# TRAFFIC CONGESTION MEASUREMENTS IN TRANSIT IP NETWORKS

Tim Bass
Science Applications International Corporation
Center for Information Protection
McLean, Virginia

## ABSTRACT

Transit IP internetworks supporting USAF bases are highly congested during duty-hours and attempts to access to the Internet often fail. Often, during these connection failures, transit networks are in a state known as *congestion collapse* and connections time out when a user attempts to reach the Internet. This paper presents the results of a simple IP congestion analysis between four USAF bases and the Internet, graphically illustrating the *congestion domains*. We demonstrate that congestion in DISN links are primarily responsible for the congestion collapse. A skeleton framework for creating similar tests is included and suggestions for improvements to the simple test script are provided.

## INTRODUCTION

> "When too many packets are present in (a part of) the subnet, performance degrades. This situation is called **congestion**." - Tanenbaum [1]

Access to commercial, government and educational sites on the Internet has become a mission requirement for numerous functional areas in the USAF and other Department of Defense organizations. However, the Internet is a store-and-forward queuing system and IP datagrams traverse the Internet via transit domains which are not under the administrative control of either the sender nor the recipient. The routers which control transit backbone traffic do not reserve memory or other resources prior to receiving IP datagrams [2]. It follows, when the packet load of an IP network is within the design capacity of the network, datagrams free of transmission errors are delivered. However, as the packet load increases beyond network capacity, the routers become saturated and begin dropping packets. In TCP flows, characteristic of WWW traffic, the dropped packets are retransmitted, further degrading network availability.

A congested TCP/IP network performs like a negative feedback system and **congestion increases congestion**. For example, when there are no remaining free buffers in IP routers, arriving packets are discarded. The sending device normally times out in this situation and retransmits the TCP segment, perhaps attempting retransmission many times. Congestion at the receiving router forces the sending router to buffer the undeliverable segment, creating additional congestion for neighbors of the sender. In a store-and-forward network this creates a cascading effect which can bring traffic to a virtual standstill [1], described by Comer [2] as *congestion collapse*.

The effects of congestion have become highly visible in USAF functional areas as users with mission requirements for WWW access have found traversing the Internet similar to a traffic jam in a major city. Traffic to the Internet has increased dramatically bringing ACC Internet access to a standstill during peak duty-hours. Pedestrian Internet users are unaware of the root cause of the congestion experienced. During certain times of the day, attempting to reach the Internet is useless and the transit networks may be in congestion collapse.

Many USAF bases have upgraded their IP infrastructure; however, Internet access remain extremely slow. Programs are currently bringing funding to purchase hardware and upgrade the networking infrastructure. These upgrades will not significantly improve Internet access times. In order to quantify the congestion, it is necessary to have tools which measure network performance. This is especially true as new networking infrastructure is installed.

This paper illustrates how to integrate a few simple freeware utilities to estimate the relative congestion in the transit domains providing Internet access to the functional user. The results are graphically illustrated. A simple example of the test script is included in the Appendix. Department of Defense network management personnel may perform their own tests without acquiring expensive software.

## MEASURING NETWORK DELAYS

In large queuing systems, e. g. the Internet, precise traffic analysis is difficult (at best) because of the random arrival rate and variable loading of the datagrams. There is a wide variation of traffic and congestion coupled with numerous network load factors at any singularity in time. The best estimates on the overall network congestion are ascertained from hundreds (perhaps thousands) of samples averaged over long test intervals.

Utilities are available to measure round trip times (RTTs) in IP networks. Stevens [3] uses the **Ping(8)** utility to calculate the RTT of a ICMP packet across the Internet as a function of the user data size, ranging from 0 to 1400 bytes (see [3] Figure A.3). In this paper, we are interested in the average RTT of IP packets traversing autonomous IP routing domains in the Internet. This test



0-7803-4249-6/97/$10.00 © 1997 IEEE

methodology is a widely accepted direct measurement, performed by injecting test traffic into the network and measuring the results [4].

Both the **Ping(8)** and the **Traceroute(8)** utility may be used to measure RTTs, but **Traceroute(8)** was selected. The size of the user data was selected to the default 38 bytes making the total packet size 40 bytes. Injecting a small packet into a highly congested network minimizes the effect the test packet has on congestion. It is also much easier to complete a single measurement, assuming *a priori*, smaller packets have a higher probability of not being discarded on highly congested links.

