Van Jacobsons Content Centric Networks

Christoph Schindlbeck
Betreuer: Dipl. Inform. Holger Kinkelin, Dipl. Inform. Marc Fouquet
Seminar Innovative Internettechnologien und Mobilkommunikation SS2010
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: schindlc@in.tum.de

KURZFASSUNG

Der Datenverkehr in Netzwerken wandelt sich immer mehr in eine Richtung, die die Daten selbst in der Vordergrund stellt. Dies steht im Gegensatz dazu, dass Informationen nach wie vor nicht über ihren Inhalt, sondern über ihren Speicherort angesprochen werden. Content-Centric Networks (CCN) machen sich genau dies zur Aufgabe und wollen über die „Adressierung“ von Daten anstatt Speicherorten in IP-Netzen diverse Probleme beseitigen. So sollen neben dem Beheben des Adressierungsproblems auch die verfügbaren Bandbreiten besser genutzt und die Sicherheit durch konsequenze Signierung erhöht werden und langfristig ein Ersatz des IP-Protokolls ermöglicht werden.

Schlüsselworte

CCN, Jacobson, Netzwerktechnik, Sicherheit

1. EINLEITUNG

Das Internet in seiner heutigen Form basiert auf Funktionsprinzipien, die in den 60er und 70er Jahren des letzten Jahrhunderts entstanden sind: Damals dienten Netzwerke dazu, um begrenzt vorhanden Ressourcen wie Bandspeichergeräte oder die Rechenleistung von Computern auf viele Nutzer zu verteilen. Daraus ist das Modell der IP-Kommunikation entstanden, das genau zwei Teilnehmer mit individuellen Kennzeichnungen (Adressen) auf einem weitgehend statischen Weg verbindet.[1] Einer der beiden will eine Ressource nutzen und fordert diese beim anderen an. Auf diesem Modell beruht bis heute nahezu der gesamte Datenverkehr im Internet.

Doch das Nutzungsverhalten von Computernetzen hat sich seitdem grundlegend verändert. Durch den starken Preisverfall und die damit verbundene massenhafte Verbreitung von technischen Geräten ist die Aufgabe von Netzwerken heute weniger das Bereitstellen von physischen Kapazitäten, als das Anbieten von bestimmten Inhalten. Diese Daten werden von einer ständig wachsenden Zahl von unterschiedlichen Geräten angeboten, so dass deren Organisation einen immer größeren Aufwand bedeutet. Wenn ein Internetnutzer zum Beispiel ein Video auf Youtube ansehen will, interessiert er sich nicht dafür, wo dieses gespeichert ist. Doch um an die gewünschten Daten zu kommen, muss der Anfragende dazu wissen, *wo* er danach suchen muss, anstatt nach dem *Inhalt selbst* zu fragen.

Daraus ergeben sich mehrere kritische Punkte, die im Umgang mit Daten beachtet werden müssen:

- **Verfügbarkeit:** Um einen schnellen, zuverlässigen Datenverkehr sicherzustellen, werden derzeit spezialisierte, den anwendungsspezifischen Gegebenheiten angepasste Infrastrukturen wie Peer-to-Peer-Netze oder Content Delivery Networks (z.B. Akamai) genutzt.
- **Sicherheit:** Die Datenintegrität wird bisher in erster Linie danach beurteilt, wo die Daten herkommen und wie die Verbindung zum Sender gesichert ist.
- **Ortsabhängigkeit:** Daten liegen an festen Stellen. Um auf sie zugreifen zu können, muss zuerst ein Zusammenhang zwischen den Inhalten und ihrem Speicherort gezogen werden (in der Regel durch das Domain Name System DNS), was viele Ressourcen verschlingt.

Dazu kommt, dass viele Datenpakete, wie etwas beliebte Videos auf Youtube, große Downloads oder über das Internet verteilte TV-Sendungen, die alle nur auf wenigen Servern vorgehalten werden, vielfach die selben Wege durch das Netz zurücklegen. Dadurch werden bei jeder Nutzung die Server des Hauptverteilers beansprucht und zudem fließt der Datenverkehr vielfach auf den gleichen Wegen.

Um diese Probleme anzugehen, hat ein Team um den Netzwerkspezialisten Van Jacobson am Palo Alto Research Center ein völlig neues Konzept der Vernetzung von Rechnern entwickelt: Die Content Centric Networks (CCN)[6]. Hierbei ist der Ansatz, Daten nicht nach ihrem Speicherort zu adressieren, sondern ihnen Namen zu geben, die sich dann ansprechen lassen.

