slow-start threshold any goodput differences result from loss/recovery processing because all flows use the same algorithm to accelerate to the initial maximum transfer rate. Lower overall congestion leads to fewer losses per flow, which means that all flows achieve higher goodputs and that alternate congestion control algorithms have fewer opportunities to invoke their loss/recovery procedures.

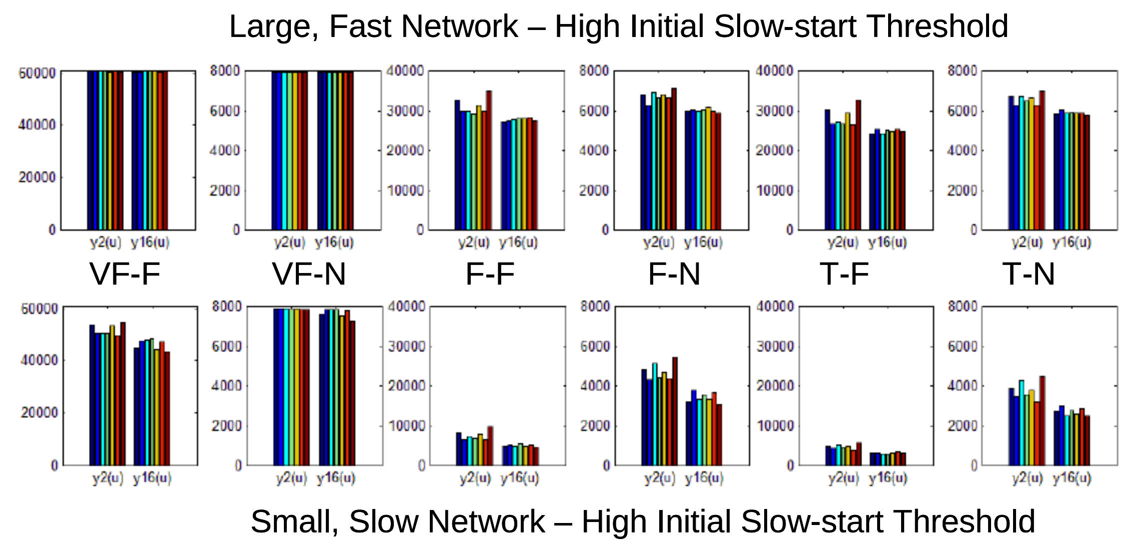


Figure 9-11. Average Goodputs (pps) on Movies under Combinations of Path Class and Interface Speed (Large Fast Network vs. Small Slow Network)

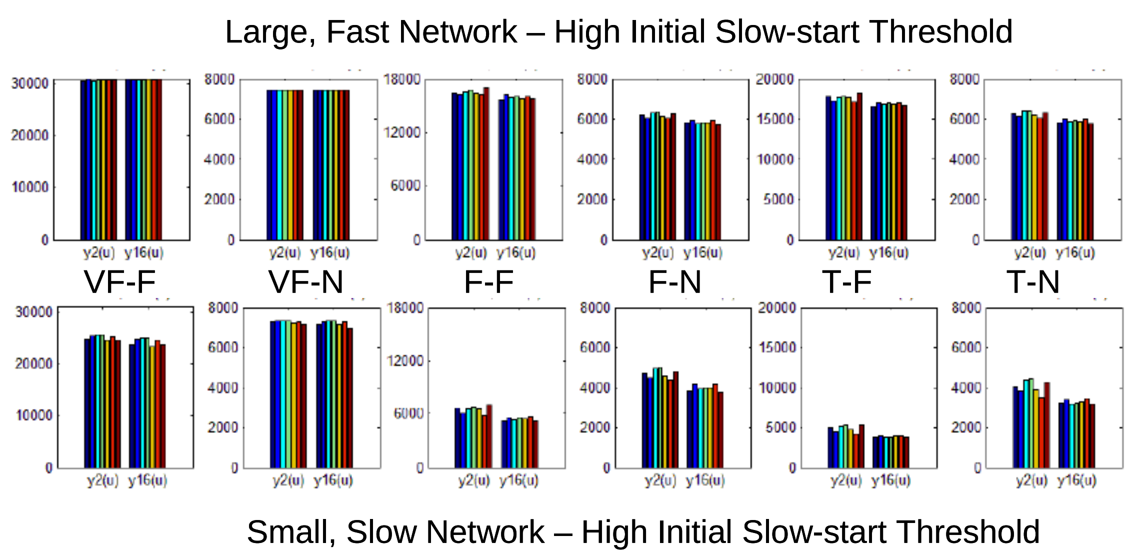


Figure 9-12. Average Goodputs (pps) on Service Packs under Combinations of Path Class and Interface Speed (Large Fast Network vs. Small Slow Network)

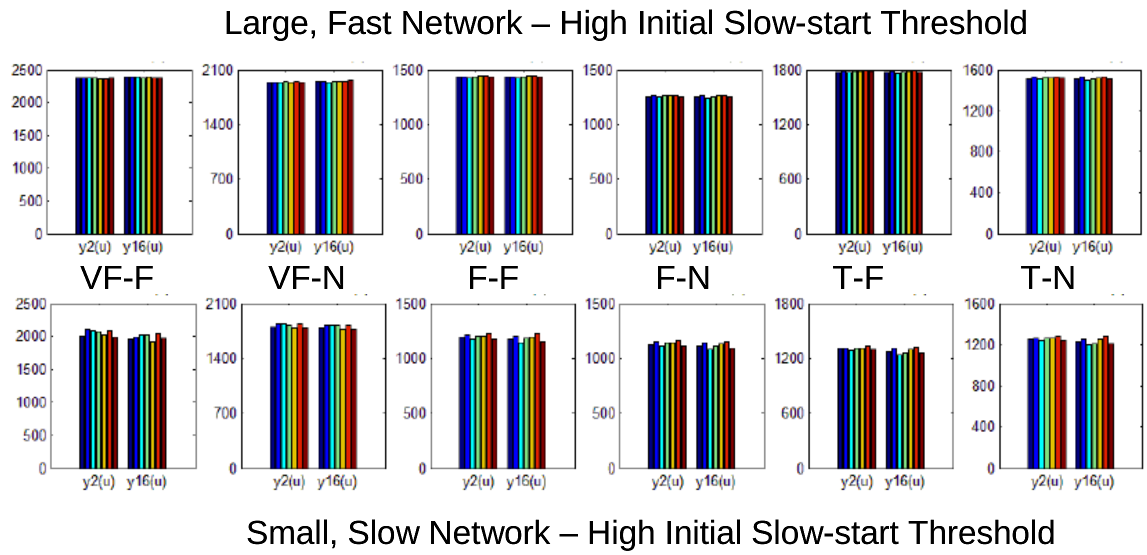


Figure 9-13. Average Goodputs (pps) on Documents under Combinations of Path Class and Interface Speed (Large Fast Network vs. Small Slow Network)

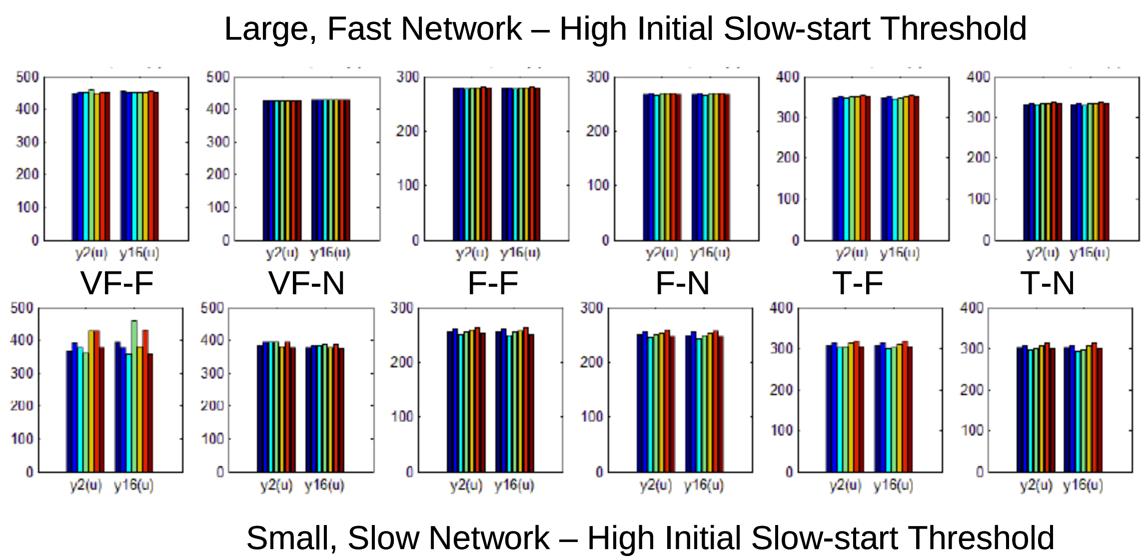


Figure 9-14. Average Goodputs (pps) on Web Objects under Combinations of Path Class and Interface Speed (Large Fast Network vs. Small Slow Network)

Given the similarity in goodput for flows with the same file size, regardless of whether using standard TCP or alternate congestion control procedures, we decided to see if factors other than file size influenced goodput on flows. To investigate, we conducted a principal components analysis (PCA) of the average goodput data across all flow groups. Fig. 9-15 plots the resulting information, which reveals four main groups: (1) a group where network speed is higher (x1 = 1), (2) a group where network speed is lower (x1 = -1), (3) a group where propagation delay is higher (x2 = 1) and (4) a group where propagation delay is lower (x2 = -1). Within each group, two subgroups appear: (1) a subgroup where file sizes are larger (x5 = 1) and (2) a subgroup where file sizes are smaller (x5 = -1). Thus, PCA reveals that differences in flow goodput are influenced mainly by network speed, propagation delay and file size – not by congestion control algorithm.

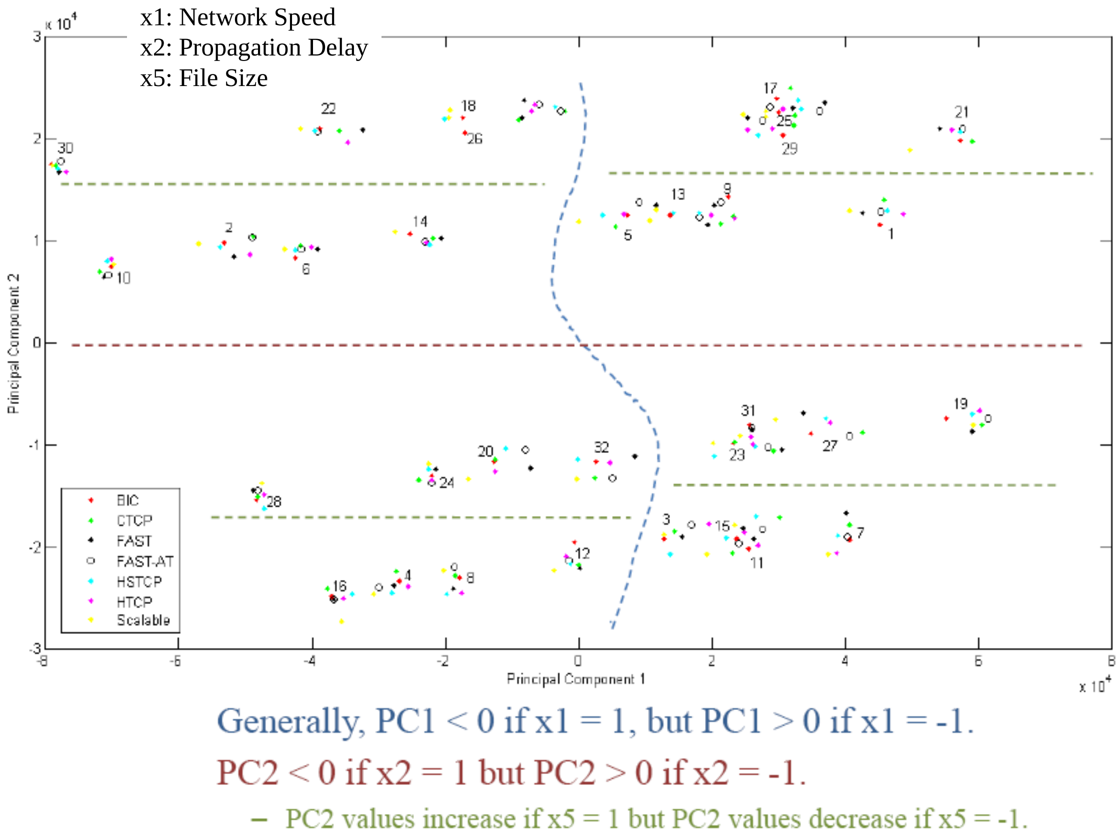


Figure 9-15. Principal Component 1 (x axis) vs. Principal Component 2 (y axis) from Average Goodput Data in a Large, Fast Network with High Initial Slow-Start Threshold – the blue dashed line separates (but not crisply) PC1 values for a network with higher (left) and lower (right) propagation delays, the red dashed line separates PC2 values with higher (top) and lower (bottom) network speeds, and the green dashed lines subdivide the two PC2 areas (one above and one below the red line) by file size: larger (above the green lines) and smaller (below the green lines)

