Seminararbeit - R3: Resilient Routing Reconfiguration

Thomas Bersch

Zusammenfassung

Die Wiederherstellung der Erreichbarkeit von Netzknoten nach dem Ausfall eines Netzwerk-Links ist eine wichtige Aufgabe heutiger Routing-Protokolle. Weiterführende, oder daraus resultierende Probleme werden von heutigen Verfahren jedoch oft außer Acht gelassen. Routing Resiliency Rekonfiguration (R3) von Wang et al. [WWMA+10] ist ein neuer Ansatz, der es ermöglicht einige dieser Probleme zu berücksichtigen. R3 kann zum einen Staufreiheit nach Link-Ausfällen garantieren, ist flexibel an verschiedene Leistungsanforderung anpassbar und ist robust gegenüber variierender Netzbelastung oder sich ändernde Netztopologien. Außerdem erfordert R3 nur geringen Berechnungs- und Speicheraufwand für die eingesetzten Router.

1 Einleitung

Heutige Computernetze, wie beispielsweise das IP-basierte Internet, bestehen in der Regel aus vielen einzelnen Netzknoten. Diese sind nicht notwendigerweise direkt, sondern über mehrere Knoten hinweg miteinander verbunden. In diesem Fall spricht man von einem vermaschten Netz. Um Information über das Netz zu entfernten Knoten zu übertragen, müssen diese von den dazwischen liegenden Netzknoten solange weitergereicht werden, bis der Zielknoten erreicht wird. Die Wahl der zu verwendenden Zwischensysteme, also die Wahl eines Weges durch das Netz wird als Routing bezeichnet. Da in aller Regel mehrere Weg zur Verfügung stehen, ist die Optimierung des Weges bezüglich einer Metrik, wie z.B. der kürzeste Weg, oder die aktuelle Netzauslastung ein wichtiger Aspekt beim Routing.

Neben der Optimierung des Weges, spielt beim Routing ein weiterer Aspekt ein wichtige Rolle. Über die Zeit hinweg kann sich ein Netztopologie ständig ändern, wodurch zuvor gefundene Routen ungültig und neue gesucht werden müssen. Gründe dafür sind beispielsweise der Ausfall eines Netzknotens, das Ändern von Routing-Metriken, die aktuelle Netzauslastung etc.. Um auf solche Situationen reagieren zu können, müssen die Router die Netztopologie beobachten und Informationen über dessen aktuellen Zustand austauschen. Hierfür gibt es eine Reihe von Routing-Protokollen wie beispielsweise die im Internet eingesetzten Protokolle Routing Information Protokoll (RIP) [Malk98], Open Shortest Path First (OSPF) [Moy98] oder Intermediate System - Intermediate System (IS-IS) [Oran90].

Eine wichtige Eigenschaft in diesem Zusammenhang ist *Network Resiliency* [WWMA+10]. Mit Network Resiliency wird die Fähigkeit eines Netzwerks bezeichnet, sich schnell und problemlos nach einer Reihe von Fehlern oder Unterbrechungen zu reorganisieren und eine Datenübertragung wieder zu ermöglichen. Viele Verfahren legen dabei vorwiegend Wert auf die Wiederherstellung der Verfügbarkeit eines oder mehrerer Netzknoten. Weitergehende Probleme, die z.B. aus der Umgehung defekter Links hervorgehen können, werden häufig außer Acht gelassen. Dazu zählt beispielsweise die Mehrbelastung des verbliebenen Netzwerks durch die verminderte Netzkapazität und den umgeleiteten Verkehr. Iyer at al. [IBTD03] führt dies als eine der Hauptursachen für das Auftreten von Stausituationen in IP-basierten Netzen an. Ein weiterer, oft nicht berücksichtigter Aspekt ist die fehlende Möglichkeit, Aussagen über die Leistungsfähigkeit eines Netzes nach einem Link-Ausfall zu treffen.

Eine Schwierigkeit bei der Entwicklung von Routing-Verfahren, die auch die oben genannten Probleme berücksichtigen, ist die Vielzahl möglicher Fehlerszenarien. Bei einem Netz mit 500 Links sind bei 3 Link-Fehlern schon über 20 Millionen Szenarien möglich. Die Berücksichtigung jedes dieser Szenarien bedeutet i.d.R einen zu hohen Berechnungsaufwand. Um aber eben beispielsweise Staufreiheit nach einem Link-Fehler garantieren zu können, muss eine Aussage für jedes Szenario getroffen werden.

Das von Wang et al. [WWMA+10] entwickelte Verfahren Routing Resiliency Reconfiguration (R3) ist ein neues Verfahren, um sowohl ein Routing für den Normalbetrieb ohne Fehler, als auch ein Protection Routing für den Fall eines oder mehrere Link-Ausfälle zu berechnen und diese nach jedem Fehler auf die momentane Netztopologie anzupassen. Dazu wird ein Ansatz vorgestellt, der die Berücksichtigung aller möglichen Fehlerszenarien vermeidet. Weiterhin wird gezeigt, dass das Verfahren Staufreiheit durch Link-Ausfälle garantieren kann. Zudem wird gezeigt, dass das Verfahren effizient in Hinsicht auf Berechnungs-Overhead und Speicherverbrauch ist, es flexibel an verschieden Leistungsanforderungen (z.B. zur Einhaltung von Service Level Agreements) angepasst werden kann und robust gegenüber Änderungen der Netzauslastung oder der Netztopologie ist.

