Go Back-N : a bit of practice

Consider a go-back-n sender and a go-back receiver that are directly connected with a 10 Mbps link that has a propagation delay of 100 milliseconds. Assume that the retransmission timer is set to 3 seconds. If the window has a length of 4 segments, draw a time-sequence diagram showing the transmission of 10 segments (each segment contains 10000 bits):

- when there are no losses
- when the third and seventh segments are lost
- when the second, fourth, sixth, eighth, ... acknowledgements are lost
- when the third and fourth data segments are reordered (i.e. the fourth arrives before the third)

Go Back-N : a bit of practice

Consider the following situation. A go-back-n sender with Sequence number encoded as 2 bits has a sending windows of 4 segments.

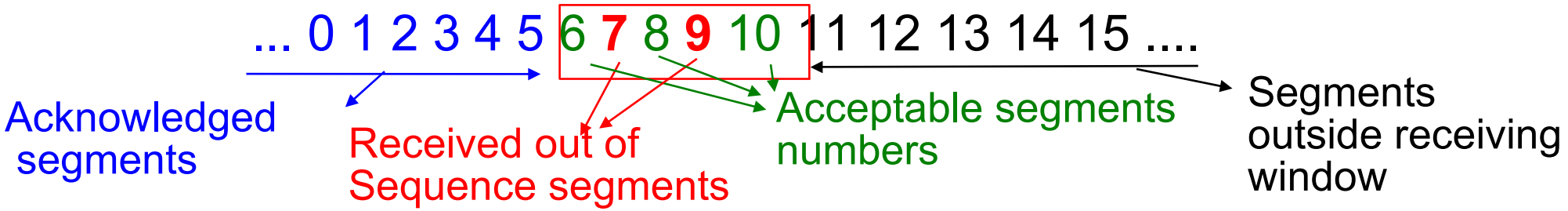
The sender sent a full window of data segments. All the Segments have been received correctly and in-order by the receiver, but all the returned acknowledgements have been lost.

Show by using a time sequence diagram (e.g. by considering a window of four segments) what problem can happens i this case.

Can you fix the problem on the go-back-n sender ?

Selective Repeat

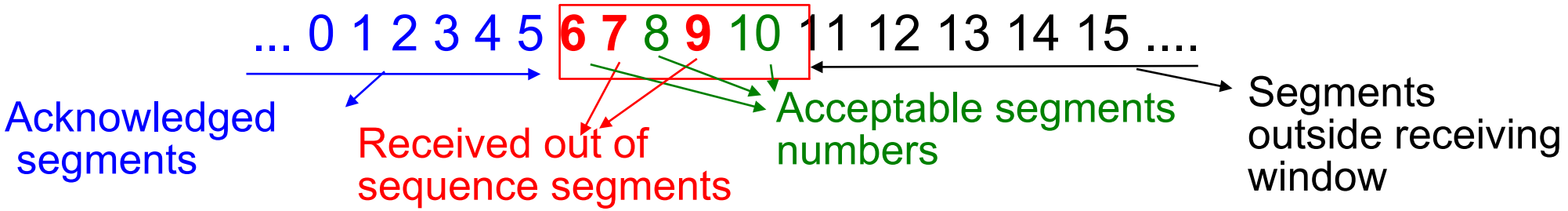
- Receiver
  - Uses a buffer to store the segments received out of sequence and reorder their content
  - Receiving window



- Semantics of the control segments
  - OKX
    - The segments up to and including sequence number X have been received
  - NAKX
    - The segment with sequence number X was errored
- Sender
  - Upon detection of an errored or lost segment, sender retransmits only this segment
  - may require one retransmission timer per segment

