Network Protocol Design and Evaluation

Exercise 3

Stefan Rührup

University of Freiburg
Computer Networks and Telematics
Summer 2009
Exercise 3

Task 1  *SDL and Message Sequence Charts*

A server responds to request messages ("Req") and sends Ping messages in regular intervals of 4 sec according to the specification given below. Assume that requests arrive at time 1, 3, 5, 6, 7, 15 and that processing a request takes 2 sec. Draw a MSC of the server for time 0-16. Consider the alternative timer setting `expirytime := NOW + period;` and draw a MSC for this variant.
DCL expirytime Time := 0.0;
DCL period Duration := 4;
TIMER T;

expirytime := expirytime + period;
SET (expirytime, T)

IDLE

process(Req)  T

IDLE

expirytime := expirytime + period;
SET (expirytime, T)

Ping
A note on timers

- Timer events are handled like incoming messages

- “When an inactive timer is set, a Time value is associated with the timer. Provided there is no reset or other setting of this timer before the system time reaches this Time value, a signal with the same name as the timer is put in the input port of the agent. The same action is taken if the timer is set to a Time value less than or equal to now” (from the SDL Specification)
Task 1

Wrong!

Processing time of the Req messages is not considered.
Task 1

The server is busy for 2 sec each for processing Req messages.
Task 1

Alternative Variant
expirytime := NOW + period;
Exercise 3

Task 2  Alternating Bit Protocol

Consider the Alternating Bit Protocol for an unreliable channel. Lost messages are retransmitted after a timeout. What happens if we allow that messages arrive out of order?

- Describe a scenario, where out-of-order arrival leads to a failure of the protocol. What kind of failures may happen?

- Try to fix the protocol and give a formal specification of your new protocol.
Alternating Bit Protocol, Sender

process sender

get data

SEND1

Msg1

set timer

Ack1

get data

SEND0

Ack0

SEND1

T

Ack0

get data

SEND0

Ack1

SEND1

T
Alternating Bit Protocol, Receiver

process receiver

WAIT1

Msg1
accept
Ack1
WAIT0

Msg0
Ack0
WAIT1

WAIT0

Msg0
accept
Ack0
WAIT1

Msg1
Ack1
WAIT0
Error Scenario

 MSC abp_failure

 Sender

 Msg1

 Ack1

 Receiver

 accept Msg1

 accept Msg0

 accept duplicate Msg1

 Msg1 remains unacknowledged
Sequence numbers are added to the messages. The sender repeats transmitting a message until it is acknowledged with a matching sequence number.

**process sender**

```
DCL c Integer;

C := 1;
get data
SEND
Msg(c)
set timer
```

- `c' := 0;
- `c := c + 1;
- `c := c';

```
[ c' = c ]
T
```

```
[ c' = c + 1 ]
```

**process receiver**

```
DCL c' Integer;

C' := 0;
WAIT
Msg(c)
```

```
[ c = c' + 1 ]
ELSE
accept
```

```
c' := c;
Ack(c')
```

```
Ack(c')
```

```
-
Out-of-order scenario

**msc** abp+

**Sender**
- Send **Msg(1)**
- Send **Msg(2)**
- Send **Msg(3)**

**Receiver**
- Accept **Msg(1)**
- Accept **Msg(2)**
- Ignore **Msg(1)**, as **c** != **c' + 1**

Wrong ack, retransmit **Msg(3)**

**Notes**
- **Ack(1)**
- **Ack(2)**
- **Ack(3)**
- **Wrong ack, retransmit Msg(3)**