## TEST CONFIGURATION

For each site, **Traceroute(8)** was used to trace the path from a computer on a base to the host site, WWW.CNN.COM. Using reverse DNS lookup and the **whois(8)** program, the administrative domains of each router hop was determined for transit network in Department of Defense (DoD) domain(s). In general, the autonomous domain transit path was either;

- AFB ↔ AFIN ↔ DISA ↔ INTERNET; or
- AFB ↔ AFIN ↔ DLA-DISA ↔ INTERNET.

The **Traceroute(8)** program was terminated when the trace was two router hops past the DISA gateway, into the commercial Internet. For purposes of these tests we were primarily interested in DoD related congestion. Also, because the network was highly congested, testing additional commercial router hops would have significantly reduced the number of samples during the averaging period.

## UNCERTAINTIES AND ERRORS

All measurement methodologies, even very well designed metrics, have uncertainties and errors [4]. The series of tests performed here were certainly no exception. For example, both UDP and ICMP datagrams are connectionless and during periods of high congestion datagrams are discarded. The **Traceroute(8)** program defaults to three UDP probes for each router hop measured; yet in numerous probes during congested periods, it is possible all three datagrams are discarded by the network. This creates a situation where a decision must be made on how these events are managed by the test script. This includes assigning a default RTT metric when all probes are discarded.

Historically, router software assigns a low priority to ICMP packets. This introduces uncertainty in the latency as ICMP packets are returned by the routers. Normally, routers with a very high CPU utilization will forward packets slower than routers with plenty of excess CPU capacity. Also, if routes within transit networks change dynamically during the test period effecting the path length or hop count, the sampling program should compensate for this change.

During the testing periods, the paths for each test remained static, greatly simplifying the procedures. I should note, however, this is not always the case. Tests were repeated during subsets of time intervals for both duty-hour tests and non-duty hour tests. Test averages correlated better than one percent over a three day period. Therefore, for purposes of discussion, the tests provide a good baseline as well as a realistic indication of the average congestion in the transit domains.

## MEASURED TEST RESULTS

Four bases were selected by USAF personnel for the tests. At each base the test program repeated the delay measuring process, averaging the results for time intervals of as little as four duty-hours to a many as twelve non-duty hours. In selected tests, 18,000 samples were averaged.

In all graphs, the y-axis represents the RTT of a 40 byte packet in milliseconds. The labels below the x-axis are directly under rectangles representing a transit IP service provider. The darker line in the figures represent the test results during duty-hours. The lighter line illustrates non-duty hour RTTs.

## LANGLEY AFB TEST RESULTS

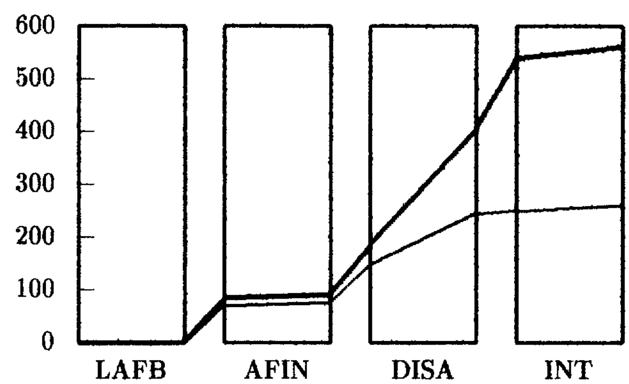


Fig. 1. Langley AFB Delay vs. Domain

Off duty-hour test results for Langley AFB were obtained over a twelve hour time period from 2000 hours to 0800 hours during a normal week day. The total number of samples averaged were over 9000. Different tests over duty-hours at Langley AFB were made and the test results of the individual tests were within a few milliseconds. The duty-hour data (always the highest) shown in Fig. 1 represents a single test of 469 samples. Internet congestion was too high for thousands of test samples during duty-hours. There were four router hops across Langley AFB in these tests. The average RTT across all four hops was 2.5 *ms*. At the time of this measurement, the Langley AFB IP backbone was a meshed 10BaseFL topology.

As a side note, the FDDI backbone link at Langley AFB averaged between 0.2 and 0.4 percent utilization. The most heavily congested 10BaseFL backbone links average about 5 percent utilization and peak between 8 and 9 percent. As the measurements indicate, the base network is not causal to Internet reachability problems.