Selective Repeat

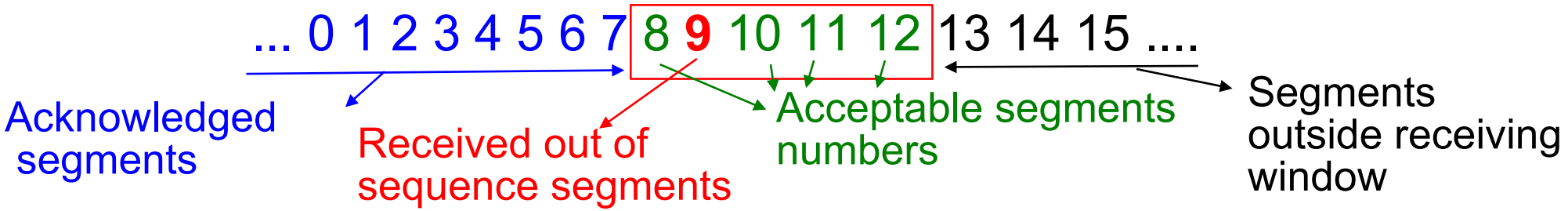
- Receiver
  - Uses a buffer to store the segments received out of sequence and reorder their content
  - Receiving window



- Semantics of the control segments
  - OKX
    - The segments up to and including sequence number X have been received
  - NAKX
    - The segment with sequence number X was errored
- Sender
  - Upon detection of an errored or lost segment, sender retransmits only this segment
  - may require one retransmission timer per segment

Selective Repeat

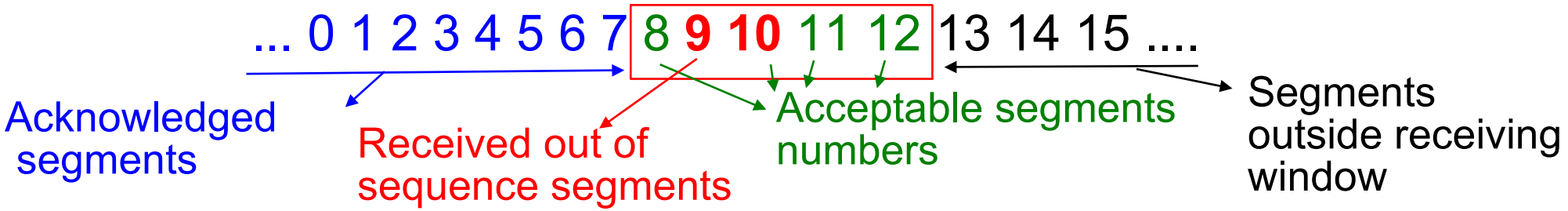
- Receiver
  - Uses a buffer to store the segments received out of sequence and reorder their content
  - Receiving window



- Semantics of the control segments
  - OKX
    - The segments up to and including sequence number X have been received
  - NAKX
    - The segment with sequence number X was errored
- Sender
  - Upon detection of an errored or lost segment, sender retransmits only this segment
  - may require one retransmission timer per segment

Selective Repeat

- Receiver
  - Uses a buffer to store the segments received out of sequence and reorder their content
  - Receiving window



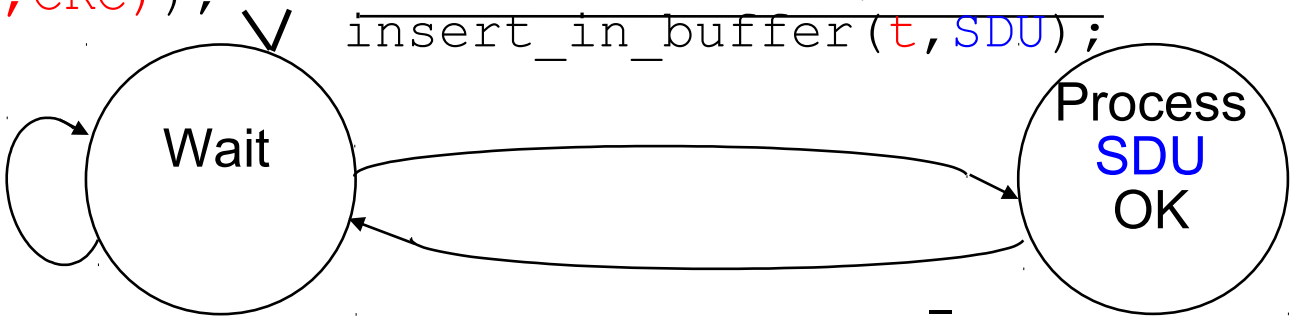
- Semantics of the control segments
  - OKX
    - The segments up to and including sequence number X have been received
  - NAKX
    - The segment with sequence number X was errored
- Sender
  - Upon detection of an errored or lost segment, sender retransmits only this segment
  - may require one retransmission timer per segment

