CSci551 Spring 2006 Thursday Section Homework 2

Assigned: February 9, 2006. Due: noon, March 1, 2006.

You are welcome to discuss your homework with other students, but each student is expected write his or her final answer independently.

Answers should be *short* and *to the point*. Complete sentences are not required. It is possible answer the homework in one page of text, and if your answers are much more than that you should ask yourself if you're answering clearly and succinctly. We reserve the right to deduct points on answers that are too long or that miss the point.

This homework must be submitted electronically. Choice of formats is:

- Simple text (ASCII). This format is *strongly* preferred since it can be most easily marked up and returned to you.
- PDF. Please use PDF *only* if you have figures in your text.
- Postscript or HTML. Use these only if you have figures and cannot generate PDF.

Note that MS-Word or other proprietary formats are *not* accepted. See the course web page for details about how to generate postscript or PDF if you start with these formats. (See the course web site for information about how to generate PDF.)

To submit your homework, upload it to aludra.usc.edu or nunki.usc.edu, then use the submit comment as you did for the project:

```
% submit -user csci551 -tag hw2 file1 [ file2 ... ]
```

You should have one file for the body of your homework. You may have additional files if you generate html. If you have multiple files, please list them all on the command line separately (do *not* combine them ahead of time in a tar or zip file). Just a reminder: your name and student id should appear at the top of the first page of your response.

1: This question explores how protocol headers are processed in real operating systems.

First, find the Internet Request for Comments (RFC) that specifies the ICMP protocol. a) What RFC number is the official reference for this protocol? b) What is the official specification of the message format for an ICMP "echo reply" message? (Cut and paste from the RFC.)

Find the ICMP header representation for some common operating system in the system's C header files. c) What operating system and version did you pick? d) What is the relevant structure that represents the ICMP header? (copy and paste it into your answer)

A challenge in implementing protocols is doing so *portably*. The C language leaves many details up to the compiler, including structure packing and size of integers. (Yes, in general, an "int" is not necessarily 32-bits wide! On 64-bit computers, it might be 64 bits.)

e) Does that structure have code to handle compiling on different architectures (yes or no)?

If it does, f) How does it handle different sizes of "int"? g) How does it handle potentially different structure packing rules? (For these, clearly indicate the mechanism it uses with references to specific elements in the code.)

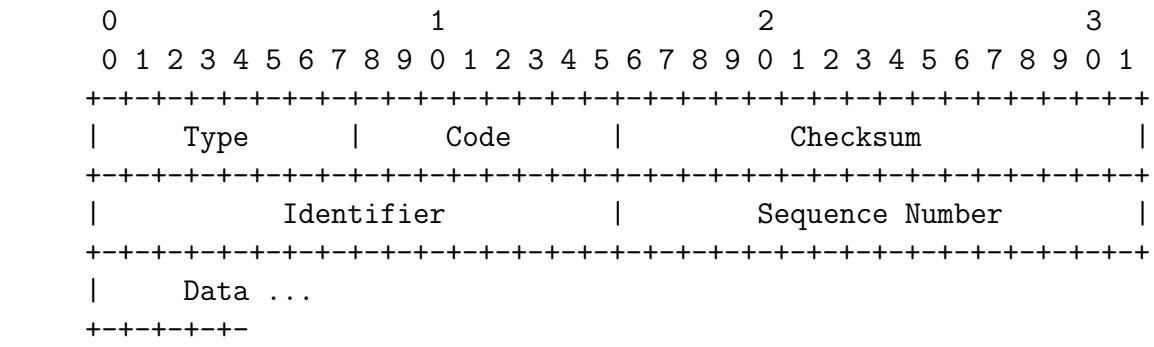
If it does not, then f) How would you modify it to handle sizes of "int"? g) How would you modify it to handle different structure packing rules? (For these, clearly indicate the mechanism it uses with references to specific elements in the code.)

Pick some *other* operating system than the one you chose in step (a). (Other means a completely different OS, not just a different version of the same OS.) h) What other OS and version did you pick? i) What is the relevant structure that represents the TCP header? (copy and paste it into your answer) j) How does the approach to handling portability taken by this other OS compare to what you found in the first OS?

Answer: see below

a) RFC 792. (Full cite, not required: Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification", RFC 792, USC/Information Sciences Institute, September 1981.)

b)



c) Linux 2.6.14 d)

```
(from /usr/include/linux/icmp.h:)

struct icmphdr {
  __u8          type;
  __u8          code;
  __u16         checksum;
  union {
        struct {
                __u16   id;
                __u16   sequence;
        } echo;
        __u32   gateway;
        struct {
                __u16   __unused;
                __u16   mtu;
```

```
        } frag;
  } un;
};
```

e) Yes

f) It uses __u16 and __u32 which are externally defined to guarantee specific data sizes.

g) It uses substructures (like struct frag) to guarantee alignment of 16-bit quantities according to the spec.

OR

c) FreeBSD 4.4

d)

```
/*
 * Structure of an icmp header.
 */
struct icmp {
        u_int8_t  icmp_type;            /* type of message, see below */
        u_int8_t  icmp_code;            /* type sub code */
        u_int16_t icmp_cksum;           /* ones complement cksum of struct */
        union {
                u_int8_t ih_pptr;                       /* ICMP_PARAMPROB */
                struct in_addr ih_gwaddr;       /* ICMP_REDIRECT */
                struct ih_idseq {
                        u_int16_t icd_id;
                        u_int16_t icd_seq;
                } ih_idseq;
                u_int32_t ih_void;

                /* ICMP_UNREACH_NEEDFRAG -- Path MTU Discovery (RFC1191) */
                struct ih_pmtu {
                        u_int16_t ipm_void;
                        u_int16_t ipm_nextmtu;
                } ih_pmtu;
        } icmp_hun;
```

e) Yes

f) Different sizes of integer are handled by defining the datatype of all the desired fields of the header by "uint16_t / uint8_t / uint32_t".

g) It uses substructures (like struct ih_pmtu) to guarantee alignment of 16-bit quantities according to the spec.

