Formale Simulation und Verifikation verteilter Algorithmen Abgabe der Praktikumsaufgabe 1

Maria Lüdemann und Birger Kamp

March 31, 2016

1 Das Modell als Petrinetz

Zur Lösung der gestellten Aufgabe haben wir zwei Petri-Netze entwickelt. Beide stellen das Modell da, allerdings wurde das Modell unterschiedlich abstrahiert.

In Variante A wurde der Algorithmus ausführlich implementiert. Dadurch gibt es eine erhöhte Anzahl von Stellen, Transitionen und Kanten.

In Variante B ist der Algorithmus komprimierter und abstrakter dargestellt. Einige der Stellen und Transitionen aus Variante A sind hier zusammengefasst zu finden.

2 Erläuterung der Modellierung

2.1 Variante A

In Abbildung 3 wurde das Petri-Netz in Quadranten eingeteilt, um es hier verständlicher beschreiben zu können.

Die Quadranten oben links und unten links sind der Sender des Algorithmus. Der Empfänger ist in den oben rechts und unten rechts abgebildet. Es fällt auf, dass das Netz über die horizontale Achse symmetrisch ist. Das liegt daran, dass die obere Hälfte sich mit dem Senden und Empfangen der Nachrichten mit positivem Kontrollbit kümmert, während der untere Teil für die Nachrichten mit negativem Kontrollbit zuständig sind. Es reicht daher eine der Hälfte in der Funktionsweise zu erklären, da die andere Hälfte sich genauso verhält, nur dass sie dabei eine andere Nachricht behandelt.

In *m1* ist das Start-Token gesetzt, es wird daher zu Beginn eine Nachricht mit positivem Kontrollbit verschickt. Sobald die Nachricht durch die Stelle *send_m1* verschickt ist, erhält der Initiator *m1* erneut ein Token, dadurch wird ermöglicht, dass die Nachricht erneut verschickt wird, falls der Sender das Empfangen des Empfängers nicht bestätigt

