Next Generation Internet
3. NAT & IPv6

INSTITUT FÜR TELEMATIK

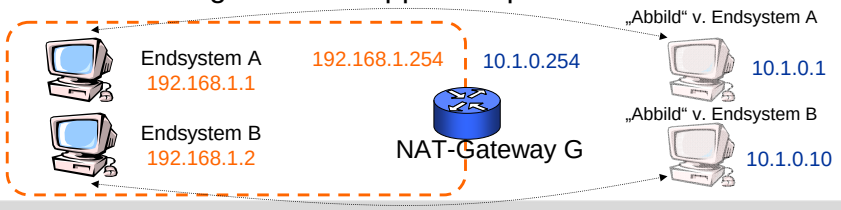
Kapitelübersicht

3.1 Network Address Translation

- Motivation
  - Bildung privater Netze („Intranets“) mit IPv4-Adressen aus einem der „privaten“ Adressräume: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 [RFC1918]
  - Adressenknappheit: Zu wenig öffentliche IPv4-Adressen
- NAT (Network Address Translation) [RFC3022], ursprünglich [RFC1631] (1994)
  - Bijektive Abbildung („Binding“) zwischen IP-Adressen (öffentlich ↔ privat)
  - Umsetzen der Adressen im IP-Paket (u. ICMP-Paket) erfolgt „transparent“ durch NAT-Gateway
    - Erfordert Pool an öffentlichen Adressen
    - Anpassen der Prüfsummen (ggf. auch TCP/UDP) notwendig
    - Anwendungsabhängige Anpassung notwendig!
      → Application Level Gateway (ALG)

Reines NAT

- Zuerst 1991 beschrieben
- Reine 1:1 Übersetzung, Basic NAT
  - spart keine IPv4-Adressen ein
  - pro Datenstrom zustandslos
- Heutzutage primärer Einsatzzweck in Unternehmen
  - Verbindung sich überlappender privater Adressräume

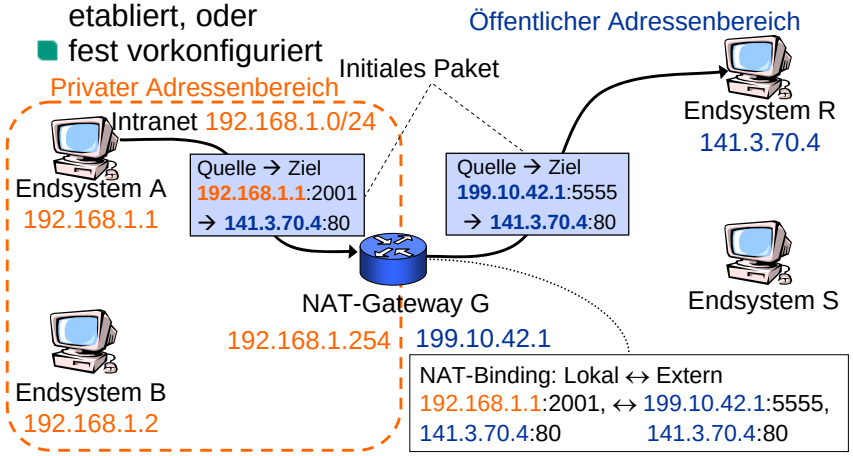


Network Address Port Translation

- NAPT (Network Address Port Translation)
  - Zusätzlich zu „Basic NAT“: Umsetzung von Transport-IDs: TCP/UDP-Ports bzw. ICMP Query ID
  - Bijektive Abbildung zwischen (Adresse,Port)-Tupeln
    - erfordert Zustandshaltung → Crash des NAT-GW führt zu Abbruch der Kommunikation
    - erfordert das Initiieren der Kommunikation „von innen“ heraus
    - Erreichbarkeit von außen nur für freigeschaltete/konfigurierte Adressen
  - Adressenbedarf wird auf die (eine) öffentliche Adresse des Gateways reduziert
  - Gateway muss u.U. entscheiden, wann Kommunikation beendet ist (Lebenszeit des Bindings?)
  - Interne Netzstruktur wird vor dem Internet verborgen

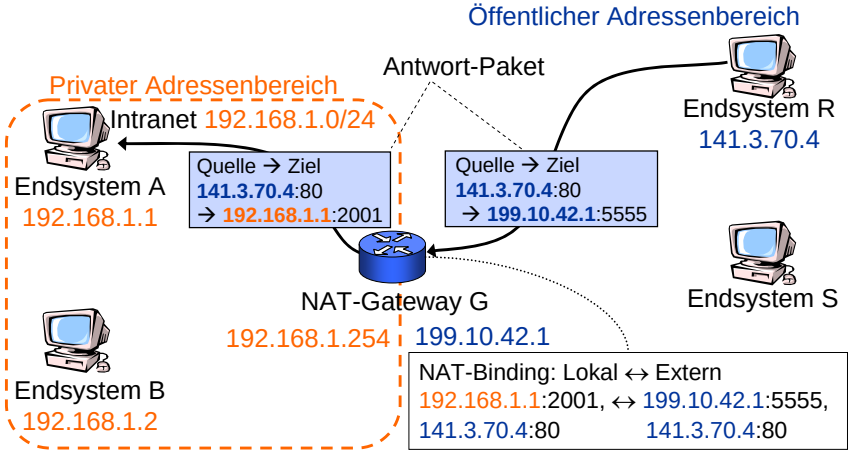
NAPT Beispiel (1)

- Binding
  - wird durch initiales Paket aus dem Intranet heraus etabliert, oder
  - fest vorkonfiguriert



NAPT Beispiel (2)

- Für Paket von extern muss ein Binding existieren!



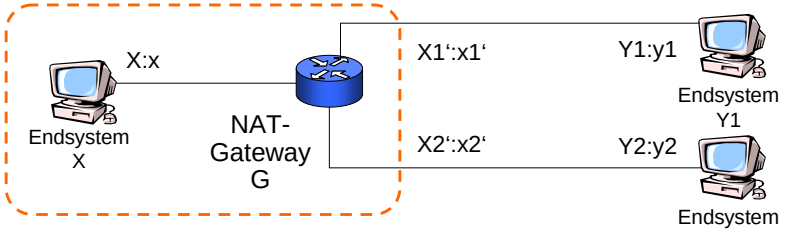
Probleme durch NAT Binding

- Verbindungsloses Konzept/Soft-State
  - Wie lange gilt das Binding? So lange wie Sitzung dauert...
  - Zustand wird nach Timeout gelöscht (TCP: 4 min, UDP: ?)
  - Folge: Abändern von Protokollen bzw. Generieren von Zusatzverkehr, um Binding aufrecht zu erhalten
- Spontane Erreichbarkeit von außen nicht möglich, da das Binding fehlt
- Problem für Protokolle, die dynamisch Nutzdatenströme auf neue bzw. von neuen Ports erzeugen, z.B. VoIP (SIP+RTP)
- Insbesondere ein Problem für zwei Endsysteme hinter verschiedenen NATs (z.B. Peer-to-Peer-Netze) [RFC2663]
  - UNilateral Self-Address Fixing (UNSAF): Feststellen und Festlegen der „externen“ Adresse, meist mit Hilfe eines externen öffentlichen UNSAF-Servers

NAT-Varianten

- Viele verschiedene NAT-Typen
  - NAT, NAPT → NAT44, NAT64, NAT46, NAT444 usw.
  - Outbound NAT (Initiierung d. Komm. von innen heraus), Two-way NAT
  - Twice NAT (modifiziert Quell- und Zieladressen)
    - bei Adresskollisionen zwischen internen und externen Adressen
    - erfordert Split-DNS
  - Alte Klassifikation: Symmetric NAT, Full Cone, Restricted Cone, Port-Restricted Cone, usw.