2 Grundlagen

Neben der pfadbasierten Auffassung von Routing, ist es auch möglich Routing als Fluss [ApCo03] und das zugrunde liegende Computernetzwerk als Flussnetzwerk zu betrachten. Häufig ist ein Pfad (Route) von Sender zu Empfänger nicht eindeutig, sondern es existieren mehrere Pfade für das Paar aus Sender und Empfänger. Für die Datenübertragung kann nicht nur einer dieser Pfade, sondern auch mehrere unterschiedliche Pfade gleichzeitig genutzt werden. Jeder Pfad überträgt dann einen gewissen Anteil der zu übermittelnden Informationen. Fasst man das Netzwerk als gerichteten Graph $G = (V, E)$ auf, wobei V der Menge der Router und E der Menge der Netzwerkverbindungen zwischen den Routern entspricht, lässt sich so ein Flussnetzwerk definieren. Die Werte $r_{ab}(e)$ geben dazu jeweils den Anteil einer Übertragung von $a \in V$ nach $b \in V$ an, der über die Verbindung $e \in E$ geroutet wird. Als Menge $\{r_{ab}(e) | a, b \in V \wedge e \in E\}$ zusammengefasst beschreiben diese Werte so das gesamte Routing r. Der Beitrag zur Auslastung einer Verbindung e entspricht dem Produkt $d_{ab} r_{ab}(e)$, wobei d_{ab} dem aktuell von Sender a initiierten Verkehr zu Empfänger b entspricht und $r_{ab}(e)$ dem wie oben beschriebenen Anteil. Ein so definiertes Routing gilt als korrekt, wenn es die in [WWMA+10] definierten Bedingungen erfüllt. Nachfolgend sind diese Bedingungen nocheinmal aufgeführt.

Definition 1 *Ein Routing r ist korrekt, wenn r die nachfolgenden Bedinungen erfüllt:*

$$\begin{array}{ll} \textit{[R1]} & \forall i \neq a, b : \sum_{(i,j)\in E} r_{ab}(i,j) = \sum_{(j,i)\in E} r_a b(j,i) \\ \textit{[R2]} & \sum_{(a,i)\in E} r_{ab}(a,i) = 1 \\ \textit{[R3]} & \forall (i,a) \in E : r_{ab}(i,a) = 0 \\ \textit{[R4]} & \forall e \in E : 0 \leq r_{ab}(e) \leq 1 \end{array} \tag{1}$$

Die erste Bedingung entspricht dem für Flussnetzwerke üblichen Flusserhalt und fordert, dass jeder Zwischenknoten genauso viel weiterleitet wie an ihn gesendet wird. Die zweite Bedingung stellt sicher, dass der gesamte von einer Quelle initiierte Verkehr weitergeleitet wird, während die dritte Bedingung verbietet, dass ein Teil davon wieder zur Quelle zurückgesendet wird. Die vierte Bedingung fordert, dass der weitergeleitete Anteil zwischen 0% und 100% liegt.

Alternativ zum klassischen IP-Routing gibt es auch andere Verfahren zur Weiterleitung von Dateneinheiten innerhalb eines Netzwerks. Ein Beispiel, welches auch in [WWMA+10]

als Implementierungsgrundlage von R3 verwendet wird, ist Multiprotolkoll-Label-Switching (MPLS) [RoVC01]. MPLS ist ein Verfahren, welches eine verbindungsorientierte Datenübertragung auch in ansonsten verbindungslosen Netzen ermöglicht. Während beim klassischen Routing auf Schicht 3 jeder Router erneut den nächsten Knoten (Next Hop) für eine Dateneinheit bestimmen muss, wird bei MPLS zu Beginn einer Übertragung jede Dateneinheit einer Weiterleitungsklasse (Forwarding Equivalent Class - FEC) zugeordnet, die den Pfad (Lable Switch Path - LSP) durch das Netz bestimmt. Dazu wird ihr eine kurze Kennung fester Länge, ein sog. Label zugeordnet. Die nachfolgenden Knoten können anhand dieses Labels aus einer Tabelle (Incoming Lable Map - ILM) den nächsten Knoten bestimmen. Eine Auswertung des kompletten Paketkopfes ist nicht mehr notwendig. MPLS biete außerdem die Möglichkeit einem Paket auch mehr als ein Label zuzuordnen. Dazu werden die Labels auf einem Stack, dem Lable-Stack abgelegt. Ein Router kann neue Labels auf dem Stack ablegen. Für die Auswertung des nächstens Routers wird immer das oberste Stackelement herangezogen. Ist der nächste Router der aktuelle Router, muss das Label wieder vom Stack entfernt werden. Der Label-Stack von MPLS wird in [WWMA$^+$10] zu Implementierung von R3 verwendet.

Solange die urspüngliche Netztopologie besteht, kann das reguläre Routing (d.h bereits berechnete Routen) verwendet werden. Bei Link-Ausfällen ist dieses Routing nicht mehr gültig und muss angepasst werden. Dazu existieren unterschiedliche Verfahren, sog. Routing Protection Schemes. Ein häufig genutztes Verfahren ist das in [ShBr10] für IP definierte Fast-Rerouting. Wird bei der Paketübertragung ein Netzwerkfehler von einem Router erkannt, muss für den ursprünglichen Pfad eine Alternative gefunden werden, die die ausgefallenen Netzkomponenten nicht verwendet. Solange ist keine Übertragung möglich. Die Dauer dieser Unterbrechung setzt sich dabei zusammen, aus der Zeit bis der Fehler erkannt wird, der Zeit bis der lokale Router auf den Fehler reagiert, die Zeit bis die Informationen über den Fehler an die anderen Router weitergeleitet ist, der Zeit zur Neuberechnung der Routing-Tabellen und der Zeit um die Routing-Tabellen in die Hardware zu übertragen. Um nun diese Ausfallzeit möglichst kurz zu halten, verwendet Fast-Rerouting eine schon vorab berechnete alternative Route um die ausgefallenen Komponenten zu umgehen. Dadurch kann die Ausfallzeit eines Netzknoten und damit die Verzögerung von Dateneinheiten auf die Erkennung des Fehlers am lokalen Router und das aktivieren der Alterniv-Route beschränkt werden. Eine Information der anderen Router im Netz ist nicht zwingend erforderlich. Für MPLS gibt es eine ähnlich arbeitenden Variante die in [PaSA05] beschrieben wird.