## BARKSDALE AFB TEST RESULTS

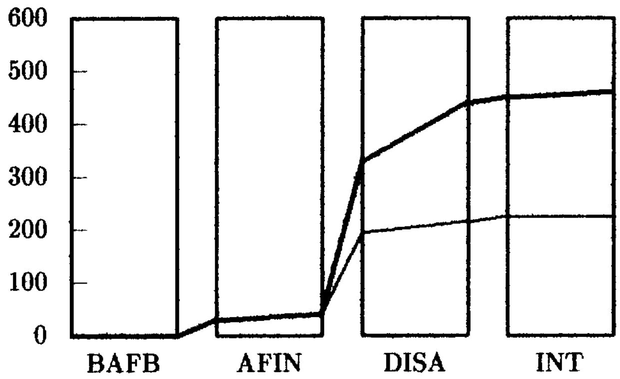


Fig. 2. Barksdale AFB Delay vs. Domain

Off duty-hour test results for Barksdale AFB were obtained over a twelve hour time period from 2000 hours to 0800 hours. The total number of samples averaged for one specified test period was over 18,000. The data shown for the duty-hour graph represents one test of 525 samples. There were two router hops across Barksdale AFB for these series of tests. The average RTT across the two router hops on base was 2.2 *ms*.

## DAVIS-MONTHAN AFB TEST RESULTS

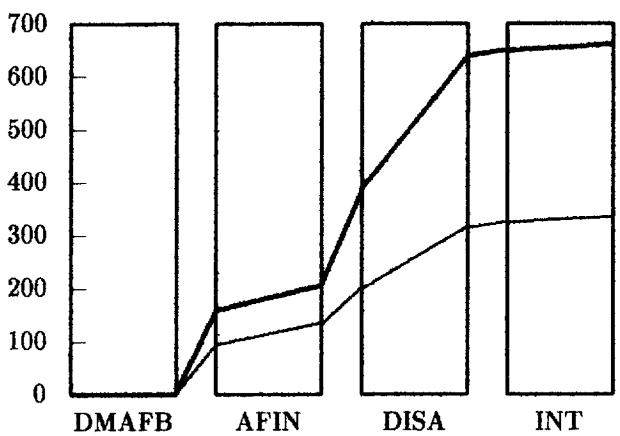


Fig. 3. Davis-Monthan AFB Test Results

Off duty-hour test results for Davis-Monthan AFB were obtained over a six hour period from 1930 hours to 0230 hours. 5,000 samples averaged in the testing period. Data shown for duty-hours testing represents 525 samples. There were two router hops across Davis-Monthan AFB for these series of tests. The average RTT across the two hops was 0.7 *ms*. During the testing period, the Davis-Monthan IP backbone transport was OC3-ATM.

## MINOT AFB TEST RESULTS

Off duty-hour test results for Minot AFB were obtained over a six hour period from 1930 hours to 0230 hours. The total number of samples averaged for this test was well over 5,000. The data shown for duty-hours testing represents five samples. The test for Minot AFB was tried

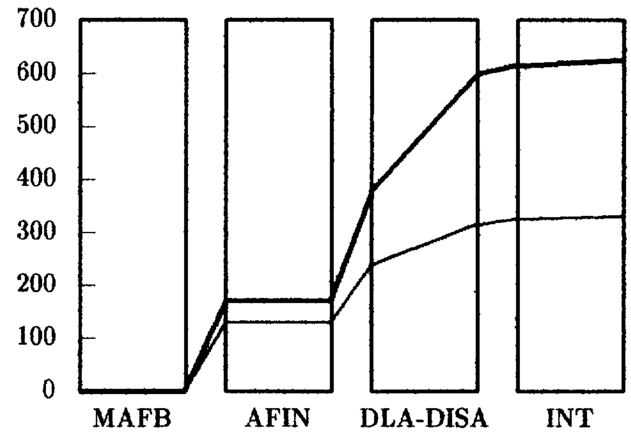


Fig. 4. Minot AFB Test Results

repeatedly during duty-hours; however, during the test period the network was extremely congested. Hence, establishing a good connection with the Internet from Minot AFB was almost impossible. There were two router hops across Minot AFB with an average RTT of 2.1 *ms*.

## DISCUSSION

In all test results the RTT across the base backbones were small. Current backbones at the bases do not exhibit noticeable congestion. Relative to the congestion across DISN, USAF base backbone congestion is negligible.

