

TT1 Praktikum 1 & 2 : Ausarbeitung

Carsten Noetzel, Armin Steudte

09.11.2011

Inhaltsverzeichnis

1 Projektschritt 1 - Aufbau	1
2 Projektschritt 2 - Vergleich Babel und OLSR	1
2.1 Babel	1
2.1.1 Informationsaustausch	1
2.1.2 Mesh - Konfiguration	2
2.1.3 Loop-Verhinderungsstrategie	3
2.2 OLSR	3
2.2.1 Informationsaustausch	3
2.2.2 Mesh - Konfiguration	3
2.2.3 Loop-Verhinderungsstrategie	3
3 Projektschritt 3 - Vergleich der Übertragungsqualität	3

Abbildungsverzeichnis

1 Projektschritt 1 - Aufbau

Im ersten Praktikumstermin wurden die Router konfiguriert und mit den nötigen Protokollen versehen. Die Versuchsdurchführung im zweiten Praktikumstermin wurde in Zusammenarbeit mit André Harms und Oliver Steenbruck durchgeführt und die Ergebnisse dokumentiert.

2 Projektschritt 2 - Vergleich Babel und OLSR

2.1 Babel

2.1.1 Informationsaustausch

Jeder Router A hält beim Babel-Protokoll die Kosten zu einem Nachbarrouter B in der Form $C(A, B)$ vor. Die Metrik einer Route bestimmt sich über die Summierung aller Kosten zwischen den Knoten die auf der Route liegen. Das Ziel des Algorithmus liegt darin für jede Quelle S einen Baum der Routen, mit den niedrigsten Metriken zu S zu berechnen.

Babel nutzt *Hello-Messages* zur Durchführung des *Neighbour-Discovery-Processes*. Hello-Messages werden periodisch über alle Interfaces des Knoten gesendet um so die Nachbarknoten zu entdecken. Zusätzlich werden Hello-Messages in Verbindung mit den *IHY-Messages* (I Heard You Messages) genutzt, um die *Bidirectional Reachability* und die Metrik zwischen Sender- und Empfängerknoten zu ermitteln. Der Empfänger antwortet auf Hello-Messages mit den IHY-Messages. Diese erhalten die, mit Hilfe der Hello-Messages ermittelten, Laufzeitverzögerungen aus Sicht des Senders der Hello-Messages. Mit Hilfe der Laufzeitverzögerungen findet dann die Bewertung der Qualität der Links, sowohl in Sende- als auch Empfangsrichtung statt.

2.1.2 Mesh - Konfiguration

Die lokalen Mesh-Konfigurationen werden über einen verteilten Bellman-Ford Algorithmus berechnet.

Hierzu hält jeder Router A zwei Werte vor, zum einen die geschätzte Distanz zu S (bezeichnet als $D(A)$) und den Next-Hop-Router zu S (bezeichnet als $NH(A)$). Zu Beginn des Informationsaustausches ist $D(A)=\text{unendlich}$ und $NH(A)$ ist nicht definiert.

Periodisch sendet jeder Knoten B ein Update seiner Routen an alle seine Nachbarn mit dem Inhalt $D(B)$, welcher die Distanz zu S angibt. Erhält ein Nachbar A von B ein Update, so prüft A ob B für S als Next-Hop-Router eingetragen ist. Ist dies der Fall wird als Distanz $D(A)$ die Summe aus $C(A,B)$ (= Kosten von A nach B) und $D(B)$ (= Kosten von B zu S) gesetzt. Damit ist im Knoten A die Metrik von A nach S über B aktualisiert.

Im Fall, dass B nicht als Next-Hop-Router für S auf A eingetragen ist, vergleicht A die Summe aus $C(A,B)$ (= Kosten von A nach B) und $D(B)$ (= Kosten von B zu S) mit dem gegenwärtigen Wert von $D(A)$. Ist $C(A,B)+D(B)$ kleiner als $D(A)$ ist die angebotene Route besser als die bisher eingetragene und $NH(A)$ wird auf B und $D(A)$ auf $C(A,B)+D(B)$ gesetzt.

Im Pseudocode sieht dies in etwa so aus:

```
1 receiveRouteupdate( $D(B)$ ) from B for S
2 If  $NH(A) = B$  Then
3      $D(A) = C(A,B)+D(B)$ 
4 Else
5     If  $C(A,B)+D(B) < D(A)$  Then
6          $NH(A) = B$ 
7          $D(A) = C(A,B)+D(B)$ 
8     EndIf
9 EndIf
```

Durch den Austausch der Updates wird Mesh-Konfiguration festgelegt und die Knoten wissen an welchen Knoten sie Pakete weiterleiten müssen, um ein bestimmtes Ziel mit möglichst geringen geschätzten Gesamtkosten zu erreichen.

2.1.3 Loop-Verhinderungsstrategie

2.2 OLSR

2.2.1 Informationsaustausch

2.2.2 Mesh - Konfiguration

2.2.3 Loop-Verhinderungsstrategie

3 Projektschritt 3 - Vergleich der Übertragungsqualität