Das CCN-Protokoll ist dabei nicht ein weiteres Protokoll, das auf dem Internetprotokoll aufbaut, sondern soll dieses weitgehend ersetzen. Um dies möglich zu machen, hat Jacobson bei der Entwicklung einen ähnlichen Ansatz verfolgt, wie beim Entwurf des IP. So soll gewährleistet werden, dass beide Protokolle koexistieren können und ein Umstieg fließend erfolgen kann. Im Gegensatz zum relativ statischen IP kann ein CCN aber sehr leicht von der Verwendung mehrerer Übertragungswege profitieren.

[1] Loadbalancer und andere Faktoren können zwar den Weg von Paketen beeinflussen, aber häufig nehmen Pakete, die zum selben Datenstrom gehören, denselben Weg. Zudem kommt es in der Regel nicht zu parallelen Verbindungen zwischen zwei Kommunikationspartnern.

Im folgenden soll zunächst der grundlegende technische Aufbau von CCNs erläutert werden, mit der Protokollstruktur sowie Erklärungen zum Datentransport und Routingprotokollen. Anschließend folgt ein Abschnitt zur Datensicherheit in CCNs. Zuletzt soll das Konzept noch kritisch betrachtet und Probleme des Entwurfs festgestellt sowie die Aussichten auf Erfolg betrachtet werden.

2. DAS CCN-PROTOKOLL

Die gesamte Kommunikation in einem CCN besteht aus zwei verschiedenen Typen von Paketen: Zum einen gibt es Interest-Pakete, die signalisieren, dass ein Teilnehmer bestimmte Daten sucht. Zudem gibt es noch die eigentlichen Datenpakete, die zum jeweiligen Interest passen. Diese Daten müssen nicht im Vorhinein existieren, sondern können auch dynamisch nach Anfrage erzeugt werden, wie etwa Webseiten mit veränderlichen Inhalten. Wenn ein Datenpaket verschickt wird, zehrt es alle zugehörigen Interest-Pakete auf dem Weg zum Anfragenden auf. Der Aufbau der Pakete wird in Abbildung 1 skizziert.

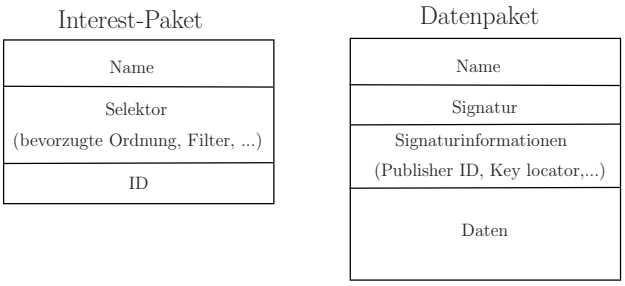

Abbildung 1: Aufbau der Pakete

2.1 Namen

Beide Paketarten sind durch einen gleichlautenden Namen eindeutig festgelegt. Die Erzeugung dieser Namen ist ein wesentliches Bestandteil des Konzepts. Wie beim IP (Netz, Subnetz, Host) basiert es auf hierarchischen Strukturen. Dadurch lassen sich die Inhalte leicht in Form von Baumstrukturen verwalten. Ein Beispiel hierfür findet sich in Abbildung 2.

Suchen können daher schnell und effizient durch ein Vergleichen der Namenspräfixe realisiert werden. Ist der angeforderte Inhalt noch nicht verfügbar, so kann er auch erst nach Empfangen eines Interest-Pakets generiert werden. Zudem können die Namen auch vom Kontext abhängen. Beispielsweise könnte der Name `/Nachrichten/heute` die aktuellen Nachrichten liefern.

Die vom CCN-Protokoll verwendeten Namen müssen keinerlei Formansprüchen genügen. Insbesondere müssen sie weder menschenlesbar sein, noch irgendeine Form auf Protokollebene besitzen. Auch die Länge der Namensteile spielt keine Rolle, genauso wie die Hierarchieebenen. Es ist lediglich wichtig, dass die jeweiligen Präfixe soweit bekannt sind, dass Interests an die richtigen Stellen weitergeleitet werden können. Die Form muss lediglich so gewählt sein, dass Protokolle auf höherer Ebene sie verstehen können. Dies ist ein entscheidender Beitrag zur Datensicherheit (siehe Kapitel 4).

2.2 Verwalten von Anfragen

Das grundlegende Funktionsprinzip ähnelt dem des bekannten Internetprotokolls: Pakete, die über eine Schnittstelle eingehen, werden analysiert und anschließend wird, abhängig vom Ergebnis der Analyse, eine Aktion ausgeführt. Es gibt dabei folgende drei Strukturen:

- Forwarding Information Base (FIB)
- Inhaltsspeicher (ein Puffer)
- Pending Interest Table (PIT)

Kann ein Interest nicht direkt beantwortet werden, so enthält die FIB die bekannten Quellen, die möglicherweise eine passende Antwort bereitstellen können. Dies erfolgt ähnlich wie beim IP, allerdings mit dem Unterschied, dass die ausgehenden Anfragen über verschiedene Netzwerkschnittstellen erfolgen können. Dadurch kann ein CCN von mehreren Verbindungen profitieren.

Der Puffer verfolgt eine andere Strategie wie beim IP: Werden die Datenpakete dort nur so lange gespeichert, bis sie weiter verschickt werden, so passiert in einem CCN das Gegenteil: Sämtliche Datenpakete bleiben auch nach dem Verschicken an den Interessenten weiter gespeichert. Erst wenn der Speicher voll ist, werden die Pakete gelöscht, die am längsten nicht mehr nachgefragt wurden (Least Recently Used-Ersetzung, LRU).

In der PIT werden die noch nicht beantworteten Interest-Pakete verwaltet. In CCNs werden nur die Interests geroutet. Datenpakete, die als Antwort darauf verschickt werden, verfolgen den Weg abschnittsweise entlang den PIT-Einträgen zurück. Wann immer ein Datenpaket auf einem Zwischenknoten zur Quelle eines entsprechenden Interest weitergeleitet wird, wird dessen Eintrag in der PIT gelöscht, da die Nachfrage beanwortet wurde. Erfolgt auf einen Interest keine Antwort, so kann der Eintrag in der PIT auch durch einen Timeout entfernt werden.

Geht ein Interest über eine Netzwerkschnittstelle ein, so wird zunächst eine Suche nach der längsten Übereinstimmung des Namens mit Einträgen in drei Speicherstrukturen durchgeführt. Findet sich ein identischer Eintrag im Pufferspeicher, so wird das Antwortpaket verschickt. Passiert das nicht, so wird zunächst verglichen, ob in der PIT bereits ein gleichlautender Name enthalten ist. Ansonsten wird das Paket über die FIB weitergeleitet, falls dort Hinweise zu finden sind, welche anderen Quellen befragt werden könnten und das Interest in der PIT abgespeichert. Enthält auch die FIB keinen übereinstimmenden Präfix, so kann die Anfrage als letzter Ausweg noch an die lokale Broadcastdomain weitergeleitet werden, bevor die Suche aufgegeben wird.

Wenn sich in der PIT bereits ein identisches Interest befindet, so wird lediglich der Sender der neuen Anfrage als zusätzliches Ziel hinzugefügt und das neue Interest verworfen.

Datenpakete können sehr einfach behandelt werden. Existiert ein entsprechender Eintrag in der PIT, so wird das Paket an die vermerkten Ziele weitergeleitet und der PIT-Eintrag gelöscht. Gibt es ein Paket dieses Namens im Pufferspeicher, so handelt es sich um ein Duplikat, das verworfen werden kann. Findet sich lediglich ein Ergebnis in der FIB, so ist das Paket nicht angefordert worden und kann ebenfalls vernichtet werden.

3. DATENVERKEHR

CCN ist darauf ausgelegt, unregelmäßige und unzuverlässige Verbindungen zu nutzen. Interest- und Datenpakete können daher auf dem Weg zu ihrem Ziel verloren gehen. Wird ein Interest nicht innerhalb eines gewissen Zeitrahmens befriedigt, so muss sich der Anfragende Teilnehmer selbst darum kümmern, das Paket neu anzufordern.

3.1 Datentransport

Da in einem CCN Anfragen über verschiedene Übertragungswege verschickt werden können, kann es vorkommen, dass ein Interest einen Knoten auf verschiedenen Wegen erreicht oder im Kreis läuft. Um zu verhindern, dass ein Antwortpaket auf beiden Wegen zurückgeschickt wird, enthalten Interestpakete einen zufällig generierten Erkennungswert. Kommt ein zweites Interest mit einem identischen solchen Wert an, so wird es verworfen.

Bei Datenpaketen ist dies nicht notwendig, da doppelte Pakete wie beschrieben verworfen werden. Somit können sie auch nicht im Kreis laufen.

3.1.1 Sicherstellen des Datenflusses

Auf ein Interest kommt im CCN höchstens ein Datenpaket. Ist dessen Größe nicht ausreichend, um die gesamten Daten zu transportieren, so muss ein neuer Interest gesendet werden, der den nächsten Teil in Relation zum ersten benennt. Dadurch gibt es ein Gleichgewicht zwischen Frage und Antwort. Wie beim TCP ist es allerdings möglich, mehrere Interests zu verschicken, bevor eine Antwort auf das erste kommt. Dadurch kann der Anfragende steuern, wie schnell er Daten empfangen kann. Geht ein Paket bei der Übertragung verloren, so kommt es nicht zu Stauungen, da alle weiteren Interest- und Datenpakete davon unabhängig sind.

Kommt es bei TCP zu Paketverlusten, so wird von den Endknoten die Größe des Übertragungsfensters dynamisch geändert, da davon ausgegangen wird, dass ein dazwischen liegender Router die Pakete aufgrund von Überlastung verworfen hat. Diese Problem wird von CCNs anders angegangen. Dort ist jeder einzelne Knoten im Netz selbst dafür verantwortlich, wie er mit hohen Lasten umgeht. Durch das LRU-Ersetzungsprinzip bleiben häufig benötigte Pakete im Netz, anstatt dass wie bei TCP alle Pakete eine FIFO-Queue durchlaufen müssen. So kommt es seltener zu Verspätungen beim Datenfluss.

3.1.2 Sequenzierung

Wie bei TCP ist es auch in einem CCN notwendig zu spezifizieren, welches Datenpaket der Empfänger als nächstes haben will. Dafür gibt es bei TCP das ACK-Feld im Paketheader: Sie geben die Sequenznummer des nächsten gewünschten Pakets an. Durch das Namensschema bei der Benennung von CCN-Paketen ist hier ein Ansatz, der auf einem reinen Zahlensystem beruht, nicht möglich.

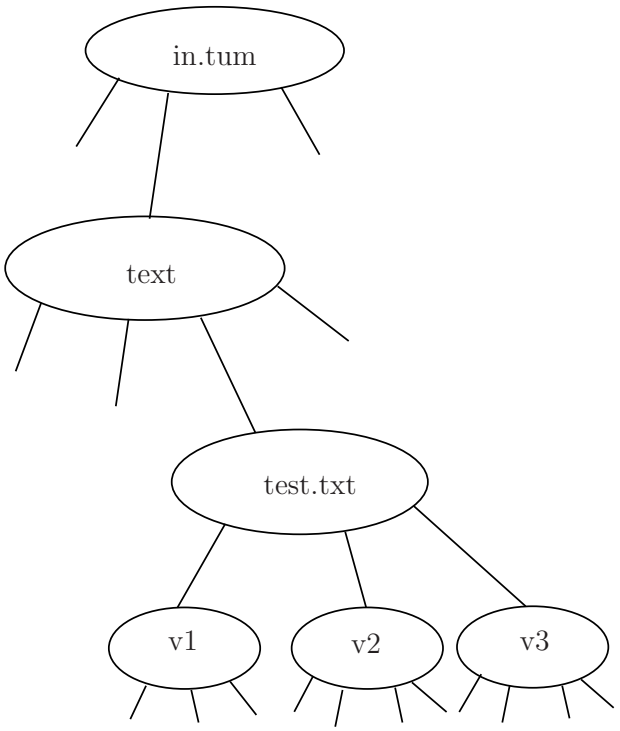

Abbildung 2: Hierarchisches Namensschema

Stattdessen wird hier die hierarchische Struktur beim Aufbau der Namen ausgenutzt. Auch wenn die Namen eine Bedeutung auf höherer Ebene haben, so ist es für den Transport der Daten irrelevant, wie dieser Name tatsächlich aussieht. Aus Sicherheitsgründen kann die Benennung sogar verschlüsselt sein.

Wenn der vollständige Name eines Pakets nicht bekannt ist, so kann der Name in Relation zu einem Vorgängerpaket angegeben werden. Durch die Traversierung des Namens anhand einer Baumstruktur ist es etwa möglich, den ersten Kindknoten einer Ebene anzufordern, oder dessen rechten Nachbar. Ein Beispiel (siehe Abbildung 2) wäre es, von einer Datei `in.tum/text/test.txt` die aktuellste (von einer unbekannten Anzahl) Version anzufordern, indem man ein Interest mit dem Namen `in.tum/text/test.txt/rechtesKind` versendet. Darauf würde als Antwort das erste Paket der aktuellsten Version folgen.

Auch wenn das eigentliche Aufbauschema dieser Namen nicht direkt im Protokoll spezifiziert wird, sollen sich hier noch Standards entwickeln, die zu einem leicht umsetzbaren und transparenten System führen, um Inhalte anzusprechen.

3.1.3 Nutzung verschiedener Netzwerkschnittstellen

CCN-Pakete können sich, wie bereits beschrieben, nicht im Kreis bewegen. Daher können, im Gegensatz zu IP, mehrere Netzwerkschnittstellen gleichzeitig (beispielsweise UMTS

neben einer WLAN-Verbindung) ohne zusätzlichen Aufwand benutzt werden. Dies soll eine Reihe von Vorteilen mit sich bringen: So soll neben einer höheren Geschwindigkeit auch eine höhere Stabilität der Verbindung und erweiterte Mobilität erreicht werden. Erreicht man beispielsweise mit einem mobilen Gerät die Grenzen des vom bisher verwendeten WLAN abgedeckten Bereichs, so etwa kann eine Datenübertragung etwa über UMTS nahtlos über die verbleibende Verbindung fortgesetzt werden.

Darüber hinaus sollen alle an einer Übertragung beteiligten Knoten durch das Erhalten von Paketen mit gleichen Namenspräfixen über verschiedene Schnittstellen Informationen erhalten, was die „beste“ Schnittstelle ist, um Interests mit diesem Präfix weiterzuleiten. Diese Präferenzen für bestimmte Schnittstellen werden in der FIB für jeden Präfix einzeln gespeichert. Je nach gewählter Strategie können einzelne Interests dann wahlweise nur über den „besten“ Weg oder über mehrere weitergeleitet werden. Zudem werden für die einzelnen Anschlüsse auch Attribute vorgehalten wie *isContentRouter* oder *BroadcastCapable*.

Zusammen werden diese Informationen als *Strategy Layer* des CCN-Protokolls bezeichnet. Die Testimplementierung von Jacobson [3] wählt in der Standardeinstellung die Strategie, Interests zuerst an alle Broadcast-fähigen Schnittstellen zu senden und dann, falls nötig, alle weiteren sequentiell durchzuprobieren. So soll erreicht werden, dass Daten zuerst im lokalen Netz gesucht werden und nur die Interests, die so nicht erfüllt werden können, über Router weitergeleitet werden.

Schnittstelleneinträge in einer FIB kommen auf verschiedene Weisen zustande. Wird eine Datenquelle mit einem CCN verbunden, so führt sie eine *register*-Operation aus, um den verbundenen Knoten mitzuteilen, zu welchen Präfixen Inhalte bereitgehalten werden. Die verbundenen Knoten speichern dies dann in ihrer FIB ab. Zusätzlich wird in einem Flag gespeichert, ob diese Informationen auch weiterverbreitet werden sollen. Dies kann durch sogenannte *announcement agents* geschehen, die die lokale FIB nach solchen Präfixen durchsuchen. Übertragen werden können diese Informationen wieder über CCN oder etwa auch über IP.

3.2 Routing

Das Routing von CCNs soll problemlos über die vom IP bekannten Strukturen möglich sein. Jacobson sieht sein Protokoll als idealen Weg, geroutete Informationen zu vermitteln. Auch das soll ein Hilfsmittel sein, CCN-Infrastruktur neben herkömmlichen IP-Netzen aufzubauen und diese schrittweise abzulösen.

3.2.1 Lokales Routing

Beim Intra-Domain Routing sind etwa das OSPF- [1] und das IS-IS-Protokoll [2] verbreitet. Bei IS-IS werden adjazente Knoten in Abhängigkeit ihrer MAC-Adresse (OSI-Layer 2 [4]) verwaltet, aber Inhalte über IP-Präfixe auf OSI-Layer 3 angesprochen. CCN soll sich sehr ähnlich wie IP darauf verhalten. Auch wenn die Struktur der angesprochenen Namenspräfixe bei CCN sich deutlich von ihren Pendants bei IP unterscheiden, soll durch Ausnutzung eines *type label value (TLV)*-Feldes sichergestellt werden, dass CCN-fähige Router problemlos mit bestehenden Netzen zusammenarbeiten. Die Spezifikation sieht vor, dass unbekannte TLV-Wert ignoriert werden, wodurch die Kompatibilität gewährleistet ist.

Durch das Konzept werden in einem CCN prinzipbedingt sehr gute Werte für benötigte Bandbreiten und Latenzzeiten erreicht, da seltener gleiche Daten die gleichen Strecken zurücklegen müssen.

Ein unterschiedliches Verhalten von IP und CCN beim Routing gibt es, wenn ein Router mehrere unterschiedliche Teilnehmer kennt, die einen bestimmten Namenspräfix auflösen können. Da bei IP Daten im Kreis laufen können und daher Routing in Form von Spannbäumen erfolgen muss, kann nur ein einziger Knoten angesprochen werden, um Pakete weiterzuleiten. Dagegen leitet ein CCN das Interest an alle möglichen Stellen weiter. Dies liegt daran, dass in einem CCN nicht der gesamte Datenbestand mit einem bestimmten Präfix auch über eine einheitliche Adresse erreichbar ist. Ermöglicht wird das durch die Eigenschaft, dass sich Pakete nicht im Kreis bewegen können.

Auch wenn CCNs keine Spannbäume für Routingeinträge berechnen müssen, sind die Routinginformationen vollständig. Um die Funktionalität herzustellen, sollen lediglich Änderungen der jeweiligen Implementierungen der Routingprotokolle nötig sein, nicht jedoch Änderungen an den Protokollen selbst.[2]

3.2.2 Netzübergreifendes Routing

Kämem bei einer gewissen Anzahl von Kunden eines Internetproviders CCNs zum Einsatz, läge es für den ISP nahe, dies auch zu unterstützen. Dadurch könnte der durchgehende Datenverkehr bei häufig abgerufenen Daten deutlich reduziert werden.

Allerdings existieren bisher noch keine konkreten Wege, wie der Datenverkehr in diesen Dimensionen geroutet werden könnte. Insbesondere was den Datenaustausch zwischen zwei CCN-fähigen Domains betrifft, wenn zwischen diesen ein Router, der dazu nicht in der Lage ist betrieben wird, ist noch ein Problem. Zwar kann man CCN-Pakete etwa auch über einen UDP-Tunnel versenden, aber das wirft Probleme auf und widerspricht auch dem Grundprinzip des CCN.

4. SICHERHEIT

Das Sicherheitskonzept von CCNs richtet wie das Netzwerk selbst den Fokus auf die Daten. Daher werden nicht (wie etwa beim Einsatz von https in IP-Netzen) die Verbindungen zwischen zwei Teilnehmern gesichert (was im Widerspruch dazu stehen würde, dass Daten über mehrere Wege übertragen werden und sogar von mehreren Quellen kommen können), sondern die Daten selbst. Da Daten nicht nur auf bestimmten Servern bereit gestellt sind, sondern auch Duplikate an anderen Positionen im Netz existieren, besteht eine erhöhte Gefahr, dass diese manipuliert werden. Daher werden grundsätzlich alle Daten mit Zertifikaten validiert, um sicherzustellen, dass sie auch aus der richtigen Quelle stammen. Darüber hinaus können die Daten und sogar, wie

[2]Laut Jacobson gilt dies allerdings nur für Link-State-Protokolle wie IS-IS oder OSPF, nicht jedoch für Protokolle, die auf Distanzvektoren beruhen wie RIP.

bereits beschrieben, die Namen selbst auch noch verschlüsselt werden.

4.1 Validierung

Bei der Signierung der Datenpakete werden neben den Daten auch die Namen selbst mit einbezogen. So soll sichergestellt werden, dass die Inhalte und deren Namen zusammenpassen. Außerdem wird es dadurch unnötig, die Namen zusätzlich mit der Signatur der Daten zu verschlüsseln, wie es bei anderen Validierungsmethoden erforderlich ist.

Die Signaturen selbst basieren auf den gewöhnlichen Public-Key-Strukturen, so dass jeder Knoten im Netzwerk die erhaltenen Daten auch überprüfen kann. Die Signaturart kann der Veröffentlichende je nach Einsatzzweck und benötigter Sicherheit selbst festlegen. Jedes CCN-Datenpaket enthält Informationen, wie der zum Validieren nötige Schlüssel zu erhalten ist. Allerdings ist keineswegs gewährleistet, dass Daten mit einem bisher unbekannten Schlüssel auch tatsächlich von der Person veröffentlicht wurden, die als Quelle benannt wird. Die Signatur stellt lediglich sicher, dass alle Pakete ursprünglich von derselben Person stammen und lässt so einen Schluss auf die Konsistenz zu.

4.2 Trust Management

Das Konzept der mit den Inhalten in die Signatur einfließenden Namen bringt einen sehr nützlichen Nebeneffekt mit sich: Wenn ein Public Key selbst auf diese Weise in ein Datenpaket gesteckt wird, entsteht dadurch ein Zertifikat. So kann auf einfache Art eine Infrastruktur geschaffen werden, bei der Inhalte andere Inhalte validieren.

Neben dieser Struktur eines Netzwerks von vertrauenswürdigen Quellen können auch weiter herkömmliche Modelle wie die bekannte PKI verwendet werden oder neue, an CCNs angepasste. Ein gut zur CCN-Struktur passendes Modell ist das von SDSI/SPKI. Hierbei wird es ermöglicht, dass Schlüssel aus einem gemeinsamen *namespace* sich gegenseitig authentifizieren können. Beispielsweise könnte ein Unternehmen seine Angestellten validieren oder umgekehrt, von einem vertrauenswürdigen Mitarbeiterschlüssel auf seine Kollegen und das Unternehmen geschlossen werden.