NAT Varianten – neue Klassifikation



freizügig

einschränkend

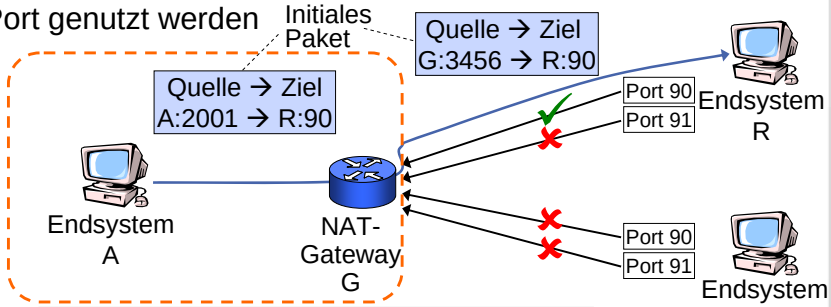
Abbildungsfunktion [RFC4787] Filter

- Endsystem-unabhängig
- Adressabhängig
- Adress- und Port-abhängig

- Endsystem-unabhängig
- Adressabhängig
- Adress- und Port-abhängig

NAT-Varianten: Beispiel Symmetric NAT

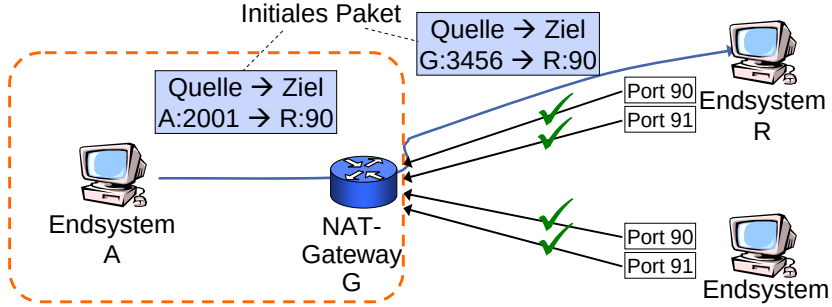
- Restriktivste Variante: Binding je 5-Tupel (Q-IP, Q-Port, Z-IP, Z-Port, Protokoll),
- Binding kann nur von gleichem Endsystem mit passendem Port genutzt werden



NAT-Bindings:	NAT-Filter:
Lokal ↔ Extern	Extern erlaubter Zugriff v.
A : 2001 ↔ G : 3456	R : 90

NAT-Varianten: Full Cone

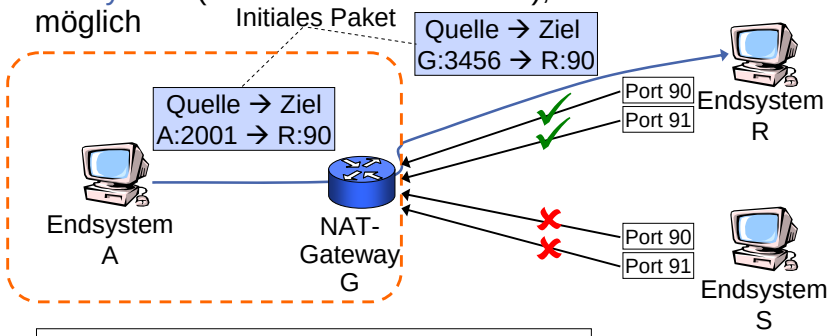
- Liberale Variante: Gleiche externe Adresse für gleiches Paar Quell-Adresse/Quell-Port, Nutzung eines bereits etablierten Bindings durch andere Systeme möglich



NAT-Bindings:	NAT-Filter:
Lokal ↔ Extern	Extern erlaubter Zugriff v.
A : 2001 ↔ G : 3456	* : *

NAT-Variante: Restricted Cone

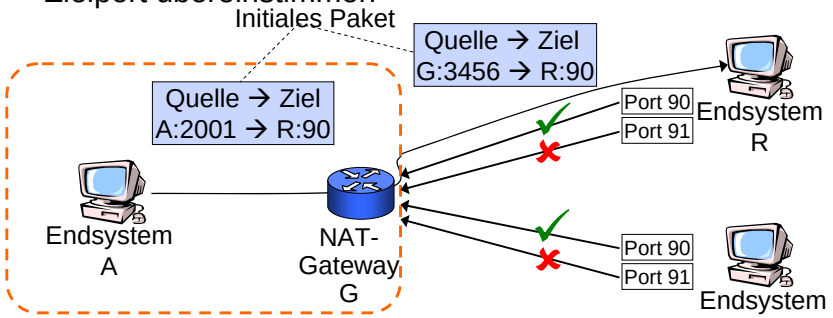
- Binding in Abhängigkeit von Quelladresse/-Port, eingehende Pakete eingeschränkt auf das gleiche Endsystem (Ziel des initialen Pakets), aber andere Ports möglich



NAT-Bindings:	NAT-Filter:
Lokal ↔ Extern	Extern erlaubter Zugriff v.
A : 2001 ↔ G : 3456	R : *

NAT-Variante: Port-Restricted Cone

- Diensteorientiert: Nutzung eines etablierten Bindings von unterschiedlichen Adressen aus, Quellport muss mit Zielport übereinstimmen



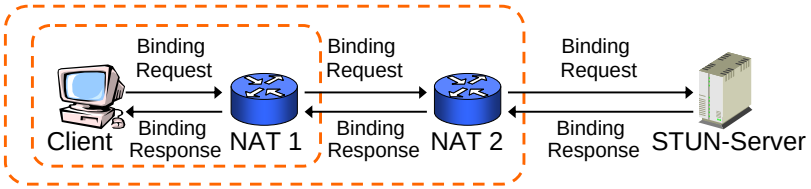
NAT-Bindings:	NAT-Filter:
Lokal ↔ Extern	Extern erlaubter Zugriff v.
A : 2001 ↔ G : 3456	* : 90

NAT-Discovery: STUN

- STUN – Session Traversal Utilities for NAT → UNSAF-Lösung [RFC5389]
  - Client/Server-Protokoll
    - STUN Client:
      - läuft in Anwendung, die ankommende Daten per UDP erwartet, z.B. VoIP Client
      - Anwendung ersetzt private Adressen im Nutzdatenteil vorab
    - STUN Server: kann
      - manuell konfiguriert werden
      - per DNS SRV herausgefunden werden
- STUN ermöglicht das Feststellen
  - des Vorhandenseins eines NAT-Gateways
  - der NAT-Konfiguration für UDP/TCP
  - der verwendeten Adressenabbildung
  - und erlaubt Etablieren und Aufrechterhalten des Bindings

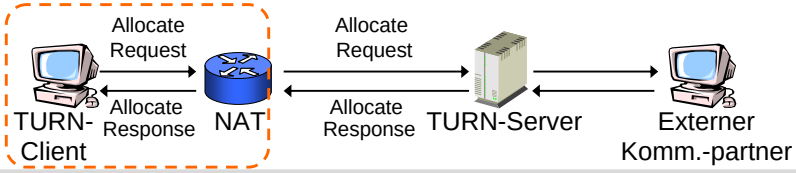
STUN

- Ermöglicht eingehende UDP-Pakete für bestimmte NAT-Varianten
  - nicht für TCP und nicht für symmetrisches NAT
  - primärer Einsatzzweck für VoIP/RTP
- Benötigt STUN-Server im öffentlichen Netz (Default-Port: 3478)
- Binding Request/Response