Leider kann auch mit FRR nicht sichergestellt werden, dass es aufgrund ausgefallener Links und dadurch umgeleiteten Netzwerkverkehr zu Stausitutation kommt. Wang et al. [WWMA$^+$10] definiert dazu das folgende Problem:

Definition 2 *(Routing Resiliency) Das Problem Routing Resiliency besteht darin, sowohl ein Routing r als auch ein Protection Routing p zu berechnen, so dass die maximale Link Auslastung für bis zu F Link-Fehler 100% nicht überschreitet.*

Die bisherigen Arbeiten zu diesem Thema lassen sich in zwei Gruppen unterteilen. Eine Gruppe behandelt *Routing im Fehlerfall*, während sich die andere mit *Routing für sich ändernde Netzauslastungen* beschäftigt. Das Hauptaugenmerk bei Routing im Fehlerfall liegt in der Regel auf der Minimierung Ausfallzeit einzelner Netzkomponenten. Ein häufig verwendeter Ansatz [AtZi08, LCRA$^+$07, ShBr10] ist die Vorberechnung von Alternativ-Routen und das schnell Umschalten im Fehlerfall. Dieser Ansatz bietet jedoch noch keine Möglichkeit um Stausituationen zu vermeiden oder die voraussichtliche Leistung nach einem Link-Ausfall zu bestimmen. Alternativ zur Vorberechnung können bestehende Routen auch nach jedem Fehler angepasst und optimiert werden [WXQY$^+$06]. Der Vorteil besteht in der besseren Anpassbarkeit der neuen Routen an die aktuelle Netztopologie, wodurch diese optimal ausgelastet

werden können. Der Nachteil liegt in der schlechteren Reaktionszeit, da mitunter die Reorganisation des Routings für die neue Netztopologie sehr aufwendig zu berechnen und einzurichten sein kann. Der anderer Aspekt mit dem sich die zweite Gruppe von Arbeiten befasst, ist die Tatsache, dass sich die Verkehrssituationen in Netzen wie dem Internet sehr häufig ändern und man daher ein Routing benötigt, welches dies berücksichtigt. Auch hierzu gibt es unterschiedliche Ansätze. Ein Möglichkeit ist die Optimierung von Routen auf Grundlage von zuvor beobachtetem Netzwerkverkehr [AgNB05, RoTZ03].

Mit R3 hat Wang et al. ein Verfahren vorgestellt, das beide Aspekte berücksichtigt und zudem in der Lage ist, Staufreiheit für eine Vielzahl von Fehlerszenarien zu garantieren (den Beweis führt Wang et al. ebenfalls in [WWMA+10]).

3 Resilient Routing Reconfiguration

Wie in Abschnitt 1 erwähnt, besteht ein Problem beim Entwickeln von Routing-Protection-Schemes in der Vielzahl möglicher Fehlerszenarien und der daraus resultierenden Unkenntnis der genauen Netztopologie. Für ein Netz mit $|E|$ Links und bis zu F Link-Fehlern entspricht die Anzahl möglicher Fehlerszenarien $\sum_{i=1}^{F} \binom{|E|}{i}$. Die Grundidee von R3 besteht nun darin, nicht jede dieser möglichen Topologien zu berücksichtigen, sondern das Problem in eine besser handhabbare Form zu transformieren.

Angenommen ein Link e auf einem Pfad zwischen zwei Routern a und b in einem Netz $G = (V, E)$ fällt aus, so muss das verbleibende Netz $G' = (V, E \backslash \{e\})$ den Verkehr des ausgefallenen Links zusätzlich übertragen. Diese Zusatzbelastung hängt von der aktuellen Auslastung des ausgefallenen Links ab, kann aber höchsten so groß wie die Kapazität c_e des Links sein und wird als *Virtual Demand* bezeichnet. So lässt sich jedem Link ein Virtual Demand entsprechend seiner Kapazität zuordnen. Werden alle Virtual Demands mittels Konvexkombination zusammengefasst, beschreibt dies den gesamten umzuleitenden Verkehr im Fehlerfall, unabhängig vom konkreten Fehlerszenario. Formal kann man den Virtual Demand eines Links e als Variable x_e auffassen. Für bis zu F Link-Fehler überdeckt dann die Menge

$$X_F = \{x | 0 \leq \frac{x_e}{c_e} \leq 1 (\forall e \in V), \sum_{e \in E} \frac{x_e}{c_e} \leq F\} \tag{2}$$

den gesamten umzuleitenden Verkehr für jedes beliebige Fehlerszenario. Anders ausgedrückt, für jedes mögliche Fehlerszenario mit maximal F fehlerhaften Links beschreibt X_F den gesamten umzuleitenden Verkehr. Anstelle eines Routings für eine konstante Verkehrsmatrix d bei variabler Netztopologie zu berechnen, kann dadurch ein Routing für die Ursprungstopologie bei variablem Verkehrsaufkommen $d + X_F$ berechnet werden. Beschrieben wird das Verkehrsaufkommen $d + X_F$ durch die Summe des aktuellen Verkehrs d und der Menge von Virtual Demands X_f und entspricht $d + X_f = \{d + x | x \in X_f\}$.

Noch einmal zusammengefasst bedeutet dies: Durch die Verlagerung des Problems von der unbekannten Netztopologie (aber bekanntem Verkehrsaufkommen) hin zu unbekanntem Verkehrsaufkommen (aber bekannter Netztopologie), wird das Aufzählen aller möglichen Fehlerszenarien zur Berechnung eines Routings sowie Protection Routings unnötig.

Zu beachten ist allerdings, dass im Gegensatz zur endlichen Anzahl möglicher Netztopologien die Anzahl möglicher neuer Verkehrsmatrizen $d + X_F$ unendlich ist. Dies stellt jedoch kein Problem dar, da sich die Menge dieser Verkehrsmatrizen mittels (2) als endliche Menge linearer Bedingungen formulieren lässt. Dadurch kann lineare Optimierung zur Berechnung und Optimierung eines Routings r und eines Protection Routings p verwendet werden. Die Aufzählung aller möglichen Verkehrsmatrizen ist dazu nicht nötig.

R3 nutzt nun diese Überlegungen als Grundlage. Prinzipiell gliedert sich R3 in zwei Teile, einer vom Netzbetrieb unabhängigen Vorberechnung (Offline-Phase), die sowohl ein Routing r als auch ein Protection Routing p basierend auf den obigen Überlegungen berechnet und einer Rekonfiguration (Online-Phase) von r und p nach jedem Link-Ausfall während des eigentlichen Netzbetriebs.