In experiments reported in Chapter 8, we found that under conditions with higher congestion flows using several alternate congestion control algorithms (e.g., BIC, HSTCP and Scalable TCP) had significantly higher goodput than competing TCP flows. Given the generally lower overall congestion when simulating a larger, faster network, can such differences still be discerned? To investigate, we used scatter plots and per-condition bar graphs, as introduced in Sec. 8.3.2. Fig. 9-16 gives seven scatter plots, each showing TCP goodput (y axis) vs. goodput of an alternate (as labeled) congestion control algorithm for movies transferred on very fast paths with a fast interface speed. The scatter plots show no significant difference in goodput for TCP flows vs. flows using alternate congestion control algorithms. Fig. 9-17, which gives differences in goodput between TCP flows and alternate congestion control algorithms under each of 32 simulated conditions, also shows no significant differences. The lack of differences can be attributed to the fact that very fast paths exhibit little congestion, which means that few losses occur and so one should expect little difference in flow goodputs regardless of congestion control algorithm.

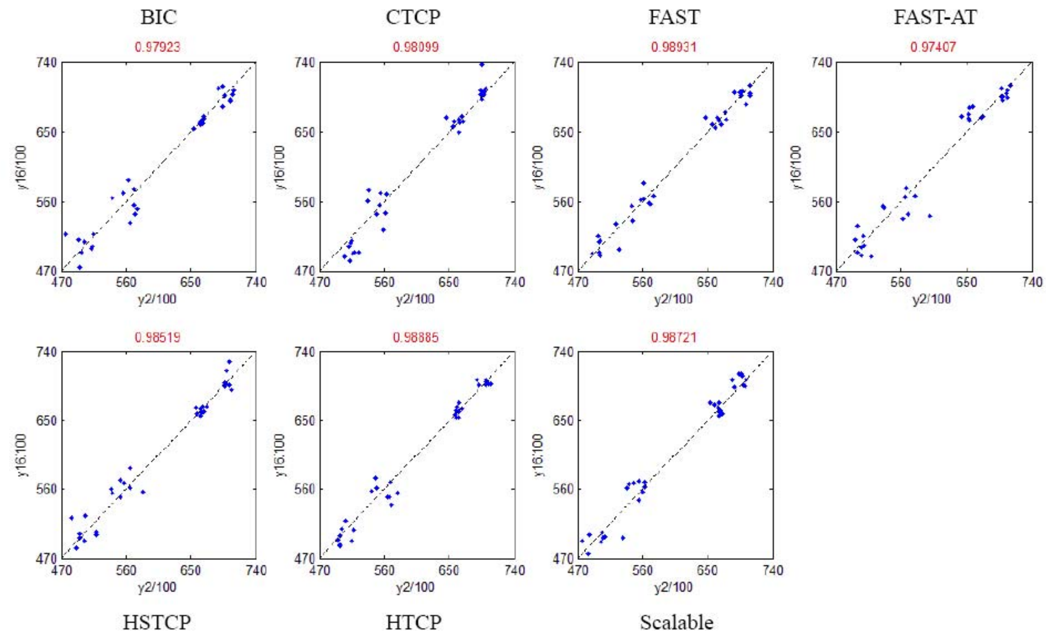


Figure 9-16. Goodput on TCP Flows (y axes give y16u/100 pps) vs. Non-TCP Flows (x axes give y2u/100 pps) for Movies on Very Fast Paths with Fast Interfaces in a Large, Fast Network with High Initial Slow-Start Threshold

When we examine path classes with higher likelihood of congestion, BIC, HSTCP and Scalable TCP flows have a goodput advantage over standard TCP flows on very large files – i.e., movies. For example, Fig. 9-18 shows related scatter plots that reveal the tendency of alternate congestion control algorithms to have better goodputs than TCP flows. As in Chapter 8, the effect is most pronounced for BIC, HSTCP and Scalable TCP. This occurs because large files have a tendency to accumulate more losses on more congested paths, which allows for the loss/recovery procedures of the alternate congestion control algorithms to be activated more often. As previously shown, BIC, HSTCP and Scalable TCP tend to resist lowering transmission rate on sporadic losses, so flows using those regimes achieve significantly higher goodputs vs. TCP flows, which reduce their transmission rate in half on each loss. Fig. 9-19 suggests that the advantage of the alternate congestion control algorithms over TCP tends to increase with increasing congestion, at least until congestion becomes so pervasive that all flows suffer significant reductions in goodput.

The advantage of alternate congestion control algorithms decreases with decreasing file size because there are fewer packets on each flow to incur losses. This effect can be seen in the scatter plots in Fig. 9-20 for service packs sent over fast paths with fast interfaces and in the accompanying bar graphs plotted in Fig. 9-21. Notice that Fig. 9-21 confirms that alternate congestion control algorithms can achieve better goodputs than TCP flows as congestion increases, as seen in conditions 26, 18, 27, 9, 15 and 17.

Table 9-12 gives a summary of goodput differences as percentages for each of the 24 flow groups measured. Differences under a smaller, slower network with a high initial slow-start threshold are reported (taken from Table 8-30) in three columns: (1) **AMONG ALTs** gives the range of percentage difference between flows using the alternate congestion control algorithms with the highest and lowest average goodput; (2) **AMONG TCPs** gives the range of percentage difference between TCP flows with the highest and lowest average goodput when competing with alternate congestion control algorithms; (3) **ALTs > TCPs** gives the percentage increase in average goodput for flows using alternate congestion control algorithms over competing TCP flows (note that when given in red, TCP flows achieved higher average goodput). A similar set of three columns reports goodput differences under a large, fast network with high initial slow-start threshold.

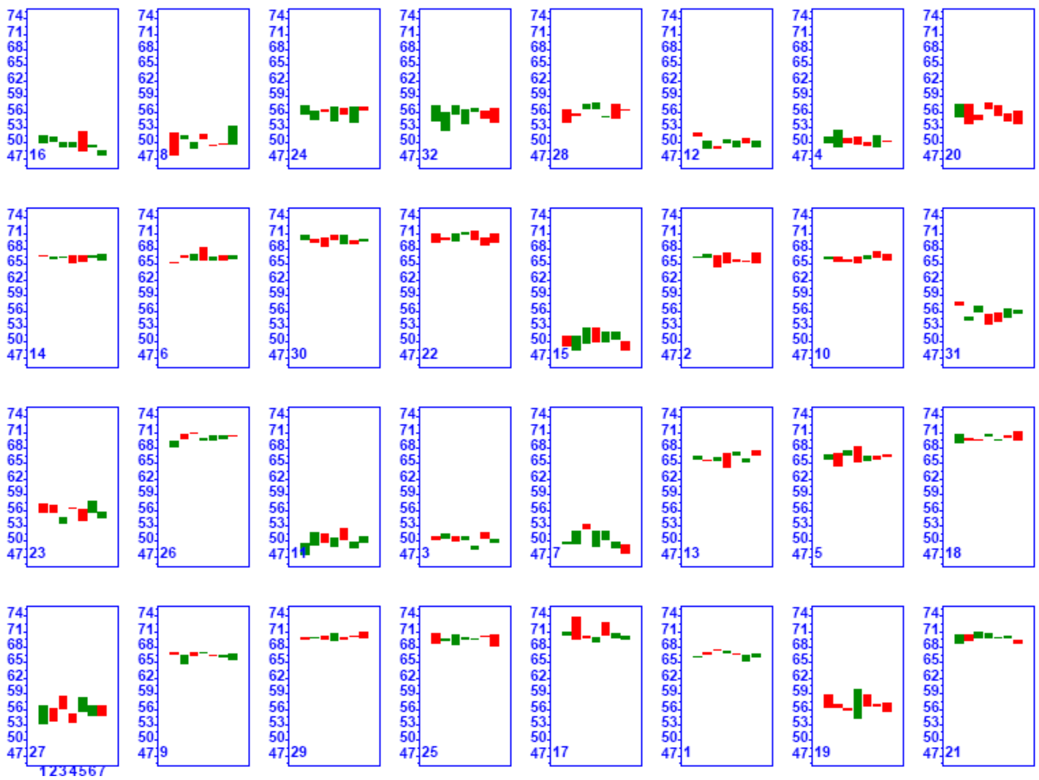


Figure 9-17. Bar Graphs (one for each simulated condition) plotting Goodput Differences (pps/1000) on TCP Flows vs. Non-TCP Flows for Movies Transferred on Very Fast Paths with Fast Interfaces in a Large, Fast Network with a High Initial Slow-Start Threshold (Each graph contains seven bars, one per congestion control algorithm, ordered left to right by algorithm identifier. Each bar plots the magnitude of the difference in average goodput for TCP flows – y16(u) – versus competing alternate flows – y2(u). If the bar is red, y16(u) is greater; if the bar is green, y2(u) is greater. The 32 bar graphs are sorted from least to most congestion by condition, as indicated in the lower left-hand corner of each plot.)

