

Aufgabenblatt 3

(zum 14. November 2012)

Gruppe 08

Björn Stabel 222128

Tim Strehlow 316594

Friedrich Maiwald 350570

3.1 Auswahl

a) Implementieren Sie den dreiphasigen Auswahlalgorithmus auf Bäumen aus der Vorlesung und testen Sie ihn auf verschiedenen Baumtopologien.

Wir haben den Algorithmus implementiert. Die Knoten sind zuerst weiß, nach der Explosionsphase werden sie orange. Nach Ende der Kontraktionsphase werden sie rot dargestellt, am Ende wenn sie über den Gewinner Bescheid wissen werden sie blau. In allen getesteten Fällen ist die Nachrichtenanzahl auch immer $3n-4+k$.

Ein Sonderfall ist, wenn ein oder mehrere der Initiator-Knoten Blätter sind. In diesem Fall müssen das die Initiator-Knoten selber merken, und senden sofort ihre ID als Kontraktionsnachricht aus. Knoten die noch nicht in der Explosionsphase waren und schon eine Kontraktionsnachricht erhalten (also Eltern dieser Initiator-Blätter), speichern dieses Maximum, starten die Explosion auf allen anderen Kanälen und warten auch dort auf die Rückmeldung der Kontraktionsphase. Da wir uns die Explosionsnachrichten der Initiator-Blätter sparen (sie könnten ja auch von den Kontraktionsnachrichten überholt werden), kommen wir auch hier auf $3n-4+k_i$ Nachrichten, wobei k_i nur für die Initiatoren steht, die keine Blätter sind.

b) Testen Sie den Algorithmus auf einer Nicht-Baum-Topologie, zum Beispiel einem Hypercube oder einem Ring. Was passiert? Wie erklären sich Ihre Beobachtungen?

Wir haben den Algorithmus auf einem Ring getestet, und erwartungsgemäß hat nur die Explosionsphase geklappt. Danach passiert nichts weiter, weil es in diesen Topologien keine Blätter gibt, die die Kontraktionsphase einleiten. Selbst wenn es zusätzlich ein Blatt-Knoten geben würde, wäre damit nicht gesichert, dass der Algorithmus korrekt funktioniert, da der Nachrichtenverlauf der Kontraktionsphase auf die Baum-Topologie aufbaut und daher auf zyklischen Topologien wie Ring oder Hypercube nicht funktioniert.

3.2 Terminierung

Folgender Algorithmus zur Feststellung der verteilten Terminierung im Atommodell wird vorgeschlagen: Jede Nachricht erhält einen global eindeutigen Namen. In einer Kontrollwelle werden jeweils die Namen der gesendeten und der empfangenen Nachrichten in zwei Mengen S und E kumuliert. Terminierung wird gemeldet, wenn nach Abschluss der Kontrollwelle die beiden Mengen übereinstimmen ($S = E$). Ist dieser Algorithmus korrekt? Beweisen Sie Ihre Antwort!

Dieser Algorithmus wird in einem Atommodell korrekt funktionieren. Wenn am Ende der Kontrollwelle die Elemente der Mengen S und E übereinstimmen, bedeutet das Folgendes: Alle Nachrichten, die von Knoten gesendet wurden, die danach von der Kontrollwelle besucht werden, werden auch von Knoten empfangen, die erst nach dem Empfangen von der Kontrollwelle besucht werden. Somit kann es keine Nachrichten geben, die die Kontrollwelle von der einen Seite (zeitlich vor der Kontrollwelle gesendet und nie empfangen) oder anderen Seite (zeitlich vor der Kontrollwelle empfangen aber „nie gesendet“) schneiden.

Ein Nachteil dieses Algorithmus ist es, dass in der Kontrollwelle die Namen aller gesendeten Basisnachrichten gespeichert werden, was je nach Netzgröße und Beschaffenheit des Basisalgorithmus zu sehr großen Nachrichten der Kontrollwelle führen kann.