Network Routing with Path Vector Protocols: Theory and Applications

Gregor Kopf
(kopf@informatik.hu-berlin.de)

Seminar „Internet Routing“,
Technische Universität Berlin

WS 2007/2008 (Version vom 18. Januar 2008)

Zusammenfassung

Dieses Papier ist eine Zusammenfassung von [SOB 03]. Es beschreibt einen algebraischen Ansatz zur Beschreibung von Pfad-Vektor-Protokollen. Dabei werden sowohl die theoretischen Grundlagen, als auch einige mögliche Anwendungszenarien der entwickelten Algebra beschrieben. Einige der im Originalpapier geführten Beweise werden hier nicht beschrieben. Interessierte Leser seien daher auf [SOB 03] verwiesen.

1 Einleitung

Routingalgorithmen spielen eine zentrale Rolle in grossen Netzwerken wie dem Internet. Diese Protokolle werden in zwei Kategorien aufgeteilt: Distanzvektor-Protokolle (wie RIPv1/2) und Linkstate-Protokolle (wie z.B. OSPF). In diesem Kontext kann auch BGP als Distanzvektor-Protokoll gesehen werden. Dieses Papier beschäftigt sich mit Pfadvektor-Protokollen, einer Unterklasse der Distanzvektor-Protokolle und der algebraischen Beschreibung wichtiger Eigenschaften dieser. Insbesondere die Optimalität und Konvergenz von Routingprotokollen ist Thema dieses Papiers. Im folgenden werden in Abschnitt 2 einige grundlegende Begriffe definiert, die in Abschnitt 3 benutzt werden, um eine Algebra zur Beschreibung von Routingprotokollen zu definieren. In diesem Abschnitt werden auch die Konvergenz und Optimalität auf die Monotonie und Isotonie der dort definierten Algebra zurückgeführt. In Abschnitt 4 wird beschrieben, wie man den algebraischen Ansatz nutzen kann, um Aussagen über bestimmte Arten von Pfadvektor-Protokollen zu treffen. Die dort gezeigten Zusammenhänge können beim Design neuer Routingalgorithmen von großer Hilfe sein. In Paper [GS 05] wird sogar ein Ansatz vorgestellt, der es erlaubt, Routingprotokolle in einer formalen Sprache zu spezifizieren und dann automatisch ihre Eigenschaften zu überprüfen. In Abschnitt 5 wird das Thema noch einmal abschließend zusammengefasst.

2 Grundlegende Begriffe

In diesem Abschnitt werden zunächst einige grundlegende Begriffe definiert, auf die im weiteren Verlauf dieses Papiers immer wieder Bezug genommen wird.

2.1 Netzwerk und Pfad

Als erstes betrachten wir eine formale Definition der Begriffe Netzwerk und Pfad. Intuitiv sollten diese Begrifflichkeiten sofort verständlich sein. Da aber im folgenden einige Aussagen mathematisch beschrieben und bewiesen werden sollen, ist es nötig, auch diese beiden grundlegenden Begriffe einmal zu definieren:

Definition 1 (Netzwerk) *Ein Netzwerk ist ein gerichteter Graph mit der Knotenmenge V und Kantenmenge E. Ein Tupel $(u, v) \in E$ nennt man Link. Dabei ist u der sog. Kopf des Links. v nennt man einen Out-Nachbarn von u und u einen In-Nachbarn von v. Ist $(u, v) \in E$, so bedeutet das, dass Daten von u nach v fließen können, Routingnachrichten allerdings von v nach u.*

Definition 2 (Pfad) *Einen gerichteten Graphen mit der Knotenmenge $\{u_1, u_2, \ldots, u_i\}$ und der Kantenmenge $\{(u_1, u_2), \ldots, (u_{i-1}, u_i)\}$ nennt man einen Pfad von u_1 nach u_i. Ein solcher Pfad wird abkürzend mit $u_1 u_2 \cdots u_i$ dargestellt. Sei $P = u_j \cdots u_k$ ein Pfad, so bezeichnet $u_i u_j \circ P$ das Anfügen des Links (u_i, u_j) ans vordere Ende des Pfades. Dadurch entsteht dann ein neuer Pfad $u_i u_j \circ P := u_i u_j \cdots u_k$.*

2.2 Optimaler Pfad

Der Begriff „optimaler Pfad" ist ebenfalls leicht verständlich. Der optimale Pfad von einem Knoten u zu einem Knoten d ist derjenige Pfad, dessen Gewicht in der Metrik des verwendeten Routingprotokolls am kleinsten ist. Nehmen wir einmal eine einfache Hopcount-Metrik an, so wäre der Pfad 136 durch den Graphen aus Skizze 1 optimal, der Pfad 1036 jedoch nicht. In einer anderen Metrik könnte das jedoch gerade umgekehrt sein. Insbesondere sollte man nicht vergessen, dass die Metrik eines Protokolls nicht nur den einzelnen Kanten im Graphen Gewichte zuordnet, sondern gesamte Pfade bewertet. Das Gewicht eines Gesamtpfades muss dabei nicht automatisch der Summe der Gewichte seiner einzelnen Links entsprechen! Will man beispielsweise immer den Pfad wählen, der die größte Bandbreite zur Verfügung stellt (den breitesten Pfad), sind die einzelnen Kantengewichte keineswegs zu summieren, sondern vielmehr stellt ihr Minumum das Gesamtgewicht des Pfades dar.