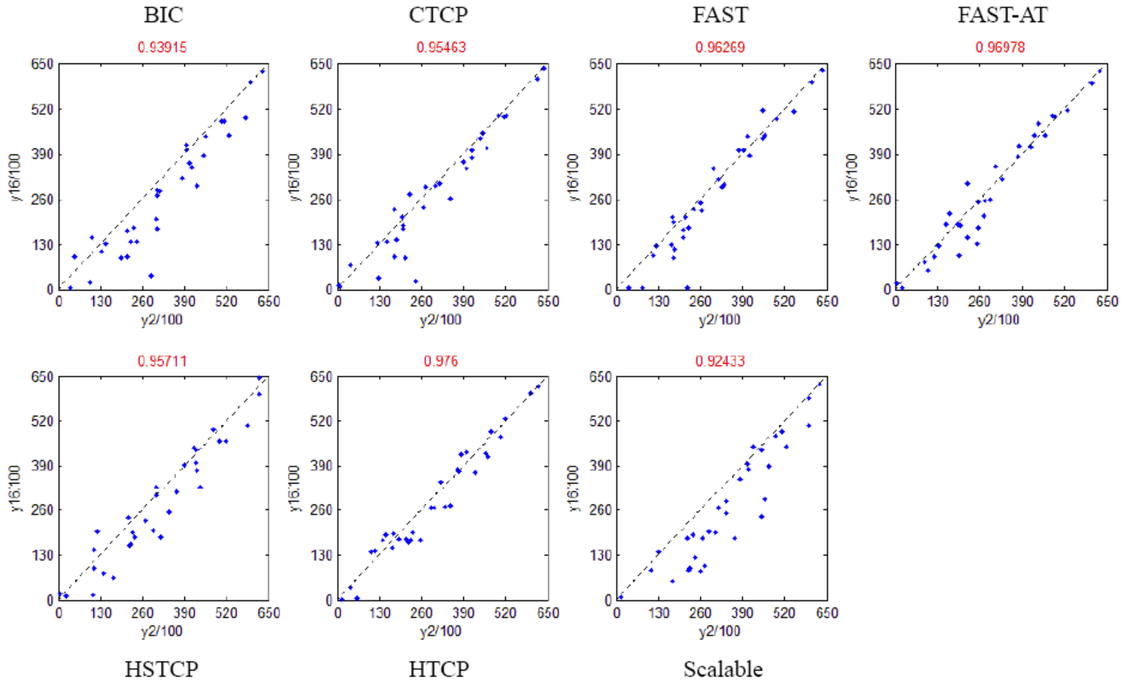


Figure 9-18. Goodput on TCP Flows (y axes give y16u/100 pps) vs. Non-TCP Flows (x axes give y2u/100 pps) for Movies on Fast Paths with Fast Interfaces in a Large, Fast Network with High Initial Slow-Start Threshold

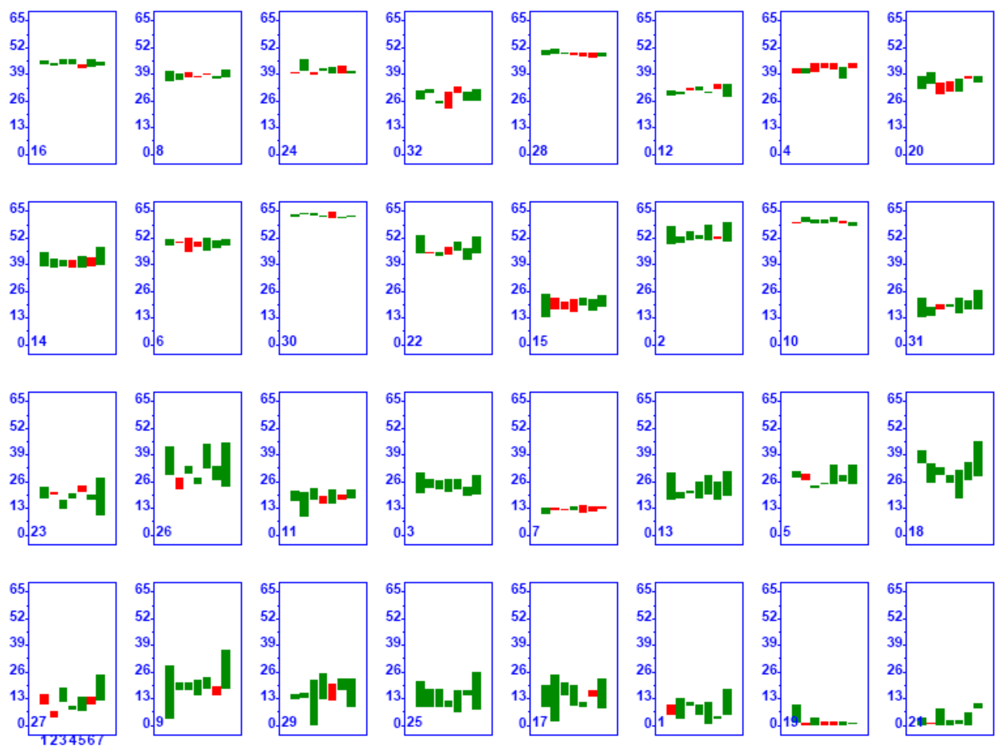


Figure 9-19. Bar Graphs (one for each simulated condition) plotting Goodput Differences (pps/1000) on TCP Flows vs. Non-TCP Flows for Movies Transferred on Fast Paths with Fast Interfaces in a Large, Fast Network with High Initial Slow-Start Threshold (green bars indicate flows using alternate algorithm have higher goodput and red bars indicate competing flows using TCP have higher goodput)

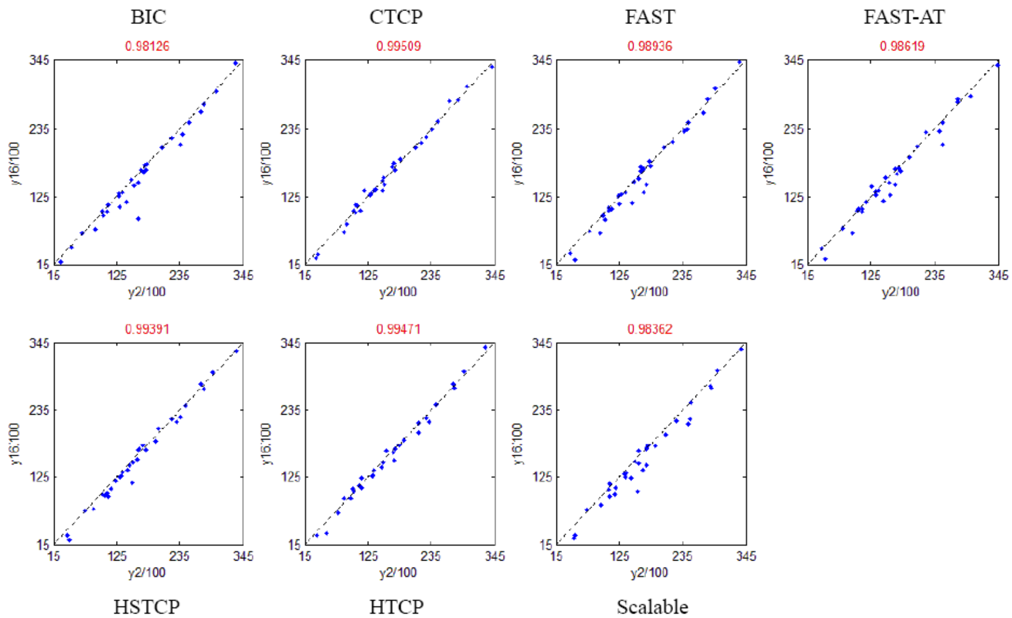


Figure 9-20. Goodput on TCP Flows (y axes give y16u/100 pps) vs. Non-TCP Flows (x axes give y2u/100 pps) for Service Packs on Fast Paths with Fast Interfaces in a Large, Fast Network with High Initial Slow-Start Threshold

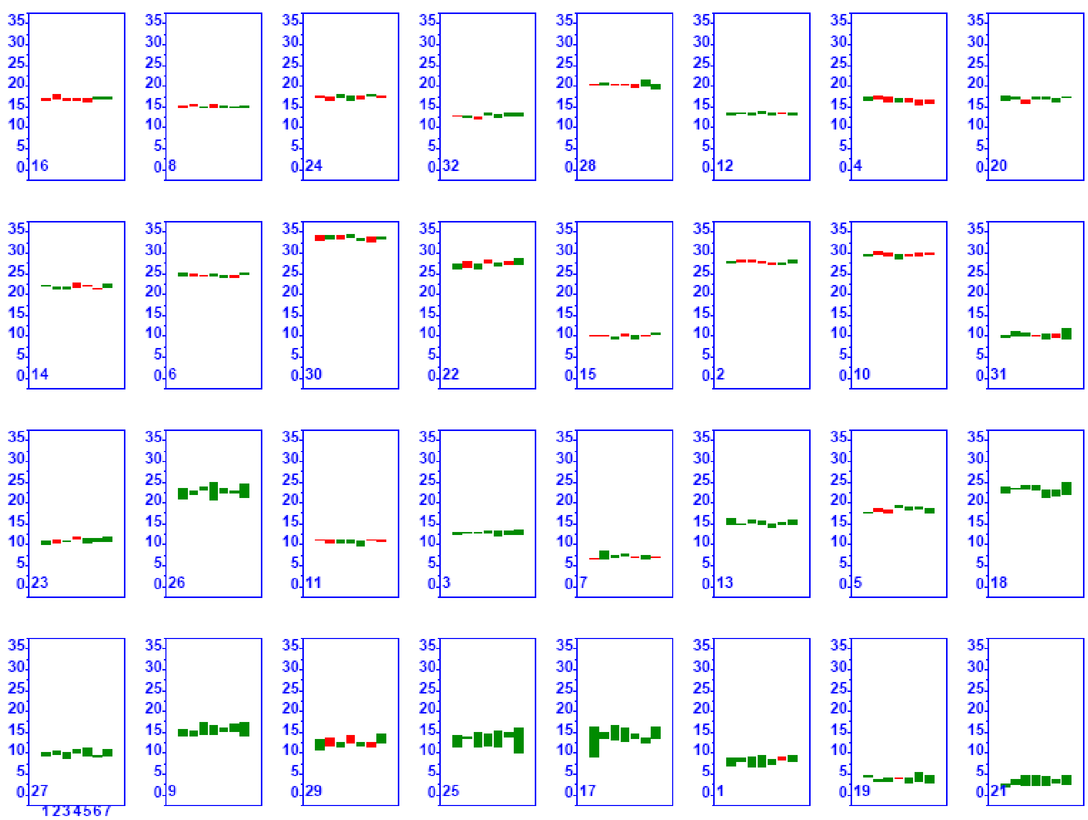


Figure 9-21. Bar Graphs (one for each simulated condition) plotting Goodput Differences (pps/1000) on TCP Flows vs. Non-TCP Flows for Service Packs Transferred on Fast Paths with Fast Interfaces in a Large, Fast Network with High Initial Slow-Start Threshold (green bars indicate flows using alternate algorithm have higher goodput and red bars indicate competing flows using TCP have higher goodput)

Examination of Table 9-12 reveals that goodput differences among alternate congestion control algorithms and among competing TCP flows narrowed as network size and speed increased. In addition, goodput improvements provided by alternate congestion control algorithms over TCP flows disappeared for most flow groups. Alternate congestion control algorithms provided improved goodputs (over TCP) only on flows where files were large (movies and service packs) and where congestion was significant (fast and typical path classes.)

Table 9-12. Range of Goodput Differences (%) for Flow Groups under High Initial Slow-Start Threshold for Small, Slow Network and for Large, Fast Network (Differences are shown: among Alternate Congestion Control Algorithms, among TCP Flows Competing with Alternate Algorithms and between Alternate Algorithms and TCP Flows)

			RANGE OF GOODPUT DIFFERENCES (%)					
			SMALL, SLOW NETWORK HIGH INITIAL SSTHRESH			LARGE, FAST NETWORK HIGH INITIAL SSTHRESH		
File	Path	Interface	AMONG ALTs	AMONG TCPs	ALTs > TCPs	AMONG ALTs	AMONG TCPs	ALTs > TCPs
M	VF	F	10	11	11	1	1	<<-1
		N	<1	8	3	0	0	<<-1
	F	F	35	16	35	20	4	12
		N	21	20	21	14	4	12
	T	F	30	11	30	23	5	15
		N	30	17	30	11	4	12
SP	VF	F	4	6	3	1	1	<-1
		N	<3	5	1	0	0	<-1
	F	F	12	8	15	4	4	4
		N	15	10	15	5	3	6
	T	F	20	6	20	6	2	5
		N	20	7	20	6	4	6
D	VF	F	5	6	<1	1	1	<-1
		N	<3	4	<2	1	1	<-1
	F	F	4	7	<2	2	2	<1
		N	5	7	<2	1	2	<1
	T	F	3	5	2	1	2	<1
		N	<4	7	2	1	2	<1
WO	VF	F	16	5	-1	2	1	<-1
		N	<5	5	<2	1	0	<-1
	F	F	5	4	<1	1	2	<-1
		N	4	4	<1	1	2	<-1
	T	F	5	6	<1	2	2	<<1
		N	5	5	<1	2	2	<<1

9.4.3 Relative User Experience

In this section, we set aside absolute differences in average goodput and consider instead relative differences. As discussed in Sec. 8.4.3, for each simulated condition, we ranked from high (7) to low (1) the average goodput – y2(u) – provided by the seven alternate congestion control algorithms and we also computed the average goodput across all seven algorithms. We took similar steps with respect to average goodput – y16(u) – among TCP flows competing with each of the alternate algorithms. Using this information, we generated seven pairs of rank matrices. One member of each pair relates to y2(u) and the other member to y16(u). (See Fig. 8-32 for a sample rank matrix.) Each matrix contains (32 conditions x 24 flow groups =) 768 cells, where each cell holds the rank (of average goodput among the seven competing algorithms) for the congestion control algorithm associated with the matrix. If the rank in a cell is rendered in green, then the goodput associated with the rank was above the average goodput for all algorithms. If red, then the goodput was below the relevant average. When a highest ranked (7) cell was farther from the average goodput than the lowest ranked (1) cell, then the cell is highlighted in green. In the reverse case, the lowest ranked cell is highlighted in red.

The columns in each matrix are divided into four vertical sections that each relate to a specific file size (movie, service pack, document and Web object). Each section contains three pairs of flow groups (labeled on the x axis) ordered by path class (very fast, fast and typical). Within each flow-group pair the ordering is by interface speed (fast and normal). The matrix rows are ordered by condition (labeled on the y axis) from least (top) to most (bottom) congested. We reproduce the matrices (Figs. 9-22 through 9-35) to show any patterns that occur. We computed the average rank for each congestion control algorithm for each file size. Similarly, we computed the average rank for TCP flows competing with each congestion control algorithm for each file size. We also determined the standard deviation in rank for each alternate congestion control algorithm, across all files sizes and considering both y2(u) and y16(u). We report these averages and standard deviations in a summary table (Table 9-13). We use the information from the summary table to generate a scatter plot (Fig. 9-36) of average rank (x axis) vs. standard deviation in rank (y axis), which reveals differences in relative user experience among the seven alternate congestion control algorithms.

Table 9-13 shows standard deviation in rank to fall and narrow significantly (0.23 to 0.73) compared with the smaller, slower network (Table 8-31), so ranks of all alternate congestion control algorithms became closer in the larger, faster network. This is congruent with other analyses of the average goodput data. The relative rank of Scalable TCP improved due to higher goodputs for movies, while differences narrowed for other file sizes. The relative rank of FAST-AT improved because the algorithm ranked very well among all file sizes except movies. The relative rank of HTCP and CTCP fell because fewer losses gave fewer opportunities to activate the TCP-friendly[2] loss/recovery procedures of the two algorithms.

[2] TCP friendliness implies that an alternate algorithm behaves similarly to TCP, e.g., reduces transmission rate in half (or nearly so) on a loss and then does not increase transmission rate very quickly. HTCP reduces transmission rate up to 50 % on a packet loss and then increases transmission rate only linearly for one second after a loss. CTCP reduces transmission rate 50 % on a packet loss and can increase transmission rate quickly, but only when the congestion window is above 41 packets and delay is not increasing on the path between a source and receiver.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
16	6	6	4	5	5	4	1	6	3	4	3	3	6	4	4	4	5	5	1	2	3	4	5	3
8	1	1	6	5	7	6	2	7	4	5	2	3	1	6	2	4	3	1	4	1	5	3	2	2
24	7	7	2	3	6	5	4	6	3	3	4	4	4	4	4	2	6	5	4	1	3	3	5	5
32	7	4	5	7	6	5	3	5	2	4	3	3	1	6	5	5	2	1	7	3	5	5	2	2
28	1	5	6	3	5	4	5	6	4	4	6	2	2	7	6	1	7	7	4	1	2	7	6	5
12	7	1	3	7	6	6	3	3	1	5	6	4	1	4	6	3	3	2	4	1	4	4	3	3
4	5	6	1	5	1	7	7	3	7	6	6	2	4	5	5	6	3	3	7	1	4	5	5	3
20	7	1	5	4	6	6	3	3	7	5	6	5	7	7	4	4	4	3	7	3	3	3	4	4
14	6	7	6	7	6	6	4	5	5	3	5	4	5	2	4	5	2	2	4	4	4	4	2	2
6	1	2	6	6	6	5	1	6	7	4	6	5	6	7	5	3	5	5	1	1	5	7	5	5
30	7	7	5	6	6	5	5	3	2	7	4	5	3	4	5	3	6	2	2	1	5	6	5	5
22	2	1	7	5	6	5	7	2	2	1	4	5	7	1	2	1	2	1	3	3	1	1	2	2
15	2	1	7	7	6	6	6	7	5	5	2	3	6	3	2	3	1	1	4	3	3	3	1	1
2	6	5	5	2	6	6	3	4	4	5	7	2	2	1	1	1	1	2	4	5	3	2	2	1
10	6	3	1	5	5	3	3	2	7	4	5	2	4	5	1	3	2	1	7	4	5	3	1	2
31	7	3	6	7	6	6	5	3	2	6	7	6	5	5	5	4	5	4	5	4	5	5	4	4
23	5	4	6	5	5	6	1	4	3	4	3	5	5	4	2	3	1	1	4	3	3	3	1	1
26	1	7	5	6	5	3	2	5	4	4	5	4	5	1	4	2	1	1	1	6	3	3	1	1
11	1	5	4	5	6	6	1	4	4	5	4	4	6	4	3	5	3	3	5	5	3	5	2	2
3	3	5	7	5	6	6	3	4	1	4	6	4	3	5	3	4	1	1	5	1	1	1	1	1
7	2	6	6	4	2	6	5	7	1	1	7	6	6	4	1	1	2	2	5	5	1	1	1	1
13	5	4	6	5	6	6	5	1	7	4	6	4	7	6	2	3	2	1	5	6	2	2	2	1
5	6	1	5	6	5	6	2	1	1	4	4	4	7	3	4	5	4	4	5	2	4	5	3	3
18	6	2	6	4	6	4	7	2	4	4	6	4	4	2	3	3	2	2	2	1	4	4	3	2
27	5	5	4	3	6	5	5	7	3	4	5	4	1	1	5	7	3	2	1	7	5	6	3	3
9	6	1	6	5	7	4	3	4	2	4	6	4	2	1	3	1	3	4	3	2	2	2	4	3
29	2	2	2	6	6	2	1	6	6	4	3	4	3	6	6	5	2	3	5	2	4	5	2	2
25	2	6	6	3	6	4	2	4	2	3	4	4	2	2	4	4	4	4	4	3	4	4	4	4
17	7	4	3	2	6	2	7	4	5	4	4	4	7	7	6	7	4	5	1	3	7	7	5	5
1	1	1	2	1	6	1	2	7	3	3	3	4	4	5	3	3	3	3	7	4	3	3	3	3
19	4	3	7	5	6	4	1	1	5	5	7	4	4	3	4	4	2	2	1	4	4	4	3	3
21	5	7	4	6	7	2	5	5	1	5	3	3	2	5	4	6	4	5	3	4	5	5	5	5
	VF F M	VF N M	F F M	F N M	T F M	T N M	VF F SP	VF N SP	F F SP	F N SP	T F SP	T N SP	VF F D	VF N D	F F D	F N D	T F D	T N D	VF F WO	VF N WO	F F WO	F N WO	T F WO	T N WO

Figure 9-22. Goodput Rank Matrix – y2(u) – BIC (Large, Fast Network, High Initial Slow-Start) Rank (7 high) in each cell denotes ordering of y2(u) for each condition (y axis) and flow group (x axis) – conditions are sorted from least (16) to most (21) congested and flow groups are ordered by file size – movies (M), service packs (SP), documents (D) and Web objects (WO) – and by path class – very fast (VF), fast (F), and typical (T) – within each file size and by interface speed – fast (F) or normal (N) – within each path class.

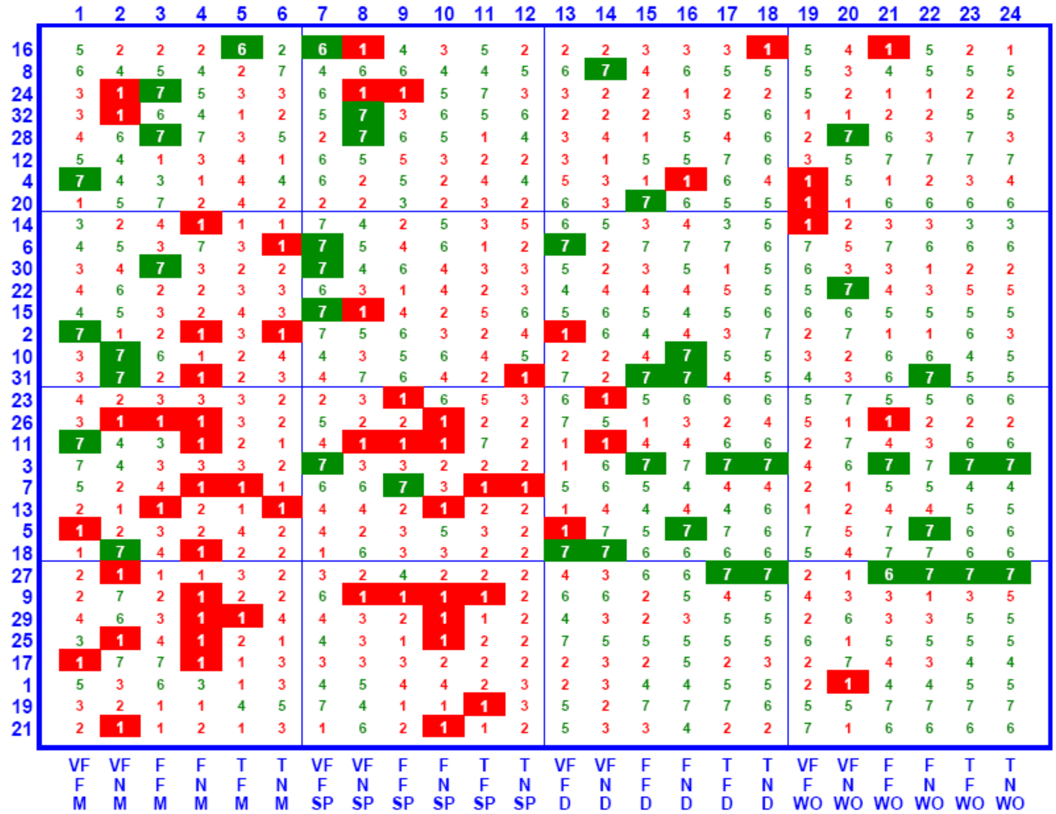


Figure 9-23. Goodput Rank Matrix – y2(u) – CTCP (Large, Fast Network, High Initial Slow-Start)