Relay NAT

- Problem: STUN hilft nicht bei symmetrischen NATs
  - STUN-Erweiterung (früher eigenständiges Protokoll TURN – Traversal Using Relays around NAT) [RFC5766]
    - gleiche Syntax wie STUN, definiert 10 neue Nachrichten
  - nutzt Server mit öffentlichen Adressen im Netz, der als Relay für Medienströme dient (UDP und TCP)
    - Initiale Kapselung von Daten notwendig (Send Indication)
      - Overhead von 44 Bytes
      - Umschalten auf Betrieb ohne Kapselung erfordert Magic Cookie in UDP-Paketen zur Erkennung der Kontrollpakete
    - Erlaubt Client nicht, Server hinter NAT zu betreiben



NAT – Vorteile/Nachteile

- Vorteile
  - schafft Abhilfe bei Adressraumknappheit
  - vermeidet Umnummerieren des Netzwerks bei ISP Wechsel
- Nachteile
  - ist aber kein Sicherheitsmechanismus
  - Verringerte Leistung
  - Zusätzliche Komponente im Pfad, d.h. NA(P)T-Gateway
  - Einschränkung der möglichen Anwendungen (Bruch des E2E-Prinzips), u.a. Einsatz von Sicherheitsmechanismen [RFC3715]
  - Zusätzliche Kosten

NAT – Weitere praktische Probleme

- Fragmente: NAT fehlt Addresseninformation und verwirft Paket
- NATs verfügen oft auch über integrierte ALGs
  - Verhalten undokumentiert, schwer vorauszusagen
  - ALGs veralten: Unterstützung für neueste Anwendungsversion?
  - Vorsicht bei „generischen“ ALGs, die ersetzen auch einfach alles in den Nutzdaten
  - Implementierung häufig beschränkt: Übersetzung von Adressen über Paketgrenzen hinweg bei Fragmentierung nicht vorgesehen
- Timeout für UDP uneinheitlich
- Portnummern können häufiger wechseln (Load Balancing macht‘s nicht besser…)
- Weiteres in [RFC3027]

NAT Resümee

- NAT hat genug Spielraum geschaffen, damit IPv6 entwickelt werden konnte
- NAT schafft eine Vielzahl neuer und schwerwiegender Probleme
  - Verlust der Transparenz und Flexibilität
  - Zusätzliche Komponente verursacht Wartungsaufwand und Kosten
  - erfordert aufgrund des UNSAF-Problems für einige Anwendungen weitere zusätzliche Komponenten wie STUN-Server und Relays mit weiteren Protokollen
    → noch mehr Komplexität!
- Bürdet Protokollentwicklern unnötigen Zusatzaufwand auf, um „NAT-freundliche“ Lösung zu entwickeln

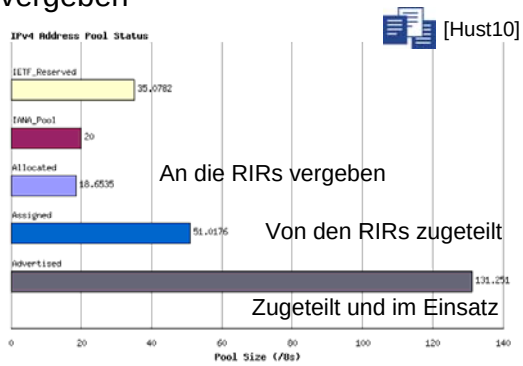
3.2 Motivation für IPv6

- **Anwachsen des Internets**
  - IP-Adressen gehen aus
  - mehr Internet-fähige Geräte, Kleinstgeräte, Sensoren
    → Bedarf wird eher steigen
- **Vereinfachtes Management**
  - Autokonfigurationsmechanismen
- **Wiederherstellung der Kohärenz**
  - Beseitigung von NAT und anderen Speziallösungen
    → ermöglicht Ende-zu-Ende-Sicherheit und Peer-to-Peer-Netze
- **Effizienteres Routing**
  - durch entsprechende inhärente Adresshierarchie
- **Hohe Datenraten**
  - Hochleistungsfähige Zwischensysteme benötigen geeignete Paketformate zur effizienten Bearbeitung

Adressenknappheit?

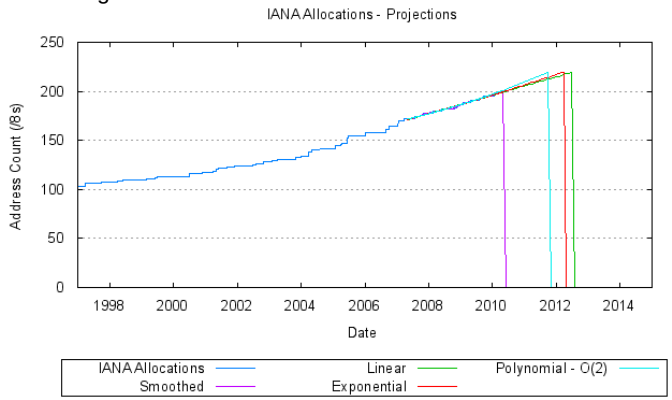
- Wie lange sind denn noch IPv4-Adressen verfügbar?
  Prognose 1994: gehen zwischen 2005–2011 aus
- Adressen werden an Regional Internet Registries (RIRs) in Form von /8-Blöcken vergeben
- Stand April 2010
  - ca. 221 /8-Unicast-Adressblöcke (mit je 2^{24} Adressen=16,7M) insgesamt verfügbar
  - ca. 9% des Adressraums noch übrig (20 /8-Blöcke)
  - keine Aussage über ungenutzten Adressraum

[Hust10]



Prognosen

- Schätzung projiziert Adressraumknappheit 09/2011 (IANA Pool erschöpft) bzw. 05/2012 (RIR Pool erschöpft) [Hust10]
  - Einige dynamische Effekte nicht berücksichtigt wie z.B. Ansturm auf die letzten verfügbaren Adressen



Host Density Ratio

- Beobachtung
  - Adressierungshierarchie beinhaltet immer Verschnitt
  - Adressraum zu voll→ keine Flexibilität mehr bzgl. Änderungen, Wachstum, Mobilität, etc.
- Wann dürfen neue Adressen angefordert werden?
- Maß für Belegungsgrad: Host-Density Ratio

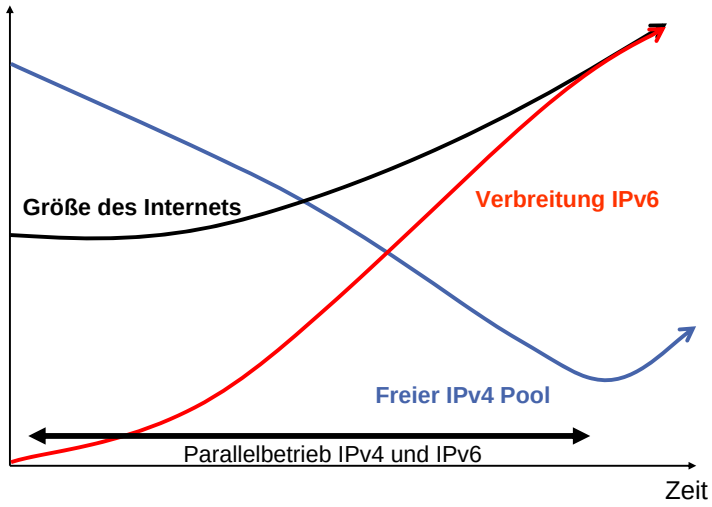
$$HD = \frac{\log(Anzahl\ vergebener\ Objekte)}{\log(max.\ Anzahl\ vergebbarer\ Objekte)}$$

[RFC3194]