Definition 3 (Optimaler Pfad) *Ein Pfad von u nach d heisst optimal, wenn er benutzbar ist und sein Gewicht kleiner oder gleich dem Gewicht eines beliebigen anderen Pfades von u nach d ist.*

Die Bedeutung von „benutzbar" wird in Abschnitt 3 näher präzisiert.

2.3 Konvergenz

Man nennt ein Routingprotokoll konvergent, wenn es die Eigenschaft hat, nach dem Entfernen oder Hinzufügen von Links im Netzwerk nach endlicher Zeit keine Routinginformationen mehr auszutauschen. Das Protokoll muss also irgendwann einen stabilen Zustand erreichen. Am besten lässt sich dieser Begriff durch ein Gegenbeispiel veranschaulichen, das im folgenden dargestellt wird: Wir betrachen das Netzwerk aus Skizze 2. Zum Zeitpunkt $t = 0$ mögen die beiden Knoten u und v gerade die Pfade P_u und P_v zum Zielknoten d erlernt haben. Diese Pfade seien in der Metrik des Routingprotokolls genau gleich gewichtet. Diese Information wird nun von jedem der beiden Knoten an den jeweils anderen weitergeleitet. Zum Zeitpunkt $t = 1$ ist also den beiden Knoten bekannt, dass der jeweils andere Knoten ebenfalls über einen Pfad zum Ziel d verfügt. Weiterhin ist den Knoten dann natürlich auch die Existenz ihres Nachbarn und damit auch der Link $l := (u, v) = (v, u)$ bekannt. Angenommen, die Metrik des Routingprotokolls gewichtet nun den Link l negativ (so dass durch Hinzufügen des Links l an einen

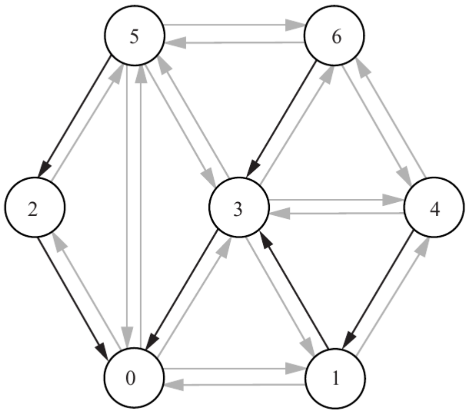

Abbildung 1: Beispielnetzwerk

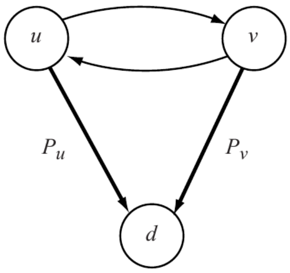

Abbildung 2: Ein Beispielnetzwerk zum Begriff Konvergenz

Pfad dessen Gewicht sinkt), so entsteht folgende Situation: Knoten u lernt, dass das Gewicht des Pfades $l \circ P_v$ kleiner ist als das von P_u. Somit ist es für ihn günstiger, die weiterzuleitenden Daten zuerst an Knoten v weiterzugeben und dann von diesem nach d transportieren zu lassen. Knoten v lernt analog, dass der kürzeste Pfad nach d nicht P_v ist, sondern über den Knoten u führt. Diese neu gewonnenen Routen mögen die Knoten dann zum Zeitpunkt $t = 2$ austauschen. Dieser Vorgang setzt sich unendlich lange fort, so dass immer weitere Routingnachrichten ausgetauscht werden. Das Protokoll konvergiert somit nicht. Selbst wenn das Gewicht der Links l null wäre, wäre die Konvergenz des Protokolls nicht sichergestellt: Durch Hinzufügen der Links l zu einem Pfad würde sich dessen Gewicht nicht ändern. Die Konvergenz des Protokolls hinge also davon ab, welchen von zwei gleich gewichteten Pfaden das Routingprotokoll bevorzugt.

3 Algebraische Beschreibung

Dieser Abschnitt widmet sich der mathematischen Formulierung der oben betrachteten Eigenschaften von Pfad-Vektor-Protokollen.

3.1 Definition der Algebra

Um das Verhalten von Routingprotokollen näher zu charakterisieren, wird im folgenden ein algebraischer Ansatz gezeigt, der ihre wichtigsten Eigenschaften zu beschreiben vermag. Auf Basis eines Netzwerks, wie es in Abschnitt 2 definiert wurde, konstruieren wir eine Algebra: Der erste Teil der Algebra beschreibt Äquivalenzklassen der Links im