Ein weiterer Sicherheitsmechanismus ist das Verfolgen von Vertrauenswegen. So kann man von bereits bekannten vertrauenswürdigen Quellen, die eine andere Quelle identifizieren, auf deren Vertrauenswürdigkeit schließen.

Hat ein Client noch keine Informationen über den Public Key zu angeforderten Daten, so lässt sich dieser immer anhand des Datenpakets finden. Dort befinden sich im Header Felder, anhand derer der Urheber zu indentifizieren ist und wo sich dessen Public Key finden lässt.

4.3 Verschlüsselung

Da Verschlüsselung das einzige Mittel ist, um in einem CCN eine Zugangskontrolle durchzuführen, ist das Protokoll in Bezug auf Verschlüsselung von Daten sehr flexibel. Es können alle gängigen Verfahren angewendet werden. Da auch die Namen der Pakete mitverschlüsselt werden können, ist es auch möglich, Schlüssel zur Dekodierung über ein CCN zu verschicken.

4.4 Netzwerksicherheit

Viele heute gängige Angriffsmethoden auf Netzwerke lassen sich in einem CCN nicht oder nur schwer durchführen. Da die Kommunikation darauf basiert, *über Daten* zu sprechen, kann auf diese Weise nicht direkt *zu einem bestimmten Host* gesprochen werden. Daher können nur schwer bösartige Pakete gezielt an bestimmte Teilnehmer verschickt werden. Ein effizienter Angriff sollte daher eher ein Denial of Service-Schema verfolgen, indem Inhalte „versteckt“ werden (d.h. nicht weitergereicht) oder das Verhindern der Zustellung von Paketen durch Überfluten von Knoten mit nutzlosen Paketen.

Durch die Signierung von Paketen und eine überprüfbare Kette von Public Keys sind Nutzer grundsätzlich in einem gewissen Umfang vor Angriffen durch manipulierte Daten geschützt.

Blinde DoS-Attacken durch Bombardieren eines Knotens mit Datenpaketen werden in CCNs durch das Gleichgewicht zwischen Interest- und Datenpaketen verhindert. Der Datenverkehr verteilt sich dabei gleichmäßig auf mehrere Teilnehmer, die Daten aber nur nach einem vorhandenen Interest weiterleiten. Ist dieses nicht vorhanden, so wird das Pakete auch nicht weitergeleitet.

Eine mögliche Angriffsart wäre das Überfluten eines Ziels mit Interests. Um das zu erreichen, müsste allerdings eine sehr große Zahl an Interests generiert werden, die alle einen individuellen Namen besitzen. Ansonsten kämen Duplikate nie beim Ziel an und würden nur eine zusätzliche Last für den Angreifer bewirken. Diesen Angriffen stehen zwei Dinge entgegen: Zum einen können Router selbstständig die Menge an Interests reduzieren, die an eine Domain weitergeleitet werden, wenn auf diese keine Antwort folgt[3]. Zum anderen kann ein Teilnehmer auch selbst verbundene Router bitten, die Menge an Anfragen mit einem bestimmten Namenspräfix zu reduzieren.

Nicht zuletzt existiert auch noch die Möglichkeit, den Lauf von Daten zu beeinflussen. So kann verlangt werden, dass Daten nur über Router verschickt werden, die deren Integrität mittels Public Key des Erstellers überprüfen.

5. PRAKTISCHE UMSETZUNG & KRITIK

Van Jacobson und sein Team am PARC haben eine Testimplementierung des CCN-Protokolls erstellt [3]. Diese ist lauffähig, hat aber noch diverse Einschränkungen.

5.1 Ergebnisse

Bei Geschwindigkeitsmessungen schneidet CCN noch etwas langsamer ab als eine TCP-Verbindung mit den selben übertragenen Daten. Dies ist zum Teil auch der Tatsache geschuldet, dass Daten für Tests in UDP-Pakete verpackt werden müssen, um über bestehende Strukturen geroutet werden zu können [6].

Wenn mehrfach gleiche Daten über ein Netzwerk verschickt werden, steigt der Aufwand in einem IP-Netz linear mit der Anzahl der Abfragen. Bei der Testimplementierung eines

[3]Bei einer solchen Zahl an zufällig generierten Interests gibt es sehr wahrscheinlich nur wenige Antworten

CCN hat sich wie erwartet gezeigt, dass für den ursprünglichen Sender der Aufwand dagegen konstant der gleiche ist, wie bei einer einzigen Anfrage.

Neben weiteren Versuchen beinhalten diese Tests auch eine Implementierung eines Voice-over-CCN-Protokolls, das im Rahmen der Tests gut funktioniert und es ermöglicht, trotz wechselnder Verbindungen ein Telefongespräch ohne Paketverluste aufrecht zu erhalten [5].

