CO633 Computer Networks 2019-20 : Programming Assignments #1 Sender and #2 Receiver

Frame Format

Specification

The data link layer splits long messages into a number of segments that are small enough for the physical layer to handle. Each message segment is packaged as a separate frame comprising of several fields, as follows:

Field	Frame delimiter (start)	Frame type	Field delimiter	Segment length	Field delimiter	Message segment	Field delimiter	Checksum	Frame delimiter (end)
Example	`<`	`E`	`-`	`05`	`-`	`hello`	`-`	`37`	`>`

Field	**Length (chars)**	**Description**
Frame delimiter	1	A single '`<`' marks the start of the frame
Frame type	1	'`D`' = data frame, but not final segment of message '`E`' = data frame containing final segment of message
Field delimiter	1	A single '`-`' separates adjacent fields within frame
Segment length	2	The number of characters in the message segment represented as a 2-digit decimal number
Field delimiter	1	A single '`-`' separates adjacent fields within frame
Message segment	0-99	Next segment of message string (can be empty)
Field delimiter	1	A single '`-`' separates adjacent fields within frame
Checksum	2	Modular checksum represented as 2-digit decimal number
Frame delimiter	1	A single '`>`' marks the end of the frame

Notes

- This frame format is designed to be read easily by humans, hence it differs in some ways from a real network protocol.
- The segment length and checksum fields always contain two digits.
  Values less than 10 have a leading zero (e.g. `07`).
- A segment length of zero (`00`) means there's no message text and the message segment field is empty (i.e. there's nothing between the two field delimiters, which must still be present).
- The segment length must not exceed 99, irrespective of the MTU (see below).
- The message segment can contain any sequence of characters supported by the Java String class, including those matching the delimiters ('`<`', '`>`', '`-`') and unprintable control codes.
- Character escaping/stuffing is **not** required for this protocol.
- The checksum is calculated from the arithmetic sum of all preceding characters in the frame except for the starting frame delimiter (i.e. from the frame type through to the final field delimiter, inclusive). Only the last two digits are recorded (e.g. if the arithmetic sum is 1234, the checksum is `34`).

- The total length of the frame including all delimiters must not exceed the MTU (*maximum transfer unit*, see class documentation).
- Any deviation from the specified format should be treated as an error. This includes frames which exceed the MTU.

Examples

Examples of correctly formatted frames are presented below to help you verify your implementation of MessageSender. Breakdowns of the working are also shown in case you had difficulty understanding the earlier specification. Example message and frame text is shown indented and using a courier font for clarity.

Short message

Suppose the following message is to be sent:

```
hello
```

Assuming the MTU is large enough for the whole message to be accommodated in a single frame, it would look like this:

```
<E-05-hello-37>
```

The working breaks down as follows:

1. `<` marks the start of the frame.
2. `E` means that this frame is the last frame for the message being sent.
3. `-` separates adjacent fields.
4. `05` is the length of the message segment.
5. `-` separates adjacent fields.
6. `hello` is the message segment itself.
7. `-` separates adjacent fields.
8. `37` is the value of the checksum.
   The arithmetic sum of the characters '`E`', '`-`', '`0`', '`5`', '`-`', '`h`', '`e`', '`l`', '`l`', '`o`', '`-`' is equal to 837.
   Only the final two digits, `37`, are recorded.
9. `>` marks the end of the frame.

Longer message

Suppose the following message is to be sent:

```
Who Framed Roger Rabbit
```

Suppose also that the physical layer MTU is 20. The message would need to be split into three separate frames, like this:

```
<D-10-Who Framed-25>
<D-10- Roger Rab-52>
<E-03-bit-22>
```

- The first two frames start with `<D` rather than `<E` because each is followed by another frame.
- The message segments are less than 20 characters long. This is because the MTU value applies to the whole frame.
- The final frame is shorter than the others because the final remaining message segment doesn't require a frame of the maximum allowed length.

Further examples

Additional examples can be found in the Testing appendix.