Netzwerk. Diese Menge wird mit L (Labels) bezeichnet. Sie beschreibt die Sicht der Routingmetrik auf die verschiedenen Links: Die Links werden von der Routingmetrik nach bestimmten Kriterien beurteilt und dabei zerfällt die Gesamtmenge der Links in Äquivalenzklassen. Darauf aufbauend gibt es eine Menge Σ (Signaturen). Diese Menge beschreibt Äquivalenzklassen aller möglichen Pfade im Netzwerk. Je nach Protokoll kann es mehr oder weniger Signaturen geben, höchstens aber so viele, wie es Pfade durch das Netzwerk gibt. Jeder dieser Signaturen muss durch die im Routingprotokoll verwendete Metrik ein Gewicht zugeordnet werden. Die Menge der Gewichte bezeichnet man mit W. Die Abbildung der Signaturen auf die Gewichte wird durch eine Funktion $f : \Sigma \to W$ realisiert. Weiterhin betrachtet man noch eine Relation $\preceq$, die eine Totalordnung auf den Gewichten darstellt (schlißlich wählt der Routingalgorithmus ja den Pfad mit dem kleinsten Gewicht). Durch eine Operation $\oplus : L \times \Sigma \to \Sigma$ modelliert man das Hinzufügen von Knoten zu einem Pfad. Die spezielle Signatur ϕ wird als „nicht benutzbarer" Pfad bezeichnet und dient als Kennzeichen für die Abwesenheit eines Routingpfades. Insgesamt ist die Algebra ein Siebentupel $(W, \prec, L, \Sigma, \phi, \oplus, f)$. Im weiteren werden folgende Schreibweisen verwendet: ist P ein Pfad, so bezeichnet $s(P)$ seine Signatur. Ist (u, v) ein Link, so bezeichnet $l(uv)$ sein Label.

Die so beschriebene Algebra kann verschiedene Eigenschaften haben, die man folgendermaßen definiert:

Definition 4 (Maximalität) *Es gilt* $\forall \alpha \in \Sigma \setminus \{\phi\} : f(\alpha) \prec f(\phi)$.
In Worten: Das Gewicht der Signatur ϕ ist echt größer als das Gewicht jeder anderen Signatur.

Definition 5 (Absorption) *Es gilt:* $\forall l \in L : l \oplus \phi = \phi$.
In Worten: Die Signatur ϕ wird durch das Hinzufügen von Labels nicht verändert.

Man geht davon aus, dass die betrachtete Algebra immer die beiden oben definierten Eigenschaften hat. Weiterhin sind noch die folgenden Eigenschaften von besonderem Interesse:

Definition 6 (Monotonie) *Es gilt:* $\forall l \in L \forall \alpha \in \Sigma : f(\alpha) \preceq f(l \oplus \alpha)$.
In Worten: Durch das Hinzufügen eines Labels zu einer Signatur wird das Gewicht der neuen Signatur nie kleiner als das der alten.

Definition 7 (Isotonie) *Es gilt:* $\forall l \in L \forall \alpha, \beta \in \Sigma : f(\alpha) \preceq f(\beta) \Rightarrow f(l \oplus \alpha) \preceq f(l \oplus \beta)$.
In Worten: Das hinzufügen des gleichen Labels zu zwei Signaturen verändert das Gewichtsverhältnis dieser Signaturen nicht.

Im weiteren Verlauf dieses Papiers wird gezeigt werden, dass die Konvergenz und Optimalität von Routingprotokollen auf Monotonie und Isotonie der zugrunde liegenden Algebra zurückzuführen sind.

3.2 Algebraischer Nachweris der Eigenschaften von Routingprotokollen

Bei der Verwendung von Routingprotokollen in der Praxis sind vor allem Konvergenz und Optimalität interessante Eigenschaften. Dieser Abschnitt behandelt deren algebraische Formulierung inklusive der nötigen Beweise.

3.2.1 Optimalität

Die Wahl eines optimalen Pfades durch ein Netzwerk ist für ein Routingprotokoll in der Praxis eine der wichtigsten Aufgaben. Im folgenden wird gezeigt, dass die Optimalität von Routingprotokollen nur von der Monotonie und Isotonie der zugrunde liegenden Algebra abhängt: Ein Routingprotokoll wählt genau dann optimale Pfade durch das

Netzwerk, wenn die verwendete Algebra sowohl monoton als auch isoton ist. Zuerst betrachten wir dazu folgende Struktur:

Definition 8 (In-Baum) *Ein In-Baum ist ein gerichteter Graph mit den folgenden drei Eigenschaften:*

- *Genau ein Knoten (die sog. Wurzel) hat keine Out-Nachbarn*
- *Alle anderen Knoten haben genau einen Out-Nachbarn*
- *Es gibt von jedem Knoten einen Pfad zur Wurzel*

T_d *bezeichnet einen In-Baum mit der Wurzel* d*.*

Innerhalb eines In-Baumes gibt es eindeutige Pfade. Daher ist für die Optimalität eines Routingprotokolls die Wahl eines geeigneten In-Baumes von großer Wichtigkeit.

Definition 9 (Optimaler In-Baum) *Einen In-Baum mit Wurzel* d *nennt man optimal, wenn er die folgenden Bedingungen erfüllt:*

- *Für alle Knoten* u*: Wenn* u *zum In-Baum gehört, dann ist der einzige Pfad von* u *zu* d *im In-Baum ein optimaler Pfad.*
- *Für alle Knoten* u*: Wenn* u *nicht zum In-Baum gehört, dann gibt es keinen optimalen Pfad von* u *zu* d*.*

In-Bäume sind die Strukturen, die man anzutreffen erwartet, wenn man Daten allein auf Grund ihrer Zieladresse weiterleiten möchte. Die Wurzel des Baumes repräsentiert dabei das Ziel der zu routenden Pakete. Abbildung 3 zeigt einen in-Baum mit Wurzel 0.

