ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Fakultät für Angewandte Wissenschaften, Institut für Informatik

Multiple Path Routing

Seminar: Peer-to-Peer Netzwerke

von

Sebastian Wagner

Seminarleiter : **Prof. Dr. Christian Schindelhauer**
Betreuer : **Arne Vater**

Datum der Abgabe: **24. Juli 2009**

Inhaltsverzeichnis

1 Einleitung

Heutzutage existiert eine Vielzahl unterschiedlicher Netzwerkprotokolle für den Datenaustausch. Ihre Routing-Algorithmen lassen sich grob in eine der folgenden Kategorien unterteilen: Link-State-Routing-Protokolle, Distanzvektor-Protokolle und Pfadvektor-Protokolle. Allen Protokollen gemeinsam ist das Konvergenzproblem. Diese tritt ein wenn ein zuvor berechneter Routingpfad, über den der Datenaustausch stattfinden soll, nicht benutzt werden kann. Diverse Gründe dafür sind beispielsweise Serverausfälle. In dieser Zeit, in der eine Neuberechnung stattfinden muss, ist das Routing inkonsistent. Eben diese Situationen führen zu einem erhöhten Zeit- und Resourcenverbrauch, wodurch es zu einer Verzögerung oder gar zum Abbruch der Übertragung kommen kann.

Im Folgenden wird der *Failure-Carrying Packets* (FCP) Routing-Alg. vorgestellt, welcher in der Lage ist Konvergenzprobleme vollständig zu beseitigen. Der Algorithmus ermöglicht es, ohne die vollständige Kenntnis der Netzwerkstruktur, Daten immer verlustfrei zu übertragen. So ist es selbst bei nicht realisierbaren, vorberechneten Routingpfaden möglich konvergenzfrei neue Routen zu berechnen. In der Ausarbeitung wird gezeigt, dass der Mehraufwand der die Konvergenzfreiheit garantiert, vernachlässigbar ist und nur zu einer kleinen Verlustrate bei der Übertragung führt. Zudem bliebt das Protokoll leicht steuerbar, was eine einfache Routingberechnung zu Folge hat. Dies ist gerade bei Applikationen bei denen der Paketverlust mit enormen Performance-Einbußen gleichzusetzen ist von großer Bedeutung (z.B. VoIP). Um also den Paketverlust so gering wie möglich zu halten muss ein funktionierender Pfad gewährleistet sein, über den die Daten versendet werden können. Bei herkömmlichen Routing-Algorithmen ist dazu der ständige Austausch von aktuellen Routingtabellen nötig, was unweigerlich zu Konvergenzen führt. Es gibt mehrere Möglichkeiten Konvergenzperioden so gering wie nur möglich zu halten, z.B. durch vorher berechnete Backup-Pfade. Kommt es dann zu einem fehlerhaften Link, kann dieser Backup-Pfad dazu benutzt werden um einen Ausweichpfad zum Ziel zu bekommen.

Im Folgenden wird nun das grundlegende FCP Protokoll vorgestellt und dessen Funktionsweise erläutert.

2 FCP

2.1 Funktionsweise

Im Folgenden wird genauer auf die Funktionsweise des FCP-Algoritmus eingegangen.

Beim FCP Ansatz besitzen alle Knotenpunkte einer Route denselben Netzwerkplan. Es ist somit nicht notwendig den zu versendenden Paketen zusätzliche Informationen über die Netzwerkstruktur mitzugeben. Stattdessen werden die Pakete dahingehend erweitert, dass Kenntnisse über fehlende oder defekte Links hinzugefügt werden. Defekte Links sind dadurch charakterisiert, dass keine zum Datenaustausch funktionsfähige Verbindung zwischen zwei Knotenpunkten besteht. Während der Paketweiterleitung können die Router ihre Pfadberechnung daran orientieren und es ist kein zwischenzeitlicher Austausch von Routingtabellen nötig. Zur genauen Beschreibung des FCP Protokolls wird im Folgenden davon ausgegangen, dass sich die Netzwerkstruktur nicht ändert. Wie die Verbreitung und Aktualisierung einer solchen Netzwerkstruktur (Map) beim FCP-Algoritmus funktioniert, wird in einem anderen Abschnitt beschrieben.