Selective Repeat : Receiver

- State variable
  - `next` : sequence number of expected data segment
  - `last` : last received in-sequence segment

```
Recvd(D(t,SDU,CRC))
AND NOT(IsOK(CRC,SDU))
-----------------------
discard(SDU);
send(C(NAK,t,CRC));
```

```
Recvd(D(t,SDU,CRC))
 AND IsOK(CRC,SDU)
-----------------------
insert_in_buffer(t,SDU);
```



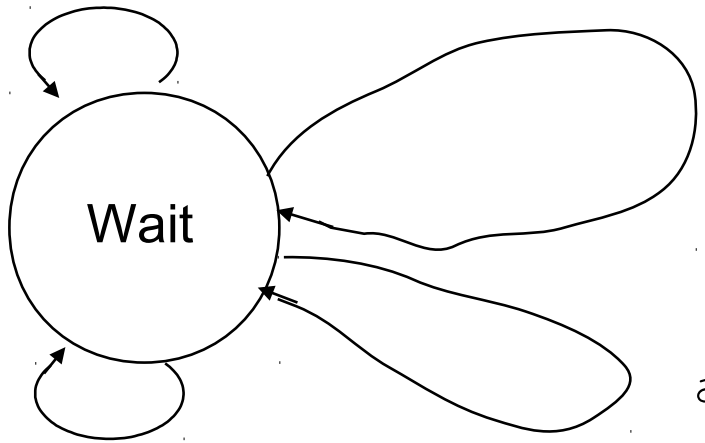
For all in sequence segments inside buffer

```
    Data.ind(SDU);
slide the sliding window;
update next and last
send(C(OK,(next-1)));
```

Selective Repeat : Sender

- State variables
  - `base` : sequence number of oldest unacknowledged segment
  - `seq` : first free sequence number
  - `w` : size of sending window

```
Recvd(C(?,?,CRC))
and NOT( CRCOK(C(?,?,CRC)))
```