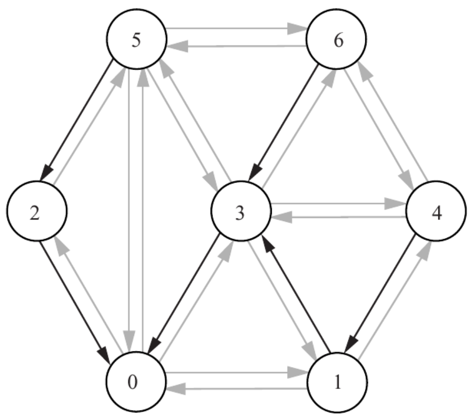

Abbildung 3: Die dunkelen Pfeile zeigen einen in-Baum mit Wurzel 0

Neben optimalen Pfaden gibt es auch noch sog. lokal optimale Pfade. Ein Pfad P kann jedoch nur lokal optimal im Bezug auf eine gegebene Pfadmenge V sein.

Definition 10 (Lokal optimaler Pfad) *Sei* P *ein Pfad von* u *nach* d *und* V *eine Menge mit Pfaden, wobei jeder Pfad in* V *bei einem out-Nachbarn von* u *beginnt und bei* d *endet. Man konstruiert eine Menge* $\overline{V}$ *wie folgt:* $\overline{V} := \{uv \circ Q : Q \in V, u \text{ ist kein Knoten von } P\}$*. Den Pfad* P *nennt man nun lokal optimal im Bezug auf* V*, wenn er ein benutzbarer Pfad aus* $\overline{V}$ *ist und sein Gewicht kleiner oder gleich dem jedes anderen Pfades in* $\overline{V}$ *ist.*

Im Unterschied zu einem optimalen Pfad, ist ein lokal optimaler Pfad im Bezug auf eine Pfadmenge V also ein Pfad, der „nur" besser (oder gleich) als alle anderen Pfade in einer Menge $\overline{V}$ ist. Dieses Konzept mag vorerst verwirrend erscheinen, ist aber doch sehr praxisnah. Ein kleines Beispiel mag das verdeutlichen: Die Menge V enthält eine Menge von vorgegebenen Pfaden; in der Praxis könnte das die Menge von Routen sein, die den Nachbarn des Knoten u bekannt sind. Auf diese Routen hat der Knoten allerdings keinen weiteren Einfluss. Also kann u die Routenwahl nur dadurch optimieren, dass er denjenigen Nachbarn auswählt, bei dem die Gesamtroute (inklusive der Strecke von u zu diesem Nachbarn) am günstigsten ist. Genau diese Vorgehensweise wird durch die Bildung der Menge $\overline{V}$ beschrieben: $\overline{V}$ enthält alle möglichen Pfade, die Knoten u zum Transport der Daten wählen könnte (wie gesagt hat Knoten u keinen Einfluss auf die bereits bekannten Pfade in V). Damit ist ein lokal optimaler Pfad im Bezug auf eine Pfadmenge V also der beste Pfad, der sich aus Sicht des Ursprungsknotens u zum Transport der Daten anbietet. Gegeben einen In-Baum T_d definiert man $V_u(T_d)$ als die Menge aller Pfade im Baum T_d, die bei einem Out-Nachbarn von u beginnen und bei d enden. Darauf aufbauend kann man einen lokal optimalen In-Baum definieren:

Definition 11 (Lokal optimaler In-Baum) *Einen In-Baum T_d nennt man lokal optimal, wenn er die folgenden Bedingungen erfüllt:*

- *Für alle Knoten u: Wenn u zu T_d gehört, dann ist der einzige Pfad von u nach d in T_d optimal im Bezug auf $V_u(T_d)$.*
- *Für alle Knoten u: Wenn u nicht zu T_d gehört, dann gibt es keinen lokal optimalen Pfad von u nach d im Bezug auf $V_u(T_d)$.*

Um die Optimalität eines Routingprotokolls auf die Monotonie und Isotonie der verwendeten Algebra zurückzuführen, beweisen wir zuerst einen Hilfssatz:

Satz 1 *Ist die Algebra sowohl monoton als auch isoton, dann gibt es für jedes Knotenpaar (u, d) einen optimalen Pfad P von u nach d, so dass jeder Teilpfad von P mit Endknoten d selbst ein optimaler Pfad ist.*

Beweis: Der Beweis wird hier nur skizziert: Angenommen, die Algebra wäre monoton und isoton. Weiterhin gebe es auf jeden Pfad von u nach d einen Knoten, so dass der Teilpfad der bei diesem Knoten beginnt nicht optimal ist. Wir betrachten den optimalen Pfad $uu_2 \cdots u_k \circ P$ von u nach d. Hierbei sei $uu_2 \cdots u_k$ der längste Teilpfad, so dass alle Pfade, die bei einem seiner Knoten beginnen und bei d enden selbst optimal sind. Demnach ist der Pfad P, der bei u_k beginnt und bei d endet kein optimaler Pfad. Offenbar ist $u_k \neq d$. Daher betrachten wir einen optimalen Pfad $u_k u_{k+1} \circ Q$ von u_k nach d. Da die Algebra monoton ist, kann kein Knoten vor u_k Teil von Q sein. Somit gibt es einen Pfad $uu_2 \cdots u_k u_{k+1} \circ Q$. Wir wissen, dass $f(s(u_k u_{k+1} \circ Q)) \prec f(s(P))$ (denn $u_k u_{k+1} \circ Q$ sollte ja optimal sein und P war nicht optimal). Aus der Isotonie der Algebra folgt dann aber für alle $1 \leq i < k$: $f(s(u_i \cdots u_k u_{k+1} \circ Q)) \preceq f(s(u_i \cdots u_k \circ P))$. Damit sind also für $1 \leq i < k$ alle Teilpfade $u_i \cdots u_k u_{k+1} \circ Q$ optimal. Per Annahme war $u_k u_{k+1} \circ Q$ auch optimal. Das widerspricht allerdings der Annahme, $uu_2 \cdots u_k \circ P$ wäre der optimale Pfad von u nach d, bei dem $uu_2 \cdots u_k$ der **längste** Teilpfad wäre, so dass alle Pfade, die bei einem seiner Knoten beginnen und bei d enden selbst optimal sind.

