Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Christian Scheideler
Johannes Nowak

Wintersemester 2006/2007
Übungsblatt 4
16. Mai 2007

Netzwerk-Algorithmen

Aufgabe 1

Ziel dieses Problems ist es zu zeigen, dass NTO universell stabil ist. Wir beweisen dies durch vollständige Induktion. Betrachte einen beliebigen (w, λ)-beschränkten Gegner mit $\lambda = 1 - \epsilon$ für ein $\epsilon > 0$.

(a) Zeigen Sie als Induktionsanfang, dass ein Paket an seiner ersten Kante höchstens w/ϵ-mal aufgehalten werden kann. Nehmen Sie an, das sei nicht der Fall, d.h. es gibt ein Paket P, das mehr als w/ϵ-mal an seiner ersten Kante aufgehalten wird, und sei e diese Kante. Sei t der Zeitschritt, an dem P zum $w/\epsilon + 1$-ten mal an e aufgehalten wird. Wir gehen rückwärts in der Zeit von t bis wir einen Zeitpunkt t' erreichen, an dem kein Paket an der Kante e gewartet hat, dessen erste Kante e ist. Betrachten Sie das Zeitintervall $I = [t' + 1, t]$ und argumentieren Sie, dass alle Pakete, die e während I durchlaufen haben, auch während I injiziert worden sind. Verwenden Sie diese Tatsache um zu zeigen, dass $\lambda(|I| + w) \geq |I|$ ist, was zu einem Widerspruch unserer Annahme führt. Also braucht jedes Paket höchstens $(w+1)/\epsilon$ Zeit, um die erste Kante zu durchlaufen.

(b) Angenommen, Sie haben bereits gezeigt, dass jedes Paket für das Durchlaufen von d Kanten höchstens

$$\sum_{i=1}^{d} \frac{w+1}{\epsilon^i}$$

Zeit benötigt. Nehmen Sie dann an, es gäbe ein Paket, das mindestens $(w+1)/\epsilon^{d+1}$-mal an seiner $(d+1)$-ten Kante aufgehalten wird. Zeigen Sie mit ähnlichen Argumenten wie in (a), dass dann für ein gewisses Zeitintervall I gelten muss, dass

$$\lambda \left(\sum_{i=1}^{d} \frac{w+1}{\epsilon^i} + |I| + w \right) \geq |I|$$

Leiten Sie daraus einen Widerspruch ab.

Also kann dieselbe worst-case Zeitschranke für NTO wie für SIS gezeigt werden.

Aufgabe 2

Zeigen Sie, dass es für $\lambda \geq 0.76$ ein Netzwerk und einen Gegner gibt, so dass NTG instabil ist. (Hinweis: verwenden Sie dieselbe Strategie wie für FIFO.)

Aufgabe 3

Schreiben Sie ein Programm in der Subjects-Umgebung, das bei Eingabe d einen d-dimensionalen Hypercube generiert.