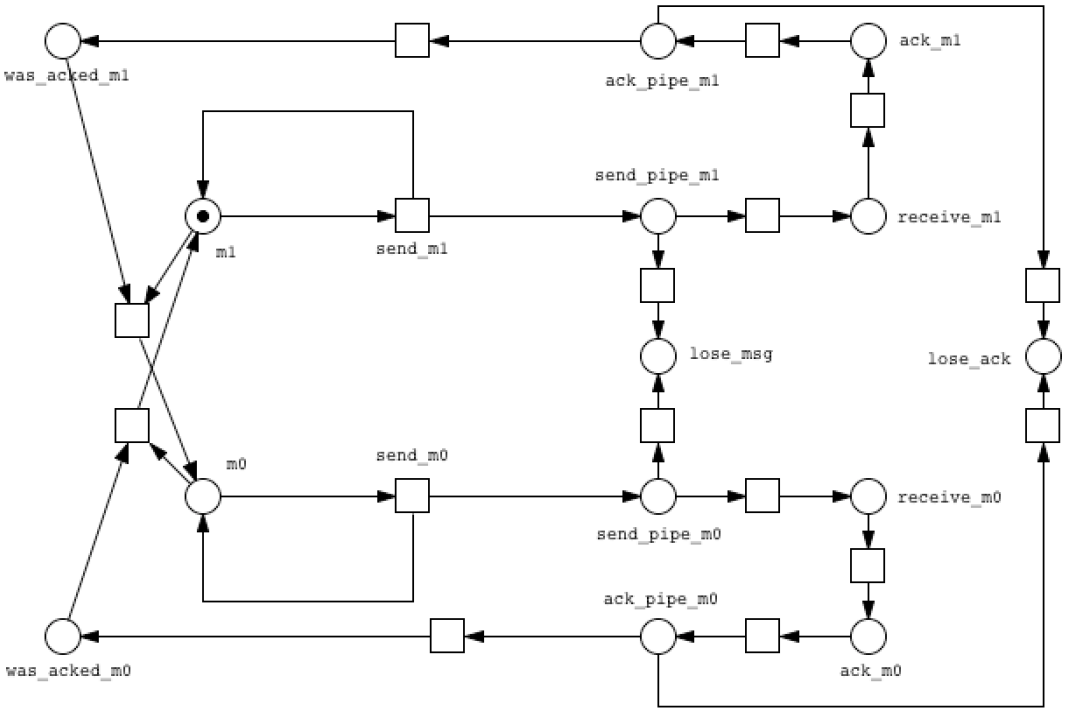

Figure 1: Variante A

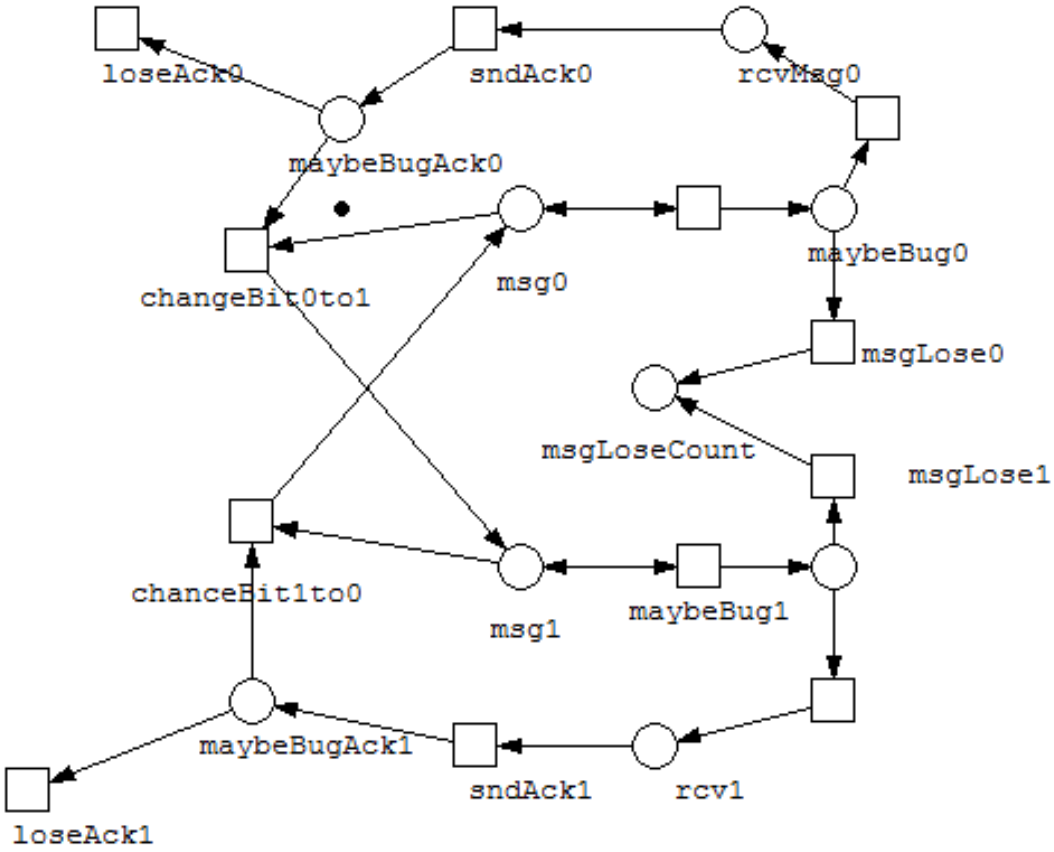

Figure 2: Variante B

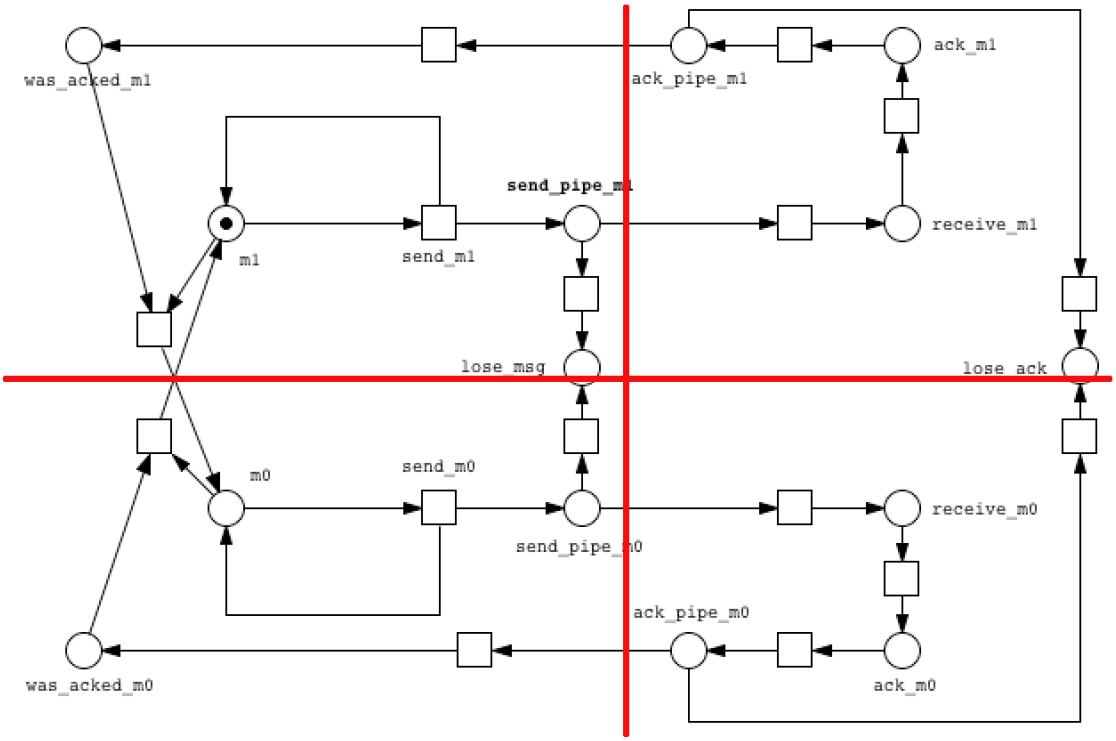

Figure 3: Variante A, in Quadranten eingeteilt

bekommt. Außerdem wird die Nachricht "physikalisch" auf die Leitung gelegt, die hier durch die Stelle *send_pipe_m1* repräsentiert wird. Dort kann es passieren, dass die Nachricht verloren geht. Sollte die Nachricht nicht verloren gehen, wird sie vom Empfänger in *receive_m1* erhalten.

Der Empfänger wird nun den Erhalt der Nachricht an den Sender bestätigen. Dazu möchte der Empfänger eine Nachricht senden in der Stelle *ack_m1*. Die Bestätigungs-Nachricht wird auf die "physikalische" Leitung gelegt, die hier durch die Stelle *ack_pipe_m1* dargestellt wird. Dort kann es passieren, dass die Bestätigungsnachricht verloren geht, und der Sender diese nicht erhält. Sollte der Sender die Bestätigung erhalten haben, merkt er sich dies in der Stelle *was_acked_m1*.

Sobald der Sender wieder sendebereit ist - wenn also ein Token in *m1* ist - kann der Sender wechseln zum Senden von Nachrichten mit negativem Kontrollbit bzw. der Stelle *m0*. Dort beginnt der Sende-Empfangs-Kreislauf wieder wie bereits für *m1* erklärt.

2.2 Variante B

Diese Modellierung beinhaltet zwei autarke Schleifen die das Senden, eventuelle Verlieren und empfangen der Nachrichten modellieren. Diese Schleife gibt es je für die Nachrichten mit dem Kontrollbit 1 und dem für 0.