Mit Hilfe dieses Satzes lässt sich nun der eigentlich interessante Satz beweisen:

Satz 2 (Optimalitätssatz) *Ein Pfad-Vektor-Protokoll mit einer monotonen Algebra wählt genau dann optimale Pfade, wenn die Algebra isoton ist.*

Beweis: Wir zeigen zuerst die direkte Implikation: Angenommen, T_d sei ein lokal optimaler In-Baum, aber kein optimaler In-Baum. Dann gibt es einen Knoten u mit einem optimalen Pfad zu d, so dass entweder u nicht zu T_d gehört oder der einzige Pfad in T_d von u zu d nicht optimal ist. Sei $u_n u_{n-1} \cdots u_1$ (mit u_n = u und $u_1 = d$) ein optimaler Pfad von u nach d, so dass alle seine Teilpfade mit Ziel d ebefalls optimal sind. Die Existenz eines solchen Pfades wird durch Satz 1 sichergestellt. Für alle zu T_d gehörigen u_k sei P_k der Pfad in T_d von u_k zu d. Sei nun $i \geq 1$ der kleinste Index für den entweder u_i nicht zu T_d gehört oder P_i kein optimaler Pfad ist. Wenn P_i aber kein optimaler Pfad ist, dann ist $f(s(u_i u_{i-1} \cdots u_1)) \prec f(s(P_i))$. P_{i-1} und $u_{i-1} \cdots u_1$ sind beide optimale

Pfade, wobei P_{i-1} in $V_{u_i}(T_d)$ liegt. Wegen der Monotonie der Algebra, kann u_i nicht in P_{i-1} enthalten sein. Aus der Isotonie der Algebra folgt: $f(s(u_{i-1}\cdots u_1)) = f(s(P_{i-1})) \rightarrow f(s(u_i u_{i-1}\cdots u_1)) = f(s(u_1 u_{i-1} \circ P_{i-1}))$. Insbesondere ist $f(s(u_i u_{i-1}\cdots u_1)) \neq \phi$, womit u_i zu T_d gehören muss. Dann ist allerdings $f(s(u_i u_{i-1} \circ P_{i-1})) \prec f(s(P_i))$, was der Annahme widerspricht, dass P_i ein lokal optimaler Pfad Im Bezug auf $V_{u_i}(T_d)$ sei. Somit muss T_d also ein optimaler In-Baum sein.

Die andere Richtung der Implikation soll mit Hilfe von Abbildung 4 gezeigt werden: Wenn die Algebra nicht isoton ist, dann gibt es ein $l \in L$ und $\alpha, \beta \in \Sigma$, für die gilt:

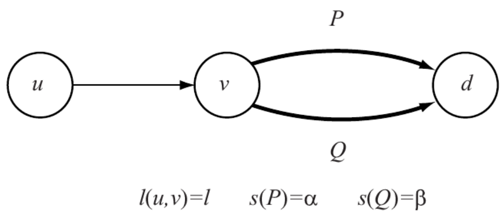

Abbildung 4: Skizze zum Beweis. Die dicken Linien stehen für Pfade, die dünnen für Links.

$f(\alpha) \preceq f(\beta) \wedge f(l \oplus \beta) \prec f(l \oplus \alpha)$. In Abbildung 4 hat der Pfad P die Signatur α, der Pfad Q die Signatur β. Der Link (u, v) hat das Label l. Der In-Baum, der den Pfad P und nicht den Pfad Q enthält, ist ein lokal optimaler In-Baum, weil $f(s(P)) = f(\alpha) \preceq f(\beta) = f(s(Q))$. Somit ist der Pfad von u zu d im lokal optimalen In-Baum gerade $uv \circ P$ (so er denn existiert). Dies ist jedoch kein optimaler Pfad von u zu d, denn $f(l \oplus \beta) = f(s(uv \circ Q)) \prec f(s(uv \circ P)) = f(l \oplus \alpha)$.

3.2.2 Konvergenz

Die Beobachtungen aus Abschnitt 2.3 legen nahe, dass die Konvergenz eines Protokolls von der Monotonie der zugrunde liegenden Algebra abhängt: Ist die Algebra nicht monoton, so kann das Hinzufügen eines Labels zu einer Signatur das Gewicht der so neu entstandenen Signatur verringern. Dadurch ist es möglich, dass das Protokoll immer wieder vermeintlich neue Routen findet. Doch selbst eine monotone Algebra garantiert nicht für die Konvergenz des Protokolls: Ist das Gewicht eines Links null, so hängt die Konvergenz des Protokolls davon ab, welcher von zwei gleichwertigen Pfaden bevorzugt wird. In freien Netzwerken ist allerdings dies jedoch irrelevant. Dieser Zusammenhang soll hier nicht bewiesen werden. Für den Beweis wird auf das Originalpapier [SOB 03] verwiesen.