3.1 Vorberechnung (Offline-Phase)

Aufgabe der Vorberechnungsphase ist es ein möglichst optimales Routing r für eine gegebene Verkehrsmatrix d, sowie ein entsprechendes Protection Routing p für den Virtual Demand X_F zu berechnen. Optimierungsziel ist dabei die Minimierung der maximalen Link-Auslastung (MLU) für die ursprüngliche Netztopologie und das kombinierte Verkehrsaufkommen $d + X_F$ aus aktuellem Verkehr d und Virtual Demand X_F. Wie Eingangs schon erwähnt lässt sich dieses Problem mittels linearer Optimierung effizient lösen.

Formal entspricht das Problem dem nachfolgenden linearen Optimierungsproblem:

$$
\begin{aligned}
&\textbf{minimize}_{(r,p)} && MLU \\
&\textbf{subject to:} && \\
&\text{[C1]} && r = \{r_{a,b}(e) | a, b \in V, e \in E\} \text{ ist ein Routing nach (1)} \\
& && p = \{p_l(e) | l, e \in E\} \text{ ist ein Routing nach (1)} \\
&\text{[C2]} && \forall x \in X_F, \forall e \in E : \\
& && \frac{\sum_{a,b \in V} d_{a,b} r_{ab}(e) + \sum_{l \in E} x_l p_l(e)}{c_e} \leq MLU
\end{aligned}
\tag{3}
$$

Während [C1] sicherstellt, dass r und p die Bedingungen für ein gültiges Routing einhalten fordert [C2], dass die MLU für jeden Link und jeden Virtual Demand $x \in X_F$ nicht überschritten wird. In dieser Form des linearen Optimierungsproblems besteht weiterhin die Schwierigkeit, dass es sich bei X_F um eine nicht endliche Menge handelt und sich daraus unendlich viele Bedingungen ergeben.

Nun kann aber die Dualität von linearen Optimierungsproblemen ausgenutzt werden um das ursprüngliche Problem (3) in das nachfolgende einfachere lineare Optimierungsproblem mit polynomialer Anzahl an Bedingungen zu transformieren.

$$
\begin{aligned}
&\textbf{minimize}_{(r,p,\pi,\lambda)} && MLU \\
&\textbf{subject to:} && \\
& && r = \{r_{a,b}(e) | a, b \in V, e \in E\} \text{ ist ein Routing nach (1)} \\
& && p = \{p_l(e) | l, e \in E\} \text{ ist ein Routing nach (1)} \\
& && \forall e \; in E : \frac{\sum_{a,b \in V} d_{a,b} r_{ab}(e) + \sum_{l \in E} \pi_e(l) + \lambda_e F}{c_e} \leq MLU \\
& && \forall e, l \in E : \frac{\pi_e(l) + \lambda_e}{c_l} \geq p_l(e) \\
& && \forall e, l \in E . \pi_e(l) \geq 0 \\
& && \forall e \in E : \lambda_e \geq 0
\end{aligned}
\tag{4}
$$

Die Transformation zwischen (3) und (4) basiert vorwiegend auf der Dualität von linearen Optimierungsproblemen und wird in dieser Arbeit nicht näher behandelt. Für eine genaue Beschreibung der Transformation siehe [WWMA+10].

3.2 Rekonfiguration (Online-Phase)

Da sich im Fehlerfall, z.B. durch den Ausfall eines Links während des Betriebes die Netztopologie ändert, muss das von der Offline-Phase für das gesamte Netzwerk ermittelte Protection Routing p so angepasst werden, dass p keine ungültigen (defekten) Links verwendet.

Nachdem ein Link-Fehler auf einem Link e von einem Router erkannt wird, aktiviert dieser sofort sein Protection Routing p und leitet den Verkehr entsprechend einer Routing-Alternative ξ_e um. Da p allerdings nicht für die Topologie in diesem konkreten Fehlerfall, sondern auf der ursprünglichen Netztopologie definiert wurde, muss p zuvor entsprechend des Fehlers angepasst werden, um sicherzustellen das der defekte Link e nicht mehr verwendet wird. Erreichen lässt sich dies durch eine einfache Anpassung (re-scaling) der verbliebenen Links, die den Verkehr von e nun mittragen müssen. Die Anpassung erfolgt nach folgender Formel:

$$\xi_e(l) = \frac{p_e(l)}{1 - p_e(e)};\ \forall l \in E \backslash \{e\} \tag{5}$$

Damit wird sichergestellt, dass der gesamte Verkehr auch weitergeleitet wird und somit Routing-Bedingung [R2] in (1) gültig bleibt.

Um den fehlerhaften Link e nun generell nicht mehr zu benutzen, müssen sowohl Routing r als auch Protection Routing p angepasst werden. Dazu kann einfach der vormals an e zugeteilte Verkehr auf die Routing-Alternative ξ_e verteilt werden. Der über einen Link $l \in E\backslash\{e\}$ geroutete Verkehr $r_{ab}(l)$ erhöht sich dann um den an ihn, durch die Umleitung ξ_e zugeteilten Anteil $r_{ab}(e)\xi_e(l)$. Entsprechendes gilt auch für das Protection Routing p.

3.3 Erweiterungen

Wang et al. beschreibt in [WWMA+10] einige Erweiterungen mit denen der Standardansatz von R3 erweitert werden kann.

Eine Erweiterung ist der Umgang mit variablen Verkehrsmatrizen, anstelle der wie bisher angenommen konstanten Verkehrsmatrix d. Dazu wird die konstante Verkehrsmatrix in (2) durch die konvexe Hülle einer Menge verschiedener Verkehrsmatrizen $d_1, ..., d_n$ ersetzt. Damit wird ein größerer Bereich von Verkehrsmustern erfasst.

Die nächste Erweiterung ist die Behandlung zusammenhängender Link-Ausfälle. Link-Ausfälle treten nicht immer einzeln und unabhängig voneinander auf, sondern können in einem gewissen Zusammenhang stehen. Als Beispiel wird in [WWMA+10] der Ausfall einer, von mehreren IP-Links gemeinsam genutzten Komponente angeben, oder das gleichzeitige Abschalten mehrerer Links für Wartungsarbeiten durch den Netzbetreiber. Wang et al. unterscheidet dazu die beiden Klassen, Shared Risk Link Group (SRLG) zu der das erste Beispiel gehört und Maintenance Link Group (MLG) zu der das zweite Beispiel gehört. Eine Shared Risk Link Group besteht aus Links, die aus technischen Gründen immer gleichzeitig unterbrochen werden, da sie dieselben Komponenten nutzen, z.B. dieselbe Glasfaserleitung. Eine MLG besteht aus Links die vom Netzbetreiber zusammengefasst und bei Wartungsarbeiten zusammen abgeschaltet werden. R3 kann nun ergänzt werden um solche Fehlercharakteristiken zu berücksichtigen.