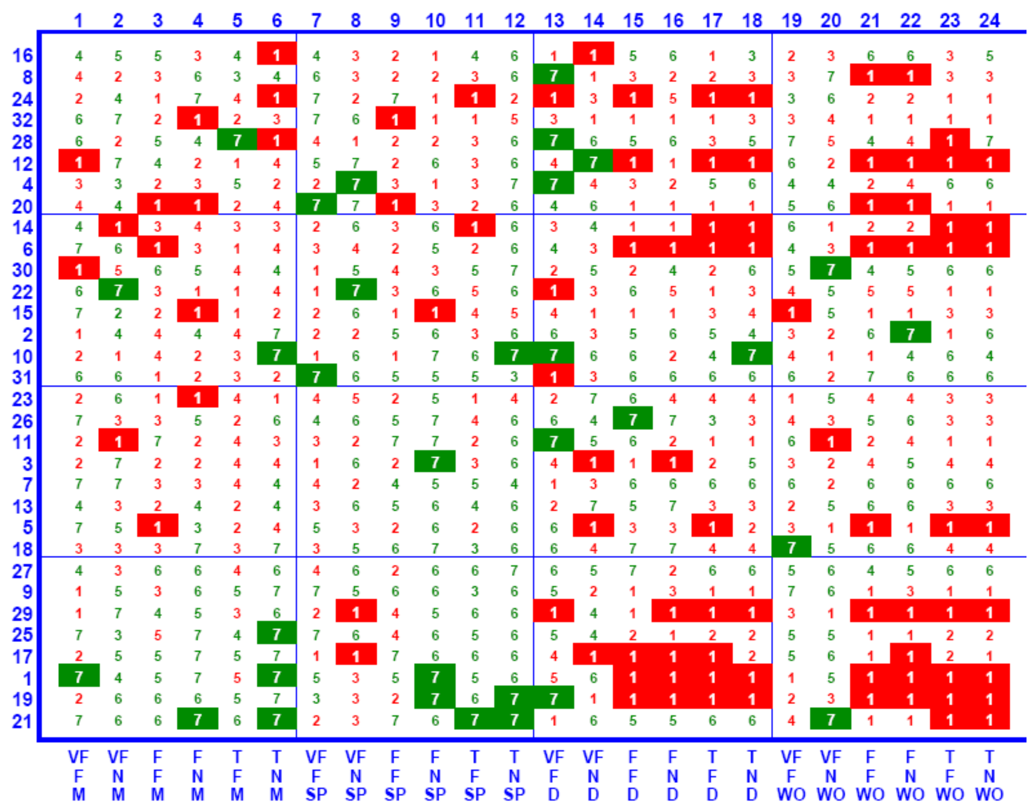


Figure 9-24. Goodput Rank Matrix – y2(u) – FAST (Large, Fast Network, High Initial Slow-Start)

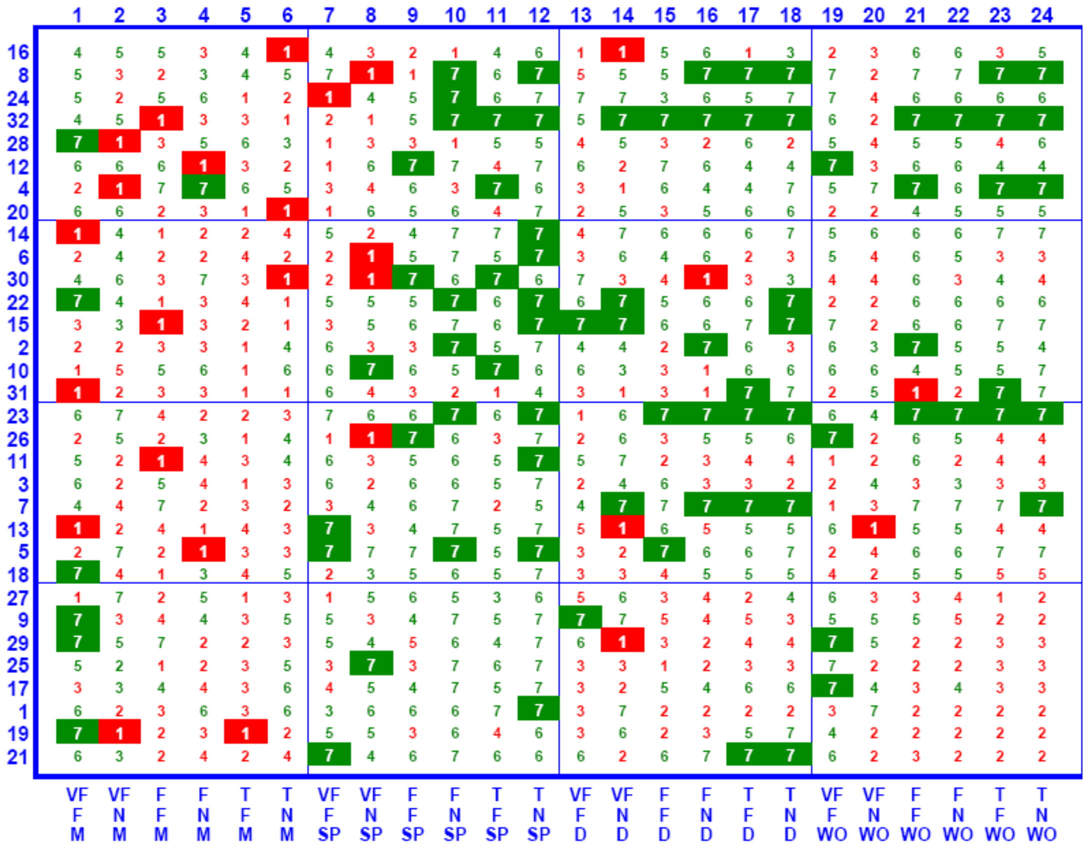


Figure 9-25. Goodput Rank Matrix – y2(u) – FAST-AT (Large, Fast Network, High Initial Slow-Start)

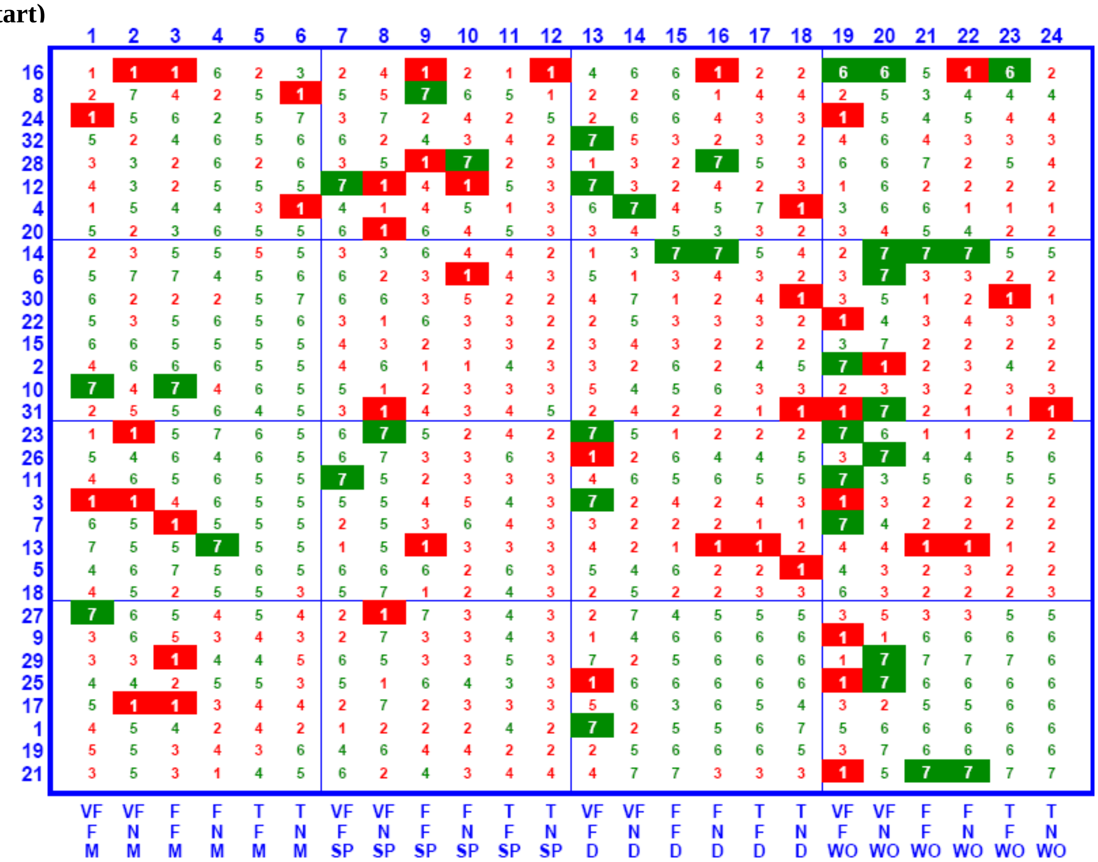


Figure 9-26. Goodput Rank Matrix – y2(u) – HSTCP (Large, Fast Network, High Initial Slow-Start)

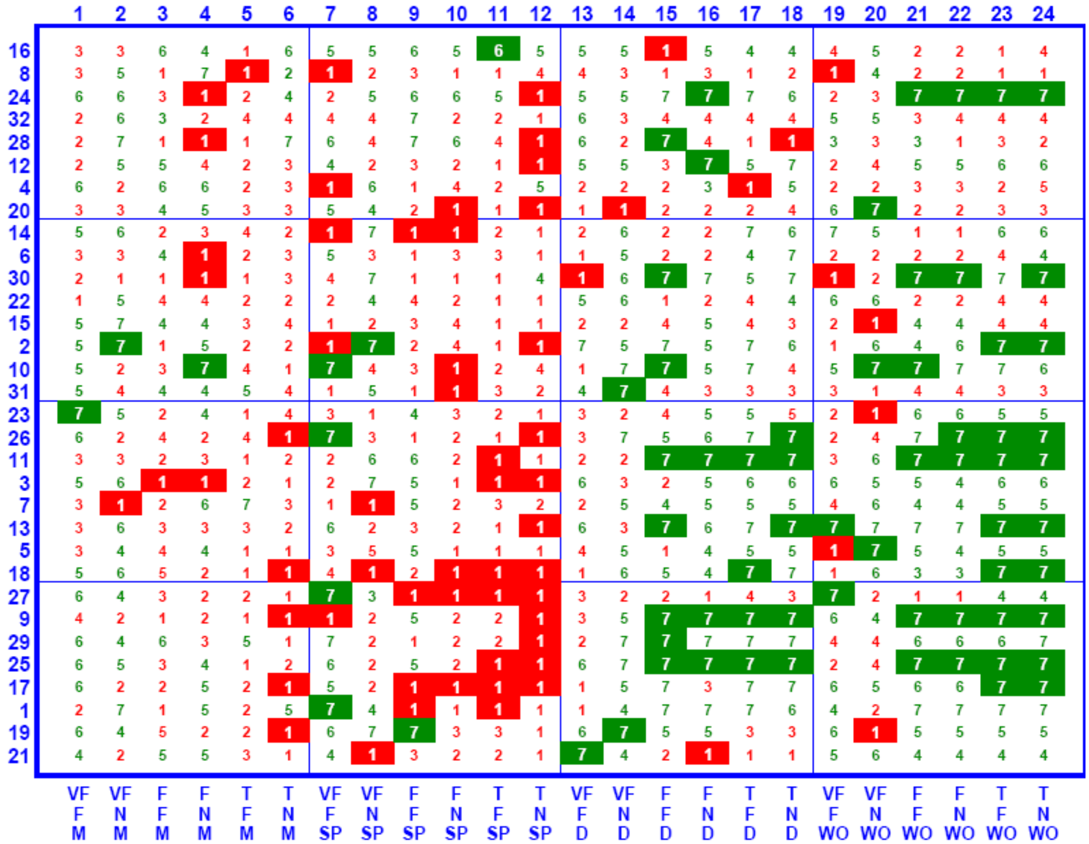


Figure 9-27. Goodput Rank Matrix – y2(u) – HTCP (Large, Fast Network, High Initial Slow-Start)

