

# Protocol with Timer

## **Abstract**

This example shows how the CP-net from “Simple Protocol” can be modified so that the retransmission of messages is controlled by a simple timer – modelled by means of two places and two transitions. The CP-net is not using time stamps and hence it should not be confused with the “Timed Protocol”.

## **Developed and Maintained by:**

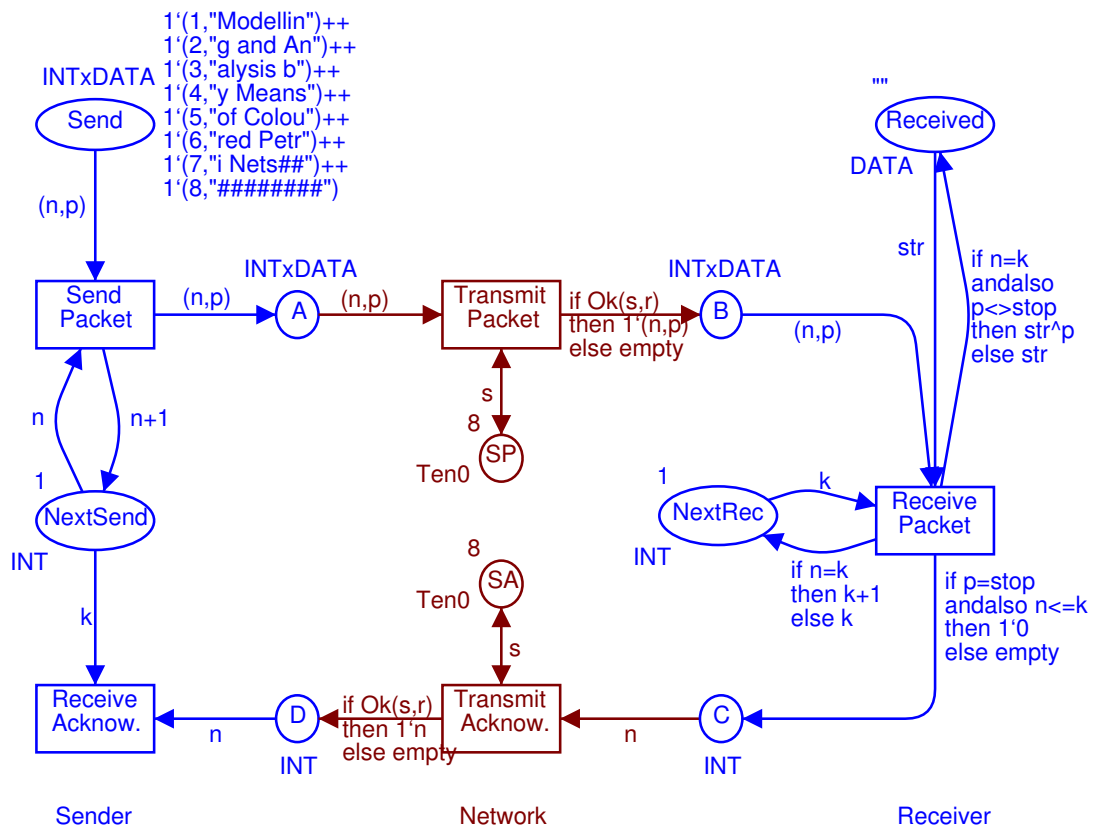
Kurt Jensen, Aarhus University, Denmark ([kjensen@daimi.au.dk](mailto:kjensen@daimi.au.dk)).

## CPN Model

The protocol part of the CP-net is shown below. It is very similar to the CP-net for the “Simple Protocol”. However, there are two differences. The protocol is optimistic, in the sense that it assumes that each packet will be successfully received and hence immediately continues to send the succeeding packet. This can be seen from the arc expression at the arc from *Send Packet* to *Next Send*. Moreover, the receiver now only sends an acknowledgment when it receives a packet containing "#####" (and already has received all the preceding packets). This can be seen from the arc expression at the arc from *Receive Packet* to *C*. By convention the acknowledgment contains the value 0. When the acknowledgment reaches the sender, the token at *Next Send* is removed and hence the transmission stops.

It is obvious that this simple scheme does not work alone. We also need to include a mechanism to start a retransmission – in the case where one or more packets get lost (or overtake each other). We achieve this by means of the simple timer shown on the next page. When transition *Send Packet* sends the

### Protocol Part



last packet (the one with "#####") it adds a token with value 0 to place *Count* (in the timer). This enables transition *Clock Tick*. Each occurrence of this transition increases the count (i.e., the value of the token at *Count*). When the count reaches a pre-set limit (determined by the token at *Limit*) transition *Clock Tick* ceases to be enabled. Instead transition *Alarm* becomes enabled. When it occurs it stops the timer and resets the value of the token at *Next Send* to 1, i.e., starts a retransmission of the entire set of packets. The two transitions in the timer may occur concurrently to the transitions in the protocol part (i.e., the other five transitions). A retransmission will occur if and only if *Alarm* occurs before *Receive Acknowledgment*. This will happen if the protocol part is "too slow" or if one or more packets get lost or overtake each other. By adjusting the timer limit it can be made more or less likely that the timer raises an alarm before the protocol part has finished.

### Simple Timer

