Network Troubleshooting Tools

1 Background

In the lecture we have looked at various techniques for using common tools to make measurements of network conditions. Here we get some practice with some of these techniques.

2 Procedure

2.1 Measuring throughput with `ttcp`, `ping`

1. Choose a partner.
2. Use the `ttcp` program to measure throughput between your machines. Take several measurements. Take turns in being the client (`ttcp -r -s`) and server (`ttcp -t -s` ⟨*ip-address*⟩). Remember to start the client first. `ttcp` gives throughput in bytes per second. Record the throughput measurements here in bits per second, indicating whether you were the client or server.
3. Ping your partner with two different sized packets. Do this a number of times. Calculate the throughput between your machines.

$$TP = 16 \times \frac{P_l - P_s}{t_l - t_s} \quad \text{bits per second}$$

where:

$$\begin{aligned} P_l &= \textbf{l}\text{arge packet size} \\ P_s &= \textbf{s}\text{mall packet size} \\ t_l &= \text{ping time for } \textbf{l}\text{arger packet} \\ t_s &= \text{ping time for } \textbf{s}\text{maller packet} \end{aligned}$$

Do your calculations here:

✎

How does this compare with your measurements using `ttcp`? Explain discrepancies. Try big packets: `ping -s 8872 gw`

4. Use the method described in the lectures (using `ping`) to measure the bandwidth between the gateway and `nickpc.tyict.vtc.edu.hk`. The formula is:

$$TP = 16 \times \frac{P_l - P_s}{t_{2l} - t_{2s} - t_{1l} + t_{1s}} \quad \text{bits per second}$$

where:

$$\begin{aligned}
P_l &= \boldsymbol{l}\text{arge packet size} \\
P_s &= \boldsymbol{s}\text{mall packet size} \\
t_{1l} &= \text{ping time for } \boldsymbol{l}\text{arger packet to the near link} \\
t_{1s} &= \text{ping time for } \boldsymbol{s}\text{maller packet to the near link} \\
t_{2l} &= \text{ping time for } \boldsymbol{l}\text{arger packet to the far link} \\
t_{2s} &= \text{ping time for } \boldsymbol{s}\text{maller packet to the far link}
\end{aligned}$$

Do your calculations here:

✎

5. Compare the ping times to the gateway and to `nickpc`. Do they surprise you? Explain what you see.

2.2 Ethereal and DHCP

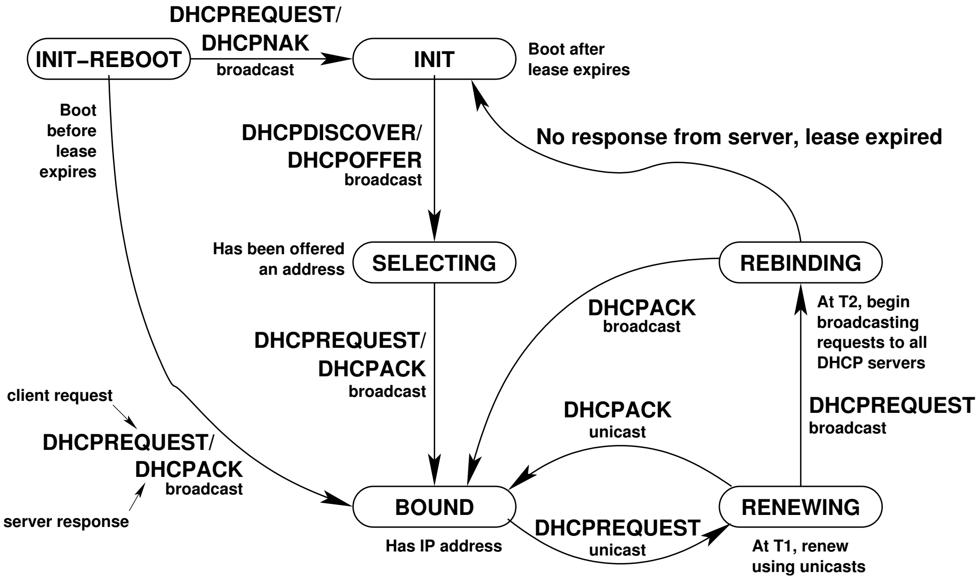

Figure 1: A state diagram showing states of a DHCP client. Note that T is the lease time, $T1 = \frac{T}{2}$, $T2 = \frac{7T}{8}$. See also table 1 on page 5 from the DHCP RFC 2131 (available in full at `/home/nfs/ietf/rcf/rfc2131.txt`), which sumarises DHCP messages.

1. Start up `ethereal`:

   ```
   $ ethereal &
   ```

2. Choose Capture → Start
3. For the Filter, enter: `port 67 or port 68`
4. Select (turn *on*) the check box item Update list of packets in real time, but make sure that the item Enable network name resolution is turned *off*.
5. Click on OK, and wait until you have captured at least 20 packets, preferably more.
6. Examine the exchanges between the DHCP servers and clients. Expand the ⊞ for the *Bootstrap Protocol*. Note that the *transaction ID* is the same for a DHCP session between client and server.
7. Figure 1 shows the states that a DHCP client passes through. Examine this together with the data from `ethereal`.
8. List the IP addresses of DHCP server(s). See table 1 on page 5 for the messages that come from servers. .... □
9. List the IP addresses of some clients that were successfully given an address:

✐**10.** Which port do clients use? . . . .

✐**11.** Which port do servers use? . . . .

12. What is the lease time? (Examine the `Bootp Flags` in the `DHCPACK` message)

✐ . . . .

13. Can you see any unauthorised DHCP servers? (These will be any server that is not

✐ `ictlab`!) . . . .

14. Identify a path through the state diagram in figure 1 on the previous page that you can see from your `ethereal` data. You can recognise a session by its *transaction ID*. Examine the *Bootstrap Protocol* section in the middle pane of the `ethereal` window.

✐

15. List the IP addresses of any clients that were denied a requested address; can you see why?

✐

Message		Use
DHCPDISCOVER	—	Client broadcast to locate available servers.
DHCPOFFER	—	Server to client in response to DHCPDISCOVER with offer of configuration parameters.
DHCPREQUEST	—	Client message to servers either (a) requesting offered parameters from one server and implicitly declining offers from all others, (b) confirming correctness of previously allocated address after, e.g., system reboot, or (c) extending the lease on a particular network address.
DHCPACK	—	Server to client with configuration parameters, including committed network address.
DHCPNAK	—	Server to client indicating client's notion of network address is incorrect (e.g., client has moved to new subnet) or client's lease as expired
DHCPDECLINE	—	Client to server indicating network address is already in use.
DHCPRELEASE	—	Client to server relinquishing network address and cancelling remaining lease.
DHCPINFORM	—	Client to server, asking only for local configuration parameters; client already has externally configured network address.

Table 1: DHCP Messages: this is "table 2" from RFC 2131; the RFC is available in full from `ictlab` at `/home/nfs/ietf/rcf/rfc2131.txt`.