

Course ISE 431: Distributed Information Systems

Recitation 4 Exercise

Michael J. May

November 10, 2009

1 In-process invocation

Describe the experiments you performed to reach results for the following two questions: the number of repeats of the operation being measured, the number of the experiments, how you selected the final result, and its accuracy.

- A How long does a procedure call with no arguments and no body take?
- B How long does a procedure call with 10 arguments and no body take?

2 Inter-process, intra-machine invocation

Create a client program and a server program, described in the following. Use some sensible port number for the server, unique to yourself.

Write a program which, when given "client" as an argument does the following repeatedly: calls "send" with a dummy data argument of size datasize (given as a second program argument), followed by "receive from" with a 4-byte data argument.

When given "server" as an argument it is to repeatedly call recvfrom with a data argument of size datasize (given as the second program argument), then make a reply using sendto with a 4-byte data argument. Note that this server does not perform any useful processing upon the arriving requests. Nor does it do a lookup to identify a (dummy) procedure or method to call.

- Run a client and server together AT THE SAME MACHINE, and test that they work together.

- A How long does a single request-reply interaction take when datasize is
 - (a) 4 bytes?
 - (b) 1024 bytes?

State the accuracy of your answers, and how you arrived at them from the measurements.

- B How do you think the elapsed time for a local request-reply interaction is spent? Give a qualitative answer.

3 Inter-machine invocation

Run the server program at a dedicated workstation and the client program at your own workstation

- A How long does a single request-reply interaction take when datasize is

- (a) 4 bytes?
- (b) 1024 bytes?

State the accuracy of your answers, and how you arrived at them from the measurements

B `sendto` uses UDP. This places an 8-byte header onto an IP packet, which in turn uses a 20-byte header. An Ethernet packet carries 18 bytes of its own header and checksum. The minimum Ethernet packet size is 64 bytes. The Ethernet bandwidth is 10Mbits/sec.

What is the total hardware transmission time (time "on the wire") for the packets in a single remote request-reply interaction?

- (a) 4 bytes?
- (b) 1024 bytes?

How do you think the rest of the elapsed time for a remote request-reply interaction is spent? Give a qualitative description, in terms of differences to your answer to 2B.

4 Inter-machine asynchronous invocation

- Adapt your previous program, to produce the following. As a client, the program repeatedly calls `sendto` but not `recvfrom`. As a server, it repeatedly calls `recvfrom` and does not reply. Make the server report if any messages are lost.
- Run a client and server at separate, dedicated machines. NB You may find that `sendto` fails for lack of buffer space after a certain number of sends. Try to accommodate this by reducing the number of attempted sends each time.

A Do messages get lost? If so, how? Under what conditions did you find losses? How did you detect losses?

B How long does each client request take if `datasize` is

- (a) 4 bytes
- (b) 1024 bytes?

5 TCP and Connected Sockets

- Do `netstat /?`, to find out how to show per-protocol packet statistics.
- Adapt your previous program so that the client and server use a single pair of connected sockets for request-reply communication.

A How long does a single request-reply interaction take when `datasize` is

- (a) 4 bytes?
- (b) 1024 bytes?

B Use `netstat` to count the number of packets involved in a request-reply exchange. How do you think the elapsed time for a remote request-reply interaction is spent? Give your answer in terms of whether/how it differs from your answer to 3B, and in particular refer to the fact that TCP is a reliable protocol.

6 TCP for Request/reply Interactions?

- A If you were designing an RPC system under Unix, would you use connected or unconnected sockets? Why? What are the merits of the other choice?
- B If the server takes a long time to reply, then does TCP send additional "are you still there?" or acknowledgement messages? Describe an experiment performed to check whether/how the TCP protocol actions vary with the amount of processing performed by the server. Comment on your results. Attach your program and any related information.