Definition 12 (Freies Netzwerk) *Ein Netzwerk heisst frei, wenn für alle Kreise $u_n, \ldots, u_1, u_0$ mit $u_0 = u_n$ gilt: $\forall w \in W \setminus \{f(\phi)\}\ \exists i : 0 < i \leq n \forall \alpha \in \Sigma : f(\alpha) = w \rightarrow f(l(u_i, u_{i-1}) \oplus \alpha) \neq w$.*
In Worten: Ein Netzwerk heisst frei, wenn es in jedem Kreis ein Label gibt, an den man jede beliebige Signatur α (ausser ϕ) mit Gewicht w anhängen kann und die so neu entstehende Signatur nicht das Gewicht w hat.

Zusammen mit Monotonie bedeutet Freiheit, dass das Durchlaufen eines Kreises das Gewicht des durchlaufenen Pfades erhöht. Wenn die Algebra streng monoton ist, dann ist offenbar jedes Netzwerk frei.

Satz 3 (Konvergenzsatz) *Ein Routingprotokoll konvergiert dann, wenn die zugrunde liegende Algebra monoton ist und das Netzwerk frei ist oder das Routingprotokoll zwischen zwei gleich gewichteten Pfaden immer den mit der kleineren Zahl an Links bevorzugt.*

Anmerkung: Ist eine Algebra streng monoton, so ist das Netzwerk immer frei.

Tabelle 1: Beispielalgebren

W	$\oplus$	ϕ	$\preceq$	Optimaler Pfad
$\mathbb{R}^+ \cup \{\infty\}$	$+$	∞	$\leq$	kürzester
$\mathbb{R}^+ \cup \{\infty\}$	min	0	$\geq$	breitester
$[0,1]$	$\cdot$	1	$\geq$	zuverlässigster
$\{(d,b) \mid d \in \mathbb{R}^+, b \in \mathbb{R}^+ \cup \{\infty\}\} \cup \{\phi\}$	$(d_1 + d_2, \min(b_1, b_2))$	ϕ	$d_1 < d_2 \vee d_1 = d_2 \wedge b_1 \geq b_2$	breitester kürzester

4 Anwendungen der Algebra

Im folgenden werden einige praktische Anwendungen des oben definierten algebraischen Ansatzes gezeigt.

4.1 Praktische Beispiele

Tabelle 1 zeigt einige bekannte Routingstrategien und die entsprechende Belegung der Mengen, Operationen und Relationen in der Algebra.

In der ersten Zeile wird normales Shortest-Path-Routing beschrieben. Die zweite Zeile entspricht Widest-Path-Routing: Hier wird derjenige Pfad bevorzugt, der die meiste Kapazität bietet; die Kapazität eines Pfades entspricht dabei dem Minimum der Kapazitäten seiner Links. Der zur dritten Zeile gehörige Routingalgorithmus wählt bevorzugt zuverlässige Pfade; die Zuverlässigkeit eines Pfades ist hier definiert als das Produkt der Nicht-Ausfall-Wahrscheinlichkeiten seiner einzelnen Links. Die vierte Zeile modelliert eine Kombination der Strategien die in den ersten beiden Zeilen dargestellt wurden. Hier wählt der Routingalgorithmus denjenigen Pfad, aus der Menge der kürzesten Pfade, der die größte Kapazität hat. In allen Fällen ist die Algebra sowohl monoton als auch isoton. Das Routingprotokoll wählt also optimale Pfade. Um die Frage der Konvergenz der Protokolle zu klären, untersuchen wir zuerst die Freiheit der Netzwerke: Im Fall von Shortest-Path-Routing ist ein Netzwerk genau dann frei, wenn in jedem Kreis mindestens ein Link existiert, der länger als null ist. Gehen wir davon aus, dass jeder Link im Netzwerk ein Gewicht größer null hat, so konvergiert das Routingprotokoll in jedem Fall.

Beim Widest-Path-Routing wird jedoch ein Netzwerk allein durch die Anwesenheit eines Kreises unfrei: Das mehrmalige Durchlaufen eines Kreises wird in der Metrik des Routingprotokolls nicht anders bewertet werden als das einmalige Durchlaufen. Hier kommt es also darauf an, welcher von mehreren gleichgewichteten Pfaden bevorzugt wird. Bevorzugt das Protokoll den Pfad mit dem kleinsten Hopcount, so konvergiert es.

Routingprotokolle, die zuverlässige Pfade bevorzugen, konvergieren dann, wenn jeder Kreis im Netzwerk mindestens einen Link enthält, dessen Ausfallwahrscheinlichkeit von 0 verschieden ist. Kann diese Eigenschaft nicht sichergestellt werden, so sollte das Protokoll unter gleich zuverlässigen Pfaden denjenigen mit dem geringsten Hopcount bevorzugen.

Für das zusammengesetzte Verfahren aus der vierten Zeile gilt bzgl. Konvergenz genau das, was für Shortest-Path-Routing gilt.