Eine weitere Möglichkeit von R3 besteht darin, unterschiedliche Verkehrsklassen unterschiedlich zu behandeln. Dazu kann für jede Klasse, die in der Regel gewissen Service Level Agreements (SLA) entspricht festgelegt werden, wie viele zeitlich überlappende Link-Ausfälle sie tolerieren kann.

Die letzten beiden vorgestellten Erweiterungen befassen sich mit dem Verhältnis zwischen Leistung im Normalbetrieb und Fehlerfall und mit dem Verhältnis zwischen Netzauslastung und Verzögerungszeit. Da bei Ausfall eines oder mehrere Links die Netzkapazität gemindert wird und der Verkehr des defekten Links zusätzlich übertragen werden muss, sind entsprechende Redundanzen in Form von Kapazitätsreserven zu berücksichtigen. Will man im Fehlerfall weiterhin eine hohe Leistung ermöglichen, muss diese Redundanz dementsprechend groß sein. Umgekehrt bedeutet dies, dass im Normalbetrieb nicht die bestmögliche Auslastung erreicht werden kann. Um die Auslastung für den regulären Betrieb zu beeinflussen bzw. zu verbessern,

wird das lineare Optimierungsproblem (4) um zusätzliche Bedingungen ergänzt, mit denen die Differenz zwischen Leistung im Normalbetrieb und dem Optimum beeinflusst werden kann. Auf ähnliche Weise kann auch die durchschnittliche Verzögerungszeit für eine Ende-zu-Ende Übertragung beeinflusst werden. Hierzu wird (4) ebenfalls durch zusätzliche Bedingungen ergänzt.

4 Implementierung

Im nachfolgenden Abschnitt soll eine mögliche Implementierung von R3, wie sie von Wange et al. in [WWMA+10] vorgeschlagen wird vorgestellt werden.

Die Grundlage von R3 bildet die flussbasierte Darstellung des Routings. Ein solches Routing-Schema wird von heutigen Routern allerdings noch nicht unterstützt, weshalb für eine Implementierung bestehende Verfahren daraufhin angepasst oder neue implementiert werden müssen. Wang et al. verwendet MPLS (Linux MPLS) als Grundlage und gib eine darauf aufbauende Erweiterung für flussbasiertes Routing an, genannt MPLS-ff.

Bei reinem MPLS wählt der Router beim Eintreffen eines Paketes anhand dessen Label das zu verwendende Interface, über welches das Paket weitergeleitet werden soll. Dazu führt der Router eine Datenstruktur in Form einer Tabelle (ILM), die für jedes Label eine entsprechende *next-hop label forwarding instruction (NHLFE)* enthält. Da für jedes Label nur ein Eintrag vorhanden ist, wird der gesamte Verkehr der mit demselben Label gekennzeichnet ist auch immer über dasselbe Interface geroutet. Um nun flussbasiertes Routing mit MPLS zu ermöglichen, muss MPLS so erweitert werden, dass der zu einem Label gehörende Datenverkehr anteilig auf mehrere Interfaces verteilt werden kann. MPLS-ff erweitert dazu die Datenstruktur so, dass sie mehrere NHLFE Einträge enthalten kann. Außerdem wird jedem NHLFE ein Faktor (*next-hop splitting ratio*) zugeordnet, anhand dessen der Router die Pakete wie in Abschnitt 3 beschrieben, auf die diesem Label zugeordneten Interfaces aufteilen kann.

Damit Pakete die zu einer TCP-Verbindung gehören nicht unnötig über unterschiedliche Wege durch das Netz geschickt werden, wodurch Laufzeitunterschiede entstehen können, verwendet Wang et al. in [WWMA+10] eine Hashfunktion um die Pakete auf die zur Verfügung stehenden Links zu verteilen. Die Hashfunktion erhält als Eingabe die Quell- und Ziel-IP, die Quell- und Ziel-Ports sowie eine ID, abhängig vom aktuellen Router. Ziel der Hash-Funktion ist es, Paketen die zur selben TCP-Verbindung gehören auf einem Router immer denselben Hash-Wert zuzuordnen. Für unterschiedliche Router kann der Wert allerdings unterschiedlich sein. So wird erreicht, dass die Pakete einer TCP-Verbindung denselben Weg nehmen.

Mithilfe der Erweiterung MPLS-ff kann nun R3 implementiert werden. Die Vorberechnung des *Protection Routing p* und die Einrichtung von MPLS-ff kann von einem zentralen Server erledigt werden. Die Rekonfiguration während des Betriebes muss jeder Router selbst durchführen. Zu Fehlererkennung während des Betriebs kann die Sicherungsschicht (Layer 2) auf Fehlerereignisse beobachtet werden. Auch andere Fehlererkennungsverfahren wie z.B. [KYGS05] sind möglich. Wird ein fehlerhafter Link erkannt, werden alle anderen Router mittels ICMP über den Fehler informiert, sodass diese ihr Protection Routing entsprechend anpassen können. Dazu muss jeder Router die *next-hop splitting ratio* für die zu dem ausgefallenen Link gehörenden Alternativ-Links anpassen.

Um Pakete im Fehlerfall umzuleiten, wird der Label-Stack von MPLS verwendet. Erkennt ein Router einen Linkfehler, wird die für das Label des Paketes definierte Alternativ-Route aktiviert. Dazu legt der Router ein weiteres Label auf den Label-Stack des Paketes und ordnet es damit der Alternativ-Route zu. Das Paket wird dann entsprechend diesem Label weitergeleitet. Erreicht ein Paket wieder die Ursprungsroute, wird das neue Label vom Label-Stack entfernt und das Paket kann auf normalem Weg weitergeleitet werden.

