

# Chapter 1

## Performance Analysis

Hundreds of tests were done using a socket connection Perl script. The Perl script would establish a connection and transmit configuration parameters for both the client and server. The script would wait a given time to verify the settings were accepted on the server and use apache bench as a web benchmark. Apache bench contains statistical information. For many tests, the webpage was retrieved up to 5 times and if the mean and median were not within a given threshold, the test would be invalidated.

These results were collected and verified with many tests. The results below are of the average bandwidth given in kilobytes per second. A diagram of the setup will help visualize the test scenario. These results will be presented more formally in the next chapter. The computers in the testbed and used for these results were 10 machines (with the same configuration) running Linux Fedora Core One with 600 MHz Pentium III processors and 256 megabytes of memory with a modified 2.4.24 Linux kernel. Kernel modifications were in the INET network implementation, and additional proc values were added to enable access to these kernel values.

### 1.1 Baseline results

These are the baseline results with no bandwidth limiting, weighting different paths, or buffering incoming packets. The remainder of the chapter tests deals with these different scenarios. These results are straight baseline results. Below in Fig 4.1 shows the four different types of connections.

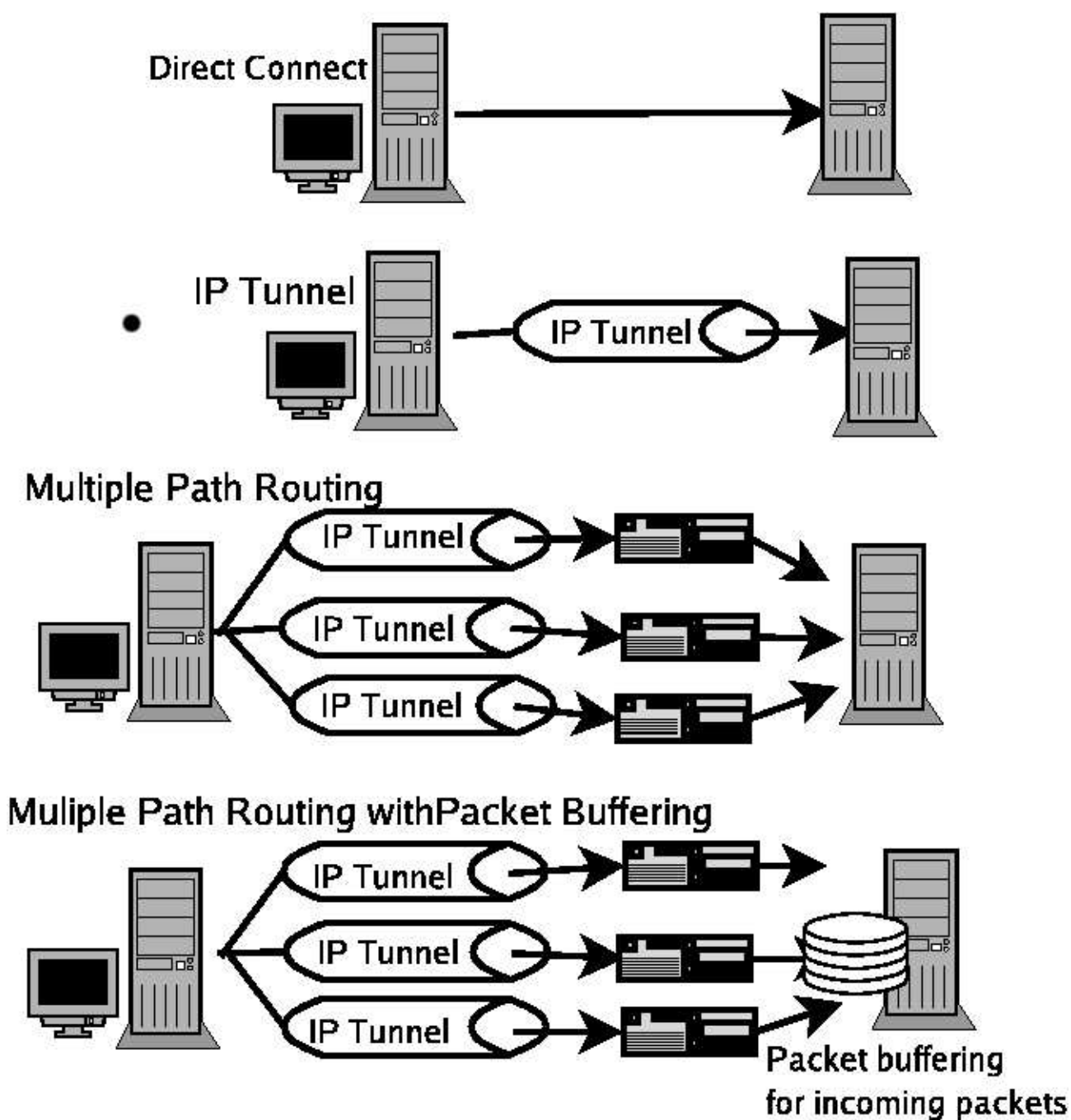


Figure 1.1: Types of connections

Direct connection is a connection without any modifications. IP Tunnel is a connection using packet with two IP headers. The first header goes to the intermediate proxy (or gateway) and is stripped off. The second header contains the destination's IP address and is routed to the destination by the intermediate proxy.

Multiple path routing uses IP tunneling and kernel modifications to switch the intermediate server. Multiple Path Routing with Packet Buffering has a buffer before the TCP processing kernel code. This buffer sorts out of sequence packets.

Below are the results both in table 4.1 and fig 4.2 displaying the control bandwidth for the different types of connections.