NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2012-2013

CSC302 – NET-CENTRIC COMPUTING

Nov/Dec 2012 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 6 questions and comprises 8 pages (including Appendix)
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. Questions DO NOT carry equal marks.

---

1. (a) Explain how competing computers access the shared channel in CSMA/CD networks, under what conditions a collision may occur, and how long it takes for a transmitting computer to be sure it has seized the channel without any collision.

(8 marks)

(b) Explain what are flow control and congestion control, respectively. Furthermore, describe a situation where flow control is needed but congestion control is not, and another situation where congestion control is needed but flow control is not.

(7 marks)

2. Suppose machine A is communicating with machine B using the sliding window protocol in Assignment 1 (attached in the **Appendix** on pages 7-8). Answer following questions:

(a) Suppose B has received frames 0, 1, 2, 3, 4, and 6 from A. Which frame(s) can B deliver to the network layer? Which frame(s) can B save in the buffer? Which sequence number can B (positively) acknowledge A? What sequence numbers are within the receiving window of B?

(5 marks)

Note: Question No.2 continues on Page 2

(b) Under what circumstances will machine A receive a negative acknowledgement from machine B? When machine A receives a negative acknowledgement from machine B, under what conditions will machine A retransmit a frame? Explain why such conditions are required using a concrete scenario.

(5 marks)

3. Consider the subnet in Figure Q3. Distance vector routing is used, and the following vectors have just come into router C:

- from B: (6, 0, 8, 12, 6, 2);
- from D: (12, 14, 6, 0, 9, 10);
- from E: (2, 6, 3, 9, 0, 4).

The measured delays to B, D and E, are 8, 6, and 10, respectively. What is C's new routing table? Give both the outgoing line to use and the expected delay in your answer.

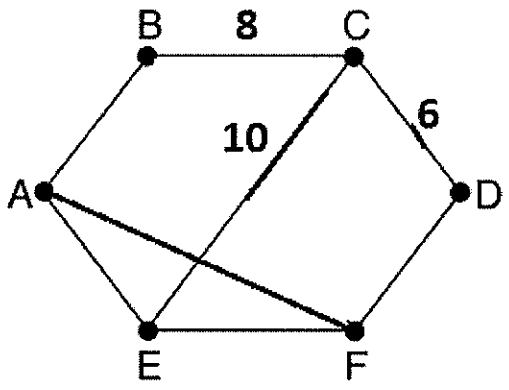

Figure Q3

(15 marks)

4. Consider a TCP machine sending windows of 65,535 bytes over a 1 Gbps channel. What line efficiency can be achieved if the round-trip time of this channel is 200ms and 500ms, respectively? Explain your answers.

(10 marks)

5. Figure Q5a shows the browser display of an HTML document. Figure Q5b gives the source code of the HTML document and embedded JavaScript. The HTML document contains a form for user registration. A "First/Given name" text box collects the user's given name. A "Last/Family name" text box collects the user's family name. There is a "Gender" radio group that collects the user's gender. There are two options: "Male" and "Female". The "Create 6-digit PIN" text box collects the PIN number from the user. The "Confirm 6-digit PIN" text box requires the reinput of the same number and confirms the PIN.

Complete the source file in Figure Q5b by filling in the boxes **A** to **L**, each with a single statement or an expression to implement the following functionalities:

First, the "Create 6-digit PIN" must be a six digit number. If the input is not valid, an alert message is generated and the input in the "Create 6-digit PIN" text box is focused and selected.

Second, the text of "Confirm 6-digit PIN" must be the same as that of "Create 6-digit PIN", and if not, an alert message is generated and the input in the "Confirm 6-digit PIN" text box is focused and selected. Whenever there is a change in the "Create 6-digit PIN", we need to ensure that the new PIN input should be confirmed.

Third, encryption is applied to guarantee the security in delivering the PIN to the server. When the "Submit" button is clicked, the 6-digit PIN number is "masked" (bitwise XORed) with a random number (201212 in this question) before it is sent out.

