



9. Blatt Praktikum Protokolldesign WS 07/08

Beachte: Du hast für dieses Blatt 2 Vorlesungswochen Zeit. Wegen der Weihnachtsferien ist die Abgabe dann am 15. Januar 2008.

Die nächste Diskussionsgruppe nach den Weihnachtsferien findet statt am 7. Januar 2008 und behandelt das 8. Aufgabebblatt. Die nächste Zentralübung nach den Weihnachtsferien ist am 15. Januar 2008.

Aufgabe 1: (150 Punkte) Implementierung eines vereinfachten BGP-Routers im P2P-Netzwerk

Nachdem wir auf dem 7. Übungsblatt eine einfache Forwarding-Tabelle implementiert haben, werden wir nun das Füllen dieser Tabelle mit Hilfe eines echten Routing-Protokolls vornehmen. Dieses Routing-Protokoll ist an das im Internet verwendeten Pfad-Vektor Protokoll BGP¹ angelehnt, das mit dem 8. Aufgabenblatt eingeführt wurde.

Nimm an, dass jeder Knoten in dem P2P-Netzwerk ein eigenes Autonomes System (AS) sei. Dabei sei die AS-Nummer durch die Knoten-ID repräsentiert.

Das zu verwendende vereinfachte BGP-Protokoll wird im Folgenden spezifiziert:

BGP-Nachrichten

Routingprotokolle wie z.B. BGP dienen dazu, Erreichbarkeits-Informationen zwischen Routern auszutauschen. Im Falle von BGP geschieht dies zwischen zwei Knoten die über eine BGP-Sitzung miteinander verbunden sind. Für unser P2P-Netz gehen wir davon aus, dass jeder Knoten eine BGP-Sitzung mit allen seinen direkten Nachbarn unterhält und das diese Sitzung implizit mit dem HELLO-Handshake aufgesetzt wird.

Um BGP-Nachrichten zwischen zwei Nachbarn austauschen zu können soll ein neuer Nachrichtentyp genutzt werden, der als generischer Transportmechanismus zwischen zwei Nachbarn dient. Diese Nachrichten können auch zum übertragen anderer Informationen als BGP-Nachrichten benutzt werden. Da dieser Transportmechanismus nur für die Kommunikation zwischen Nachbarn ausgelegt ist enthalten seine Nachrichten weder Absender oder Empfänger noch eine TTL. Da dieser Mechanismus generisch ist, muss der **Application:**-Parameter die Anwendung spezifizieren, von der dieser Transportmechanismus benutzt wird. Die Nachrichten mit Hilfe derer dieser Transportmechanismus implementiert ist sehen so aus:

```
LOCALTRANSPORT P2P/0.2\r\n
Application: <app>\r\n
Content-Length: <body size>\r\n
\r\n
<payload>
```

Die Behandlung von LOCALTRANSPORT-Nachrichten funktioniert ähnlich wie die von Handshake-Nachrichten: Sie werden niemals weitergeleitet, enthalten deshalb auch weder Absender oder Empfänger noch eine TTL. Allerdings dürfen auch LOCALTRANSPORT-Nachrichten erst *nach* einem erfolgreichen

¹Border Gateway Protocol

HELLO-Handshake versandt werden. Ansonsten stellt dies eine Protokollverletzung dar auf die mit der entsprechenden Fehlerbehandlung zu reagieren ist.

Wir werden den LOCALTRANSPORT-Mechanismus nutzen um BGP-Informationen zwischen Nachbarn auszutauschen. Die eigentliche BGP-Information wird dann in mit `\r\n` abgeschlossenen Zeilen im Body der Nachricht übertragen.

Folgende Informationen werden für BGP mittels LOCALTRANSPORT übertragen:

Melden neuer und Ändern von bestehenden besten Routen: Die Nachricht zum Melden einer neuen oder zur Aktualisierung einer bestehenden Route sieht folgendermassen aus:

```
LOCALTRANSPORT P2P/0.2
Application: BGP
Content-Length: <bytes>

ANNOUNCEMENT <ziel> <as-pfad>
```

Zurückziehen von Routen: Die Nachricht zur Mitteilung, dass keine Route mehr zu *einem vorher gemeldeten* Ziel mehr bekannt ist:

```
LOCALTRANSPORT P2P/0.2
Application: BGP
Content-Length: <bytes>

WITHDRAW <ziel>
```

Fehlermeldungen: Fehlermeldungen sehen so aus:

```
LOCALTRANSPORT P2P/0.2
Application: BGP
Content-Length: <bytes>

NOTIFICATION <fehlermeldung>
```

Auf eine NOTIFICATION-Nachricht hin ist die Verbindung umgehend zu beenden!