Beispiel 1 *Angenommen O1 und O2 wollen beide mittels einer TCP-Verbindung Daten über das Netzwerk an D1 übertragen. Sei weiter angenommen, der normale Label Switch Pfad von R1 zu R3 führt über R2, wie in Abbildung 1 gezeigt. Trifft ein Paket bei Router R1 ein, so ordnet er diesem Paket ein Label zu, z.B. das Label 100 für den Pfad R1-R2-R3 und leitet das Paket entsprechend weiter.*

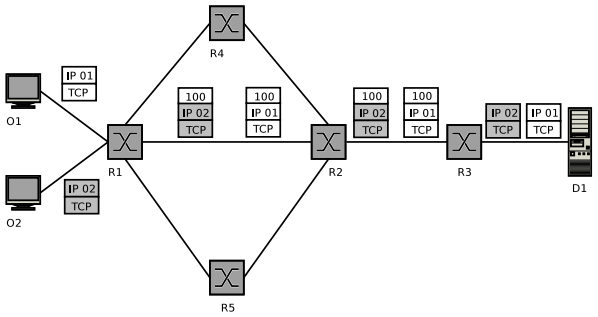

Abbildung 1: Routing ohne Link-Fehler

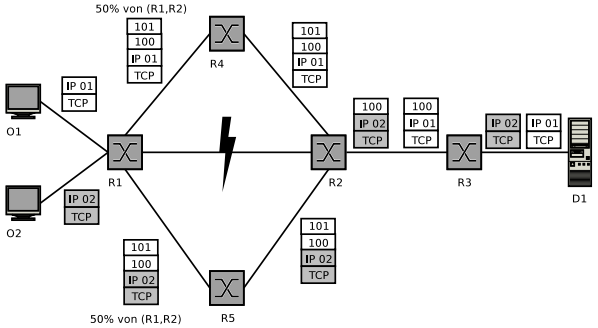

Abbildung 2: Routing im Fehlerfall

Kommt es nun zu einem Link-Ausfall zwischen R1 und R2, wie in Abbildung 2 dargestellt, so muss R1 sein Protection Routing aktivieren und die Pakete entsprechend über die Alternativ-Routen umleiten. Dazu ordnet R1 anhand seiner ILM den Paketen ein neues Protection Label, beispielsweise 101 zu und legt es auf den Label Stack der MPLS Dateneinheit. Für dieses Label existieren angenommen zwei NHLFE in der ILM von R1, die besagen, dass jeweils 50% des Verkehrs von (R1,R2) über (R1,R4) und (R1,R5) übertragen werden. Um nun Pakete für dieselbe TCP-Verbindung auch auf demselben Weg zu transportieren, berechnet R1 wie oben beschrieben einen Hashwert. Anhand dieses Wertes teilt R1 entsprechend die Pakete für O1 an den nächsten Router R4 und die Pakete von O2 an den nächsten Router R5 zu. R4 und R5 leiten die Pakte dann anhand des Labels 101 an R2 weiter, wo das oberste Label 101 wieder vom Stack entfernt und auf normalem Weg weiter zu R3 geleitet wird. Ein ähnliches Beispiel findet sich auch in [WWMA$^+$10]

5 Evaluierung

Der nachfolgende Abschnitt gibt einen Überblick über die in [WWMA$^+$10] vorgestellte Evaluierung von R3. Zur Evaluierung wurden zum einen Simulationen und zum anderen die ebenfalls in [WWMA$^+$10] vorgestellte, auf Linux MPLS basierende reale Implementierung von R3 verwendet.

Die verwendeten Netztopologien zur Simulation sind zum einen die Topologie eines Tier-1 Internet Service Providers (ISP) nachfolgend US-ISP genannt, und zum anderen die durch das Rocketfuel-Projekt [SpMW02] aufgezeichneten Netztopologien von Level-3, SBC und UUNet, sowie einer mittels GT-ITM generierten Backbonetopologie. Zur Evaluierung der experimentellen Implementierung von R3 wurde mittels Emulab die Backbone-Topologie des Abilene Netzwerks emuliert.

Netz	Knoten	Links
Abilene	11	28
Level-3	17	72
SBC	19	70
UUNet	47	336
Generated	100	460
US-ISP	-	-

Tabelle 1: Übersicht über die Netztopologien [WWMA+10]

Als mögliche Fehlerszenarien wurden alle Einzel-Linkfehler sowie alle möglichen Zwei-Link-Fehler berücksichtigt. Da die Anzahl möglicher Szenarien für mehr als zwei Fehler extrem groß wird, wurden insgesamt 1100 verschiedene Drei- und Vier-Link-Fehler berücksichtigt, die zufällig ausgewählt wurden. Gemessen wurde die Verkehrsdichte an der Stelle im Netz mit den geringsten Kapazitätsreserven. Ermittelt wurde außerdem das Verhältnis dieser gemessenen Verkehrsdichte und optimalem flussbasiertem Routing. Je näher dieser Wert an 1 liegt, desto besser ist die Leistung des verwendeten Verfahrens.

Verglichen wurden die beiden Routingverfahren OSPF und MPLS-ff im Zusammenhang mit den Protection-Verfahren Constrained Shortest Path First (CSPF) [Zieg07], OSPF reconvergence (recon) [WWMA+10], Failure-Carrying-Packet (FCP) [LCRA+07], Path Splicing (PathSplicing) [MEFV08], R3 und Flow-based optimal link detour routing (opt) [WWMA+10].