4.2 Überprüfung der Konvergenz eines Protokolls

Möchte man für ein gegebenes Routingprotokoll überprüfen, ob es zu optimalen Pfaden konvergiert, so genügt, es die Monotonie und Isotonie der zugrunde liegenden Algebra sowie die Freiheit des zugrundeliegenden Netzwerks zu zeigen. In vielen Fällen ist es natürlich möglich, direkt ein mathematisches Argument anzugeben, das diese Eigenschaften bestätigt. Betrachtet man beispielsweise die erste Zeile aus Tabelle 1, so sieht man schnell, dass die Algebra monoton ist, weil die Gewichte auf positive Zahlen und 0 beschränkt sind und die $\oplus$-Operation gerade der gewöhnlichen Addition entspricht. Die Isotonie dieser Algebra ist ebenso offensichtlich. Doch selbst in einem beliebig komplizierten Fall lassen sich Monotonie und Isotonie einer Algebra durch „probieren" überprüfen: Um beispielsweise auf Monotonie zu testen, berechnet man für jede Signatur α und jedes Label l das Gewicht von $l \circ \alpha$ und vergleicht es mit dem Gewicht von α. Isotonie lässt sich analog überprüfen. Die Zeitkomplexität dieser naiven Algorithmen ist polynomiell in der Zahl der Links und der Zahl der Signaturen. Im Gegensatz dazu benötigen kombinatorische Ansätze zum Nachweis der Konvergenz im Allgemeinen eine Rechenzeit, die polynomiell zur Anzahl der Pfade im Netz, also exponentiell zu dessen Größe wächst.

Zum Nachweis der Freiheit eines Netzwerks kann man folgende Konstruktion betrachten: Man bildet zu jedem Gewicht w eine Menge L_w, in der all diejenigen Labels l vertreten sind, für die es mindestens eine Signatur α gibt, so dass $f(l \oplus \alpha) = w$ für alle $l \in L_w$ gilt. Besteht ein Kreis im Netzwerk dann nur aus Links, die alle ein Label aus einer bestimmten Menge L_w tragen, so ist das Netzwerk nicht frei; denn dann gibt es die Signatur α, deren Gewicht sich beim Durchlaufen des Kreises nicht ändert. Anders herum: Gibt es keinen Kreis, dessen Links ausschließlich Labels aus einem Bestimmten L_w haben, so ist das Netzwerk frei. Zur Bestimmung der Mengen L_w lässt sich folgender Algorithmus angeben:

Algorithmus 1 (Labelmengen)

1. *for* α *in* $\Sigma \setminus \{\phi\}$ *do begin*
2. $w := f(\alpha)$
3. $L_w := \emptyset$
4. *for* l *in* L *do begin*
5. *if* $w == f(l \oplus \alpha)$ *then* $L_w \leftarrow l$
6. *end*
7. *end*

Es ist hierbei wichtig zu beachten, dass die Menge der Signaturen nicht gleich der Menge aller möglichen Pfade durch das Netzwerk sein muss, sondern vielmehr diese Pfade in Äquivalenzklassen unterteilt. Im folgenden Abschnitt wird dies noch einmal verdeutlicht.

4.3 Customer-Provider & Peer-Peer Beziehungen

Aus wirtschaftlichen Gründen verwendet man zwischen verschiedenen autonomen Systemen im Internet oft policybasiertes Routing. Kunden bezahlen ihre Provider dafür, Pakete über sie austauschen zu dürfen. Daher sind die Provider bemüht, den Kunden zum Austausch möglichst vieler Pakete über ihre Router zu bringen. Zwei autonome Systeme können aber nicht nur in einem Customer-Provider-Verhältnis stehen, sondern auch sogenannte Peers sein. Das bedeutet, dass die beiden Systeme direkt über einen Link kommunizieren und sich die dafür anfallenden Kosten teilen. Um seine Kunden gut erreichbar zu machen und zum Transport vieler Pakete zu animieren, werden Provider also ihren Kunden alle Routinginformationen mitteilen, die ihnen bekannt sind. Kunden sind allerdings bemüht, ihre Kosten zu minimieren. Ein autonomes System kann durchaus Kunde mehrerer Provider sein und wird seinen Providern zwar die Routen zu seinen eventuellen Kunden mitteilen, jedoch keinesfalls die Routen zu seinen anderen Providern oder Peers. Würde ein Autonomes System dies tun, so würde es bei seinem Provider dafür bezahlen müssen, dass dieser über das System selbst Pakete mit einem anderen Provider oder einem Peer des Systems austauscht. Einen Link, der von einem System zu seinem Provider führt, nennt man Provider-Link. Einen der vom Provider zum Kunden führt Customer-Link. Einen Link von einem Peer zu einem anderen nennt man Peer-Link. Einen Pfad, der mit einem Customer-Link beginnt, nennt man Customer-Pfad. Provider-Pfad und Peer-Pfad sind analog definiert. In 5 ist beispielsweise der Link von Knoten 5 zu Knoten 2 ein Customer-Link, während der von Knoten 2 zu Knoten 5 ein Provider-Link ist. Knoten 2 und Knoten 3 sind über Peer-Links miteinander verbunden.