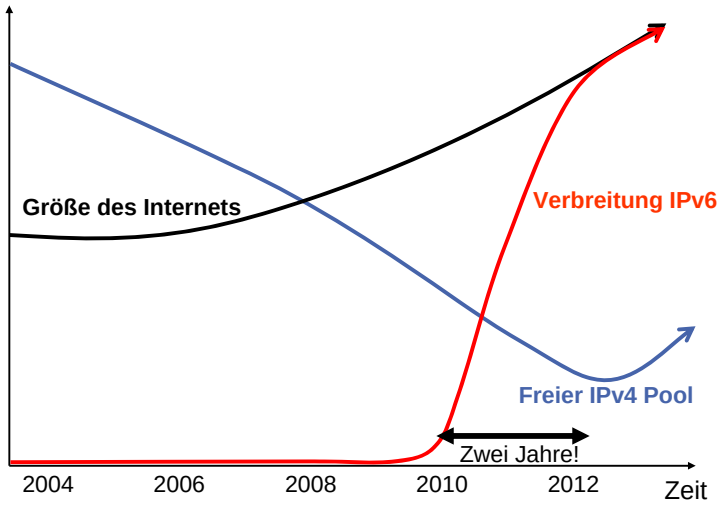
  - RFC 3194: HD-Ratio sollte möglichst <85% sein
  - RFC 4692: HD-Ratio von 94% für IP-Netze eher realistisch
  - Derzeit IPv4 (Mai 2010):
    - 98,2%=log (220,922 - 20)/log(220,922)
    - 96,9% (inkl. 16/8 Reserve)

→ IPv4 Adressraum extrem eng gepackt!

Was hätte passieren sollen...



Was passierte...



Ja und?

- Einsatz und Verbreitung von IPv6 wird länger dauern als angenommen
  - aber zwischenzeitlich werden die IPv4-Adressen ausgehen! Neukunden für ISPs ohne IPv4?
  - auch NAT-basierte Lösungen brauchen IPv4-Adressen
- Reine IPv6-Netze werden entstehen
  - aufgrund operationeller Einfachheit und Kostenersparnis
- Großteil der Inhalte wird aber nach wie vor noch nur über das IPv4-Internet zugänglich sein
  - ein Übergang zu IPv6 dauert seine Zeit
  - reine IPv6-Rechner benötigen Zugang zu über IPv4 bereitgestellten Inhalten
  - IPv4-Systeme benötigen Zugang zu IPv6-Servern

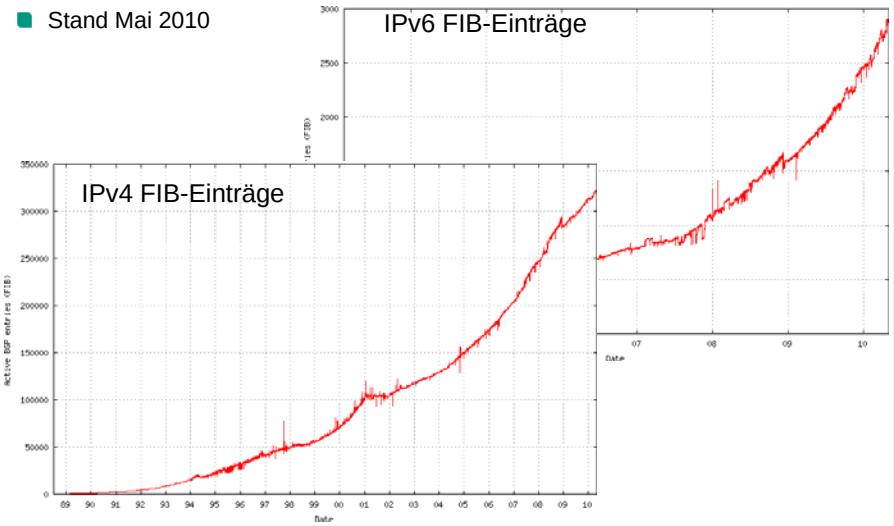
Weitere NAT-Lösungen...

für längere Co-Existenz benötigt, derzeit diskutiert:

- NAT64 v6/v4-Übersetzung NAT64/DNS64
  - Zugriff von IPv6only-Systemen auf nur in IPv4 verfügbaren Inhalte
- Carrier Grade NAT/Large Scale NAT
  - private Adressen innerhalb von ISPs (evtl. sogar gemeinsam genutzt)
  - ISP nutzt nur wenige öffentliche IP-Adressen
  - Robustheit, Skalierbarkeit?
- „Address+Port“: gemeinsame Nutzung einer IP-Adresse
  - innerhalb von ISPs, Kunden erhalten gleiche IP-Adresse, aber anderen Port-Bereich
  - Ports sind dann Teil der „IP-Adresse“
  - viele Probleme, u.a. Fragmentierung, Management, Logging usw.

IPv6 Einsatz derzeit...

- Stand Mai 2010



Keine andere Alternative…als IPv6

- IPv4-Adressen werden bald ausgehen
  - Reserveblock relativ klein (16 /8), praktisch nicht einsetzbar
    - Win95 – Windows Server 2003 akzeptieren diese Adressen nicht
  - Privater Adressraum und NAT helfen auch nicht ewig weiter (Mehrdeutigkeit, Komplexität)
    - für große Firmen/Provider reichen RFC1918-Adressen nicht, brauchen offizielle /8-Präfixe [Dura06]
- Stabile Alternative zu entwickeln dauert >5 Jahre
  - IPv6 ist inzwischen ausgereift (Entwicklung seit 1992)
  - Keine extrem große Änderungen gegenüber IPv4
  - Zu wenig Zeit, um weitere Alternative zu entwickeln
  - Daher: besser jetzt direkt anfangen, IPv6 einzusetzen, um notwendiges Wissen zu erwerben

Kurze Geschichte von IPv6

- 1991:
  - Gründung der Arbeitsgruppe ROAD (Routing and Addressing) [SAMI06]
  - Mailingliste „Big-Internet“
- 1992: Vorschläge, u.a. [Brad06]
  - The „P“ Internet Protocol (PIP)
  - TCP and UDP with Bigger Addresses (TUBA)
  - Simple Internet Protocol (SIP)
- 1993:
  - IPNG Area: Management des „IP the next generation“-Prozess
  - SIP+PIP = SIPP (Simple Internet Protocol Plus)
  - ALE (Address Lifetime Expectations) Working Group
- 1995:
  - RFC 1752, Empfehlung für das IPng-Protokoll
  - RFC 1883 IPv6 Specification
- 1998: RFC 2460 Internet Protocol, Version 6 (IPv6) Specification
  (derzeit immer noch Draft-Standard,
  aber auf dem Weg zum „Full-Standard“) → www.ipv6-to-standard.org
- Sept 2007: IPv6 WG geschlossen, 6MAN WG gegründet

Neuerungen in IPv6 (1)

- Erweiterte Adressierung
  - Erhöhung der Adresslänge von 32 Bit auf 128 Bit
  - Einführung von Anycast-Adressen
    (Kommunikation zum Mitglied einer Gruppe)
  - Jede Schnittstelle besitzt Link-Local-Unicast-Adresse
  - Kryptographisch-generierte Adressen (CGAs)
  - Unique Local IPv6 Unicast Addresses (ULAs)
  - Multicast-Adressen enthalten Reichweite (Scope)
    - kein „Missbrauch“ des Hop Limit/TTL-Feldes mehr wie bei IPv4

Neuerungen in IPv6 (2)

- Schnelle Bearbeitung in Routern durch vereinfachtes Paketformat
  - Standard-Paketkopf mit fester Länge und nur 8 Feldern (13 bei IPv4)
  - Verschieben von Optionen in flexible Paketkopferweiterungen
  - Kopferweiterungen sind an 64-Bit-Grenzen ausgerichtet
  - Keine (Kopf-)Prüfsumme → UDP-Prüfsumme jetzt zwingend
  - Keine Hop-by-Hop-Fragmentierung
    → Nur Ende-zu-Ende-Fragmentierung plus Path-MTU-Discovery
  - IPv6-Spezifikation erfordert Link-MTU >= 1280 Bytes (IPv4: 68 Bytes)