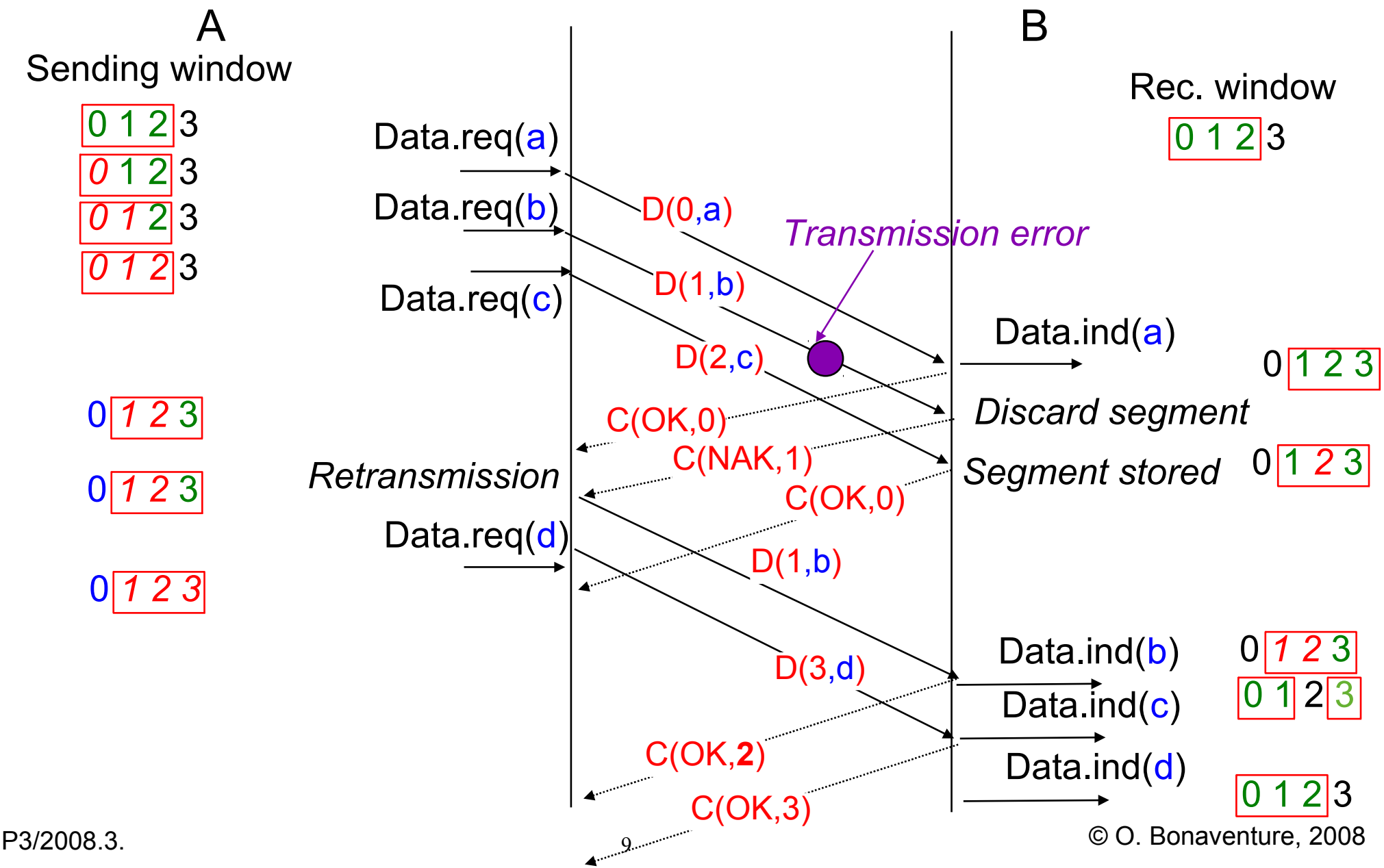

```
Recvd(C(OK,t,CRC))
and CRCOK(C(OK,t,CRC))
For all segments i<=t
   cancel_timer(t);
slide sliding window to
the right;
```

Data.req(SDU)