In most circumstances, AFIN links contribute a minimal amount of delay to Internet access and do not indicate heavy congestion. As expected, the link between ACC bases and AFIN contribute to delay proportional to the bandwidth. Davis-Monthan AFB, for example, had a 56 kbps circuit to AFIN and shows the highest RTT of all the ACC-AFIN links sampled. Barksdale AFB had a full T-1 to AFIN, showing the lowest delay, also expected.

DISN links are highly congested during duty-hours and show the highest level of congestion. In fact, during many time intervals within duty-hours, relative to Internet access, the DISN links are in a state of congestion collapse. Attempts to access the Internet from USAF bases are often impossible.

The USAF loads DISN with more IP traffic than any other branch of the service [5]. Load factors are actually higher than indicated because the graphical data does not adequately illustrate congestion and delay in DISN and DISN gateways. This is due to the connectionless nature of IP datagrams used for testing. The congestion is actually higher than indicated in the graphs because a failure of one traceroute cycle (three UDP probes) were **intentionally** averaged as a zero RTT in order to underestimate the congestion! Increasing the default RTT for packets which are discarded in the network due to congestion would increase the RTT times illustrated. More tests are appropriate to ascertain accurate default RTT values if a more exact congestion estimate is required. Selecting the correct value for default RTTs when packets are discarded due to congestion merits further research and discussion. For the objective of these tests, illustrating transit domains where congestion was a problem, further refinement was not deemed necessary.

The RTT estimates in this paper have been intentionally *underestimated* to illustrate the relative degree of congestion in DISN. Functional areas in the USAF have mission requirements to access information archived on WWW sites in the Internet; however, the current degree of congestion encountered by users attempting to reach the Internet is unacceptably high, impairing the mission. Moreover, congestion often increases to the threshold of *congestion collapse* during peak periods in an average work day. Connection attempts to the Internet are abandoned. This is causal to a situation analogous to highway commuters who choose to remain home vis-a-vis fighting highly congested rush hour traffic.

## CONCLUSION

Internet access to WWW sites from the USAF bases tested does not meet mission requirements. The largest contributor to the congestion was the DISN network and the gateways between AFIN and DISN. Congestion across individual base backbones was insignificant. Gateway links between AFIN and the bases contribute to congestion, but to a much lesser degree than DISN. Upgrading USAF links between the bases and AFIN will help lower congestion; however, congestion collapse in DISN will continue if dramatic improvements in DISN does not occur.

## ACKNOWLEDGMENTS

This work was funded, in part, prior to the author joining SAIC, via contract #F19630-93-D-001, the PRC Super-Mini Program. I am deeply appreciative for the excellent team relationships fostered with PRC professionals over the past four years, especially Mr. Ray Landers and Mr. Bill Daus. A special thanks is in order for USAF BNCC (AFNCC) personnel at Barksdale, Davis-Monthan, and Minot AFB for providing remote access to servers to perform these tests with very short notice. RAF Squadron Leader Jim Lindsay's editorial comments on the first draft of this report was extremely helpful, motivational, and much appreciated.

## APPENDIX: PERL SCRIPT SUGGESTIONS

The series of tests in this paper were performed over a three day period which included writing and testing the scripts, securing remote access to servers on four ACC bases, modifying and configuring scripts, and running the tests. Because of the short time frame, little effort was concentrated in making the scripts auto-configurable for specific bases. Configuring the scripts was performed manually.

From the purist programmers perspective the scripts could be written more compactly. However, many USAF personnel are not trained in writing scripts, including PERL scripts; so I intentionally unrolled the loops to make reading the code easier for script writing novices. For these, and similar, network management tests, PERL is a very valuable tool to wrap around networking utilities and this test is but one example of the usefulness of PERL in Network Control Centers and network management.

As previously mentioned, the script should take into consideration the possibility of increased path lengths and a changing topology due to dynamic IP routing protocols. In these tests, we were careful to manually watch for this possibility, but an automated technique within the script is a much needed improvement. Automating the task of determining the domain of the routers is also a possible enhancement. This is certainly true if non-prototype test scripts are to be distributed to numerous sites and the tests performed by a wide range of personnel with varying levels of script writing and internetworking expertise.