Neuerungen in IPv6 (3)

- Unterstützung von Ressourcenreservierung/ Verkehrslenkung
  - Flussmarke (Flow Label) und Verkehrsklasse (Traffic Class) pro IPv6-Paket
- ICMPv6: Erweiterung von ICMP
  - Zuvor getrennte Protokolle direkt in ICMP integriert
    - IGMP und ARP, d.h. Gruppenverwaltung und Adressauflösung
  - Neighbor Discovery (als Teil des neuen ICMP) [RFC4861]
    - Adressauflösung (IPv6- auf MAC-Adressen), inkl. Erkennung doppelter Adressen und Detektion von Ausfällen
    - Erkennen des nächsten Routers sowie des Netzwerk-Präfixes

Neuerungen in IPv6 (4)

- Automatische Systemkonfiguration [RFC4862] (Stateless Autoconfiguration)
  - Zustandslos: Präfix über Router Advertisements plus EUI-64-Interface-ID
  - Zustandsbehaftet: traditionelles DHCP (Dynamic Host Configuration Protocol)
- Bessere Unterstützung mobiler Systeme MK (MobileIPv6)
  - Bewegungserkennung und Adressenzuweisung durch automatische Systemkonfiguration
  - Die Option Binding Update im Destination-Options-Header ermöglicht die direkte Umleitung der IP-Pakete an den aktuellen Standort

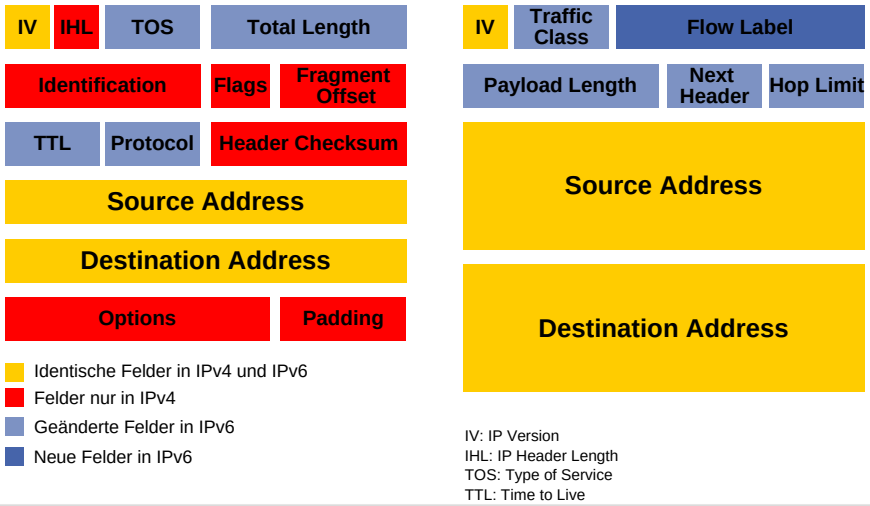
Neuerungen in IPv6 (5)

- Berücksichtigung von Sicherheitsaspekten
  - Zwingende Unterstützung von Sicherheitsmechanismen für alle IPv6-Knoten! (aber kein zwingender Einsatz)
  - Unterstützung von Authentifizierung und Datenintegrität
    - Authentifizierung/Integritätssicherung: Authentication Header (AH)-Erweiterungskopf
    - Verschlüsselung: Encapsulating Security Payload (ESP)-Erweiterungskopf
    - auch für IPv4 verfügbar...
  - Secure Neighbor Discovery (SEND)
    - verhindert „ARP-Spoofing“, d.h. das Hijacking von IP-Adressen
    - verhindert Missbrauch des Autokonfigurationsmechanismus für Angriffe

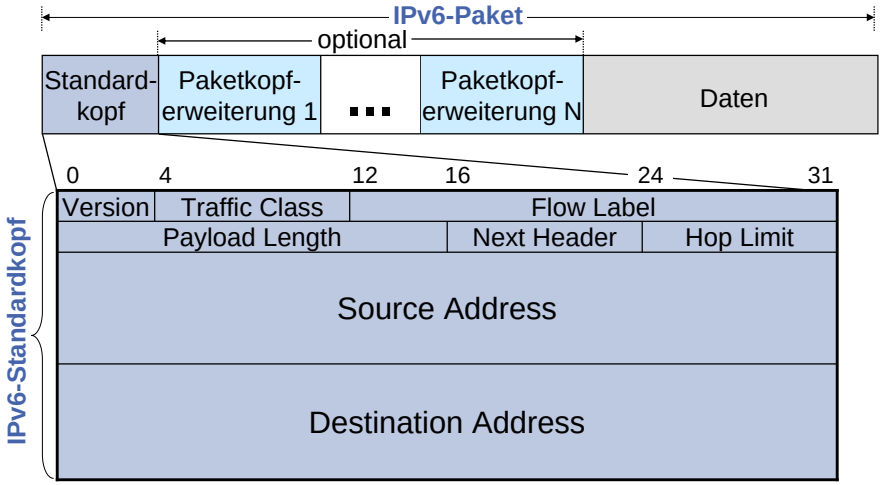
Neuerungen in IPv6 (6)

- Vereinfachung der Administration bzgl. Adressen
  - Vermeidung der Renummerierung von Subnetzen bei Änderung des ISP
  - Multi-Homing: Verwendung mehrerer Adressen gleichzeitig, aber skalierbar →Shim6 Working Group
- Nachteile
  - Zusatzaufwand durch IP-Paketkopf nun mindestens 40 Bytes (IPv4: 20 Bytes)
    - z.B. IP-Telefonie: 20% statt 11% Overhead bei 8 Bit/8 KHz unkomprimiert
  - Adressen noch weniger handhabbar → DNS zwingend
  - Nicht abwärtskompatibel (IPv4 ist keine IPv6-Variante)
  - Zusätzlicher Aufwand für Betriebspersonal

Vergleich Kopffelder IPv6 und IPv4

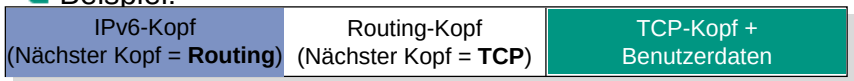


IPv6-Paketformat



IPv6-Paketkopferweiterungen (1)

- Feld „Nächster Kopf“ (Next Header): Typ der nachfolgenden Paketkopferweiterung
  - Beispiel:



Übertragungsreihenfolge

- Definition von sechs Kopferweiterungen. Empfohlene Reihenfolge:
  1. IPv6-Kopf
  2. Knoten-zu-Knoten-Optionen (Hop-by-Hop Options)
  3. Optionen für Zwischenziele gemäß Routing-Kopf (Destination Options (1))
  4. Routing (Routing Header)
  5. Fragmentierung (Fragment Header)
  6. Authentifizierung/Integritätssicherung (AH – Authentication Header)
  7. Verschlüsselung (ESP – Encapsulating Security Payload)
  8. Optionen für endgültiges Ziel (Destination Options (2)) höhere Schicht, z. B. TCP oder UDP

IPv6-Adressdarstellung (1)

- Textuelle Repräsentation von IPv6-Adressen (128 Bit) [RFC4291]
  - Dotted-Decimal wie bei IPv4 nicht mehr sinnvoll
  - Hexadezimaldarstellung von 8 durch Doppelpunkte getrennte 16-bit-Worte
    z.B. `2001:0db8:0204:0001:0206:5bff:fe30:bbd2`
- Vereinfachungen
  - Führende Nullen unterdrücken
    z.B. `2001:db8:204:1:206:5bff:fe30:bbd2`
  - Ein oder mehrere Gruppen von 16-bit Nullwerten können durch zwei direkt aufeinanderfolgende Doppelpunkte abgekürzt werden (aber nur ein einziges mal)
    z.B. `fe80::206:5bff:fe30:bbd2`
  - Kanonische Schreibweise wünschenswert