5.1 Simulation

Abbildung 3 zeigt die in [WWMA+10] ermittelten Messergebnisse der US-ISP Topologie für alle Einzel-Fehler Events, alle Zwei-Fehler Events und 1100 zufälligen Drei-Fehler Events. Angegeben ist jeweils das Leistungsverhältnis für ein Szenario. Die Werte wurden entsprechend ihres Leistungswertes aufsteigend sortiert. In allen drei Grafiken lässt sich die gleiche Gruppierung erkennen. OSPF+R3, OSPF+opt, MPLS+R3 liegen immer nahe beieinander und sind dem Optimum am nächsten. PSPF+CSPF, OSPF+recon und FCP bilden ebenfalls eine Gruppe, liegen ebenfalls eng zusammen und sind zumindest für die Extremwerte am weitesten vom Optimum entfernt. PathSplicing liegt in der Regel zwischen den beiden Gruppen. In Zahlen ausgedrückt, ist die erste Gruppe für Einzel-Fehler Events bis zu 30% schlechter gegenüber dem Optimum, PathSplicing ist bis zu 100% schlechter und die restlichen Verfahren sogar um bis zu 260%. Für Zwei-Fehler Events ist die dritte Gruppe bis zu $3,7$-mal schlechter als im Optimalfall möglich wäre und bis zu 94% über dem höchsten Wert der ersten Gruppe. Bei Drei-Fehler Events beträgt der Abstand hingegen mindestens 50%

Weiterhin interessant sind die Ergebnisse für die SBC Topologie, siehe Abbildung 4. Die Grafik zeigt ein deutlich besseres Leistungsverhältnis für MPLS-ff+R3 gegenüber allen anderen Verfahren. Nach Wang et al. liegt dieses Verhalten in der gemeinsamen Optimierung von Routing und Protection-Routing von MPLS-ff+R3 begründet und zeigt dessen Vorteile.

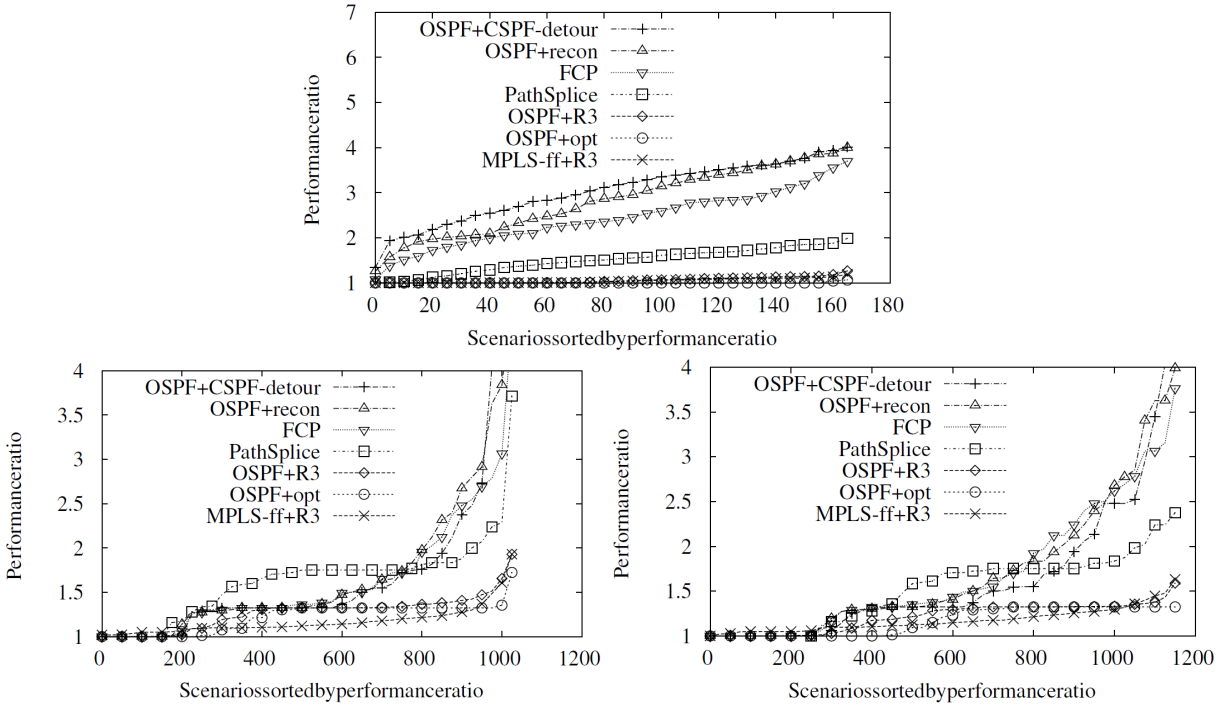

Abbildung 3: Einzel-, Zwei- und zufällige Drei-Fehler Szenarien (US-ISP)[WWMA+10]

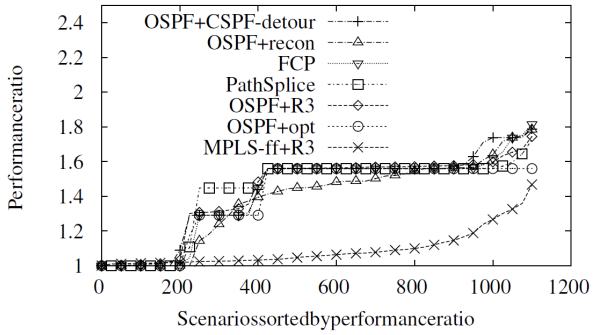

Abbildung 4: Leistungsverhältnis für Drei-Fehler Szenarien (SBC) [WWMA+10]

5.2 Reale Implementierung

Neben der Simulation des Verhaltens verschiedener Routing-Protokolle in verschiedenen Topologien wurde in [WWMA+10] auch die Leistungsfähigkeit von R3 durch eine für Linux implementierten Variante von R3 evaluiert. Bewertet wurde dabei die Zeit für die Vorberechnung (Offline-Phase) von r und p, den zusätzlichen Speicheraufwand von MPLS-ff und die Leistungsfähigkeit des Routings.

Tabelle 5.2 zeigt die ermittelten Zeiten der Offline-Phase. Gelöst wurde das zugrunde liegende lineare Optimierungsproblem (4) mit ILOG CPLEX 10.0. Die Laufzeiten nehmen mit zunehmender Knoten- und Linkanzahl zu, schwanken aber über der Anzahl der Fehler.