2: In class we talked about naming and addressing with respect to networks.

Consider a network with a firewall, where the topology is:

INSIDE—FIREWALL—OUTSIDE

"Inside" represents a network with hosts that are protected by the firewall. The firewall is a router that filters traffic that passes through it, allowing some to pass through and stopping others. "Outside" is the hard, cold Internet.

IP addresses identify interfaces (attachment points), not hosts. a) Why does network identification of interfaces useful when connecting to the firewall to configure it?

Suppose we deploy a web server in parallel with the firewall (i.e., it is connected to both the internal and external networks). This web server is on both the internal and external network. b) Why would user requests from the general Internet be unable to connect to the web server? (Assume that it is functioning correctly. Think about how naming could cause a problem.)

Answer: a) Identification of interfaces helps because it means that one can log in to the router to its "inside" or "outside" interface. Thus someone coming from the inside can be given more control than someone coming from the outside.

(If one could only identify hosts, then one could only connect to the firewall itself, not the internal interface of the firewall.)

This answer assumes that the firewall can block requests from the outside. (Again, it does this because it can distinguish which traffic comes in on which interface.)

b) If the user uses the name for the internal interface of the web server, then the user's traffic from outside the Internet might be blocked by the firewall.

(In more detail: the web server has two interfaces, inside and outside. Let's assume the inside one has IP address 11.0.0.1 and the outside has 12.0.0.1. The web server's DNS record may list both of these records, binding www.example.com to both 11.0.0.1 and 12.0.0.1. If the users' browser gets both records and then tries only the first one it gets, and the first one is 11.0.0.1, then traffic must go from outside the network to interface 11.0.0.1. This traffic might have to go through the firewall and through the internal network, but the firewall is probably configured to drop most traffic to the internal network that originates from the outside network.)

3: Consider a TCP connection between USC and Karlsrühe, Germany. Assume there is a dedicated link with basic Ethernet speeds (10Mb/s) the round-trip time is 150ms, and that we're using 1500B-long segments. (For simplicity, assume here the 1M is 1 000 000, not a power of two.) a) How big a TCP window, in segments, is required to send at full rate, assuming no packet losses, that every segment is acknowledged? b) How long, in RTTs, will it take for TCP to saturate this link, assuming no losses and that the receiver ACKs every segment and the initial window size is 1 segment?

Now assume the receiver uses delayed ACKs and Reno TCP (otherwise continue with the assumptions above). c) (2pts) How big will the window be now be after the second RTT completes?

d) (2pts) If the loss rate is 4%, will TCP be able to keep the link saturated? (Say both yes or no and briefly justify your answer.)

Answer: a) The bandwidth delay product, $.15s \times 10Mb/s \times 1\text{segment}/1500B \times 1B/8b =$

25, 125 segments.

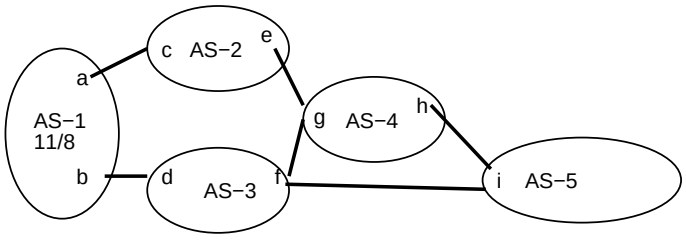
b) 1, 2, 4, 8, 16, 32, 64, 128 $\Rightarrow$ 8 RTTs

c) 1 packet, 1 ACK; 2 segments, 1 ACK $\Rightarrow$ 3 segments. (Alternative answer, assuming 1 more RTT happens, 5 segments.) (It doesn't increase strictly 1, 2, 4... because TCP only increases by 1 segment per *ACK*, not per *ACKed segment*.)

Half credit: 4 segments (wrong answer since it ignores delay acks).

d) No, it will not keep the link saturated. With 4% loss there will be typically 4 packets lost each RTT if the window is fully open. TCP cannot easily recover from that many losses per RTT.

4: Consider the following set of ASes, assuming they're all using BGP routing.



For this question, please write routes in the format "address: next-hop-router (AS-path)", where AS path reads from the last-hop-AS to the origin-AS. a) What route(s) does AS-1 announce directly to AS-2? b) What are all the routes that AS-2 will consider to get to 11/8?

c) What are all the routes that AS-3 will consider to get to 11/8?

d) Assuming no special actions are taken, will traffic from AS-5 go to AS-1 via AS-2 or AS-3? e) Why?

Suppose AS-1 wanted traffic to it from AS-5 to go through the alternate AS (i.e., the AS that was not your answer in part c). One way it could accomplish this policy is to use AS-PATH inflation. g) To which AS must it change what it announces, AS-2 or AS-3? h) What should its new announcement to this AS be?

Answer: points: a) 1, b) 4, c) 1, d) 3, e) 1, f) 2.

a) 11/8: a (1) (Common wrong answer: AS-1 says both.)

b) 11/8: c (2 1) and 11/8: e (2 4 3 1)

c) 11:8 d (3 1) and 11/8: f (3 4 2 1) and 11/8: f (5 4 2 1)

d) through AS-3 e) Because BGP routing prefers shortest paths in terms of AS-hops

e) to AS-3

f) 11/8: b (1 1 1).

This announcement would cause AS-5's traffic to prefer the route 11/8: h (4 2 1) to 11/8: e (3 1 1 1).