IPv6-Adressdarstellung (2)

- IPv4-mapped/IPv4-compatible Adressen
  - Die letzten 32 Bit können in der IPv4-üblichen dotted-decimal Schreibweise dargestellt werden, z.B.
    `::ffff:141.3.71.6` (anstatt `::ffff:8d03:4706`)
- Präfixschreibweise wie bei CIDR, d.h. Präfixlänge kann angehängt werden, z.B.
  `2001:db8:204::/48`
- Schreibweise mit eckigen Klammern
  - URLs, z.B. [RFC3986]
    `http://[2001:db8:204:1:290:27ff:fe72:b48]:8088/`
    `ldap://[2001:db8::7]/c=GB?objectClass?one`
  - Secure Copy, z.B.
    `scp user@[2001:db8:204::1]:datei.txt kopie.txt`

IPv6-Adressierungsarchitektur

- Adresstypen [RFC4291]
  - Unicast
    - Identifikator für ein einzelnes Interface
  - Anycast
    - Identifikator für eine Menge von Interfaces
    - Paket an solche Adresse wird an eins aus dieser Menge ausgeliefert (üblicherweise das „nächstgelegene“)
  - Multicast
    - Identifikator für eine Menge von Interfaces
    - Paket an solche Adresse wird an alle aus dieser Menge ausgeliefert
- IPv6-Adressen sind Netzwerkschnittstellen (Interfaces) und nicht Netzknoten zugeordnet
- Es gibt keine Broadcast-Adressen mehr!!
  - deren Funktion wird durch Multicast-Adressen übernommen

IPv6 Adresstypen

- Adresstyp wird durch führende Bits einer Adresse festgelegt

	Führende Bits	als Präfix
Unspecified Address	00...0 (128 Bit)	::/128
Loopback Address	00...1 (128 Bit)	::1/128
Multicast	11111111	ff00::/8
Link-Local Unicast	1111111010	fe80::/10
Global Unicast	(alles übrige)	

- Besondere Adressen
  - IPv4-mapped Address ::ffff:0:0/96
    zur Darstellung von IPv4-Adressen im IPv6-Format, z.B. ::ffff:141.3.70.14
  - IPv4-compatible Address::/96
    (sollte nicht mehr verwendet werden, wurde früher für automatisches Tunneln von IPv6-Paketen über IPv4-Netze eingesetzt, z.B. ::141.3.70.14)

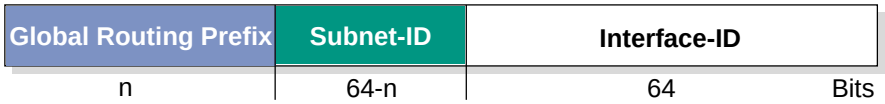
Scoped Address Architecture

- Bereichsbegrenzung (Scope) bei IPv6 fester Bestandteil der Architektur [RFC4007]
- Multicast-Adressen besitzen expliziten Bereich, der in der Adresse kodiert ist
- Momentan für Unicast-Adressen definiert
  - Global Scope
  - Link-Local Scope

→ Im Folgenden: Global Scope und Link-Local Scope Unicast-Adressen

Adressformat – Global Unicast

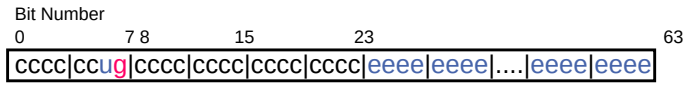
- Struktur [RFC3587]



- Global Routing Prefix: ähnlich wie derzeit CIDR bei IPv4
- Subnet-ID: eindeutige Kennzeichnung eines Subnetzes
- Interface-ID (IID): eindeutige Identifikation einer Schnittstelle innerhalb eines Subnetzes
  - Kann z.B. auch kryptographisch generiert werden (s. CGA – Cryptographically Generated Addresses)
- Derzeit zugeordneter Bereich für Global Unicast Adressen: 001_2 = 2000::/3
  - Sollte nicht als Formaterkennungspräfix verwendet werden!

Bedeutung der Interface-ID

- Ziel: Eindeutigkeit im Subnetz für automatisches Generieren der IP-Adresse (Auto-Konf.)
- Wird nicht zur Ermittlung der MAC-Adresse bzw. Zustellung im LAN verwendet! (häufiges Missverständnis) → das ist die Aufgabe von Neighbor-Discovery
- Grundlage bilden eindeutige IEEE EUI-64-Adressen:

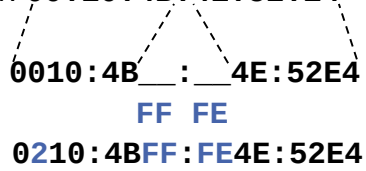


(„c“ Company, „u“ universal/local bit, „g“ individual/group bit, „e“ id assigned by company)

- IID kann auch manuell oder zufällig zugewiesen oder kryptographisch generiert werden

Generieren modifizierter EUI-64 Adressen

- Generieren modifizierter EUI-64-Adressen aus IEEE 802 48-Bit MAC-Adressen [RFC4291]
  - Einfügen von **FFFE** in der Mitte ab dem 25. Bit (direkt vor den e-Bits der MAC-Adresse)
  - u-Bit setzen (da Bedeutung des Bits in IPv6 gegenüber EUI-64 umgekehrt wegen spezieller Adressen wie ::1)
  - Bsp.: HW-Addr: **00:10:4B:4E:52:E4** (48bit IEEE)



→ IID: **0210:4BFF:FE4E:52E4**

- Für IEEE EUI-64-Adresse muss lediglich das u-bit invertiert werden → modifiziertes EUI-64-Format

Schutz der Privatsphäre

- Nachteil durch Generieren der Interface-ID aus IEEE-Mac-Adressen:
  - IID bleibt gleich, auch wenn Zugang gewechselt wird
  - In einigen Fällen könnten Bewegungsprofile leichter erstellt werden
    → Abhilfe: Interface-ID periodisch zufällig ändern
- IID wird zufällig generiert und mehr oder weniger häufig gewechselt [RFC4941]
  - MD5-Hash über Historie bzw. initialen Zufallswert sowie „normal generierte“ Interface-ID
  - Kollisionen werden über Duplicate Address Detection erkannt

Link-Local Unicast-Adressen

- Jedes Interface muss zumindest eine Link-Local-Unicast-Adresse besitzen
  - erreicht Systeme innerhalb eines Netzsegmentes bzw. im gleichen Subnetz
- Bilden einer Link-Local Address: fe80::/64 + Interface-ID
- Nur link-lokal gültige Adressen, relativ zum jeweiligen Interface, d.h. Interface muss ggf. angegeben werden
  - Erfordert neue Socketschnittstellen
  - Erfordert Möglichkeit der zusätzlichen Angabe (Scope Identifier), z.B. fe80::206:5bff:fe30:bbd2%eth0
  - Herausfinden anderer Link-Local-Adressen durch **`ping6 -I eth0 ff02::1`** (Angabe des Interfaces ist notwendig)
- Einsatz der Adresse bei ICMP-Kommunikation mit Router

Keine Site-Local Unicast Adressen

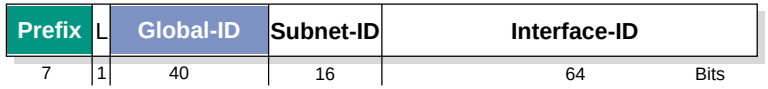
- Ursprüngliches Konzept für „Private“ Adressen, die nur innerhalb einer „Site“ gültig sind
  - können/dürfen nicht global geroutet werden
  - für unverbundene Netzwerke [RFC3879]
- Site-Locals wurden inzwischen zurückgezogen
- Probleme im Wesentlichen begründet durch
  - 1. Mehrdeutigkeit/Nicht-Eindeutigkeit der Adressen
  - 2. Unscharfe Definition einer „Site“
- Kein Einsatz von Site-Locals!
- Bessere Lösung: ULAs, siehe nächste Folie