Eine Messung des Speicherverbrauchs ergab ein moderates Ergebnis von höchstens 267KB für die Forwarding Information Base (UUNet) und höchstens 20MB für die Routing Information Base (Generated). Zur Evaluierung der Leistungsfähigkeit von R3 wurde der Durchsatz zwischen jedem Sender-Empfänger Paar des Abilene Netzwerks, die Link-Auslastungen und die aufsummierte Verlustrate an den End-Routern gemessen. Zur Fehler-Simulation wurden drei Links nacheinander abgeschaltet. Diese erkennt man an der treppenförmig steigenden Round

Netz / Anzahl Link-Fehler	1	2	3	4	5	6
Abilene	0.30	0.30	0.30	0.32	0.33	0.29
Level-3	1.80	1.97	2.56	2.71	2.46	2.43
SBC	1.46	1.76	1.75	1.76	1.92	1.91
UUNet	1010.00	572.00	1067.00	810.00	864.00	720.00
Generated	1388.00	929.00	1971.00	2001.00	1675.00	2131.00
US-ISP	21.30	21.90	21.40	20.10	22.10	21.80

Tabelle 2: R3 Offline-Phase in Sekunden [WWMA+10]

Trip Time einer Verbindung in Abbildung 5. Eine Überlastung der Links durch R3 Rerouting konnte jedoch nicht festgestellt werden.

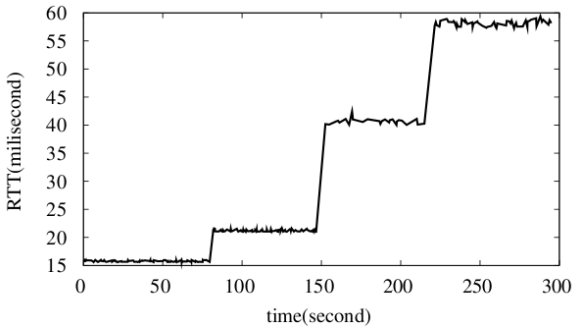

Abbildung 5: Round Trip Time (RTT) einer Verbindung

6 Zusammenfassung

In dieser Arbeit wurde das von Wang et al. in [WWMA+10] entwickelte Routing Protection Verfahren *Resilient Routing Reconfiguration* behandelt. R3 bietet die Möglichkeit sowohl Routing als auch Protection Routing zu berechnen und mittels Rekonfiguration während des Netzbetriebs auf aktuelle Fehlerereignisse zu reagieren und das Routing entsprechend anzupassen. Eine Stärke von R3 besteht darin, Staufreiheit für eine gewisse Anzahl zeitlich überlappender Fehler garantieren zu können. Den Beweis führt Wang et al. ebenfalls in [WWMA+10]. Neben der grundlegenden Arbeitsweise von R3 ist auch die von Wang et al. vorgeschlagene Implementierung auf Basis von MPLS sowie eine Zusammenfassung der Evaluationsergebnisse aus [WWMA+10] teil dieser Arbeit.

Die Arbeit von Wang et al. ist meiner Ansicht nach gut verfasst. Das Thema wird gut motiviert und die theoretischen Aspekte sind vollständig und verständlich beschrieben. Bei der Implementierung wird allerdings lediglich ausführlich auf die zweite Phase von R3, die Rekonfiguration des Netzes eingegangen. Die erste Phase kommt hier meiner Ansicht nach etwas kurz. Die Evaluierung ist wieder sehr ausführlich beschrieben.

Literatur

[AgNB05] Sharad Agarwal, Antonio Nucci und Supratik Bhattacharyya. Measuring the Shared Fate of IGP Engineering and Interdomain Traffic. In *Proceedings of the 13TH IEEE International Conference on Network Protocols*, Washington, DC, USA, 2005. IEEE Computer Society, S. 236–245.

[ApCo03] David Applegate und Edith Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, New York, NY, USA, 2003. ACM, S. 313–324.

[AtZi08] A. Atlas und A. Zinin. Basic Specification for IP Fast Reroute: Loop-Free Alternates. RFC 5286 (Proposed Standard), September 2008.

[IBTD03] Sundar Iyer, Supratik Bhattacharyya, Nina Taft und Christophe Diot. An Approach to Alleviate Link Overload as Observed on an IP Backbone, 2003.

[KYGS05] Ramana Rao Kompella, Jennifer Yates, Albert Greenberg und Alex C. Snoeren. IP fault localization via risk modeling. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, Berkeley, CA, USA, 2005. USENIX Association, S. 57–70.

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

[Malk98] G. Malkin. RIP Version 2. RFC 2453 (Standard), November 1998. Updated by RFC 4822.

[MEFV08] Murtaza Motiwala, Megan Elmore, Nick Feamster und Santosh Vempala. Path splicing. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, SIGCOMM '08, New York, NY, USA, 2008. ACM, S. 27–38.

[Moy98] J. Moy. OSPF Version 2. RFC 2328 (Standard), April 1998. Updated by RFC 5709.

[Oran90] D. Oran. OSI IS-IS Intra-domain Routing Protocol. RFC 1142 (Informational), Februar 1990.

[PaSA05] P. Pan, G. Swallow und A. Atlas. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090 (Proposed Standard), Mai 2005.

[RoTZ03] Matthew Roughan, Mikkel Thorup und Yin Zhang. Traffic engineering with estimated traffic matrices. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC '03, New York, NY, USA, 2003. ACM, S. 248–258.

[RoVC01] E. Rosen, A. Viswanathan und R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), Januar 2001.

[ShBr10] M. Shand und S. Bryant. IP Fast Reroute Framework. RFC 5714 (Informational), Januar 2010.

[SpMW02] Neil Spring, Ratul Mahajan und David Wetherall. Measuring ISP topologies with rocketfuel. *SIGCOMM Comput. Commun. Rev.* Band 32, August 2002, S. 133–145.

[WWMA+10] Ye Wang, Hao Wang, Ajay Mahimkar, Richard Alimi, Yin Zhang, Lili Qiu und Yang Richard Yang. R3: resilient routing reconfiguration. In *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, SIGCOMM '10, New York, NY, USA, 2010. ACM, S. 291–302.

[WXQY+06] Hao Wang, Haiyong Xie, Lili Qiu, Yang Richard Yang, Yin Zhang und Albert Greenberg. COPE: traffic engineering in dynamic networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, New York, NY, USA, 2006. ACM, S. 99–110.

[Zieg07] Mark Ziegelmann. *Constrained Shortest Paths and Related Problems - Constrained Network Optimization.* VDM Verlag, SaarbrÃ¼cken, Germany, Germany. 2007.