Der nachfolgende Pseudo-Code beschreibt die Grundzüge des FCP Protokolls.

```
Initialization: pkt.failed_links = NULL
Packet Forwarding:
  while (TRUE)
    path = ComputePath(M - pkt.failed_links)
    if(path == NULL)
      abort("No path to destination")
    else if(path.next_hop == FAILED)
      pkt.failed_links = path.next_hop
    else
      Forward(pkt, path.next_hop)
      return
```

[LCR+07]

Zu Beginn wird ein kürzester Pfad aus allen verfügbaren Links berechnet. Die Menge **M** enthält initial alle Links und `packet.failed_links` beschreibt die Menge aller momentan nicht erreichbaren Links. Falls kein Pfad berechnet werden konnte, also `path==NULL`, wird mit der Ausgabe "No path to destination" die Routenberechnung abgebrochen. Andernfalls erfolgt eine Paketweiterleitung zum nächsten Knoten. Anschließend wird getestet ob der im Pfad nächste Link erreichbar ist. Ist dies nicht der Fall, so wurde ein fehlerhafter Link gefunden, welcher in die Liste `pkt.failed_links` aufgenommen wird. Dieser Algorithmus wird so lange ausgeführt bis das Paket sein Ziel erreicht hat oder kein Pfad zum Ziel mehr existiert, was zum Abbruch führt. Fehlerhafte Links werden bei diesem Verfahren nicht in Routingtabellen aufgenommen, sondern dienen nur der Paketweiterleitung.

Abbildung 1 veranschaulicht die Funktionsweise des Routing Algorithmus noch einmal grafisch. Wie in der Abbildung zu sehen bestehe das Beispielnetz-

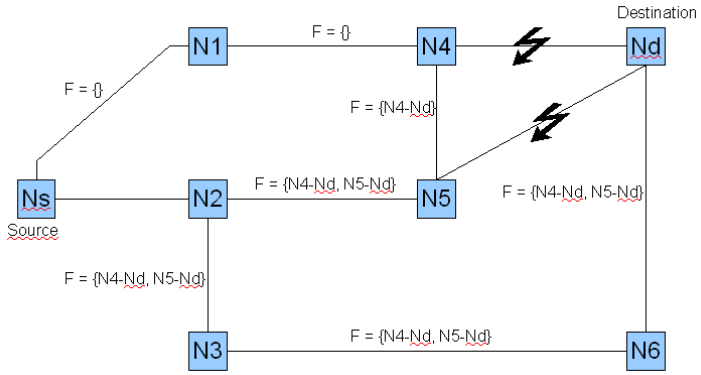

Abbildung 1: Beispiel für FCP-routing

werk aus acht verschiedenen Routern, wobei N_s den Source-Router und N_d den Destination-Router repräsentiert. Vom Source-Router ist es nun möglich über die Zwischenrouter N1-N6 Pakete an den Destination-Router zu übermitteln. In dem Beispiel ist zu beachten, dass jeder Router dieselbe Map des Netzwerkes besitzt und die beiden fehlerhaften Links (hier durch Blitze kenntlich gemacht) zur Laufzeit den Routern nicht bekannt sind. Beispielhaft soll nun ein Datenpaket von N_s nach N_d durch Einsatz des FCP Protokolls übertragen werden. Der Source-Router N_s berechnet zu Beginn aus der Liste **M** den kürzesten Pfad zum Destination-Router N_d. Wir nehmen dabei an, dass dieser Pfad über die Router N1 und N4 zu N_d verläuft. Von N1 erfolgt eine erfolgreiche Weiterleitung nach N4. Die Menge **F**, der fehlerhaften Links bleibt dabei leer. Im nächsten Übertragungsschritt verhindert allerdings der fehlerhafte Link die Weiterleitung des Pakets an den Zielrouter N_d. Hier tritt also der Fall ein, dass eine alternative Route berechnet werden muss und der fehlerhafte Link zwischen N4-N_d wird in die Liste **F** aufgenommen. Die weitere Berechnung erfolgt nun unter Berücksichtigung von **F**, wobei wir annehmen, dass die neue Route über den Router N5 nach N_d erfolgt. Wieder erhalten wir nach N5 einen neuen Eintrag in **F** und sind gezwungen eine Neuberechnung des Routingpfades vorzunehmen. Schlussendlich führt nur der Pfad P=N1;N2;N3;N6; zu N_d, über den das Datenpaket verlustfrei übertragen werden kann.