Unique Local IPv6 Unicast Addresses

- Neuer Typ von IPv6 Unicast Adressen, die [RFC4193]
  - für lokale Kommunikation gedacht sind (innerhalb einer „Site“)
  - üblicherweise nicht im globalen Internet geroutet werden
  - aber global eindeutig sind
- Eigenschaften von ULAs
  - Global eindeutiges Präfix (zumindest mit hoher Wkt)
    - Wohlbekanntes Präfix (**`FC00::/7`**) zum leichten Filtern
  - Keine Konflikte
    - beim Verbinden oder Zusammenlegen zweier „Sites“
    - bei gelegentlichen Lecks durch DNS oder Routing
  - ISP-unabhängig, ohne dauerhafte oder intermittierende Konnektivität
  - Anwendungen behandeln diese Adresse wie „normale“ global gültige Adressen

Unique Local Adressen – Aufbau

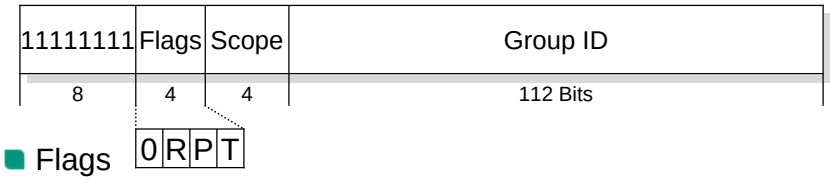
- Aufbau:



  - Prefix **`FC00::/7`** (belegt 0,781% des Adressraums)
    - damit stehen 2050 für eine geschätzte Weltbevölkerung von 9.3 Milliarden immer noch 236-/48-Präfixe pro Person zur Verfügung
  - L=1, wenn Präfix lokal zugewiesen, L=0 reserviert für zukünftige Methoden
  - Global-ID: wird zufällig generiert, z.B. mit Hilfe von rightbits(SHA-1(64-bit NTP Zeitstempel | EUI-64 ID), 40)
- ULAs haben globalen Geltungsbereich
- Site kann mehrere solcher Adressen gleichzeitig verwenden

Multicast-Adressen

- Struktur [RFC4291]



  - Flags
    - Transient-Bit
      - 0 = permanente, „wohlbekannte“ (von der IANA vergebene) Gruppenadresse
      - 1 = dynamisch vergebene, „transiente“ Gruppenadresse
    - P-Flag für Unicast-Prefix-based IPv6 Multicast [RFC3306]
    - R-Flag für eingebettete Rendezvous-Point-Adressen [RFC3956]

Multicast-Adressen (2)

- Scope:
  16 mögliche Gültigkeitsbereiche/Reichweiten
  - 0 und 15 sind reserviert
  - 1: interface-local
  - 2: link-local
  - 4: admin-local
  - 5: site-local
  - 8: organization-local
  - 14: global
  - Rest bisher nicht vergeben
- Spezielle Adressen
  - All-nodes: ff02::1 (link local), ff01::1 (interface-local)
  - All-routers: ff02::2 (link local), ff05::2 (site-local)
  - Solicited-Node: ff02:0:0:0:1:ffxx:xxxx
    (x: Low order 24-bits der Unicast/Anycast-Adresse)

Multicast-Adressen → Link Layer

- Abbildung von einer IP-Multicast-Adresse auf Link-Layer-Adresse?
  (z.B. nötig für Neighbor Discovery)
- Beispiel IPv6 → auf Ethernet MAC-Adressen:
  33 + 33 + die vier letzten Oktetts der IPv6-Multicast-Adresse, z.B. [RFC2464]
  - All-nodes (Link-Local): ff02::1 → 333300000001
  - Solicited Node: ff02::1:ff0c:5e44 →3333ff0c5e44
- Ethernet-Switch kann nur effizient Pakete verteilen, wenn er die Gruppenzuordnung kennt
  - Multicast Listener Discovery-Snooping notwendig
  - Switch muss Schicht-3-Protokoll beherrschen!

Anycast-Adressen

- Zustellung eines IP-Pakets an einen Zielknoten aus einer Gruppe von Zielknoten (identifiziert durch die Anycast-Adresse)
- IP-Paket wird dem nächstgelegenen (nach Routingmetrik) Zielknoten zugestellt
  - Routing sucht den nächstgelegenen Zielknoten
  - Quelle hat keine Einflussmöglichkeit auf die Wahl des Zielsystems
  - Eintragen von „Host-Routen“, ggf. aggregierbar, sonst Skalierbarkeitsprobleme für globale Anycastadressen
- Anycast-Adressen sind syntaktisch nicht unterscheidbar von Unicast-Adressen

Zusammenfassung Adressen

- Global Unicast
  - 64-bit Netzwerkteil (Global Routing Prefix + Subnet)
  - 64-bit Interface Identifier
- Unique Local Unicast
  - Spontan generierte Adresse, eingeschränkter Geltungsbereich
- Link-Local Unicast
  - besitzt jedes Interface
  - erreicht On-Link-Systeme (gleiches Subnetz)
- Multicast
  - Gruppenadresse, Auslieferung an alle Mitglieder
- Anycast
  - Gruppenadresse, Auslieferung an (irgend)ein Mitglied

3.3 ICMPv6

- Koordination und Kontrolle des IP-Verkehrs durch neues ICMP (Internet Control Message Protocol)
  - Aufgaben des alten ICMP, alten IGMP und alten ARP
  - weitere neue Aufgaben
- Arten von ICMP-Nachrichten
  - Fehlernachrichten
  - Informationsnachrichten
    - Echo-Nachrichten (Ping)
    - Verwaltung von Multicast-Gruppen (früher IGMP, jetzt MLD)
  - Neighbor Discovery: IPv6 Knoten am selben Link können mit ND [RFC4861]
    - Die Anwesenheit anderer „On-Link“-Systeme erkennen
    - gegenseitig die Link-Layer Adresse bestimmen
    - Router finden
    - Erreichbarkeitsinformation über Pfade zu aktiven Nachbarn erhalten

Konfigurationsmechanismen

1. Automatisches Konfigurieren von IP-Adressen [RFC4862]
   a) Link-lokale Adresse:
      – wird für Kommunikation zur Konfiguration mit ICMP benötigt
      – zur Kommunikation mit On-link-Systemen, d.h. direkten Nachbarn im gleichen Subnetz
   b) Globale Adresse:
      – zur Kommunikation mit Off-Link Destinations benötigt
2. Erkennung von Adresskonflikten (Duplicate Address Detection)
3. Auffinden von Routern und Konfigurationsparametern (Router Discovery)
4. Auffinden von On-link-Systemen (Adressauflösung)
5. Erkennung von nicht mehr erreichbaren Nachbarn (Neighbor Unreachability Detection)

Verwendung von Multicast

- Einsatz von Multicast für
  - Duplicate Address Detection
    - Solicited-Node Multicast Address
  - Router Discovery
    - All routers ff02::2
    - All nodes multicast address ff02::1 (ohne MLD)
  - Neighbor Discovery
    - Solicited-Node Multicast Address
- Multicast Listener Discovery (MLD)
  - Zur Verwaltung von Gruppenmitgliedschaften in einem Subnetz
  - Muss für alle Multicast-Adressen ausgeführt werden, insbesondere für die Solicited-Node Multicast Address
  - Nicht notwendig für Beitritt zu ff02::1
  - Link-Lokale Adressen benötigen keine Zustände im Router
  - Einsatz bei ND: ermöglicht Optimierung für Snooping Switches...