Der Body von BGP-Nachrichten kann aus mehreren ANNOUNCEMENT und WITHDRAW-Zeilen bestehen und auch gemischt sein (ausser NOTIFICATION). Bei der Berechnung der `Content-Length` ist das Zeilenende (`\r\n`) als 2 Byte mit zu berücksichtigen!

BGP-Routen

Routen in BGP bestehen in erster Linie aus einem Ziel und einem Weg zu diesem Ziel. Der Weg besteht normalerweise aus einer Folgen von AS-Nummern (hier Knoten-IDs).

Es kann *pro Nachbar* höchstens eine Route zum selben Ziel geben. Das liegt daran, dass ein Knoten immer genau die Route weitergibt, die er auch selbst benutzt. Und das ist hier immer die beste die er kennt!

Wenn wir mehrere Nachbarn haben, haben wir auch mehrere Routen zum selben Ziel, alle über verschiedene Nachbarn. Diese Redundanz ist notwendig, falls eine Route wegfällt. In diesem Fall können wir stattdessen eine der anderen Routen nehmen.

Die beste Route zu einem Ziel ist die, die den kürzesten AS-Pfad hat.

Der AS-Pfad dient auch dazu, Routing-Schleifen zu vermeiden. Da wir den ganzen Pfad kennen ist es leicht zu vermeiden, dass ein AS im Pfad mehrfach auftritt. Wenn man eine Route erlernt und man selbst taucht im AS-Pfad bereit auf, so ist diese Route zu ignorieren.

Der Pfad wird bei der Weitergabe mittels ANNOUNCEMENT-Nachricht um die eigene AS-Nummer erweitert, da der Knoten aus Sicht seines Nachbarn Teil des Pfades ist. Ausserdem muss jeder Knoten sich zur Vermeidung von Schleifen im Pfad eintragen.

Wenn der Pfad beim Senden einer ANNOUNCEMENT-Nachricht erweitert wird, so geschieht dies immer am Anfang (links). Dies wird auch 'path prepending' genannt.

Format von BGP-Routen Da wir nicht alle Fähigkeiten des echten BGP-Protokolls implementieren wollen, soll mit der folgenden vereinfachten Form von Routen gearbeitet werden:

```
dest      ::= node-id
as-path   ::= node-id |
            node-id "," as-path
route     ::= dest _ as-path
```

Die Knoten-ID am Ende des AS-Pfades ist der Knoten, der die Route ursprünglich gemeldet hat, sollte also identisch mit dem Ziel sein. Ausserdem sollte der erste Knoten am Anfang des AS-Pfades unser direkter Nachbar sein, und zwar der, von dem wir die Route gelernt haben.

Wie oben beschrieben muss der Pfad vor dem Senden an einen Nachbarn um die eigene AS-Nummer erweitert werden.

BGP-Tabelle

Die BGP-Tabelle dient zum Speichern und Verwalten aller Routen, die wir mit der Zeit lernen. Pro Ziel ist höchstens eine dieser Routen die beste, also die, die wir benutzen.

Einträge in der BGP-Tabelle Ein Eintrag in dieser Tabelle hat folgendes Format:

```
DESTINATION NEXTHOP ASPATH BEST
```

DESTINATION ist die Ziel-Knoten-ID für diese Route. NEXTHOP ist hier der Nachbar, von dem wir diese Route erlernt haben, und auch gleichzeitig der Nachbar, über den eine Nachricht weiterzuleiten wäre, falls diese Route die beste zum gegebenen Ziel ist. ASPATH enthält den AS-Pfad zum Ziel. BEST ist ein Flag das die momentan beste Route zu ihrem Ziel markiert.

Initialisierung der BGP-Tabelle Beim Start der Knoten-Software ist die Tabelle leer. Dies gilt ebenso für die Forwarding Tabelle.

Neue Nachbarverbindungen Sobald eine neue Nachbarverbindung fertig aufgebaut ist (also nach dem HELLO-Handshake) wird eine neue Route zum neuen Nachbarn in die Tabelle eingepflegt. Diese hat als Ziel den neuen Nachbarn und einen AS-Pfad der genau aus dem Nachbarn besteht. Jetzt wird eine Routenberechnung (siehe unten) mit dem neuen Nachbarn als Ziel durchgeführt.