Das Beispiel zeigte anschaulich, dass trotz mehrerer fehlerhafter Links und ohne vorherige Kenntnis über diese, der FCP-Algorithmus in der Lage ist Datenpakete von dem Source-Router zum Destination-Router zu übertragen. Solange also noch ein Weg existiert findet ihn FCP und leitet die Datenpakete über alternative Routen weiter. Hiermit wird die Erreichbarkeit garantiert.

Bereits anhand des kleinen Beispiels lassen sich jedoch mehrere Optimierungsmöglichkeiten des primitiven FCPs erkennen. So ist problematisch, dass fehlerhafte Links, die einmal in **F** eingetragen wurde, nicht wieder entfernt werden können. Zudem sollte eine ständige Neuberechnung von fehlerhaften Links vermieden werden.

Im nächsten Abschnitt wird eben auf diese Probleme näher eingegangen und somit FCP-Optimierungsmöglichkeiten genauer vorgestellt.

2.2 Optimierungen

Um also den Neuberechnungsaufwand für die Paketweiterleitung so gering wie möglich zu halten speichert jeder Knoten einen *Bachup path*, welcher im Falle eines fehlerhaften Links genutzt werden kann. Dies spart entscheidet Rechenaufwand ein, da in unserem Beispiel die neu berechneten Pfade den Ausweichpfaden entsprechen hätten. Eine Neuberechnung wäre dann nicht nötig gewesen, bzw. ist nur noch dann nötig, wenn sowohl der *Default path* als auch der *Backup path* fehlschlägt. Dies führt also zu einer äußerst geringen Neuberechnungsrate, wenn die Anzahl der fehlerhaften Links verhältnismäßig gering ausfällt. Die Berechnung eines Ausweichpfades für einen bestimmten Knoten nimmt, im Vergleich zur Fehlererkennung von defekten Links, wenig Zeit in Anspruch. Deshalb ist es angemessen zuvor sämtliche kürzesten Wege zu berechnen und als Informationen zu speichern, um Neuberechnungen zu vermeiden.

Die nächste Optimierung ist das Speichern der defekten Links in Labels. Der Vorteil dabei ist ein bequemer Abgleich von Listen der fehlerhaften Links. Dabei werden lediglich die benachbarten Knoten um ihre Labels angefragt. Darauf hin werden die Listen mit der eigenen Liste verglichen und nur solche Links aufgenommen, die nicht in der Eigenen stehen. Wird jedoch eine neue Map eingeführt, verlieren auch die alten Labels an Aktualität und müssen erneuert werden.

2.3 Verbreitung von Netzwerkstruktur-Informationen

Der nachfolgende Abschnitt zeigt wie die Verteilung der Netzwerkstruktur-Informationen unter Verwendung des FCP-Algorithmus erfolgt. Wie zu Beginn erwähnt besitzt jeder Router die selbe Sicht des aktuellen Netzwerkstatus. Dabei ist es nötig dies zu bestimmten Zeitpunkten zu aktualisieren, da in einem Netzwerk wie beispielsweise dem Internet ständig neue Knoten hinzukommen oder wegfallen.

Um den Aufwand so gering wie möglich zu halten und trotzdem schnell reagieren zu können werden vorübergehende Änderungen nicht berücksichtigt. D.h. wenn ein Router temporär ausfällt, bleibt er immernoch in der aktuellen Map des Netzwerkes enthalten. Lediglich bei geplanten Ausfällen bzw. hinzukommen von neuen Routern werden Aktualisierungen vorgenommen, welche die alte Map updaten.