5.2 Kritik

Trotz der erfolgreichen Testimplementierung gibt es aber auch Kritik, sowohl am Konzept der CCNs, als auch an deren Umsetzung. Wie bereits in Abschnitt 3.2.2 angesprochen, gibt es noch keine Implementierung für das Routing auf Providerebene. Jacobson lässt in seiner Ausarbeitung des Konzepts offen, wie es funktionieren soll und beschreibt lediglich vage Möglichkeiten.

Das Konzept eines Content Centric Network sieht vor, dass lediglich Daten direkt angesprochen werden, nicht deren Speicherort. Allerdings widerspricht sich die Umsetzung dabei selbst. Denn um bestimmte Daten zuverlässig zu erhalten, genügt es nicht, einen Knoten im Netzwerk anzufragen, der Informationen zu Daten mit einem bestimmten Präfix hat. Theoretisch kann jeder einzelne Rechner im Netz die gewünschten Informationen bereithalten. Da allerdings nicht jeder per Broadcast gefragt werden kann, kann der Suchende die Daten möglicherweise nie erhalten, weil er keine Informationen darüber hat, über welche Schnittstelle er danach fragen kann und auch keinen anderen Knoten kennt, der mehr darüber weiß. Daher spricht Jacobson immer wieder von einer Aufteilung in Domains, die aber der Idee widerspricht, dass Daten unabhängig von bestimmten Orten auffindbar sein sollen.

Ein weiteres Problem, das im Konzept nicht ausreichend betrachtet wird, ist die Konsistenz der Daten in großen Netzwerken. So ist es durchaus möglich, dass unterschiedliche Inhalte mit identischen Namen auftauchen. Zum Teil kann dieses Problem zwar durch die Signierung der Daten behoben werden (so wird sichergestellt, dass nur berechtigte Personen gültige Daten zu bestimmten Präfixen anbieten können), allerdings ist nicht sichergestellt, dass geänderte Daten an alle Knoten durchdringen, die Informationen zum entsprechenden Präfix liefern können. So kann beispielsweise nicht gewährleistet werden, dass die Daten, die über eine bestimmte Verbindung empfangen wurden, auch tatsächlich der aktuellsten verfügbaren Version entsprechen. Und wenn etwa über verschiedene Schnittstellen gleichzeitig mehrere Interests zu Teilen der aktuellsten Version einer Datei angefordert werden, dann kann es passieren, dass durch unterschiedlichen Versionsstand eines der Kontakte die erhaltenen Daten letztlich völlig unbrauchbar sind. Bestimmte dynamisch erzeugte Daten sollten auch niemals aus dem Cache eines bestimmten Netzwerkknotens

Die größte Schwäche am Konzept hat aber mit der praktischen Realisierbarkeit im großen Rahmen zu tun. Gelingt es noch, in kleinem Maßstab ein funktionierendes Netz aufzubauen mit einigen wenigen verbundenen Rechnern und einem überschaubaren Maß an Daten, so wirft dies auf Internetebene große Probleme auf: Die beteiligten Knoten müssen große Mengen an Speicherplatz darauf verwenden, um die benötigten Datenpuffer zu realisieren. Zudem wird gerade an den großen Internetknotenpunkten wie etwa den Enden von Interkontinentalkabeln die Anzahl der benötigten Einträge in der FIB und der PIT unüberschaubar groß.

6. FAZIT

Das Konzept eines inhaltszentrierten Netzwerks ist an sich eine gute Idee. Immerhin ist es für Anwender in aller Regel irrelevant, woher die angeforderten Daten kommen, solange sie geliefert werden. Durch ein CCN kann in der Theorie sowohl die Geschwindigkeit als auch die Sicherheit im Netz deutlich steigen. Allerdings wirft die Implementierung außerhalb der überschaubaren, lokalen Broadcastbereiche unüberwindbare Hindernisse auf. Gerade die großen Knotenpunkte des Internet könnten den notwendigen Aufwand nicht bewältigen, was das Konzept zum Scheitern verurteilt, bevor es wirklich eingesetzt wurde.

7. LITERATUR

[1] IETF. RFC 2328 - OSPF Version 2.

[2] IEFT. RFC 3787 - Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS).

[3] Project CCNx. http://www.ccnx.org, 2010.

[4] ISO/IEC 7498-1. *Open Systems Interconnection: Basic Reference Model: The Basic Model*.

[5] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, J. D. Thornton, and R. L. Braynard. Voccn: Voice-over content-centric networks. *ReArch'09*, 2009.

[6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. *CoNEXT'09*.