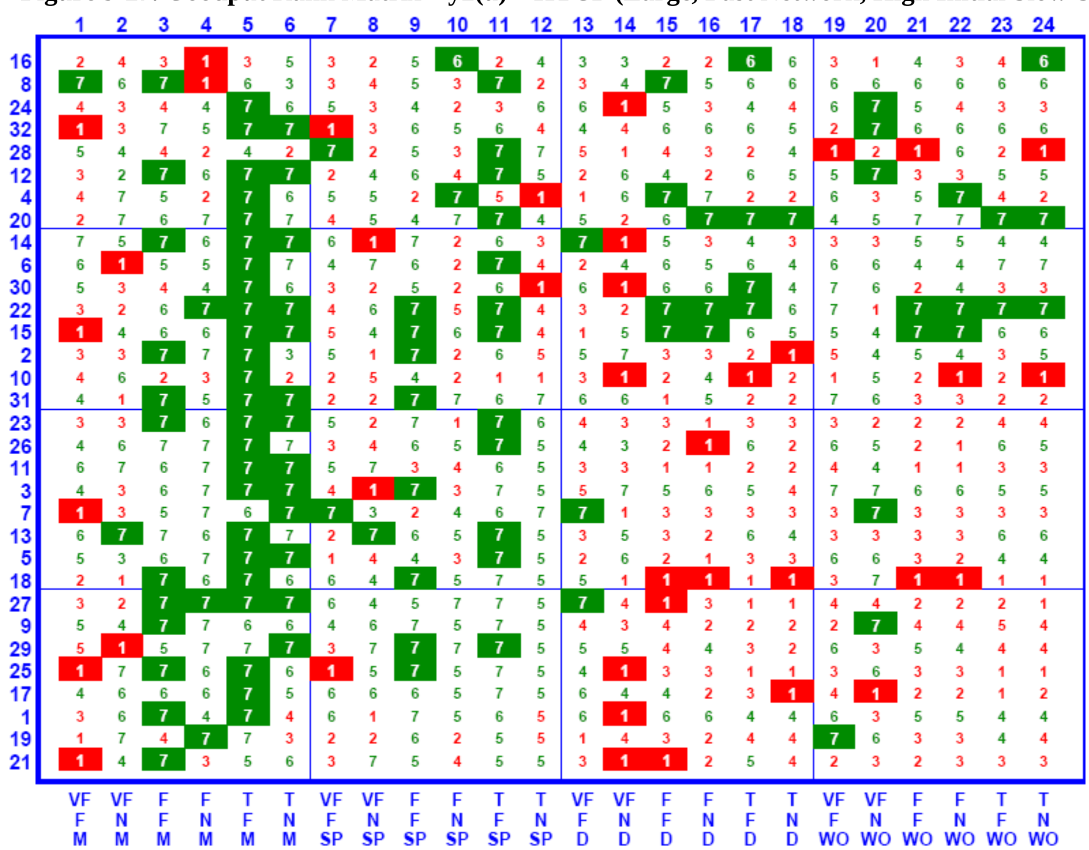


Figure 9-28. Goodput Rank Matrix – y2(u) – Scalable (Large, Fast Network, High Initial Slow-Start)

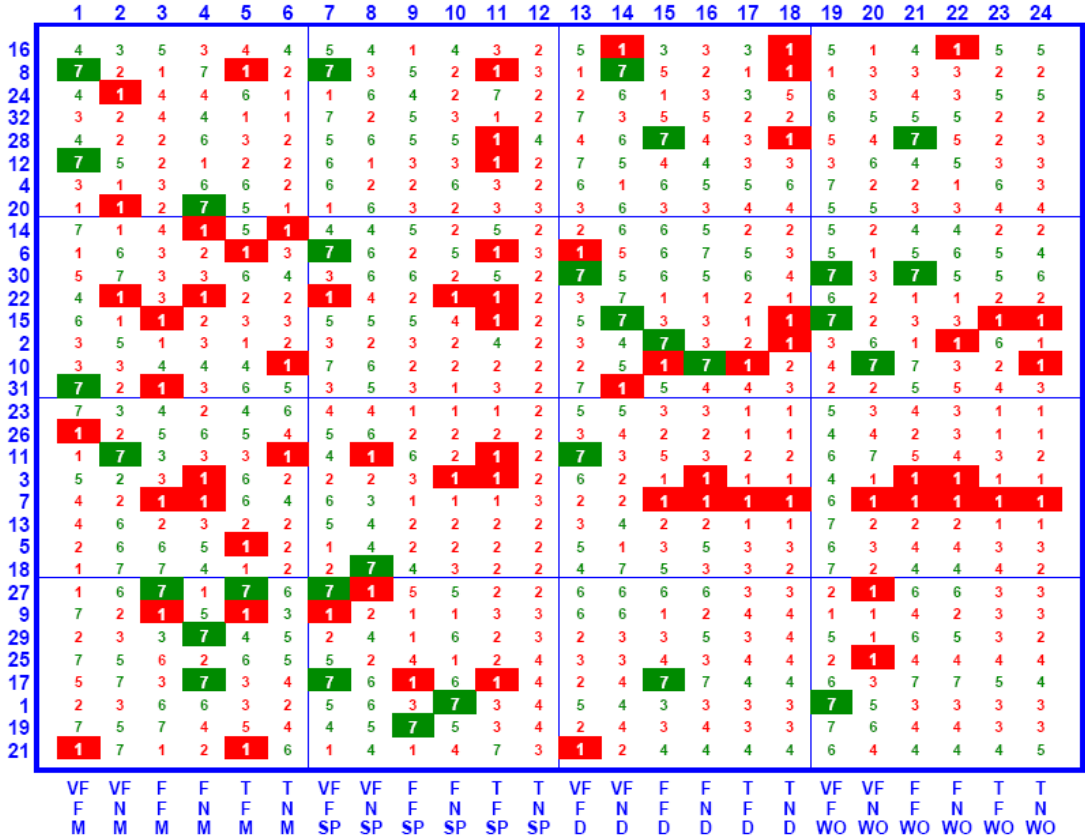


Figure 9-29. Goodput Rank Matrix – y16(u) – BIC (Large, Fast Network, High Initial Slow-Start)

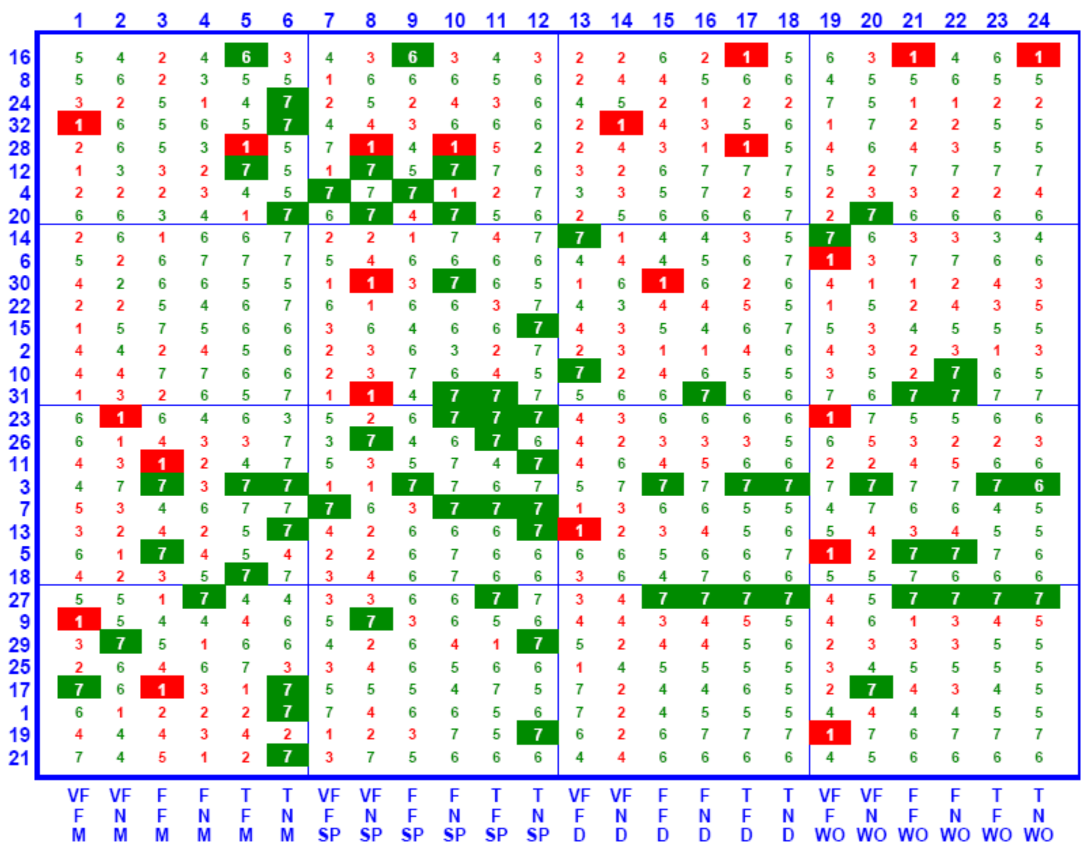


Figure 9-30. Goodput Rank Matrix – y16(u) – CTCP (Large, Fast Network, High Initial Slow-Start)

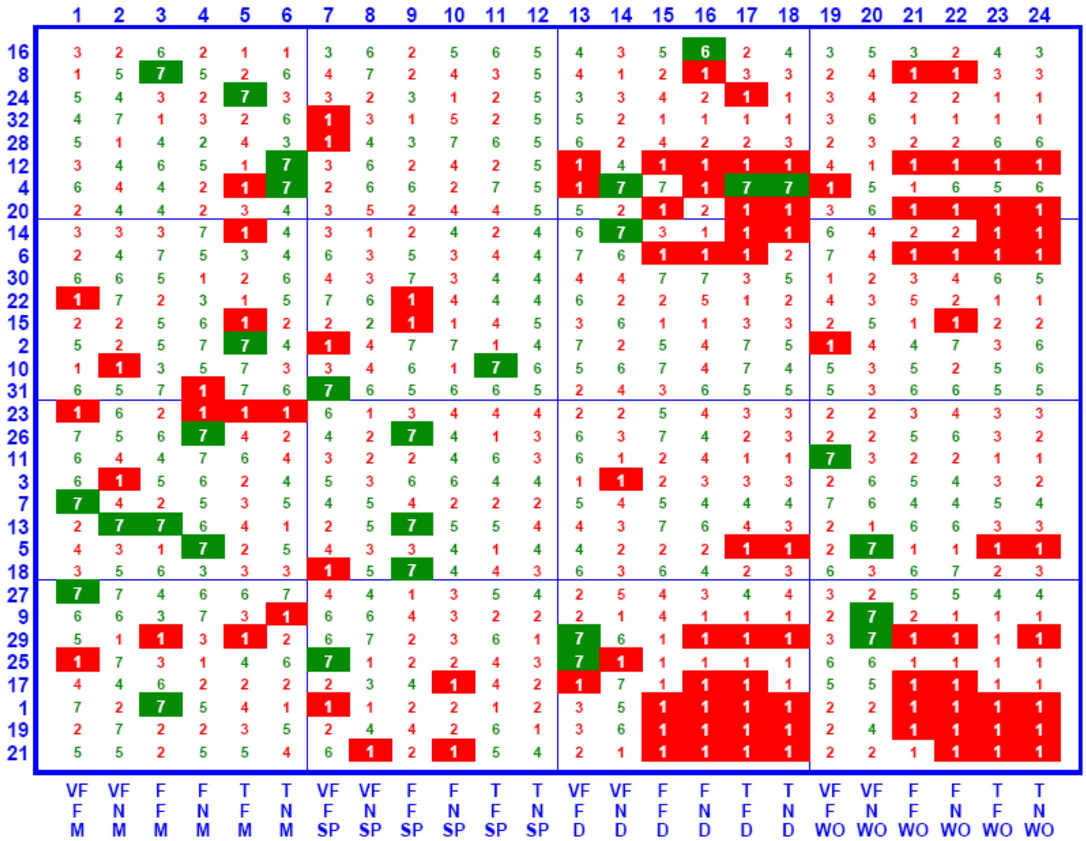


Figure 9-31. Goodput Rank Matrix – y16(u) – FAST (Large, Fast Network, High Initial Slow-Start)

