

SpoVNet – Spontaneous Virtual Networks

Fabian Stahnke

Betreuer: Heiko Niedermayer

Seminar Innovative Internet Technologien und Mobilkommunikation WS 2009/2010

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: fabian.stahnke@in.tum.de

Kurzfassung

Diese Arbeit befasst sich mit dem Projekt “SpoVNet” und seinem Ansatz, eine flexible Plattform für Anwendungen in einem komplexen Netzwerk zu schaffen. Speziell wird auf die Komponente CLIO eingegangen, welche durch Cross-Layer Informationen die Performance für durch SpoVNet verbundene Anwendungen verbessern soll.

Schlüsselworte

SpoVNet, Future Internet, Overlay, Virtual Network.

1. Motivation des Projektes

Seit der Entstehung des Internets sind die Ansprüche an die verwendeten Technologien stark gewachsen. Anforderungen wie Sicherheit, Zuverlässigkeit, Leistungsfähigkeit, Skalierbarkeit und Mobilität wurden immer wichtiger und das Thema „Future Internet“ ein relevantes Thema in der Forschung.

Aus evolutionären Ansätzen gingen bereits einige Erweiterungen wie Mobile IP (Mobilität) und IPSec (Sicherheit) hervor, jedoch dürfen solche Weiterentwicklungen die Funktionalität des bestehenden Internets nicht stören, sodass größere Entwicklungsschritte schwierig sind. Anwendungen, welche auf bestehende Dienste zurückgreifen, müssten bei jeder Veränderung dieser Dienste angepasst werden. Auch ist die Implementierung von Anwendungen, welche über unterschiedliche Netzwerktechnologien miteinander kommunizieren sollen, nicht einfach und bietet viele Fehlerquellen. [1]

Hier setzt das Projekt SpoVNet an. SpoVNet wurde gegründet im Hinblick auf die zukünftige Integration von bereits bestehenden, aber auf Anwendungsebene implementierten Diensten, in zukünftige Netzwerktechnologien. Der Ansatz sieht vor, dass eine Abstraktionsschicht geschaffen wird, welche die transparente Verwendung von neu implementierten Diensten neben bereits bestehenden Diensten ermöglicht, ohne dass Anwendungen im Application Layer dafür angepasst werden müssen. Darüber hinaus soll die Kommunikation zwischen Anwendungen durch flexible, sich selbst organisierende virtuelle Netzwerke erleichtert werden. [2]

2. Die Overlay-Architektur

Ein Overlay-Netzwerk erstellt eine logische Netzstruktur auf einem bereits bestehenden Netzwerk (Underlay). Dadurch ist es unter anderem möglich, einen vom Underlay unabhängigen Adressraum zur Verfügung zu stellen, eigene Routing-Protokolle zu verwenden und vom Underlay nicht unterstützte Netzwerkdienste zur Verfügung zu stellen. Bereits bestehende

Overlay-Netzwerke wie P2P-Netzwerke oder VPNs sind jedoch bisher statisch und teilweise umständlich zu konfigurieren. Ändern sich die Anforderungen an das Netzwerk, muss von Hand das gewünschte neue Overlay-Netzwerk aufgebaut werden. Um dies zu vereinfachen, bietet SpoVNet die Möglichkeit, das aufgebaute Overlay-Netzwerk „spontan“ an veränderte Bedingungen anzupassen. Ein Projekt, was aus SpoVNet hervorgegangen ist, heißt „ariba“. Es ist die erste Implementierung der Overlay-Architektur und der Abstraktionsschichten. Im Folgenden beschriebene konkrete Abläufe beziehen sich auf ariba.

2.1 Aufbau und Beitritt einer SpoVNet-Instanz

Eine Anwendung, welche die Vernetzung mit anderen Anwendungen benötigt, fordert beim SpoVNet-Kern ein neues Netzwerk an. Dieser erstellt einen SpoVNet-Knoten (Initiator), welcher dann eine SpoVNet-Instanz initiiert und das sogenannte Base-Overlay aufbaut. Im Base-Overlay werden vom Initiator die Eigenschaften der SpoVNet-Instanz festgelegt, wozu die SpoVNet-ID, Kryptografiefunktionen und Authentifizierungsbedingungen gehören.

Die SpoVNet-ID gibt die Zugehörigkeit zum Netzwerk an. Da ein Routing über verschiedene SpoVNet-Instanzen hinweg nicht vorgesehen ist, ist eine globale Eindeutigkeit nicht erforderlich. Um nur dem Initiator Änderungen an der bestehenden Instanz zu erlauben, wird vorgeschlagen, eine verschlüsselte ID zu verwenden. Der Private-Key würde dem Initiator eine Änderung der Parameter der SpoVNet-Instanz erlauben.

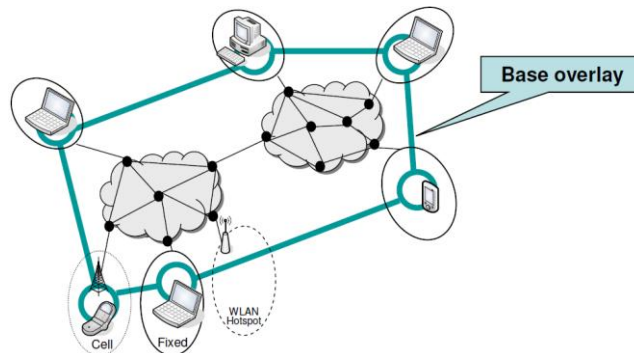


Abbildung 1 [2]

Will ein weiterer Knoten der SpoVNet-Instanz beitreten (Join Phase), muss er sich, je nach Authentifizierungsvorgaben der Instanz, authentifizieren. Ist der Knoten dazu berechtigt, dem Netzwerk beizutreten, geschieht die Authorisierung entweder durch den Initiator (centralized) oder durch einen berechtigten Knoten (decentralized authorization). Der Knoten integriert sich dann ins Base-Overlay, wodurch alle Knoten des Netzwerkes über das Base-Overlay miteinander verbunden sind. Jeder Knoten erhält zudem eine eindeutige Knoten-ID, welche ebenfalls verschlüsselt ist. Somit wird das Spoofing von Knoten-Ids verhindert.

Der Beitritt eines Knotens läuft schrittweise ab. Der Knoten kontaktiert als erstes einen Knoten, mit dem er eine direkte Verbindung aufbauen kann. Nun baut er die Verbindung zu seinen im Overlay benachbarten Knoten auf, indem eine Anfrage über das Key-Based-Routing des Overlays gesendet wird. Ist eine direkte Verbindung zum Nachbarknoten möglich, wird diese aufgebaut. Andernfalls wird der Overlapfad, welcher durch die Anfrage entstanden ist, als Verbindung verwendet. Die Knoten auf diesem Pfad werden dann als Relay verwendet.

Jeder Knoten kann von maximal einer Anwendung verwendet werden. Dabei darf eine Anwendung allerdings mehrere Knoten verwenden, um sich mit mehreren SpoVNet-Instanzen zu verbinden.

Das Basis-Overlay bietet Basisfunktionalitäten, die für den Aufbau, Betrieb und Erhalt der SpoVNet-Instanz erforderlich sind. Allen voran natürlich die Adressauflösung, welche den Lokator des Underlays auf die Knoten-ID im Overlay abbildet. Desweiteren wird das Routing über die Knoten-IDs vom Base-Overlay unterstützt. [3]

2.2 Verwendung von Netzwerkdiensten

Benötigt eine Anwendung in einer SpoVNet-Instanz einen speziellen Netzwerkdienst (bspw. Multicast), wird ein weiteres Overlay aufgesetzt, das sogenannte Service-Overlay. Im Gegensatz zum Base-Overlay müssen hier jedoch nicht alle Teilnehmer der SpoVNet-Instanz integriert sein. (Abbildung 2)

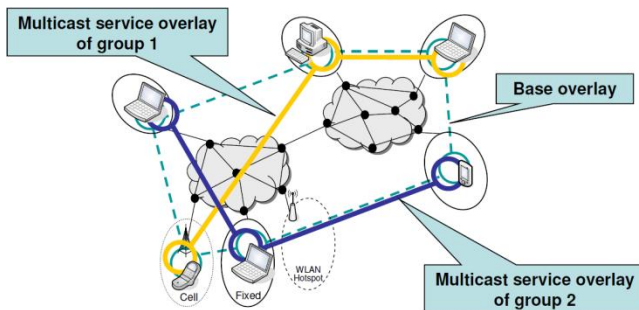


Abbildung 2 [2]

Möchte zum Beispiel ein Messenger einen Konferenz-Chat zwischen drei Teilnehmern des Messenger-Netzwerkes initiieren, wird ein Service-Overlay für Multicast zwischen den drei Teilnehmern erstellt. Über die Base-Communication-Komponente wird abhängig von den Anforderungen an die Verbindung ein geeignetes Underlay-Protokoll und ein entsprechendes Routing verwendet, um so die erforderliche Verbindung aufzubauen. [2]

2.3 Aufrechterhaltung der Verbindung

Ändert sich aus Mobilitätsgründen der Lokator eines Knotens, brechen Overlayverbindungen ab. (Abbildung 3) Durch eine teilweise erneute Durchführung der Join Phase wird ein neuer Relaypath gefunden und die Verbindung automatisch wieder aufgebaut. (Abbildung 4)

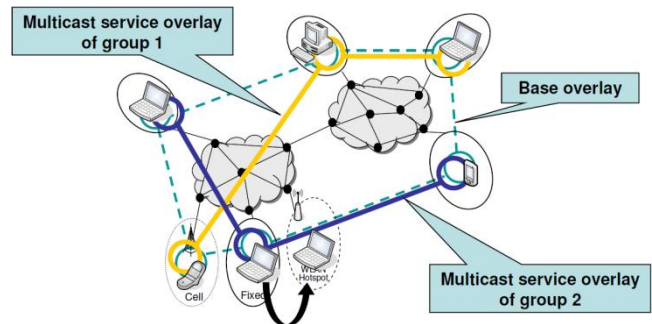


Abbildung 3 [2]

Um sofort zu erkennen, ob eine direkte Verbindung zwischen zwei Knoten möglich ist, verwendet man die Erweiterung „Connectivity Domain management“. Die Erweiterung teilt alle Verbindungen in sogenannte „Connectivity Domains“ ein. Alle Knoten in einer Domain können eine direkte Verbindung zueinander aufbauen. Durch den „Connectivity Domain Identifier“ (CDID), der für jeden Knoten in einer Domain gleich ist, können andere Knoten sofort erkennen, ob eine direkte Verbindung zu diesem Knoten möglich ist. [3]

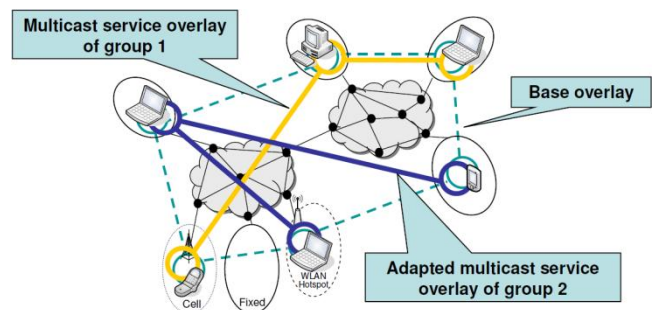


Abbildung 4 [2]

3. Die Abstraktionsschichten

Um eine transparente Nutzung von Netzwerkdiensten und Underlay-Verbindungen zu gewährleisten, müssen Abstraktionsschichten erstellt werden, die die Nutzung durch Anwendungen vereinheitlichen.

3.1 Service Abstraction

Die transparente Nutzung von Diensten wird durch die Service Abstraction gewährleistet. Eine Anwendung sieht nur, welche Dienste zur Verfügung stehen. Ob dieser Dienst aber als

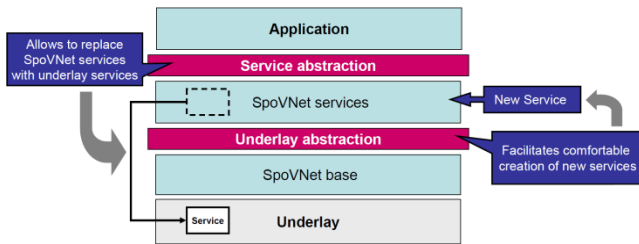


Abbildung 5 [2]

SpoVNet-Dienst auf Anwendungsebene implementiert ist oder bereits fest in die Netzwerkarchitektur integriert wurde, ist oberhalb der Service Abstraction nicht sichtbar. Dadurch können Dienste, welche zu Testzwecken oder als Erweiterung auf Anwendungsebene implementiert wurden, bei einer Weiterentwicklung des Netzwerkes direkt ins Underlay integriert werden, ohne dass eine Veränderung an den Anwendungen, die den Dienst verwenden, notwendig ist.

3.2 Underlay Abstraction

Verändert sich die Technologie oder das Protokoll der unteren Layer des Netzwerkstacks, müssten auf Anwendungsebene implementierte Dienste ebenfalls an die neuen Bedingungen angepasst werden. Darüberhinaus müsste jeder Dienst auf verschiedene heterogene Netzwerktechnologien angepasst werden, um nicht nur in einem homogenen Netzwerk zu funktionieren. Daher wird zwischen der SpoVNet-Dienstschiicht und der SpoVNet-Base die Underlay Abstraction eingeführt. Diese vereinfacht das Implementieren von neuen Diensten, welche mit unterschiedlichen und sogar sich während dem Betrieb verändernden Underlay-Bedingungen zurechtkommen. [2]

Alle über der Underlay Abstraction liegenden Funktionen verwenden für ihre Adressierung die Identifier des Overlay. Somit sind sie vom Lokator des Underlay unabhängig und somit auch unabhängig von Mobilität und Multihoming.

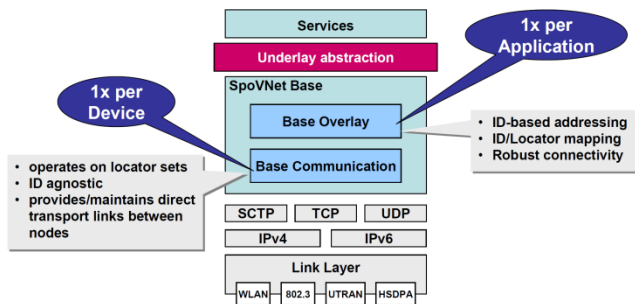


Abbildung 6 [5]

Eine Verbindung, die über einen Dienst über der Underlay Abstraction aufgebaut wurde, hat ebenso transparenten Zugriff auf Transport- und Netzwerkschichtprotokolle, was bedeutet, dass Verbindungen, welche über unterschiedliche Netzwerke aufgebaut werden, als Direktverbindung vom Dienst wahrgenommen werden. Veränderungen z.B. durch neue Relaypfade werden nicht als solche wahrgenommen und müssen somit auch nicht bei der Implementierung berücksichtigt werden.

4. CLIO/UNISONO

Bei der parallelen Verwendung mehrerer Overlays auf einem Gerät bzw. auf einem Knoten werden Cross-Layer-Informationen immer wichtiger, da sie die Möglichkeit zur Optimierung des Overlays auf die Struktur des Underlays bieten. Eine Komponente, die diese Cross-Layer-Informationen sammelt und analysiert, muss einige wichtige Voraussetzungen erfüllen, um dem Konzept von SpoVNet gerecht zu werden. Allen voran muss sie dezentral sein, um mit der Struktur eines sich selbst organisierenden Overlay-Netzwerkes vereinbar zu sein. Dadurch scheiden schon sehr viele bestehende Ansätze zur Netzwerkanalyse aus, da sie auf einer zentralen Datenerfassung basieren. Desweiteren muss die Komponente einige Aufgaben erfüllen, welche durch die Overlay-Struktur nötig werden.

Als erstes haben verschiedene Anwendungen auch verschiedene Anforderungen an das Netzwerk. Während Video-Streaming z.B. eine hohe Bandbreite voraussetzt, ist bei Voice-over-IP eine geringe Verzögerung der ausschlaggebende Faktor für die Quality of Service. Daher darf ein Overlay, auf dem unterschiedliche Anwendungen laufen sollen, nicht statisch sein und muss sich den Anforderungen dynamisch anpassen.

Um Optimierungen überhaupt durchführen zu können müssen Informationen über das Underlay gesammelt werden. Wichtige Faktoren sind hier eine schnelle Erfassung eines sich laufend verändernden Underlays und eine hohe Genauigkeit der Messwerte. Desweiteren muss das System auch mit einer Skalierung des Netzwerkes, d.h. mit einer hohen Knotenzahl bzw. einer großen Anzahl an Overlays auf einem Host zurechtkommen. Genaue und häufige Messungen parallel in vielen verschiedenen Overlays führen zwangsläufig zu einer großen Belastung des Netzwerkes allein durch die Messvorgänge. Da dies einer angestrebten Optimierung der Leistungsfähigkeit des Netzwerkes entgegensteht, ist ein möglichst geringer Einfluss der Messvorgänge auf die Leistung des Netzwerkes ein weiterer wichtiger Faktor für eine entsprechende Komponente. Die nötigen Komponenten müssen also, im Hinblick auf folgende Qualitätssichernde Leistungen entwickelt werden:

Genauigkeit, um möglichst gute Werte für die Optimierungsmaßnahmen zu erhalten. **Effizienz**, um das System nicht mit den Messungen so stark zu belasten, dass die optimierenden Maßnahmen im Endeffekt zu einer Verschlechterung der Gesamtleistung führen. **Datenschutz**, um zu verhindern, dass mit den angeforderten Messdaten weitere Instanz oder Knotenspezifische Daten übermittelt werden. **Einfachheit**, um Anwendungen durch ein möglichst allgemeines Interface den Austausch von Informationen zu erleichtern.

Mangels einer passenden, bereits bestehenden Lösung, welche allen Anforderungen in Hinblick auf SpoVNet gerecht wird, ging aus dem Ursprungsprojekt das Projekt CLIO/UNISONO hervor. UNISONO beschäftigt sich mit der Messung und Analyse, wohingegen CLIO (Cross-Layer Information Service) als Interface und Verbindungsstück für die Anfragen z.B. von Knoten fungiert. Dabei ist CLIO in zwei Teile aufgeteilt: Die Kernkomponente und die Knotenkomponente. Erstere ist für die Datenerfassung und -analyse zuständig. Es werden Daten verarbeitet, die direkt aus dem Underlay abgefragt werden können und zusätzliche Daten ergänzend durch Messungen gewonnen. Diese Daten sind dann für andere Knoten über die Knotenkomponente abrufbar. Hierfür ist ein allgemeines Interface implementiert, durch das die Daten im gewünschten Format

abgefragt werden können. Wichtig hierbei ist, dass die Daten, welche für einen Knoten gesammelt wurden, auch für andere Knoten zugänglich sind. Das heißt, dass sämtliche Daten über eine SpoVNet-Instanz auch in anderen SpoVNet-Instanzen verfügbar sind. Somit ist auch eine Overlay-übergreifende, sowie eine Instanz-übergreifende Optimierung möglich.[4]

5. Anwendungsbereiche

Wie bereits erwähnt, baut SpoVNet einzelne Instanzen auf, welche dann bestimmten Anforderungen gerecht werden. Wenn man das Projekt nun in die verschiedenen Initiativen des Future Internet einordnet, stellt man fest, dass es sich hier nicht um einen Ansatz handelt, welcher die komplette Internetstruktur gestalten soll. Vielmehr zielt der Ansatz auf verschiedene Einsatzbereiche ab, in welchen er die gewünschten Anwendungen optimiert.

Ein Beispiel sind Filesharing-Netzwerke. Ein solches Netzwerk erfordert eine flexible Anpassung an verschiedenste Rahmenbedingungen, wie unterschiedliche Netzanbindungen oder Teilnehmer, welche das Netzwerk verlassen. Darüber hinaus soll es auch schnell und unkompliziert aufzubauen sein.

Ein weiterer Anwendungsbereich sind Online-Spiele. Durch gezieltes Routing von Paketen über schnelle Routen werden geringe Latenzzeiten erzielt, welche für Online-Spiele wichtiger sind, als ein großer Datendurchsatz. Viele Online-Rollenspiele arbeiten mit Instanzen. Zwischen Teilnehmern verschiedener Instanzen müssen, wenn überhaupt, nur grundlegende Dienste angeboten werden, daher kann jede Instanz über ein eigenes Service-Overlay zusammengefasst werden.

6. Schwächen

Da die bisherigen Implementierungen noch sehr unausgereift sind, ist es schwierig, die Schwächen bezüglich Sicherheit, Skalierbarkeit, Zuverlässigkeit und anderen Bereichen einzuschätzen. Im August 2009 wurde zwar bereits ein Testnetzwerk demonstriert, allerdings war dieses noch nicht dynamisch, was eines der Hauptziele von SpoVNet ist.

Ein Problem, welches sich aus dem Bestreben nach einem zukünftigen Internet ergibt, ist natürlich, dass SpoVNet eben kein Ersatz für das aktuelle Internet darstellt. Genau das könnte jedoch auch ein Vorteil sein, denn dadurch steht SpoVNet nicht vor der Hürde des riesigen finanziellen sowie organisatorischen Aufwands, den eine Revolution des Internets bedeuten würde. Dennoch bietet es die Chance auf eine starke Verbesserung von bisherigen Services und zudem die Grundlage für weitere größere Schritte in Richtung Future Internet.

7. Zusammenfassung

Zusammenfassend lässt sich sagen, dass durch die SpoVNet-Architektur ein großes Maß an Flexibilität erreicht werden kann. Soweit die Ziele des Projektes umgesetzt werden können, sodass zuverlässige Overlay-Netzwerke möglich sind, stellt dieser Ansatz eine Erleichterung für viele vernetzte Anwendungen dar. Es ist allerdings zu beachten, dass SpoVNet im Rahmen der Future Internet Projekte nicht als Konzept für eine Weiterentwicklung der gesamten Internetstruktur darstellt. Vielmehr stellt SpoVNet

eine Möglichkeit dar, größere Netzwerke flexibel und effizient aufzubauen. Dienste, welche bisher auf statische Overlay-Netzwerke angewiesen sind, könnten durch SpoVNet auf ein flexibleres Overlay-Netzwerk zurückgreifen, welches beispielsweise effizientere Routen findet.

Jedoch steckt das Projekt noch in der Entwicklungsphase. Einige wichtige Funktionen müssen noch implementiert werden und selbst dann ist die Frage, wie erfolgreich das Projekt werden wird. Nicht nur die Technik und die handfesten Vorteile sind ein Faktor in der Wirtschaft, sondern ebenfalls die Behauptung gegenüber der Konkurrenz. An diesem Punkt hat sich in der Vergangenheit schon häufig gezeigt, dass die überlegene Technologie sich nicht gegen eine unterlegene Technologie behaupten konnte. Sei es wegen der öffentlichen Wahrnehmung oder auch aufgrund falscher wirtschaftspolitischer Entscheidungen.

Literaturverzeichnis

- [1] Handley, M. (2006, Juli). Why the Internet only just works. London, Großbritannien.
<http://www.cs.ucl.ac.uk/staff/M.Handley/papers/only-just-works.pdf>
- [2] The SpoVNet Consortium. (2007, Juli 23). SpoVNet: An Architecture for Supporting Future Internet Applications. Würzburg, Deutschland.
<http://www.tm.uka.de/~huebsch/spovnet/data/SpoVNet.pdf>
- [3] Hübsch, C., Mayer, C. P., Mies, S., Bless, R., Waldhorst, O. P., & Zitterbart, M. (2009, August). Reconnecting the Internet with ariba: Self-Organizing Provisioning of End-to-End Connectivity in Heterogeneous Networks. Karlsruhe, Deutschland.
<http://conferences.sigcomm.org/sigcomm/2009/demos/sigcomm-pd-2009-final45.pdf>
- [4] Haage, D., Holz, R., Niedermayer, H., & Laskov, P. (2009, November). CLIO - A Cross-Layer Information Service for Overlay Network Optimization. Kassel, Deutschland.
<http://www.tm.uka.de/~huebsch/spovnet/data/KiVS-2009-CLIO.pdf>
- [5] Roland Bless, O. P. (2008, Juni). The Spontaneous Virtual Networks Architecture for Supporting Future Internet Services and Applications. Heidelberg.
<http://www.tm.uka.de/~huebsch/spovnet/data/2008-06-17-FG-future-internet-org.pdf>