Reliable Transport Protocol

Benjamin Lipp

Academic Integrity:

I have read and understood the course academic integrity policy.

Alternating Bit Protocol:

This protocol uses a fist in first out buffer. When a sends a message is keeps sending the same message until it receives confirmation from b that it received the message. Then it flips the seqnums and sends the next message

Defined variables:

int a_seqnum;

Keeps track of the a seqnum sent

int b_seqnum;

Keeps track of the b seqnum received

int buffer_head;

First message in the buffer

struct msg buffer[BUFFERSIZE];

List of messages to be sent

struct msg empty_msg;

msg struct with an empty data

struct pkt a_packet;

Last packet to be sent

New functions:

int make_checksum

Takes two ints and a string (char*) and returns a int

Calculates the checksum for a packet

struct pkt make_packet

Takes two ints and a msg returns a pkt

Creates a packet from the data given

int is_valid

Takes a pkt and returns an int

Checks the pkt’s current checksum and compares it to the generated checksum from make_checksum

int flip_seq

takes a int and returns and int

if the seqnum is 0 returns 1
if the seqnum is 1 returns 0

void buffer_add

takes a msg

addes a msg to the end of the buffer

struct msg pop_buffer

returns a msg

Removes the first message in the buffer and returns it

struct msg* view_buffer

Returns a pointer to a msg

Shows the first message in the buffer

void send_message

Takes msg

Creates a packet with make_packet
Assigns that packet to a_packet
Starts the timer and sends the packet to layer3

Routines

void A_output

Takes msg

If the buffer is empty the message is sent
Else the message is added to the buffer to be dealt with later

void A_input

Takes pkt

Checks the validity of the pakcet with is_valid
If it is valid, the a_seqnum is flipped, the message is removed from the buffer and the Timer is stopped.

void A_timerinterrupt

Sends the current packet (a_packet) and starts the timer

void A_init

a_seqnum = 0;
buffer_head = 0;

void B_input

Takes pkt

Checks if the packet is valid with is_valid
If it is the b_seqnum is flipped and the payload from the packet is sent to layer 5
Then and empty message is sent to layer 3

void B_init

b_seqnum =0;

Go back N

This protocol uses a sliding window of a certain size (window_size). It transmits every message in the window without confirmation until it fills the window. To move on, the receiver needs to send confirmation. The window will then shift for each confirmed message

Defined variables

int window_size;

A int to keep track of the size of the window

int base;

The first place of the window

int next_seqnum;

The next (+1) seqnum

int expected_seqnum;

The expected current seqnum

char empty[20] = "";

Empty message data

float interrupttime;

Timer value

std::vector<pkt> buffer;

Buffer vector

New functions

int make_checksum

Takes pkt returns an int

Calculates the checksum for a given packet

struct pkt make_packet

Takes two ints and a msg returns a pkt

Creates a packet with the given data

Routines

void A_output(struct msg message)

Takes a msg

Checks if the seqnum fits in the windows size
Creates a packet with the messages as the payload
If it does the packet is sent to layer 3
If is does not the packet is added to the buffer

void A_input(struct pkt packet)

Takes a pkt

Checks if the checksum is valid by comparing the current checksum with the generated checksum
If it is the timer is stopped
Else, the timer is started again

void A_timerinterrupt()

Starts the timers and then sends the window to layer 3

void A_init()

```
next_seqnum = 1;
base = 1;
window_size = getwinsize();
interrupttime = 20.0;
```

void B_input(struct pkt packet)

Takes a pkt

Checks if packet is valid
If so, sends the window to layer 5
Then creates a new packet with an empty message and sends that to layer 3
If not valid it creates a new packet with an empty message and sends to layer 3

B_init

```
expected_seqnum = 1;
```