## BRIEF LIST OF ACRONYMS

| | |
|---|---|
| ACC | Air Combat Command |
| AFB | Air Force Base |
| AFIN | Air Force Internet |
| AFNCC | Air Force Network Control Center |
| BAFB | Barksdale AFB |
| DISA | Defense Information Systems Agency |
| DISN | DISA's IP Network |
| DLA | Defense Logistic Agency |
| DMAFB | Davis-Monthan AFB |
| DNS | Domain Name System |
| ICMP | Internet Control Message Protocol |
| INT | The Internet |
| IP | Internet Protocol |
| LAFB | Langley AFB |
| MAFB | Minot AFB |
| RTT | Round Trip Time |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| WWW | World Wide Web |

## REFERENCES

[1] Tanenbaum, A., *Computer Networks*, 3rd edition, Prentice-Hall, Eaglewood, NY, 1996.
[2] Comer, D., *Internetworking with TCP/IP, Volume I*, 2nd ed., Prentice-Hall, Eaglewood, NY, 1991.
[3] Stevens, R., *TCP/IP Illustrated, Volume 3*, Addison-Wesley, Reading, MA, 1996.
[4] Almes, G., et. al., *Framework for IP Provider Metrics*, Internet Draft, ANS, BellCore, PSC & LBL, November 1996.
[5] Morris, D., *Re: DISA Delays*, Private Communications, morrisd@nrcm.disa.mil, Jan, 17, 1997.

### BIOGRAPHY

**Tim Bass** (*bass@silkroad.com*) is a technical director with *SAIC, Center for Information Protection*, McLean, VA. Mr. Bass graduated from Tulane University in 1987 with a B.S.E. with Departmental Honors in Electrical Engineering. He played a principal role in building the SprintLink Integrated Network Management Center and the Sprint Managed Router Network organization in 1993. Recently, he completed the design and implementation of the original Base Network Control Center (BNCC) prototype for the USAF and designed and built over 20 Network Control Centers for USAF bases worldwide. His current interests are TCP/IP performance analysis, high-speed internetworking, wireless internetworks and network security issues.