Die Verteilung wird mittels eines RCP-link coordinator vorgenommen, wie er in [CCF+05] beschrieben wird. Dieser versendet dann die aktuelle Version der Map via TCP an eine bestimmte Gruppe von Routern. Danach gibt jeder die Map an seine Nachbarn weiter, solange bis alle im Netzwerk auf dem neusten Stand sind. Falls dabei ein Knoten nicht aktiv ist, erhält er die neuen Netzwerkinformationen sobald er wieder eingeschaltet ist. Dabei kann jeder die Übertragung abbrechen um mehrfache Datentransfere zu unterbinden. In Abbildung 2 sehen wir den Coordinator C und verschiedene Router, dargestellt durch blaue Rechtecke. Die Map wird zuerst an bestimmte Knotenpunket versendet (N1-N6), welche dann die weiter Verbreitung an ihre Nachbarn vornehmen sollen. Neben den Änderungen der Netzwerkstruktur wird noch die Versionsnummer und eine Zeitkonstante geliefert, welche die Gültigkeit der neuen Map bestimmt. Erhält nun ein Router eine neue Map so wird ein Timer **t** gesetzt, nach dessen Ablauf die Map ihre Gültigkeit bezieht. Der Timer richtet sich dabei nach der

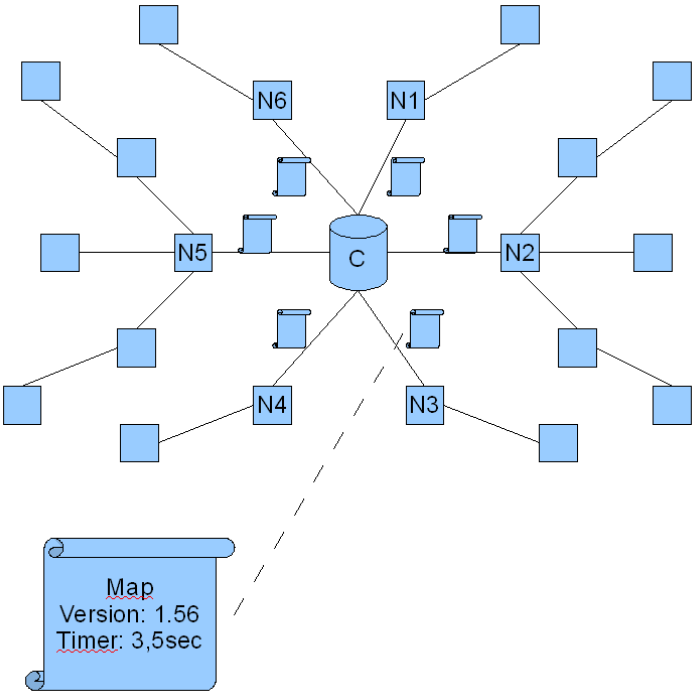

Abbildung 2: Verteilung der Map

Größe des Netzwerks und ist meist sehr vorsichtig gewählt. Jeder Knoten nimmt also eine Aktualisierung nach Ablauf des Timers vor oder er erhält erneut ein Paket mit einer noch neueren Version. Ist **t** also abgelaufen, kann das Routen mittels der neuen Map vorgenommen werden, da davon ausgegangen werden kann, dass jeder die neuen Netzwerkstruktur-Informationen erhalten hat. Ist dies dennoch nicht der Fall wird eine Methode angewendet, bei der Routing selbst mit unterschiedlichen Versionen möglich ist. Jedes Paket enthält zunächst eine Versionsnummer **s'**, die angibt welche Map verwendet wurde. Ist **s'** kleiner als die eigene Version **s**, besitzt also der Router eine aktuellere Version, so wird auf Grundlage der Map des Routers weitergeleitet. Ist jedoch die Nummer des Pakets größer, so wird ein *downgrade* vorgenommen und das Routing an die beim Router aktuellste Version angepasst. Dies kann allerdings nur passieren wenn der Timer **t** abgelaufen ist und trotzdem noch nicht alle Router ihre Map aktualisiert haben. Jedoch wird selbst dann, trotz unterschiedlicher Versionen sichergestellt, dass das Paket seinen Bestimmungsort erreicht.

Wann und wie oft eine Aktualisierung vorgenommen wird hängt von verschiedenen Faktoren ab. Beispielsweise davon wie oft ein Knoten hinzu bzw. weg kommt. Aus diesen Gründen errechnet sich schließlich eine Aktualisierungs-Periode nach der dann vorgegangen wird. In extremen Fällen kann dies schon alle 30 Sekunden sein.

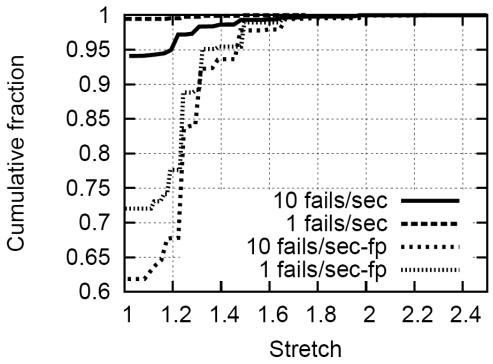

Abbildung 3: Strech-Statistik bei unterschiedlichen Fehlerraten [LCR+07]

3 Evaluation

Die Auswertungen der folgenden Test von FCP sollen nun vorher genanntes bestätigen und Parallelen zu bekannten Routingprotokollen herstellen. Dabei soll gezeigt werden, dass sowohl der Berechnungsaufwand als auch der Paketheader gering ausfallen. Der Vergleich mit traditionellem link-state Routing wie OSPF soll eine geringe Verlustrate, wie auch einen geringen Kontrollaufwand zeigen. Dabei wurde FCP mit zwei alternativen Strategien verglichen, OSPF und ein MPLS-ähnliches Protokoll. Es mussten jedoch einige Konfigurationen vorgenommen werden, damit ein Vergleich überhaupt möglich war und selbst dann gab es realitätsfremde Einschränkungen (Beispiel: Struktur des Netzwerk ändert sich nie). Allerdings ergaben einige Test, dass die Anzahl der überflüssig genutzten Knoten (hier: Stretch) selbst bei hoher Fehlerrate noch unter 2 liegt (Siehe Abbildung 3). Des Weiteren haben Tests ergeben, dass auch der Neuberechnungsaufwand bei einem Fehler pro Sekunde nur bei 2% aller Pakete aufgebracht werden muss. Die Zeit für die Berechnung eines neuen Pfades liegt dabei, unter Verwendung von moderneren Rechnern, unter einer Millisekunde. Weiter Untersuchungen haben gezeigt, dass FCP im Vergleich mit OSPF sowohl eine geringere Verlustrate, als auch einen kleineren Stretch aufweist. Es war zwar möglich OSPF dahingehend zu verbessern, dass die Verlustrate geringer ausfällt, jedoch nur unter Aufwendung eines erhöhten Kontrollflusses.

4 Fazit

Nachdem wir nun gesehen haben, dass dieses Routingprotokoll ohne Konvergenzen auskommt, geringere Verlustraten aufweist und einen kleineren Overhead bei Kontrollflüssen und Paketköpfen hat, stellt sich die Frage warum es nicht überall eingesetzt wird?

Entscheidender Grund dafür wird wohl die Implementierung in bestehende Protokolle sein. FCP stellt gewisse Ansprüche an Router, wie z.B. das Zwischenspeichern von zwei Netzwerk-Maps (die Letzte und die Aktuelle). Des Weiteren werden Koordinatoren benötigt die eine Verteilung der Maps vornimmt und dadurch ein gewisses Maß an topologischer Anordnung des Netzwerks voraussetzt. Die Paketweiterleitung auf Grundlage der Labels, die die Failed-Links enthalten, stellt weitere Anforderungen an die Router.

Speziell aus diesen Gründen ist es wohl noch nicht rentabel die Technik der *Failure-Carrying-Packets* einzusetzen.

Literatur

[CCF+05] Matthew Caesar, Donald Caldwell, Nick Feamster, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. Design and implementation of a routing control platform. In *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, pages 15–28, Berkeley, CA, USA, 2005. USENIX Association.

[LCR+07] Karthik Lakshminarayanan, Matthew Caesar, Murali Rangan, Tom Anderson, Scott Shenker, and Ion Stoica. Achieving convergence-free routing using failure-carrying packets. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 241–252, New York, NY, USA, 2007. ACM.