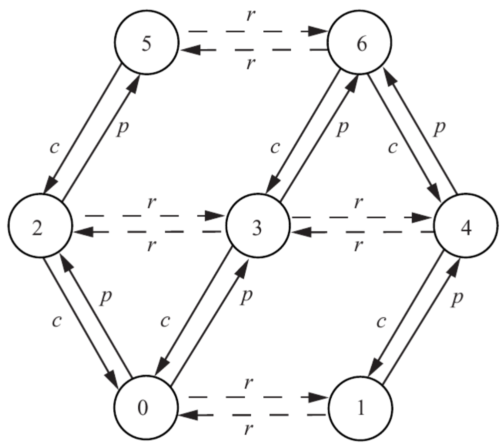

Abbildung 5: Beispiel für policy based routing

Eine algebraische Formulierung dieses Szenarios sieht wie folgt aus: $L = \{c, r, p\}$, $\Sigma = L \cup \{\varepsilon, \phi\}$, $W = \{0, 1, 2, \infty\}$. Die Relation $\prec$ ist in diesem Fall die normale $\leq$-Relation. Damit gibt es also die Labels c (Customer), p (Provider) und r (Peer). Als Signaturen stehen c (Customer-Pfad), p (Provider-Pfad), r (Peer-Pfad), ε (Trivialer Pfad, ein Pfad der nur aus einem einzigen Knoten besteht) und ϕ (unbenutzbarer Pfad) zur Vergfügung. Die Operation $\oplus$ ist wie folgt definiert:

		Signatur			
	$\oplus$	ε	c	r	p
Label	c	c	c	ϕ	ϕ
	r	r	r	ϕ	ϕ
	p	p	p	p	p

Hierbei bedeutet $c \oplus r = \phi$ beispielsweise, dass ein Peer-Pfad nicht zu einem Customer-Pfad erweitert werden kann (dass also ein Provider nicht die Peer-Pfade seiner Kunden benutzen kann). Die Funktion f definiert man folgendermaßen:

$$f(\varepsilon) = 0$$
$$f(c) = 1$$
$$f(r) = f(p) = 2$$
$$f(\phi) = \infty$$

Knoten werden also bemüht sein, zuerst Customer-Pfade (also die Pfade, die zu ihren Kunden führen) zu verwenden, so dass die Kunden dafür zahlen müssen. Die Monotonie dieser Algebra ist darin begründet, dass die $\oplus$-Operation eine Provider-Signatur nie auf eine Customer-Signatur und eine Customer-Signatur nie auf eine triviale Signatur abbildet. Damit wird das Gewicht einer Signatur durch Anwenden der $\oplus$-Operation also nie kleiner. Ebenso leicht sieht man die Isotonie der Algebra: Durch Verbinden (im Sinne von $\oplus$) eines Labels mit zwei verschiedenen Signaturen wird deren Gewichtsverhältnis nicht verändert.

Um Die Konvergenz eines solchen Policybasieren Routings nachzuweisen, untersuchen wir jetzt die Freiheit des Netzwerks. Der In Abschnitt 4.2 vorgestellte Algorithmus zum Nachweis von Freiheit erzeugt die folgenden Labelmengen:

- $L_0 = \emptyset$, da für jedes Label l gilt: $0 = f(\varepsilon) \prec f(l \oplus \varepsilon)$.
- $L_1 = \{c\}$, da $1 = f(c) = f(c \oplus c)$.
- $L_2 = \{p\}$, da $2 = f(r) = f(p \oplus r) = f(p = f(p \oplus p)$.

Somit ist ein freies Netzwerk in diesem Sinne ein Netzwerk, in dem kein Kreis nur aus Links besteht, die alle das Label c oder alle das Label p haben. Das bedeutet, dass kein System ein Provider eines seiner direkten oder Indirekten Provider sein darf. Will man das Verhältnis zwischen den Systemen nicht auf diese Art einschränken, genügt es, zu fordern, dass das Routingprotokoll im Fall von zwei gleichgewichteten Pfaden den kürzeren bevorzugt. Im BGP kommen hierzu Local Preferences zum Einsatz.

5 Zusammenfassung

In diesem Papier wurde ein algebraischer Ansatz zur Analyse der Eigenschaften von Pfadvektor-Protokollen geliefert, der sich in der Praxis vielfältig einsetzen lässt. Nicht nur die Konvergenz und Optimalität von einfachen Routingstrategien wie Shortest-Path, sondern auch Themen wir policybasiertes Routing können mit Hilfe dieses Ansatzes behandelt werden. Insbesondere ist der Nachweis dieser Eigenschaften mit Hilfe der vorgestellten Algebra leichter durchzuführen als mit den bisher verwendeten kombinatorischen Verfahren. Einen Ausblick auf ein interessantes Anwendungsgebiet dieser theoretischen Arbeit bietet [GS 05]. Hier wird ein Ansatz beschrieben, wie Routingprotokolle in einer formalen Sprache formuliert und dann automatisch auf ihre Eigenschaften hin untersucht werden können.

Literatur

[SOB 03] João Luís Sobrinho. Network Routing with Path Vector Protocols: Theory and Applications. SIGCOMM'03, August 25-29, 2003, Karlsruhe, Germany.

[GS 05] Timothy G. Griffin, João Luís Sobrinho. Metarouting. SIGCOMM'05, August 21-26, 2005, Philadelphia, Pennsylvania, USA.