(24 marks)

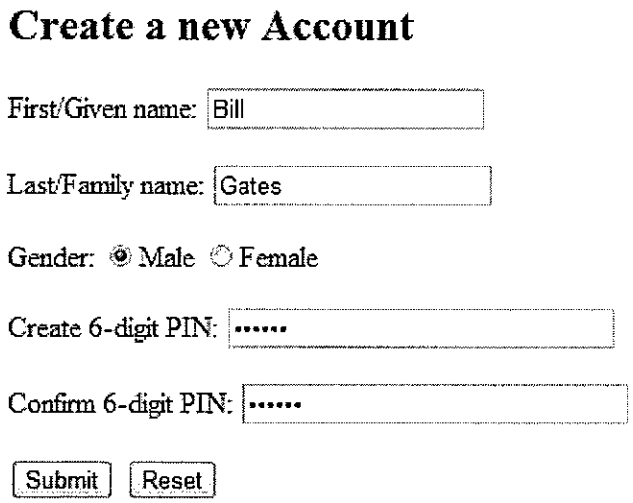

Figure Q5a

Note: Question No. 5 continues on Page 4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<title>Registration_Form</title>
<script type="text/javascript">
<!--
function checkPIN() {
        var userPIN = document.getElementById("pin");
        var pos = userPIN.value.search(      A      );
        if(      B      ) {
                alert("PIN should be six digit long. \n");
                document.getElementById("pin").value = "";
                      C      ;
                userPIN.select();
        }
              D      ;
}

function checkConfirmPIN() {
        var userPIN = document.getElementById("pin");
        var userConfirmPIN = document.getElementById("confirmpin");
        if(      E      ) {
                alert("Should match the previously entered PIN. \n");
                document.getElementById("confirmpin").value = "";
                      F      ;
                userConfirmPIN.select();
        }
}

function encryptPIN() {
        var userPIN =      G      ;
        var randomNo = 201212;
        var newPIN =      H      ^ randomNo; // ^ is bitwise XOR operator
        document.getElementById("pin").value = newPIN;
        document.getElementById("confirmpin").value = newPIN;
}
//-->
</script>
</head>
```

Note: Figure Q5b continues on Page 5

```
<body>
<form action = "Handle_Registration_Form.php" [ I ]>
<h2>Create a new Account</h2>
<p>First/Given name: <input type="text" name="givenName" id="givenName"
size="20"/></p>
<p>Last/Family name: <input type="text" name="familyName" id="familyName"
size="20"/></p>
<p>Gender: <input type="radio" name="gender" value="M" checked="checked"/>Male
<input type="radio" name="gender" value="F"/>Female</p>
<p>Create 6-digit PIN: <input type="password" name="pin" id="pin" value=""
size="30" [ J ] maxlength="6"/></p>
<p>Confirm 6-digit PIN: <input type="password" name="confirmpin" id="confirmpin"
value="" size="30" [ K ] maxlength="6"/></p>
<p><input type="submit" id="submit" value="Submit" [ L ]/>
<input type="reset" id="reset" value="Reset" /></p>
</form>
</body>
</html>
```

Figure Q5b

6. Following Question Q5, the "Handle_Registration_Form.php" program in the server side accepts form submissions from client users and updates a file named "user_information.txt" stored on the server. For each form, the file maintains an entry recording the givenName, familyName, gender and PIN, as shown in Figure Q6a. In particular, the PIN number is decrypted in the following steps:

(1) The encrypted PIN number is converted into a binary array of length 20;
(2) The binary array is "masked" with the pseudo-random number (the same number as used in clients but in the binary array format) to recover the binary PIN;
(3) The binary PIN is converted to the original decimal PIN number.

Figure Q6b gives the server PHP program. Complete the source file by filling in the boxes **A** to **M**, each with a single statement or expression to implement the above mentioned functionalities.

(26 marks)

```
givenName: Lily  familyName: Lee  gender: F  pin: 201212
givenName: Jim  familyName: Liu  gender: M  pin: 753951
givenName: Bill  familyName: Gates  gender: M  pin: 123456
```

Figure Q6a

Note: Question No. 6 continues on Page 6

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<title>Handle_Registration_Form</title>
</head>
<?php
        $givenName= $_POST[      A      ];
        $familyName= $_POST[     B      ];
        $gender= $_POST[      C      ];
        $pin= $_POST[      D      ];
        $randomNo = "00110001000111111100"; //binary representation of 201212
        $binaryEncryptPIN = array();
        $binaryPIN = array();
        $index = 0;
        while($pin > 0) {
                $remainder = $pin % 2;
                $pin =      E      ;
                $binaryEncryptPIN [$index] = $remainder;
                      F      ;
        }
        while($index < 20) {
                $binaryEncryptPIN[$index] = 0;
                $index++;
        }
        for($index = 0; $index < 20; $index++) {
           $binaryPIN[$index] = $randomNo[$index] ^ $binaryEncryptPIN[      G      ];
        }
        H
        $exp = 1;
        for($index = 0; $index < 20; $index++) {
                $originalPIN = $originalPIN + $exp * $binaryPIN[      I      ];
                      J      ;
        }
        $output = "givenName: " . $givenName . "  familyName: " . $familyName . "
                                gender: " . $gender . "  pin: " .      K      ."\n";
        $file=fopen(      L      );
        fwrite(      M      );
        fclose($file);
        ?>
<body>
</body>
</html>
```

