Linux Networking: ping

David Morgan

IP packet structure

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IP’s Data Payload</td>
</tr>
</tbody>
</table>

© David Morgan 2003
**IP’s payload**

- **Theoretically**
  - Payload can be anything
  - IP is payload-oblivious, payload-indifferent
- **Actual practice**
  - particular well-defined protocols use IP
  - IP recognizes them with a header field

---

**IP packet structure**

- Source Address
- Destination Address
- Protocol Number
- IP’s Data Payload
IP’s Payload types –
subprotocol examples and their numbers

<table>
<thead>
<tr>
<th>Src</th>
<th>Dest</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UDP</td>
<td>(17) packet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dest</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICMP</td>
<td>(1) packet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Src</th>
<th>Dest</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCP</td>
<td>(6) packet</td>
</tr>
</tbody>
</table>

… and many others
http://www.iana.org/assignments/protocol-numbers
© David Morgan 2003

ping: a utility that uses icmp

- purpose: Tests connectivity
- method: Probes an address
- output: Reports whether there is a reply
- technique: Utilizes ICMP to do its work

© David Morgan 2003
**ping usage**

[root@EMACH1 /root]# ping -c3 66.218.71.81
PING 66.218.71.81 (66.218.71.81) from 64.130.228.61 : 56(84) bytes of data.
64 bytes from 66.218.71.81: icmp_seq=0 ttl=55 time=34.5 ms
64 bytes from 66.218.71.81: icmp_seq=1 ttl=55 time=33.6 ms
64 bytes from 66.218.71.81: icmp_seq=2 ttl=55 time=34.1 ms

--- 66.218.71.81 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 33.6/34.0/34.5 ms

[root@EMACH1 /root]#

…so we know 66.218.71.81 is alive

[root@EMACH1 /root]# ping -c3 66.218.71.17
PING 66.218.71.17 (66.218.71.17) from 64.130.228.61 : 56(84) bytes of data.

--- 66.218.71.17 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss

[root@EMACH1 /root]#

…so we don’t know if 66.218.71.17 is alive

© David Morgan 2003

---

**ICMP packet structure**

<table>
<thead>
<tr>
<th>ICMP-type</th>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>header of subject/wayward IP packet or other</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP-type dependent payload</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*(ICMP: Internet Control Message Protocol)*

© David Morgan 2003
ICMP/IP packet structure

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Destination Address</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP-type</td>
<td>Code</td>
<td>Checksum</td>
</tr>
<tr>
<td>header of subject/wayward IP packet or other ICMP-type dependent payload</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IP’s payload is an ICMP packet

Some icmp message types

0 echo reply
3 destination unreachable
8 echo request
11 time-to-live exceeded

…and more (see RFC 792)
ping operation

- ping causes an icmp “echo request”
- IP (tries to) deliver it
- ICMP requires an “echo reply” packet be issued in response to receipt of an “echo request”
- So ping’d computer (tries to) reply
- Pinging computer records incoming replies

ICMP packet example

Used for echo (ping) requests…

```
[root@EMACH1 /root]# ping www.acme.com
```

<table>
<thead>
<tr>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>echo request – “are you there?”</td>
</tr>
</tbody>
</table>
ICMP packet example

… and for echo (ping) relies.

<table>
<thead>
<tr>
<th></th>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

echo reply – “yes I’m here”

ping options

- `c` specify a count
- `i` wait x seconds between requests
- `s` specify packet size
ping options

ping w/o options will ping forever (ctrl-C kills)
ping –c5
   ping 5 times and stop

ping w/o options pings at 1-second interval
ping –i5
   ping every 5 seconds

An application of ping:
crude latency test

<table>
<thead>
<tr>
<th>ping</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.ucla.edu">www.ucla.edu</a></td>
<td>L.A.</td>
<td>57ms</td>
</tr>
<tr>
<td>dandelion-patch.mit.edu</td>
<td>Boston</td>
<td>105</td>
</tr>
<tr>
<td><a href="http://www.pku.edu.cn">www.pku.edu.cn</a></td>
<td>Beijing</td>
<td>238</td>
</tr>
<tr>
<td><a href="http://www.u-tokyo.ac.jp">www.u-tokyo.ac.jp</a></td>
<td>Tokyo</td>
<td>181</td>
</tr>
<tr>
<td><a href="http://www.usyd.edu.au">www.usyd.edu.au</a></td>
<td>Sydney</td>
<td>213</td>
</tr>
</tbody>
</table>
In linux it’s a turn-off

echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts