Electronic Communications of the EASST
Volume 17 (2009)

Workshops der Wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen 2009 (WowKiVS 2009)

Selbstverwaltung im Future Internet

Achim Marikar, Jens Mödeker, Karl Jonas

10 pages

Guest Editors: M. Wagner, D. Hogrefe, K. Geihs, K. David
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Selbstverwaltung im Future Internet

Achim Marikar, Jens Mödeker, Karl Jonas

achim.marikar@fokus.fraunhofer.de, jens.moedeker@fokus.fraunhofer.de,
karl.jonas@fokus.fraunhofer.de, http://www.fokus.fraunhofer.de/go/net
Fraunhofer Gesellschaft e.V., FOKUS, Institut für offene Kommunikationssysteme,
Kompetenzzentrum NETwork Research

Abstract: Zukünftige Netze sollen in der Lage sein, eine Vielzahl verschiedener Dienste zu unterstützen. Dabei wird angenommen, dass sie oft keine statische Netzwerkstruktur haben und sich daher selbsttätig konfigurieren und automatisch an wechselnde Anforderungen sowie Netzwerksituationen und -änderungen anpassen sollen. Es ist wünschenswert, dass die benötigte Dienstqualität (QoS), im Besonderen für zeitkritische Anwendungen wie VoIP, automatisch gewährleistet werden kann. Das Netz soll zukünftigen Anforderungen gerecht werden, auch wenn diese zum Zeitpunkt der Erstellung noch nicht relevant oder bekannt sind.
In diesem Dokument wird ein möglicher Ansatz des noch jungen Forschungsprojekts Self-NET für die Erfüllung der genannten Anforderungen skizziert und zur Diskussion gestellt.

Keywords: Future Internet, Selbstverwaltung, Selbstmanagement, Selbstkonfigurierung, Selbstoptimierung, Bewusstsein, Kognitives Handeln, Self-NET

1 Einleitung

Das Future Internet stellt die Konvergenz unterschiedlicher Netzwerktypen dar. Neben der bereits begonnenen Fusion von Telefon- und Fernsehnetzen und weiterer Datennetze mit dem Internet gibt es zahlreiche weitere Möglichkeiten, das Future Internet für heutige und zukünftige Anwendungen zu nutzen. Neben Kaffeemaschinen und intelligenten Kühlschränken sollen viele weitere Geräte im Haushalt, in Unternehmen und im öffentlichen Raum an das Netz angeschlossen werden. Um eine solche Verbreitung der Netze komfortabel und auch rentabel zur erreichen, ist es wahrscheinlich, dass selbst organisierende Mesh-Netzwerke Teil der zukünftigen Infrastruktur werden.
Um diese Vision Wirklichkeit werden zu lassen, ist es notwendig, aufgrund der gestiegenen Komplexität, den Netzen einen großen Teil an Autonomie zukommen zu lassen. Eine manuelle Konfiguration ist fehleranfällig und bei dynamischen Netzwerken und stark variierender Netzwerknutzung sehr aufwendig zu administrieren. Ein Netzwerk sollte sich automatisch an die genutzten Dienste anpassen können, bei einem Datenstau oder dem Wegfall einer Route oder eines Routers automatisch reagieren und neue Netzwerkelemente selbstständig erkennen und einbinden. Über die sinnvolle Optimierung von Netzwerken gibt es bereits zahlreiche Arbeiten, [Da06], [HBP07], [Mil91], [BFH02]. Diese enthalten viele verschiedenen Ideen, wie das Future Internet gestaltet werden kann. Sie behandeln meist nur Teilaspekte oder verfolgen ein komplett neues Design (Clean Slate Approach), das auf eine Rückwärtskompatibiltät kaum Rücksicht nimmt,

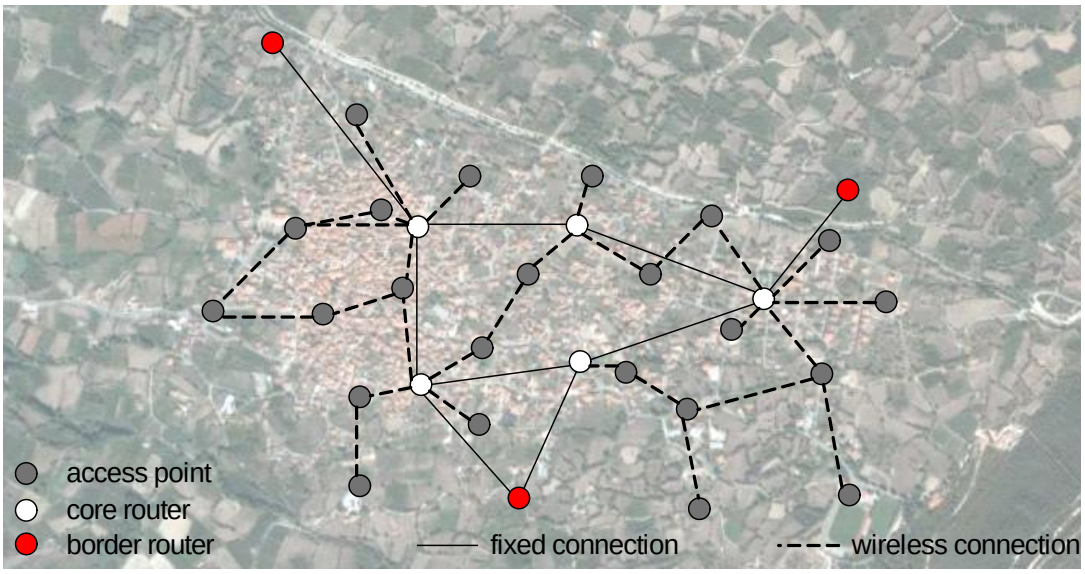

Abbildung 1: Heterogenes Netzwerk mit Mesh-Elementen zur Grundversorgung eines Dorfes

[Ca03]. Das Forschungsprojekt Self-NET [SN] hat sich als Ziel gesetzt, viele dieser Ideen zu einem Gesamtkonzept zu vereinen.

2 Anwendungsszenario

Die Komplexität zukünftiger Netze steigt. Gleichzeitig sollten der Aufwand zur Errichtung eines Netzwerkes sowie die Gesamtkosten sinken. Neben der bereits erwähnten Vernetzung nahezu jeglicher elektronischer Geräte, gibt es weitere Anwendungsfälle für das Future Internet, wie die Erschließung von noch nicht ausreichend versorgten Gebieten, der Einsatz in Katastrophengebieten oder auch die spontane Erweiterung der vorhandenen Infrastruktur bei Massenveranstaltungen.
Bei zukünftigen Netzen kann auch die Vielfalt der verwendeten Technologien weiter steigen. Ein solches heterogenes Netz kann aus festen Leitungen wie Glasfaser, DSL oder Koaxkabel oder herkömmlichem Ethernet sowie Funktechnolgien wie WLAN, WIMAX, UMTS, LTE oder DVB-T und DVB-S sowie zukünftigen Übertragungstechnologien bestehen.
Diese Vielfalt der Technologien hat jedoch den Nachteil, dass unterschiedlichste Anforderungen und Möglichkeiten zusammentreffen. So haben die Verbindungen, neben einer unterschiedlichen absoluten Bandbreite, teilweise auch Unterschiede in den Übertragungsrichtungen (Up- und Download, z.B. A-DSL) oder sind unidirektional (z.B. DVB-T). Auch die QoS-Eigenschaften sind höchst unterschiedlich. Wenn Funktionen hinzukommen, wie sie bereits aus den Mesh-Netzwerken bekannt sind, die das Anbieten von abseits gelegenen Zugangspunkten über mehrere Funkstrecken hinweg ermöglichen, erhöht sich die Komplexität des Netzwerkes nochmals. Da die Anforderungen mobiler Nutzer ständig variieren und in der Regel die Qualität der Funkverbindungen nicht konstant ist, ist ein effizientes Management und die Optimierung des Netzwer-

kes kaum noch oder gar nicht manuell durchführbar.
Um die Errichtung und den Betrieb eines heterogenen Netzwerkes, wie es in Abbildung 1 zu sehen ist, effizient und kostengünstig zu ermöglichen, ist es notwendig, dass das Netzwerk in der Lage ist, sich automatisch zu konfigurieren und kontinuierlich zu optimieren.

3 Selbstverwaltung

Die Selbstverwaltung ist einer der Bestandteile des Future Internets, die eine grundlegende Verbesserung gegenüber heutigen Netzen darstellen können. Sie beinhaltet Möglichkeiten zur Selbstkonfiguration, Selbstoptimierung und Selbstheilung sowie zum Selbstschutz [IBM05].
Über die Selbstkonfiguration ist es möglich, dass sich ein Netzwerk, ohne wesentliche manuelle Anpassungen, einrichtet und so eine ordnungsgemäße Funktion bieten kann. Hierzu müssen die grundlegenden Konfigurationsparameter von den Geräten automatisch untereinander abgesprochen und umgesetzt werden. Eine optimalere Ausnutzung der vorhandenen Ressourcen wird durch die Selbstoptimierung gewährleistet. Im Fehlerfall ist die Selbstheilung die Eigenschaft, die es ermöglicht, dass ausgefallene Router oder Links kompensiert und auch die Auswirkungen weiterer Fehler minimiert werden. Über die Fähigkeit zum Selbstschutz sollen die meisten beabsichtigten und unbeabsichtigten Angriffe auf die Funktionsfähigkeit des Netzwerkes und der angeschlossenen Endgeräte verhindert werden. Auch soll eine Beeinträchtigung des Netzwerkes durch die Fehlfunktion, Fehlkonfiguration eines Elementes verringert werden, z.B. bei Fehlern in der Implementierung.
Diese Eigenschaften dienen dazu, dem Netzwerk eine Art von Selbstbewusstsein zu geben. Dies bedeutet, dass das Netzwerk auf individuelle Situationen selbstständig und auf vielfältige Weise reagieren kann.
Im Folgenden wird besonders die Netzwerkeigenschaft Selbstoptimierung betrachtet.

4 Neue Möglichkeiten für das Future Internet

4.1 Kognitiver Zyklus

Als Basiselement zur Optimierung von Netzwerken hat Self-NET einen kognitiven Ablauf definiert. Der sogenannte MDE-Zyklus beschreibt das Zusammenspiel von Überwachung (Monitoring, M), Entscheidungsfindung (Decision Making, D) und Ausführung (Execution, E), um einem Netzwerk die Fähigkeit zur Selbstverwaltung zu ermöglicht. Durch den in Abbildung 2 zu sehenden kontinuierlich ablaufenden kognitiven MDE-Zyklus haben die Netzwerkelemente Kenntnis über die aktuelle Situation im Netzwerk. Nachdem die Anforderungen der zu übertragenden Daten ermittelt wurden, kann ein Netzwerkelement die Datenströme bei Bedarf separat behandeln. Optimiert wird dieser Zyklus über die Fähigkeit des Lernens. Über die Auswertung der Auswirkungen von vorhergehenden Änderungen kann das Netzwerkelement bei zukünftigen Entscheidungen von den Erfahrungswerten profitieren.

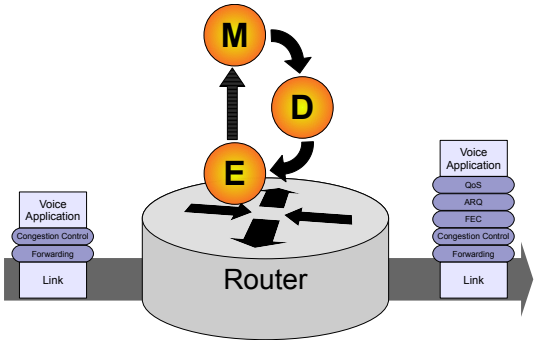

Abbildung 2: MDE-Zyklus auf einem Router

4.1.1 Monitoring

Aktive Netzwerkkomponenten wie Router müssen die Datenströme kennen, die sie verarbeiten, damit sie entsprechend auf sie reagieren können. Hierzu ist ein umfangreiches Monitoring notwendig. Neben bekannten Größen wie Latenz, Laufzeitschwankungen und Paketverlust ist der Typ der Daten (Video, Sprache, FTP, etc.) und somit deren Anforderungen an das Netzwerk von Bedeutung. Zusätzlich können Router Flags auswerten, die vom Sender oder anderen vorhergehenden Netzwerkelementen gesetzt wurden. Die Ergebnisse dieses Monitoring sind die grundlegenden Eingaben für den Decision Making-Teil des MDE-Zyklus.

4.1.2 Decision Making

Ein Netzwerkelement muss entscheiden, ob eine Modifikation an einem Datenstrom notwendig ist. In diesem Fall kann es einen Datenstrom priorisieren, über einen anderen Link umrouten oder Funktionen wie Automatic Repeat Query (ARQ) oder Forward Error Correction (FEC) hinzufügen. Die Entscheidung hängt maßgeblich von den Informationen über die derzeitige Situation des Netzwerkes und den Anforderungen des Datenstroms ab, die durch das Monitoring gewonnen wurden. Weitere Entscheidungshilfen sind die bisherigen Erfahrungen des Routers mit ähnlichen Situationen.

4.1.3 Execution

Die Umsetzung der Entscheidungen erfolgt über den Execution-Teil des MDE-Zyklus. Hier werden die Pakete tatsächlich verändert oder umgeleitet. Auch Link-lokale Funktionen wie ARQ zwischen zwei Routern als Funktion der dynamischen Protokoll-Komposition (siehe 4.3) werden von diesem MDE-Teil umgesetzt.

4.2 Unterteilung in Layer

Viele Optimierungen benötigen die Interaktion zwischen Netzwerkelementen. Solange sich ein Router nur mit seinem Nachbarn unterhalten muss, um beispielsweise ARQ auf deren Verbindung zu aktivieren, stellt dies kein großes Problem dar. Wenn jedoch Entscheidungen getroffen

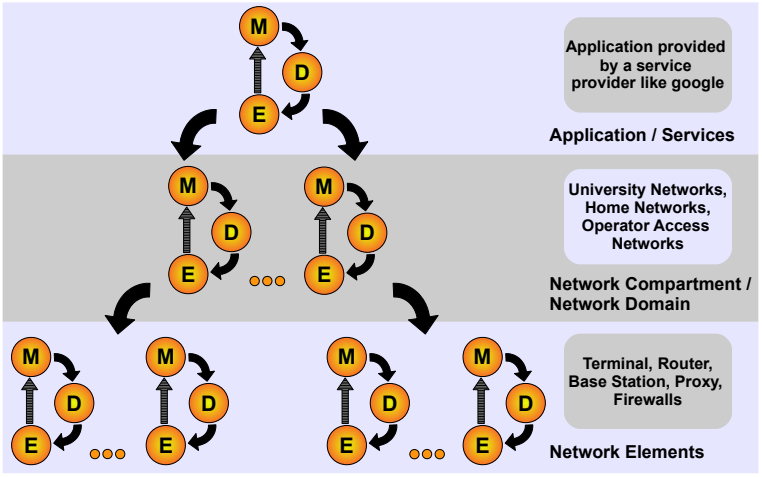

Abbildung 3: MDE-Layer

werden, die einen größeren Teil des Netzwerkes betreffen, wie das Umrouten über andere Router und Leitungen, wird dies aus der Perspektive eines einzelnen Routers sehr komplex. Um die Komplexität zu verringern, wird zwischen mehreren MDE-Layern unterschieden. Hierzu wird die in Abbildung 3 gezeigte Hierarchie verwendet. Die einzelnen Netzwerkelemente bilden den Network Element-Layer. Über dieser Ebene liegt der sogenannte Compartment-Layer, der sich um einen definierten Teil eines Netzwerkes oder ein gesamtes Netzwerk kümmert. Um Einfluss auf fremde Netze nehmen zu können, ist ein weiterer Layer notwendig. Der sogenannte Application- oder Service-Layer dient dazu, aus Anwendungen wie zum Beispiel dem IP Multimedia Subsystem (IMS) heraus, einen Datenstrom von Ende zu Ende beeinflussen zu können. Zwischen den benannten Layern sind beliebige weitere Zwischenlayer möglich, die den Verwaltungsaufwand gering halten [GAM08].

4.3 Functional Protocol Elements

Zur Umsetzung der Funktionalitäten müssen die Router erkennen können, welche Daten sie transportieren und welche Anforderungen vonseiten dieser bestehen. Da die Ports des TCP und des UDP nicht mehr zuverlässig die verwendeten Protokolle und Anwendungen angeben, wird zur Erkennung des Payloads oft Deep Inspection verwendet. Dieses Verfahren versucht, den Inhalt der Pakete zu analysieren um das Protokoll und somit die Anwendung zu bestimmen. Dies funktioniert allerdings nur mit dem Netzwerk Element bekannten Protokollen und ist sehr aufwändig. An dieser Stelle ist es wünschenswert, wenn über die übertragenden Pakete die Anforderungen der Anwendung direkt dem Netzwerk bekannt gemacht werden, wie es auf einfache Weise bereits heute mithilfe von Class of Service (CoS) und DiffServ umgesetzt wird.
Die gängigen Transportprotokolle TCP und UDP können weder an die Anwendungen angepasst, noch beim Transport variiert werden. So ist es beispielsweise nicht möglich, Congestion Control zu nutzen, aber auf ARQ zu verzichten, wovon viele moderne Anwendungen wie das IPTV profitieren würden. Neue Entwicklungen wie das Stream Control Transmission Protocol

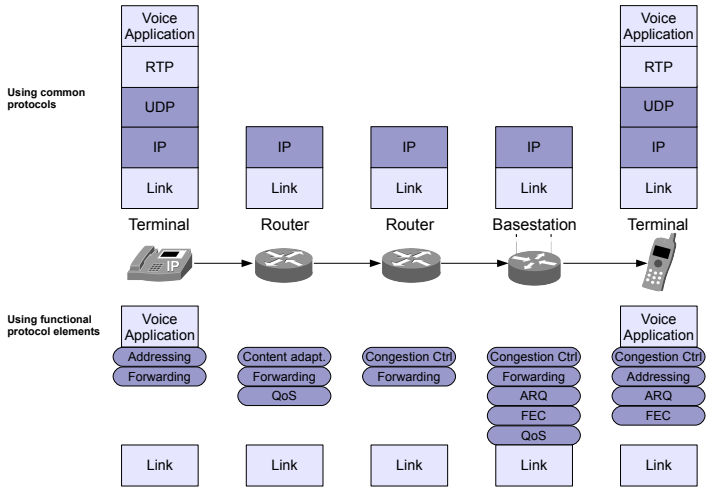

Abbildung 4: Nutzung von FPE und Variation der Komposition

(SCTP) und das Datagram Congestion Control Protocol (DCCP) versuchen einen Teil der Probleme zu kompensieren, so ist es mit DCCP beispielsweise möglich, ein Congestion Handling für Datagrammdienste zu nutzen. Allerdings ist immer noch eine Entscheidung zwischen den verschiedenen Protokollen notwendig, so können die verwendeten Eigenschaften nicht bei Bedarf beliebig verändert werden.
Angesichts dieser Schwierigkeit liegt es nahe, die benötigten Eigenschaften getrennt zu betrachten und es der Anwendung und dem Netzwerk auf Wunsch zu ermöglichen, das genutzte Protokoll aus den vorhandenen Funktionselementen selbst zusammenzusetzen [BWH+07]. Zur Umsetzung dieser dynamischen Protokollkomposition wurden die Functional Protocol Elements (FPE) entworfen. Durch das Auftrennen der Funktionen kann beispielsweise FEC für zeitkritische Datagramme aktiviert werden. Durch die Möglichkeit, einzelne FPEs nur auf Teilstrecken zu aktivieren, kann zum Beispiel ARQ auf verlustreichen Links aktiviert werden [MMa08], [KKFT02], [XJHH07]. Da Pakete im Fehlerfall schneller erneut gesendet werden können, verringert sich der Jitter gegenüber Ende-zu-Ende ARQ. Dies kommt vor allem zeitkritischen Anwendungen wie VoIP zugute, die einen geringen Paketverlust benötigen, allerdings kaum unter einzelnen verlorenen Paketen leiden.
Auch die Unterscheidung zwischen verschiedenen Arten von Nutzdaten fällt leichter. Ohne für jeden Payloadtyp eine separate Verbindung aufbauen zu müssen, ist es möglich, je nach Anforderung der Daten, das Protokoll für dieses Reihe von Paketen oder gar für ein einzelnes Paket anzupassen. Zu den Anwendungen, die von diesen Möglichkeiten profitieren, gehören auch netzwerktaugliche Spiele. Bei gängigen 3D-Spielen wie Rennsimulationen ist es notwendig, mit niedriger Verzögerung Ortsangaben zu übertragen. Diese sind nur für einen kurzen Zeitraum relevant und eine erneute Übertragung im Fehlerfall ist überflüssig. Je nach Spiel müssen jedoch viele Daten wie Karten, Texturen und weiterer Informationen übertragen werden, die vollständig und fehlerfrei empfangen werden müssen, allerdings nicht zeitkritisch sind und auch bei der

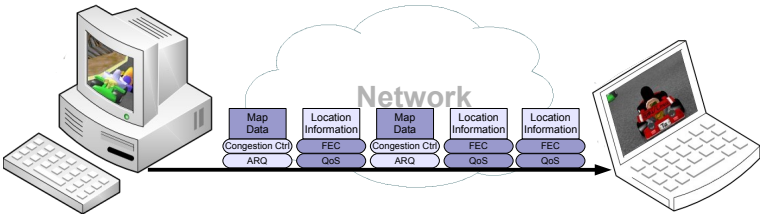

Abbildung 5: Individuelle Protokollkomposition für einzelne Flows

Übertragungsgeschwindigkeit ggf. zurückgestellt (teilweise verdrängt) werden können.
Bedeutend ist die Tatsache, dass die Konfiguration des Protokolls und somit die einzelnen Funktionen, nicht mehr von Anfang bis Ende einer Übertragung konstant sein müssen. So lässt sich die Komposition jederzeit an aktuelle Bedingungen und sogar für einzelne Teilstrecken anpassen. Bei Bedarf kann ein Router die Komposition abändern, siehe Abbildung 2. Auf diese Weise können Daten beispielsweise in Zugangsnetzen und Backbones unterschiedlich behandelt werden. Siehe auch [HBP07].

5 Verschmelzung von Netzwerklayern

Aktuelle Netzwerkstacks nutzen mehrere Layer, die jeweils Funktionen für die Datenübertragung bereitstellen. Diese einzelnen Layer sind meistens unabhängig von den darüber und darunter liegenden Schichten. Dies bedeutet, dass die verwendeten Protokolle der einzelnen Layer variabel sind und Anwendungen oft mit verschiedenen Protokollen arbeiten können. So können manche Anwendung, die standardmäßig UDP nutzen auch mit TCP arbeiten oder anstatt IP über Ethernet zu nutzen, kann ein MPLS-Layer dazwischen gesetzt werden.

Aufgrund dieser Unabhängigkeit der Layer werden viele Funktionen von verschiedenen Layern implementiert (z.B. Berechnung und Übermittlung von Checksummen). Anwendungen wie SIP nutzen eine eigene einfache Verbindungskontrolle, die bei der Verwendung von TCP überflüssig ist. Unter der Voraussetzung, dass alle Funktionen in einzelne Elemente (FPEs) zerlegt werden sollen, ist es fraglich, ob die Unterteilung in Layer, wie sie heute üblich ist, bei der Verwendung von FPEs noch notwendig ist. Eine Zusammenfassung einiger Layer, wie sie auf Abbildung 6 zu sehen ist, ist denkbar.

In heutigen Netzwerken kann IP als Standard auf Netzwerkebene betrachtet werden. Die Protokolle der weiteren Layer variieren. Um jedoch die gewünschte Rückwärtskompatibiltät zu ermöglichen, ist es notwendig, die für die Hardwarekomponenten notwendigen Layer nur soweit abzuändern, dass die Datenpakete auch weiterhin von Standardkomponenten transportiert und geroutet werden können. Für Endanwendungen wird eine Anpassung oder eine Umsetzung zwischen den Protokollgenerationen vorausgesetzt. Alternativ ist zur Kommunikation mit herkömmlichen Endanwendungen eine optionale Beibehaltung der bisherigen Layer und Protokolle denkbar, die lediglich um einige neue Funktionen ergänzt werden, die von Self-NET-fähigen Netzwerkelementen verwendet werden können.

Gängiger Netzwerk Stack:

Layer	Data Link	Network	Transport	Session / Application	
Protocol	e.g. Ethernet	IP	TCP / UDP	e.g. RTP	Payload

Stack nach der Zusammenfassung der Layer:

Layer	Data Link / Network / Transport / Session		Application
Protocol	e.g. Ethernet	Addressing / Transmission Control / etc.	Payload

Abbildung 6: Mehrere Layer können zusammengefasst werden

6 Realisierung des Future Internets

Das im Abschnitt 2 genannte Szenario kann von den oben genannten neuen Funktionen profitieren. Das verwendete Transportprotokoll lässt sich an das Netzwerk anpassen. So ist bei instabilen Funkverbindung lokal ARQ möglich. Bei Ausfall oder Überlastung einer Strecke wird automatisch eine Alternativroute gewählt. Pakete können getrennt nach Stream behandelt werden. Die Dateiübertragung nutzt die qualitativ schlechtere Strecke, damit das VoIP-Gespräch auf der besseren nicht gestört und die Sprachqualität einwandfrei ist. Ein weiterer Faktor ist, dass die neuen Funktionen wie FPEs kein komplett neues Netz benötigen. Herkömmliche Router sollen diese Pakete ohne Beeinträchtigung routen und Self-NET-fähige Router die neuen Funktionen nutzen können. Zwischen zwei Self-NET-Elementen können somit beliebige herkömmliche Netzwerkelemente vorhanden sein. Über eine gewöhnliche Protocol Translation (PT) können nicht Self-NET konforme (End-)Anwendungen einen Großteil der neuen Funktionalitäten nutzen, indem am Host, oder auf einem Netzwerkelement auf dem Übertragungsweg, herkömmliche Datenströme von TCP und UDP um das Self-NET-Protokoll ergänzt wird. So ist eine vollständige Abwärtskompatibilität gewährleistet.

7 Fazit

Zukünftige Anwendungen müssen lediglich ihre Anforderungen spezifizieren, der Protokollstack und das Netzwerk kümmern sich um deren Erfüllung. Entwickler sind nicht mehr an die Auswahl TCP oder UDP gebunden. Die Möglichkeit, die Protokollkomposition während der Übertragung zu ändern oder Funktionen ausschließlich auf Teilstrecken zu beschränken, wird auch zukünftigen Anforderungen gerecht. Über die Verteilung der Verwaltungseigenschaften auf mehrere Layer, ist die seit langem gewünschte und von modernen Diensten wie dem IMS geforderte Ende-zu-Ende QoS möglich.
Bei der Umsetzung dieser neuen Funktionalitäten sind Router die wesentlichen Elemente, die durch zusätzliche Aufgaben mehr leisten müssen als heute. Aufgrund der kontinuierlichen Leistungssteigerung bei der verwendeten Hard- und Software wird erwartet, dass dies für zukünftige Router möglich ist.

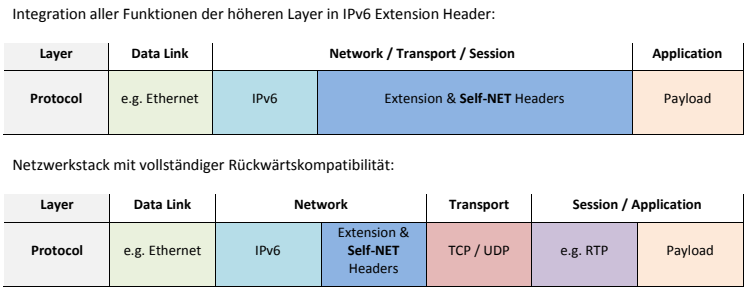

Abbildung 7: Mögliche neue Netzwerkstacks

8 Future Work

Das Projekt Self-NET befasst sich mit kognitiven Mechanismen im Future Internet. Nach dem Projektstart im Mai 2008 wurde mit der Spezifizierung dieser Mechanismen begonnen. In zukünftigen Publikationen sind detailliertere Angaben zur Umsetzung der oben genannten Ideen zu erwarten. Ziel des Projektes ist, eine Implementation der wichtigsten Self-NET-Elemente präsentieren zu können. Aktuell wird evaluiert, ob die für die FPEs notwendigen Informationen in den Extension Header von IPv6 untergebracht werden können. Über diese Erweiterung soll die gewünschte Abwärtskompatibilität gewährleistet werden. Wie zukünftige Netzwerkstacks aussehen könnten, zeigt Abbildung 7.

Literatur

[BFH02] R. Braden, T. Faber, M. Handley. From Protocol Stack to Protocol Heap – Role-Based Architecture. *Proc. Hotnets I, Princeton, NJ,*, October 2002.

[BWH+07] P. G. Bridges, G. T. Wong, M. Hiltunen, R. D. Schlichting, M. J. Barrick. Configurable and Extensible Transport Protocol. *Transactions on networking*, IEEE/ACM 2007.

[Ca03] D. Clark, et al. New Arch: Future Generation Internet Architecture. *Final Technical Report*, 2003.
http://www.isi.edu/newarch/iDOCS/final.finalreport.pdf

[Da06] S. Dobson, et al. A survey of autonomic communications. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 1(2):223–259, Dec. 2006.

[GAM08] A. Grebe, S. Abu Salah, A. Marikar. Analyse des Ende-zu-Ende QoS Routing in IP Multimedia Subsystems (IMS). *13. Mobilfunktagung Osnabrück*, 2008.

[HBP07] S. Heimlicher, R. Baumann, B. Plattner. The Transport Layer Revisited. *Communication Systems Software and Middleware*, IEEE 2007.

[IBM05] IBM. An architectural blueprint for autonomic computing. June 2005. http://www.ginkgo-networks.com/IMG/pdf/AC_Blueprint_White_Paper_V7.pdf

[KKFT02] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, S. K. Tripathi. Split TCP for mobile ad hoc networks. *Global Telecommunications Conference, 2002*, IEEE 2002.

[Mil91] I. Miloucheva. XTP-Experimental Implementation at the Technical University of Berlin for an Integrated Services Broadband Environment (BERKOM). *TRANSFER, Vol. 4, Number 4*, 1991.

[MMa08] J. Mödeker, A. Marikar, et al. System Deployment Scenarios and Use Cases for Cognitive Management of Future Internet Elements. 2008. http://www.ict-selfnet.eu/

[SN] EU ICT project Self-NET, Self-Management of Cognitive Future InterNET. http://www.ict-selfnet.eu/

[XJHH07] F. Xie, N. Jiang, Y. H. Ho, K. A. Hua. Semi-Split TCP: Maintaining End-to-End Semantics for Split TCP. *Local Computer Networks, 2007*, IEEE 2007.