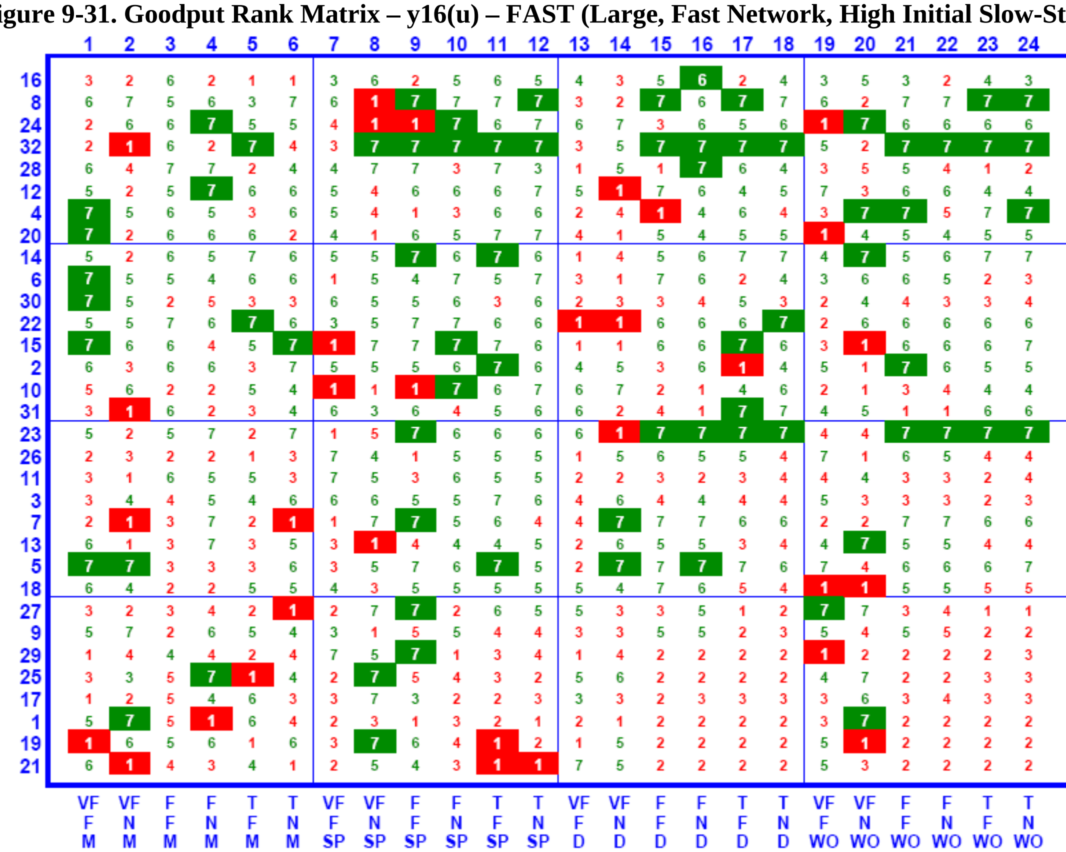


Figure 9-32. Goodput Rank Matrix – y16(u) – FAST-AT (Large, Fast Network, High Initial Slow-Start)

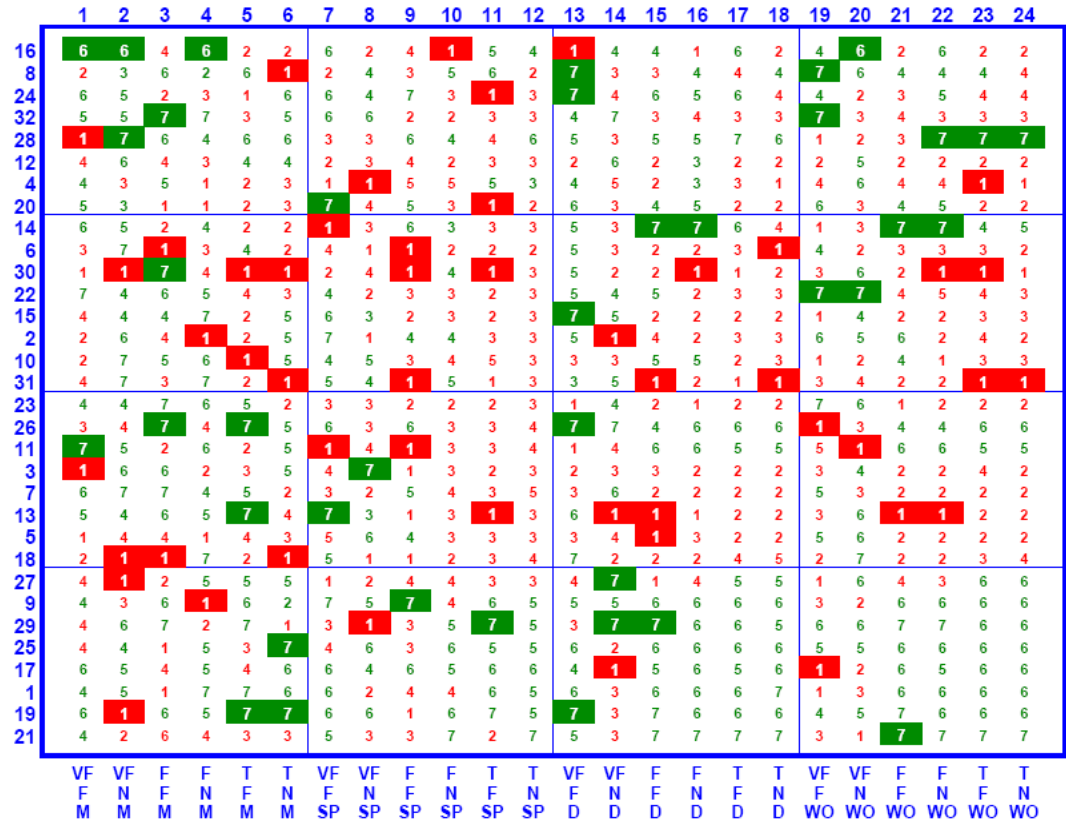


Figure 9-33. Goodput Rank Matrix – y16(u) – HSTCP (Large, Fast Network, High Initial Slow-Start)

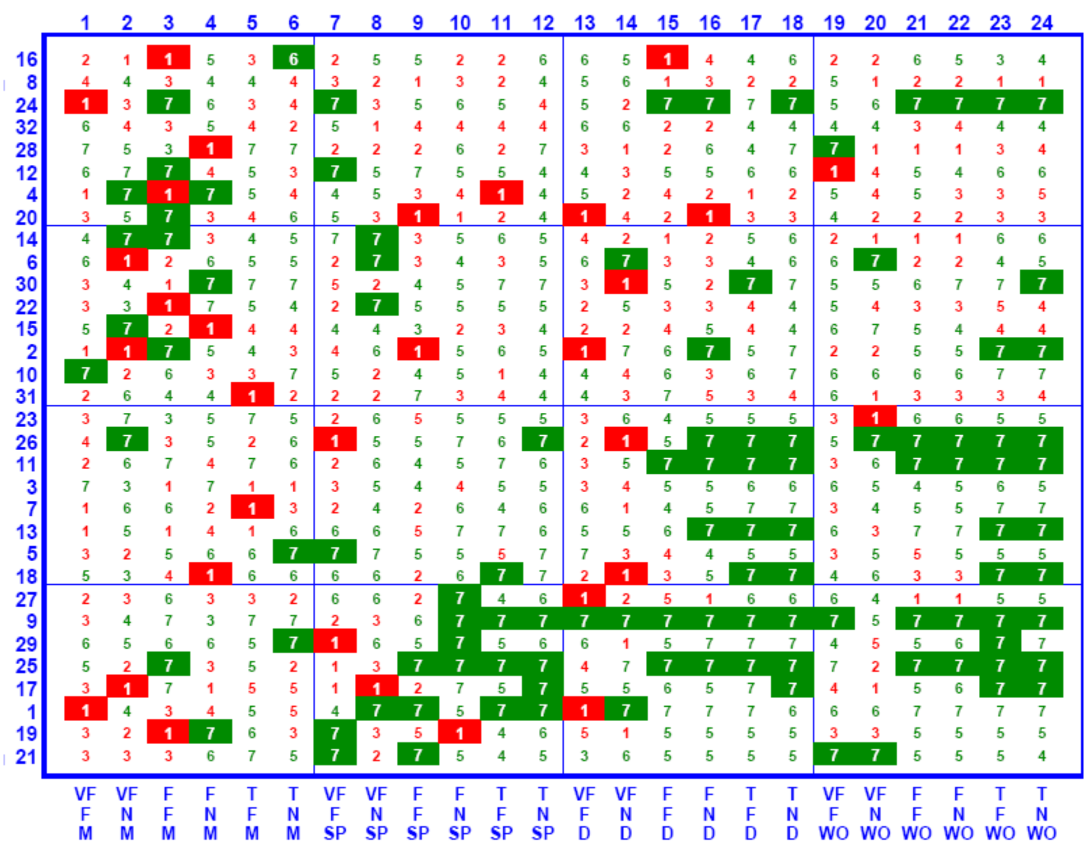


Figure 9-34. Goodput Rank Matrix – y16(u) – HTCP (Large, Fast Network, High Initial Slow-Start)

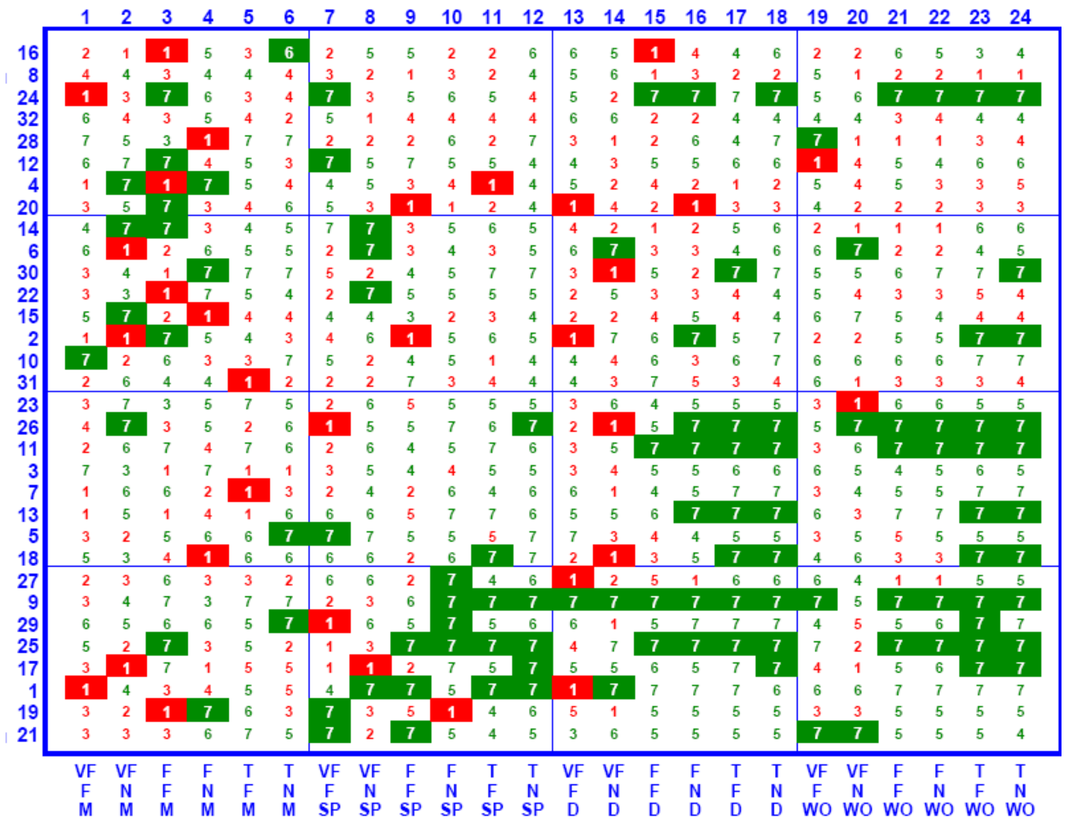


Figure 9-35. Goodput Rank Matrix – y16(u) – Scalable (Large, Fast Network, High Initial Slow-Start)

