



University
of Glasgow

Tuesday, 7 May 2019
09:30 am – 11:00 am
(1 hour 30 minutes)

DEGREES of MSc, MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

NETWORKED SYSTEMS (H)

Answer all 3 questions

This examination paper is worth a total of 60 marks.

The use of calculators is not permitted in this examination.

INSTRUCTIONS TO INVIGILATORS: Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

1. (a) Congestion control protocols use a sliding window algorithm to control the number of packets they can send. With reference to the window size, the receipt of acknowledgements, and when new packets can be sent, briefly explain the operation of a sliding window algorithm. [4]
- (b) In a sliding window protocol, how large should the window be to ensure full utilisation of a network path? Justify your answer. [3]
- (c) TCP is a sliding window protocol. It uses packet loss as a signal that congestion is occurring and that it should reduce its window. Outline two possible causes of packet loss in the Internet. For each of these two causes of packet loss, discuss whether TCP is right to use it as a congestion signal. [6]
- (d) TCP has two ways of detecting packet loss. This is signalled by either the receipt of three identical duplicate acknowledgements, or by a timeout when no acknowledgement is received. Explain why TCP needs two different ways of detecting packet loss. [2]
- (e) With the aid of a diagram, explain why TCP waits for the receipt of three identical duplicate acknowledgements before deciding that a packet was lost, rather than accepting a single duplicate acknowledgement as an indication of packet loss. [5]
2. (a) Home networks often connect to the wider Internet using a network address translation (NAT) router, rather than using a standard IPv4 router. Explain how addressing and routing for a sub-network were intended to work in the Internet architecture, and how they work in a NAT-based network. [10]
- (b) One of the key roles of the network layer is inter-networking between different link layer technologies, to allow local area networks to be combined to form a seamless wide area network. Each link layer technology may have a different maximum transmission unit (MTU), and the network layer is responsible for handling this mismatch. Describe how IPv4 and IPv6 handle inter-networking between links with different MTUs. Discuss why this behaviour was changed when IPv6 was designed. [10]
3. (a) The second laboratory exercise for the course considered the development of a web server that implements a subset of the HTTP/1.1 protocol. The code provided as part of the exercise includes a function, `read_headers()`, that is used to read the HTTP/1.1 headers from the client socket. This function is written as follows:

```
static char *
read_headers(int fd)
{
    char    buf[BUFLen];
    char    *headers    = malloc(1);
    size_t  headerLen = 0;
    ssize_t  rlen;

    headers[0] = '\0';
    while (strstr(headers, "\r\n\r\n") == NULL) {
        rlen = recv(fd, buf, BUFLen, 0);
        if (rlen == 0) {
            // Connection closed by client
```

```

        free(headers);
        return NULL;
    } else if (rlen < 0) {
        free(headers);
        perror("Cannot read HTTP request");
        return NULL;
    } else {
        headerLen += (size_t) rlen;
        headers = realloc(headers, headerLen + 1);
        strncat(headers, buf, (size_t) rlen);
    }

    if (shutdown_requested) {
        printf("shutdown requested\n");
        return NULL;
    }
}

return headers;
}

```

The parameter `fd` is the file descriptor of the socket representing the connection to the client. The global variable `shutdown_requested` indicates if the connection should be closed after this request is complete.

Discuss why the code is written in this manner, using a `while()` loop that wraps several calls to `recv()` and `realloc()` in a loop, rather than being just a single call to `recv()`. Consider both the nature of HTTP/1.1 GET requests and the need to allocate memory, and the interactions with the underlying TCP transport protocol in your answer. Explain how the function builds up complete HTTP/1.1 headers in memory. [10]

- (b) The sample web server, along with many other networked applications and services, is written in the C programming language. This language does not enforce memory safety, and indeed the first part of the exercise was to correct a buffer overflow bug in the code that prevented the server from correctly serving image files.

Mistakes in applications written in the C programming language can lead to exploitable buffer overflows, and serious security vulnerabilities. Given these potential problems, discuss why networked applications are written this way, and whether you think it is justifiable to write new networked code in memory unsafe languages. [10]