Figure Q6b

Appendix: A Sliding-Window Protocol (for Question No. 2)

```
#define MAX_SEQ 7
#define NR_BUFS ((MAX_SEQ + 1)/2)
typedef enum {frame_arrival, cksum_err, timeout, network_layer_ready, ack_timeout}
event_type;
#include "protocol.h"
boolean no_nak = true;
seq_nr oldest_frame = MAX_SEQ + 1;

static boolean between(seq_nr a, seq_nr b, seq_nr c)
{   return ((a <= b) && (b < c)) || ((c < a) && (a <= b)) ||  ((b < c) && (c < a)); }

static void send_frame(frame_kind fk, seq_nr frame_nr, seq_nr frame_expected, packet buffer[])
{
        frame s;  s.kind = fk;
        if (fk == data) s.info = buffer[frame_nr % NR_BUFS];
        s.seq = frame_nr;
        s.ack = (frame_expected + MAX_SEQ) % (MAX_SEQ + 1);
        if (fk == nak) no_nak = false;
        to_physical_layer(&s);
        if (fk == data) start_timer(frame_nr % NR_BUFS);
        stop_ack_timer();
}

void Sliding-Window-Protocol(void)
{
        seq_nr ack_expected, next_frame_to_send, frame_expected, too_far;
        int i;
        frame r;
        packet out_buf[NR_BUFS];
        packet in_buf[NR_BUFS];
        boolean arrived[NR_BUFS];
        event_type event;

        enable_network_layer(?);
        ack_expected = 0;
        next_frame_to_send = 0;
        frame_expected = 0;
        too_far = NR_BUFS;

        for (i = 0; i < NR_BUFS; i++) arrived[i] = false;
```

Note: Appendix continues on Page 8

```
while (true) {
  wait_for_event(&event);
  switch(event) {
    case network_layer_ready:
       from_network_layer(&out_buf[next_frame_to_send % NR_BUFS]);
       send_frame(data, next_frame_to_send, frame_expected, out_buf);
       inc(next_frame_to_send);
       break;
     case frame_arrival:
       from_physical_layer(&r);
       if (r.kind == data) {
          if ((r.seq != frame_expected) && no_nak)
             send_frame(nak, 0, frame_expected, out_buf); else start_ack_timer();
           if (between(frame_expected, r.seq, too_far) &&
                        (arrived[r.seq % NR_BUFS]==false)) {
              arrived[r.seq % NR_BUFS] = true;
              in_buf[r.seq % NR_BUFS] = r.info;
              while (arrived[frame_expected % NR_BUFS]) {
                 to_network_layer(&in_buf[frame_expected % NR_BUFS]);
                 no_nak = true;
                 arrived[frame_expected % NR_BUFS] = false;
                 inc(frame_expected);
                 inc(too_far);
                 start_ack_timer();
              } } }
        if((r.kind==nak) && between(ack_expected, (r.ack+1) % (MAX_SEQ+1),
                                    next_frame_to_send))
             send_frame(data, (r.ack+1) % (MAX_SEQ+1), frame_expected, out_buf);
         while (between(ack_expected, r.ack, next_frame_to_send)) {
              enable_network_layer(?);
              stop_timer(ack_expected % NR_BUFS);
              inc(ack_expected);
         }
        break;
   case cksum_err:
        if (no_nak) send_frame(nak, 0, frame_expected, out_buf);
        break;
   case timeout:
        send_frame(data, oldest_frame, frame_expected, out_buf);
        break;
   case ack_timeout:
       send_frame(ack, 0, frame_expected, out_buf);
   } } }
```

END OF PAPER

CSC302 NET CENTRIC COMPUTING

Please read the following instructions carefully:

1. **Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**

2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.

3. Please write your Matriculation Number on the front of the answer book.

4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.