Table 9-13. Summary Average and Standard Deviation in Goodput Rankings for Flows using Alternate Congestion Control Algorithms and for Competing TCP Flows (Large, Fast Network, High Initial Slow-Start Threshold)

		BIC	CTCP	FAST	FAST-AT	HSTCP	HTCP	STCP
y2(u)	M	4.70	3.01	3.93	3.39	4.30	3.33	5.23
	SP	4.05	3.41	4.34	5.05	3.57	2.76	4.71
	D	3.55	4.39	3.34	4.61	3.79	4.50	3.69
	WO	3.37	4.32	3.24	4.52	3.81	4.56	4.08
	Avg.	3.92	3.78	3.71	4.39	3.87	3.79	4.43
y16(u)	M	3.51	4.34	3.93	4.28	4.06	4.19	3.56
	SP	3.20	4.88	3.69	4.71	3.65	4.55	3.24
	D	3.49	4.53	3.07	4.12	3.94	4.65	4.10
	WO	3.54	4.54	2.93	4.23	3.86	4.80	3.96
	Avg.	3.44	4.57	3.41	4.34	3.88	4.55	3.72
y2(u) & y16(u)	Avg.	3.68	4.18	3.56	4.36	3.87	4.17	4.07
	Std.	0.48	0.63	0.49	0.49	0.23	0.73	0.64

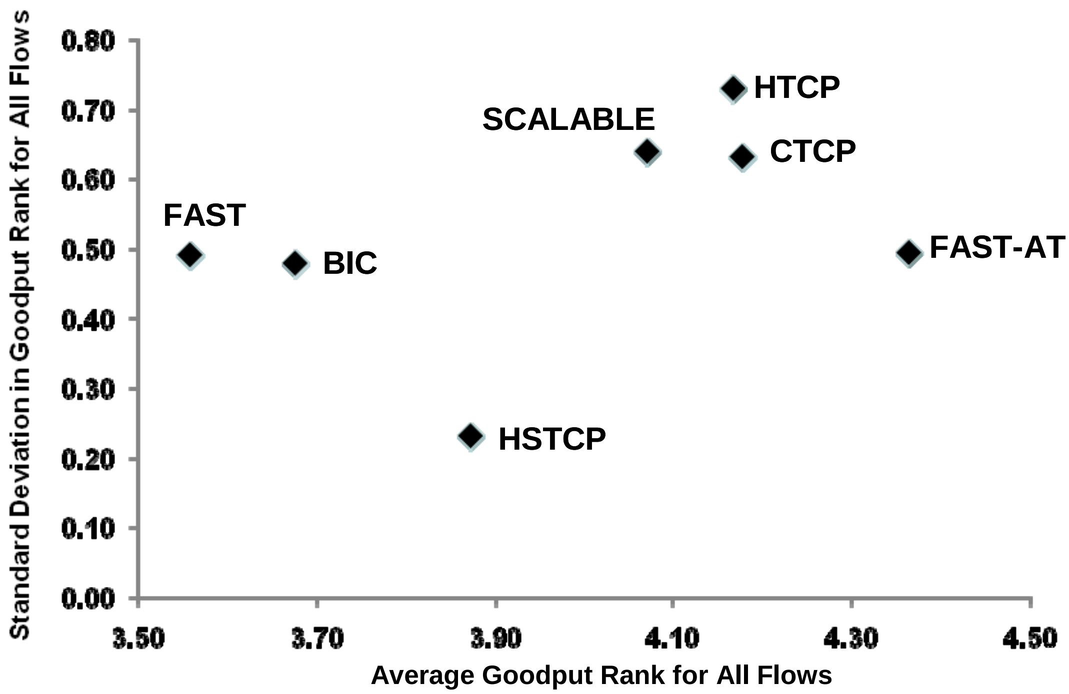


Figure 9-36. Average (x axis) vs. Standard Deviation (y axis) in Goodput Rank (Large, Fast Network, High Initial Slow-Start Threshold)

Looking at the rank matrices, summary table and scatter plot gives some impressions regarding relative goodput for flows operating under various congestion control algorithms as well as for competing TCP flows. Four of these impressions were seen and discussed before (in Sec. 8.4.3.1). First, CTCP, HTCP and FAST-AT[3] appear relatively friendly to TCP flows. Second, Scalable TCP ranks high in goodput for movies and for all file sizes under sporadic losses. Third, BIC, FAST, HSTCP and Scalable TCP are relatively unfriendly[4] to TCP flows. Fourth, HTCP ranks poorly with respect to large flows. Comparing relative ranks in a large, fast network against relative ranks in a smaller, slower network, revealed two additional impressions. First, differences in rank cover a lower range in the large, fast network (3.56 to 4.36) than was the case for a smaller, slower simulated network (3.16 to 4.63). Second, the standard deviation in ranks was much narrower in a large, fast network (0.23 to 0.73) than in a smaller, slower network (0.34 to 1.37).

Overall, then, assuming a high initial slow-start threshold, as a network becomes faster and less congested, differences in goodput offered by the alternate congestion control algorithms and competing TCP flows come closer together. Adopting a large initial slow-start threshold eliminates activation of enhanced window increase procedures

---

[3] FAST-AT reduces transmission rate 50 % on a packet loss and can increase rate quickly after that, but a falling transmission rate can cause FAST-AT to reduce the α parameter, which causes a slower increase in transmission rate when recovery occurs.

[4] TCP unfriendliness implies reducing transmission rate substantially less than 50 % following a packet loss and/or increasing transmission rate much more quickly than linearly when recovery occurs.

available in the alternate congestion control algorithms. When losses occur, differences in goodput can be discerned and attributed to loss/recovery characteristics of the various algorithms. As a network becomes less and less congested, alternate congestion control algorithms have fewer chances to invoke their enhanced loss/recovery procedures.

9.5 Findings

This experiment considered a range of files sizes (movies, service packs, documents and Web objects) being transferred across a largely uncongested network, where some (fast and typical) paths experienced more congestion than others (very fast paths) and where some flows could achieve a maximum rate of 80 x 10^3 pps, while others were constrained (by the interface speed of a sender or receiver) to at most 8 x 10^3 pps. Flows using TCP congestion control were mixed with flows using one of seven alternate congestion control algorithms. All flows adopted the same initial slow-start procedures to determine the maximum available transfer rate (i.e., all flows used a high initial slow-start threshold). In general, under these conditions (ignoring network speed and propagation delay), goodput experienced on individual flows is influenced by two main factors: (1) file size and (2) packet losses and related recovery procedures. The results of these experiments confirmed many of the findings discussed in Chapter 8.

9.5.1 Finding #1

Given a high initial slow-start threshold and the minimal congestion arising in a large, fast network, differences in average goodput narrowed in each flow group, whether using alternate or standard TCP congestion control procedures. That is, goodput differences shrank among alternate congestion control algorithms and between TCP flows and flows using alternate congestion control procedures. Assigning all flows a high initial-slow start threshold eliminated differences in increase procedures when determining the maximum available transfer rate. Increasing network speed and size reduced overall congestion by several orders of magnitude under most conditions. Lower congestion led to fewer losses, which reduced opportunities for alternate congestion control algorithms to activate enhanced loss/recovery procedures.

9.5.2 Finding #2

Under selected conditions, where file sizes were large (i.e., movies and service packs) and where congestion could appear (i.e., on fast and typical paths, which can experience sharing among more flows), differences in average goodputs could still be distinguished due to differences in loss/recovery procedures. Though the effects were somewhat muted because overall congestion was lower, the finding here is analogous to a similar finding in Chapter 8. Scalable TCP, BIC and HSTCP do not decrease their transmission rate as much as the other algorithms when a loss is detected. This means that already established flows continue to transmit at higher rates at the cost of inhibiting newer flows and also TCP flows, which cut their transmission rate in half on a loss. Thus, under congested conditions, these protocols provided higher goodput than TCP flows.

9.5.3 Finding #3

Overall, in this experiment, FAST-AT provided the best balance in relative goodput achieved on all flows. CTCP ranked second best overall, followed closely by HTCP.

FAST-AT ranked third most friendly (after CTCP and then HTCP) to TCP flows and ranked second best (after Scalable TCP) at providing goodput to flows using alternate congestion control procedures. Note that lower overall congestion narrowed significantly the differences in ranking among all the algorithms.

9.5.4 Finding #4

As seen in earlier experiments, this experiment showed that use of some alternate congestion control protocols altered selected macroscopic characteristics of the network. Here, as in Chapter 8, the characteristic changes were, in general, not statistically significant. We attribute this to two main factors: (1) overall congestion levels were kept much lower than in previous experiments and (2) FAST and FAST-AT, which have similar characteristics, where not separated in the analyses, which tended to reduce the statistical significance that might be attributed to either algorithm considered without the other. In general, the current experiments confirmed that FAST and FAST-AT tend to increase retransmission rate under higher congestion. Thus, more flows are pending in the connecting state and fewer flows complete per unit of time. In addition, Scalable TCP tends to increase buffer occupancy throughout the network. This can also lead to higher retransmission rates, to more flows pending in the connecting state and to fewer flows completing per unit time. At lower congestion levels, Scalable TCP performed worse on these metrics than FAST (and FAST-AT). At higher congestion levels, FAST (and FAST-AT) performed worse. Finally, we found again in this experiment that CTCP can exhibit a much higher average congestion window size than other congestion control algorithms. The increase appears most prominent under lower congestion levels.

9.6 Conclusions

In this chapter, we described an experiment to investigate effects on macroscopic behavior and user experience when deploying various congestion control algorithms in a large, fast, simulated, heterogeneous network, i.e., a network that includes flows operating under standard TCP congestion control procedures together with flows operating under one of seven proposed alternate congestion control algorithms. In effect, we repeated, with a few changes, half the experiments from Chapter 8. Specifically, we repeated the experiments where all flows adopted a high initial slow-start threshold. We changed the network to increase router speeds and number of sources and receivers by an order of magnitude, which also changed buffer sizes. Increasing network speed and size required more than an order of magnitude increase in computational cost, which motivated us to repeat only half the experiments from Chapter 8.

We demonstrated that, under a larger, faster network (given a high initial slow-start threshold), reduced congestion levels narrowed differences in average goodput among flows using alternate congestion control algorithms and also between flows using alternate and standard TCP congestion control procedures. Lowered congestion meant fewer losses, which reduced the opportunities for alternate congestion control algorithms to activate enhanced loss/recovery procedures. We also confirmed some findings from the experiments described in Chapter 8. First, on a loss, Scalable TCP, BIC and HSTCP reduced transmission rate less than other algorithms, so these algorithms tended to be provide higher goodput than TCP flows on larger file sizes under congested conditions. Second, under conditions with higher congestion, FAST and FAST-AT exhibited higher retransmission rates, more pending flow connections and fewer flows completing. Under conditions with lower congestion, Scalable TCP could also exhibit such undesirable network-wide properties. Third, CTCP, FAST-AT and HTCP showed better balance overall (than other alternate congestion control algorithms) with respect to relative average goodput for all flows, including both those using the alternate procedures and those using standard TCP procedures.

After completing five sets of simulation experiments (as described in Chapters 6 through 9), we accumulated sufficient information to draw some conclusions about the behaviors of the seven congestion control algorithms we studied. We also developed sufficient experience to evaluate the various methods we adopted. We turn to these topics next, where we conclude our study and identify future work.