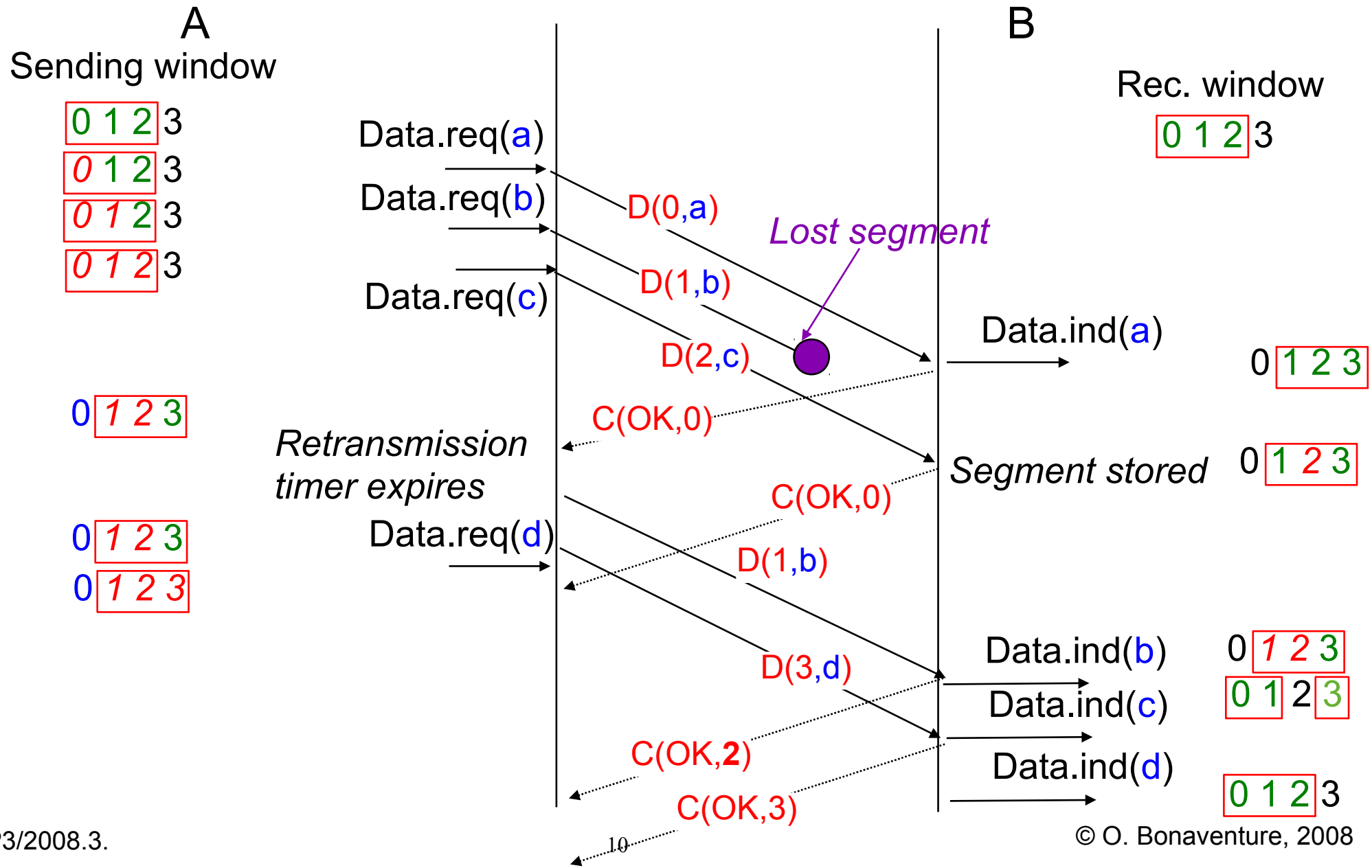
```
AND ( window not full )
start_timer(seq) ;
insert_in_buffer(SDU);
send(D(seq,SDU,CRC));
seq=(seq+1)mod C ;
```

```
[ Recvd(C(NAK,t,CRC))
and CRCOK(C(NAK,t,CRC))]
     or timer (t) expires
send(D(t,SDU,CRC)); }
restart_timer(t);
```

Selective Repeat : Example



Selective Repeat : Example (2)



Selective repeat : a bit of practice

1) Write in pseudo code of the `insert_in_buffer(t,SDU)` and `slide the sliding windows` functions for the selective repeat receiver state machine, using the variable representing the sliding windows defined earlier.

2) Same exercice as on slide 19 in previous chapter, but with a selective repeat sender and receiver.

3) Is it possible for a go-back-N receiver to interoperate with a selective repeat sender ?
Justify your answer.

4) Is it possible for a go-back-N sender to interoperate with a selective repeat receiver ?
Justify your answer.