Marking Scheme:

<table>
<thead>
<tr>
<th>Question</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/60</td>
</tr>
<tr>
<td>2</td>
<td>/10</td>
</tr>
<tr>
<td>3</td>
<td>/10</td>
</tr>
<tr>
<td>4</td>
<td>/10</td>
</tr>
<tr>
<td>5</td>
<td>/10</td>
</tr>
<tr>
<td>Total</td>
<td>/100</td>
</tr>
</tbody>
</table>

(a) ARP request or reply format.

<table>
<thead>
<tr>
<th>Destination address</th>
<th>Source address</th>
<th>Type</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 bytes</td>
<td>6 bytes</td>
<td>2 bytes</td>
<td>46–1500 bytes</td>
</tr>
</tbody>
</table>

(b) Ethernet packet format.

(c) IP packet format.
(d) ICMP header format.

(e) TCP header format.
1. (60 pts) Short questions.

   (a) (5 pts) What are the advantages and disadvantages of having a layered network architecture?

   (b) (5 pts) List the name of at least one common error detection technique used to detect bit transmission errors in networking and describe how it works.
(c) (5 pts) When an IP packet travels from source to destination across multiple links, which fields in the packet will change? (Circle all that apply.)

A. Link layer source address  
B. Link layer destination address  
C. IP source address  
D. IP destination address

(d) (5 pts) What are the main differences between datagram switching and virtual circuit switching? Which of the switching technique does the Internet adopt?
(e) (5 pts) Why does the fragment offset field in the IP header measure in eight-byte units?

(f) (5 pts) How long is an IPv6 address? Why is an IPv6 address significantly longer than an IPv4 address?
(g) (5 pts) Why is it necessary to have one IP address per network interface rather than one per host?

(h) (5 pts) Why does IP forwarding use the longest prefix matching algorithm rather than the exact matching algorithm?
(i) (5 pts) Why are IP fragments reassembled at an endpoint rather than at a router?

(j) (5 pts) Why does the TCP state transition diagram (Figure 1) include a TIME_WAIT state, in which a TCP connection must wait for 2MSL (Maximum Segment Life) before it closes the connection?
(k) (5 pts) Ben BitDiddle learned that hierarchical DNS queries are a main source of latency for web applications. What problems may arise if we shorten the DNS hierarchy and remove all the top-level hierarchy, e.g., www.google.com becomes www.google?

(l) (5 pts) Is BitTorrent a suitable protocol for online video streaming? Explain your answer.
2. (10 pts) A non-recursive DNS query is sent to find out the IP address of the DNS name www.cnn.com and here is the response.

; <<>> DiG 9.7.0-P1 <<>> +norecurss www.cnn.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48312
;; flags: qr ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 0

;; QUESTION SECTION:
;www.cnn.com. IN A

;; ANSWER SECTION:

;; AUTHORITY SECTION:
net. 4984 IN NS l.gtld-servers.net.
net. 4984 IN NS f.gtld-servers.net.
net. 4984 IN NS i.gtld-servers.net.
net. 4984 IN NS m.gtld-servers.net.
net. 4984 IN NS c.gtld-servers.net.
net. 4984 IN NS b.gtld-servers.net.
net. 4984 IN NS a.gtld-servers.net.
net. 4984 IN NS e.gtld-servers.net.
net. 4984 IN NS h.gtld-servers.net.
net. 4984 IN NS j.gtld-servers.net.
net. 4984 IN NS d.gtld-servers.net.
net. 4984 IN NS k.gtld-servers.net.
net. 4984 IN NS g.gtld-servers.net.

;; Query time: 0 msec
;; SERVER: 152.3.140.1#53(152.3.140.1)
;; WHEN: Wed Apr 30 16:16:05 2014
;; MSG SIZE rcvd: 284

(a) (5 pts) After a recursive resolver receives the response, what is the next query the resolver should send in order to find out the IP address(es) of www.cnn.com?
(b) (5 pts) From the DNS response, Ben BitDiddle claimed that CNN was likely to use a CDN service. Do you agree with him? Explain why.
3. (10 pts) Figure 2 shows a simple model of how a TCP sender adjusts its congestion window size (cwnd) to avoid congestion. Use this figure to derive the TCP throughput equation

\[
\frac{\sqrt{1.5 \cdot MSS}}{RTT \cdot \sqrt{p}}
\]

, where \(MSS\) is the maximum segment size, \(RTT\) is the TCP connection’s round trip time, and \(p\) is the loss rate.
4. (10 pts) Figure 3 shows an AS-level network topology. As we learned in class, ASes run BGP to interconnect. In this topology, AS \( D \) originates a network prefix 12.0.0.0/8. All ASes use common BGP policies to select their next hops to reach the prefix 12.0.0.0/8. Answer the following questions.

(a) (2 pts) What’s AS \( E \)’s next hop to reach the network 12.0.0.0/8?

(b) (2 pts) What’s AS \( G \)’s next hop to reach the network 12.0.0.0/8?

(c) (2 pts) An Attacker AS \( I \) wishes to hijack AS \( D \)’s traffic. To do so, AS \( I \) also announces the network prefix 12.0.0.0/8 in BGP. When AS \( G \) receives this BGP update from AS \( I \), will it change its next hop to reach 12.0.0.0/8? Explain why.

(d) (2 pts) After AS \( I \) announces 12.0.0.0/8, which ASes may change their next hops to reach the prefix 12.0.0.0/8?
(e) (2 pts) Ben BitDiddle is the network admin of AS $D$. After AS $I$ launched the attack, he received many customer calls complaining they could not reach AS $D$'s web services. Ben asked his friend, Alissa Hacker, an alumna of CompSci356, what to do. Alissa told Ben that his network’s prefix was hijacked. To regain the control of the hijacked traffic, Ben could modify the network prefix(es) AS $D$ announced in BGP. What network prefix(es) should AS $D$ announce to get its traffic back? Please specify the prefix(es) precisely. In fact, Alissa’s fix was what YouTube used when its traffic was hijacked back in 2008.
5. (10 pts) Ben BitDiddle wrote the following code to check the validity of a received packet for Lab 1, reliable transport. His TA pointed out that this piece of code contained serious bugs. Please help Ben find the bugs in his code, and correct the code inline.

```c
struct packet {
    uint16_t cksum;
    uint16_t len;
    uint32_t ackno;
    uint32_t seqno; /* Only valid if length > 8 */
    char data[500];
};
typedef struct packet packet_t;

void
rel_recvpkt (rel_t *r, packet_t *pkt, size_t n)
{
    if ((n != 8 && n < 12) || n < pkt->len) {
        fprintf(stderr, "%d [rel_recvpkt]: wrong length packet received\n", getpid());
        return;
    }
    if (cksum(pkt, n) != pkt->cksum) {
        fprintf(stderr, "%d [rel_recvpkt]: corrupted packet received\n", getpid());
        return;
    }
    packet_t * new_pkt=malloc(n);
    memset(new_pkt, 0, n);
    memcpy(new_pkt, pkt, n);

    // The rest of the code handles sending ack, storing packets at
    // the receiver buffer and output packets data. It is omitted but you can
    // assume it is implemented correctly.

    ...
}
```
This page is intentionally left blank.