# EXAMPLE: PERL SCRIPT FOR MINOT AFB

```
#!/usr/bin/perl
######################################################################
#       Author:         Tim Bass                                     #
#       Version:        0.01 for Minot AFB                           #
#       Date:           Jan. 13, 1997                                #
#       Note:           prototype not supported, use at your own risk #
######################################################################
for(;;)
{
open(LOGFILE,">>/tmp/minot.out") || die "Can't Open LOGFILE";
open(TRACE,"/usr/local/bin/traceroute -n www.cnn.com|") || die "Can't Open TRACEROUTE";
while(<TRACE>) {
chop;
if (/^ 1/) {     #print LOGFILE "$_\n\n";
                 $HOP1_COUNT++; $DELAY=&get;
                 $HOP1_TOTAL = $HOP1_TOTAL + $DELAY;
                 print LOGFILE "MINOT AFB HOP1 SUM = $HOP1_TOTAL COUNT = $HOP1_COUNT";
                 $a = $HOP1_TOTAL/$HOP1_COUNT;
                 print LOGFILE " HOP 1 AVG = $a\n";}

elsif (/^ 2/) {#print LOGFILE "$_\n";
                 $HOP2_COUNT++; $DELAY=&get;
                 $HOP2_TOTAL = $HOP2_TOTAL + $DELAY;
                 print LOGFILE "MINOT AFB HOP2 SUM = $HOP2_TOTAL COUNT = $HOP2_COUNT";
                 $a = $HOP2_TOTAL/$HOP2_COUNT;
                 print LOGFILE " HOP 2 AVG = $a\n";}

elsif (/^ 3/) {#print LOGFILE "$_\n";
                 $HOP3_COUNT++; $DELAY=&get;
                 $HOP3_TOTAL = $HOP3_TOTAL + $DELAY;
                 print LOGFILE "AFIN HOP3 SUM = $HOP3_TOTAL COUNT = $HOP3_COUNT";
                 $a = $HOP3_TOTAL/$HOP3_COUNT;
                 print LOGFILE " HOP 3 AVG = $a\n";}

elsif (/^ 4/) {#print LOGFILE "$_\n";
                 $HOP4_COUNT++; $DELAY=&get;
                 $HOP4_TOTAL = $HOP4_TOTAL + $DELAY;
                 print LOGFILE "DLA HOP4 SUM = $HOP4_TOTAL COUNT = $HOP4_COUNT";
                 $a = $HOP4_TOTAL/$HOP4_COUNT;
                 print LOGFILE " HOP 4 AVG = $a\n";}

elsif (/^ 5/) {#print LOGFILE "$_\n";
                 $HOP5_COUNT++; $DELAY=&get;
                 $HOP5_TOTAL = $HOP5_TOTAL + $DELAY;
                 print LOGFILE "DLA HOP5 SUM = $HOP5_TOTAL COUNT = $HOP5_COUNT";
                 if (/\*/) {$HOP5[5]=3000;}
                 $a = $HOP5_TOTAL/$HOP5_COUNT;
                 print LOGFILE " HOP 5 AVG = $a\n";}

elsif (/^ 6/) {#print LOGFILE "$_\n";
                 $HOP6_COUNT++; $DELAY=&get;
                 $HOP6_TOTAL = $HOP6_TOTAL + $DELAY;
                 print LOGFILE "DLA HOP6 SUM = $HOP6_TOTAL COUNT = $HOP6_COUNT";
                 $a = $HOP6_TOTAL/$HOP6_COUNT;
                 print LOGFILE " HOP 6 AVG = $a\n";}

elsif (/^ 7/) {#print LOGFILE "$_\n";
```

```
                $HOP7_COUNT++; $DELAY=&get;
                $HOP7_TOTAL = $HOP7_TOTAL + $DELAY;
                #print LOGFILE "$HOP7[5]\n";
                print LOGFILE "DISA HOP7 SUM = $HOP7_TOTAL COUNT = $HOP7_COUNT";
                $a = $HOP7_TOTAL/$HOP7_COUNT;
                print LOGFILE " HOP 7 AVG = $a\n";}

elsif (/^ 8/) {#print LOGFILE "$_\n";
                $HOP8_COUNT++; $DELAY=&get;
                $HOP8_TOTAL = $HOP8_TOTAL + $DELAY;
                #print LOGFILE "$HOP8[5]\n";
                print LOGFILE "INT HOP8 SUM = $HOP8_TOTAL COUNT = $HOP8_COUNT":
                $a = $HOP8_TOTAL/$HOP8_COUNT;
                print LOGFILE " HOP 8 AVG = $a\n";}

elsif (/^ 9/) {#print LOGFILE "$_\n";
                $HOP9_COUNT++; $DELAY=&get;
                $HOP9_TOTAL = $HOP9_TOTAL + $DELAY;
                #print LOGFILE "$HOP9[5]\n";
                print LOGFILE "INT HOP9 SUM = $HOP9_TOTAL COUNT = $HOP9_COUNT";
                $a = $HOP9_TOTAL/$HOP9_COUNT;
                print LOGFILE " HOP 9 AVG = $a\n\n";close(LOGFILE);last;

elsif (/^10/) {last;}
        }
}
sub get
        {
                $DELAY=0; @HOP=split(/ /,$_);
                for($i=2;$i<9;$i++)
                        {
                        $_=$HOP[$i];
                        if (!/\d/ || /\./) {next;}
                        $DELAY=$_;break;
                        }

                if ($DELAY==0) {$DELAY=3000;}
                return $DELAY;
        }
```

## TEST OUTPUT FOR MINOT AFB (OFF DUTY HOURS)

```
MINOT AFB HOP1 SUM = 10475   COUNT = 5562 HOP 1 AVG = 1.88331535418914
MINOT AFB HOP2 SUM = 12126   COUNT = 5562 HOP 2 AVG = 2.18015102481122
AFIN      HOP3 SUM = 712884  COUNT = 5562 HOP 3 AVG = 128.170442286947
DLA       HOP4 SUM = 1339958 COUNT = 5562 HOP 4 AVG = 240.912980942107
DLA       HOP5 SUM = 1350014 COUNT = 5562 HOP 5 AVG = 242.720963682129
DLA       HOP6 SUM = 1354896 COUNT = 5562 HOP 6 AVG = 243.598705501618
DISA      HOP7 SUM = 1748989 COUNT = 5562 HOP 7 AVG = 314.453254225099
INT       HOP8 SUM = 1535519 COUNT = 5562 HOP 8 AVG = 276.073175116864
INT       HOP9 SUM = 1513751 COUNT = 5562 HOP 9 AVG = 277.15947500899
```

Note: The PERL script above prints the results to many decimal places, which is ignored. For graphical purposes, the results have been rounded to the nearest 5th *ms*. For example: 242 is rounded down to 240 and 314 is rounded up to 315.