Anschliessend werden alle besten Routen in Form von ANNOUNCEMENT-Nachrichten an den neuen Nachbarn gesandt. Dabei wird der AS-Pfad in der Nachricht durch *voranstellen* (prepending) der eigenen AS-Nummer um einen Hop verlängert! Der Eintrag in der Tabelle wird dabei *nicht* verändert.

Empfang einer ANNOUNCEMENT-Nachricht Beim Empfang einer ANNOUNCEMENT-Nachricht erfolgt als erstes die Erkennung von Routing-Schleifen. Dazu wird nachgesehen, ob der Empfänger selbst bereits im AS-Pfad aufgeführt sind. Sollte dies der Fall sein, so ist die Nachricht zu ignorieren!!

Andernfalls wird die Route in die Tabelle eingetragen. Sollte bereits eine Route zum selben Ziel und über den selben Nachbarn existieren, so ist diese zu überschreiben. Dies heisst, dass der Nachbar seine Meinung zur alten Route geändert hat und diese durch die neue ersetzen will. Dies wird auch 'implicit withdraw' genannt, weil es einem WITHDRAW gefolgt von einem ANNOUNCEMENT entspricht.

Anschliessend wird eine Routenberechnung (siehe unten) für das Ziel aus der ANNOUNCEMENT-Nachricht Ziel durchgeführt.

Empfang einer WITHDRAW-Nachricht Wenn eine WITHDRAW-Nachricht empfangen wird, es aber keine Route zum angegebenen Ziel in der BGP-Tabelle gibt, ist die Nachricht zu ignorieren. Wenn eine WITHDRAW-Nachricht empfangen wird, und es eine Route zum angegebenen Ziel gibt, soll die Route aus der BGP-Tabelle gelöscht werden, die das selbe Ziel wie in der WITHDRAW-Nachricht angegeben hat *und* der als NEIGHBOUR den selben Knoten hat, von dem wir die WITHDRAW-Nachricht empfangen haben.

Anschliessend wird für dieses Ziel eine Routenberechnung (siehe unten) für das Ziel der gelöschten Route durchgeführt.

Verlust einer Nachbarverbindung Beim Verlust einer Nachbarverbindung müssen *alle* Routen gelöscht werden, die den Nachbarn als Next Hop referenzieren. Anschliessend sind Routenberechnung (siehe unten) für alle betroffenen Ziele durchzuführen. Der gesamte Vorgang kann behandelt werden wie eine Folge von WITHDRAW-Nachrichten für alle betroffenen Routen.

Routenberechnung

Die Routenberechnung wird für jedes Ziel getrennt durchgeführt. Wenn sich eine Änderung in der bis jetzt bestehenden besten Route zu diesem Ziel ergibt (auch dann, wenn keine Route mehr gefunden wird), so sind alle Nachbarn darüber zu informieren. Ausserdem ist die Forwarding-Tabelle anzupassen.

Eine kurze Beschreibung des Algorithmus sieht dann folgendermassen aus:

- für ein gegebenes Ziel:
 - beste Route finden (die mit dem kürzesten AS-Pfad)
 - falls die neue beste Route eine andere als die alte ist (anderer Next Hop oder anderer AS Pfad)
 - * Forwarding Tabelle updaten durch eintragen des NEXTHOPS der neuen besten Route
 - * ANNOUNCEMENT broadcasten mit neuer bester Route (eigene AS-Nummer vor den Pfad stellen!)
 - falls keine Route mehr gefunden wird, vorher aber eine existiert hat
 - * Eintrag aus Forwarding Tabelle löschen
 - * WITHDRAW broadcasten
 - falls alte und neue beste Route identisch ist
ODER
es vorher keine Route gegeben hat und auch jetzt keine Route existiert
 - * fertig.

Abzugeben sind:

- Dein Programm (sowohl Quelltext als auch ggf. kompiliert im Falle von C, C++ und Java)
- Ggf. eine Beschreibung, was funktionieren sollte, dies leider aber immer noch nicht tut...

Details zur Abgabe der Aufgaben: siehe FAQ

(unterhalb http://www.net.t-labs.tu-berlin.de/teaching/ws0708/PD_labcourse/).

Abgabedatum: Dienstag, 15. Januar 2008, 11:59h s.t.

FROHE WEIHNACHTEN UND EINEN GUTEN RUTSCH INS NEUE JAHR!