Adressenkonfiguration

- Adresskonfiguration Alternativen
  - Stateless
    - Für jedes Präfix in Präfixliste generiere Adresse durch Kombination aus Präfix und Interface-ID
    - Duplicate Address Detection (DAD) notwendig
  - Stateful
    - per DHCP zugewiesene Adresse
    - Duplicate Address Detection notwendig

Zustandslose Adressautokonfiguration

- Stateless Address Autoconfiguration vereinfacht das Anschließen von IPv6-Hosts [RFC4862]
- Für jedes IPv6-Interface sind die folgenden Schritte notwendig
  - Erzeugen einer Link-Lokalen Adresse
  - Generieren globaler Adressen
  - Für jede Adresse: Durchführen der Erkennung doppelter Adressen (Duplicate Address Detection)
- Adressen werden aus Host-lokaler Information und Router-Information gebildet:
  - Host: Interface-ID
  - Router: bekanntgegebene Präfixe
- Erfordert minimale Konfiguration der Router (falls überhaupt)

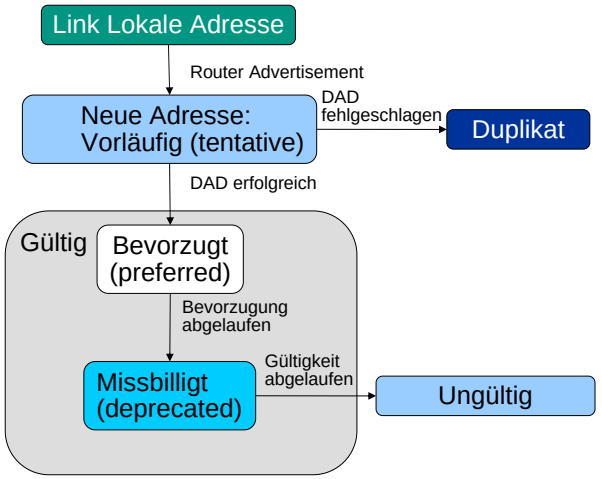
Adressautokonfiguration

- Das Bilden einer globalen Adresse benötigt die vom Router verteilten Präfixe
- Ohne Router nur Bilden von Link-lokalen Adressen möglich → ausreichend für Kommunikation zwischen On-Link-Systemen
- Zustandslos:
  - Router merkt sich keinen Zustand über vergebene Adressen an Host (im Gegensatz zu traditionellem DHCP)
  - Adressen besitzen trotzdem eine Gültigkeitsdauer (ggf. unendlich)
    - ermöglicht „sanftes“ Umnummerieren des Netzes mit Übergangsphase

Bestimmung der Link-lokalen Adresse

- Autokonfiguration der Link-lokalen Adresse
  - Erzeugen der Interface-ID aus MAC-Adresse
  - Ergänzen des Präfix fe80::0 mit IID
  - ergibt zusammen eine vorläufige (tentative) Link-lokale Adresse
  - Durchführen der Duplicate Address Detection
    - Join zur All-Nodes Multicast Address (ff02::1)
    - Bilden der Solicited-Nodes-Multicast-Address
    - Join zu dieser Adresse → MLD
    - Neighbor Solicitation an Solicited-Nodes-Multicast-Address der vorläufigen Link-lokalen Adresse, Quelladresse = „::“
      - zufällige Verzögerung
      - wird periodisch wiederholt
    - Falls kein NA innerhalb bestimmter Wartezeit empfangen, gilt Adresse als eindeutig

Adressenkonfiguration



- Solange Adresse „Preferred“ ist, können Anwendungen sie verwenden
- „Deprecated“ Adressen dürfen nicht mehr für neue „Verbindungen“ verwendet werden

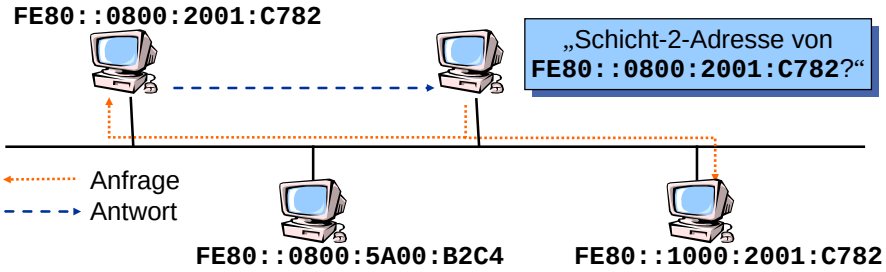
Konzeptionelle Datenstrukturen (1)

- Es gibt vier verschiedene konzeptionelle Caches in jedem Endsystem:
  - Neighbor Cache
    - Enthält direkte Nachbarn, zu denen kürzlich gesendet wurde
    - „On-Link“ Unicast IP-Adresse → MAC-Adresse
  - Destination Cache
    - Kürzlich benutzte On-Link/Off-Link-Ziel-IP-Adressen
    - Abbildung Ziel-IP-Adresse → Next-Hop-IP-Adresse
  - Präfix-Liste
    - Gelernte On-Link-Präfixe aus Router-Advertisements
    - Wird zur Bestimmung von On-Link-Systemen verwendet
    - Gültigkeitszeitraum (auch unendlich möglich z.B. für Link-Local-Adresse)
  - Default Router Liste
    - Liste von möglichen Routern (mit Gültigkeitszeitraum)
    - Default Router wird aus dieser Liste bestimmt

Neighbor Cache – Zustände

- INCOMPLETE
  - Adressauflösung noch im Gang, MAC-Adresse noch nicht bestimmt
- REACHABLE
  - Nachbar war vor kurzem (innerhalb von wenigen 10s) erreichbar
- STALE
  - Nicht bekannt, ob Nachbar noch erreichbar. Erreichbarkeitstest wird zurückgehalten, bis mit dem Nachbar kommuniziert wird.
- DELAY
  - Nicht bekannt, ob Nachbar noch erreichbar, aber Verkehr wurde kürzlich zu ihm geschickt. Ermöglicht Abwarten von Feedback höherer Schichten.
- PROBE
  - Neighbor Solicitation im Gange

Adressauflösung durch Neighbor Discovery (ND)



- Problem: Ermittlung der Schicht-2-Adresse zu einer (On-Link) IPv6-Adresse
- Lösung: Anfrage mittels Neighbor Solicitation (NS)
  - Rechner hören auf Solicited-Node Address
  - Neighbor Solicitation an die Solicited Nodes Address des Zielrechners – kein(!) Broadcast
  - Zielrechner übermittelt seine Schicht-2-Adresse an den anfragenden Rechner in Neighbor Advertisement (NA) (Unicast)

Adressauflösung – Übermittlung von Link-Layer-Adressen

- Link-Layer-Adressen werden in entsprechenden Optionen der Neighbor Discovery-Nachrichten übermittelt
  - Source Link-Layer Address Option (SLLAO)
    - Enthält Link-Layer-Adresse des Senders des NS, RS oder RA
  - Target Link-Layer Address Option (TLLAO)
    - Enthält Link-Layer-Adresse des Ziels (Verwendung in NA und Redirect)
  - MAC-Adressen des Link-Layer-Rahmens werden für Adressauflösung nicht ausgewertet
  - Verwendung des gleichen Mechanismus unabhängig von des jeweiligen Schicht-2-Typs (im Gegensatz zu ARP)

Adressauflösung – Details (1)

- Adressauflösung wird nicht für Multicast-Adressen durchgeführt
- ND stellt bidirektionale Erreichbarkeit fest
- Anfragender Knoten
  - Neighbor Solicitation an Solicited-Node Address
    - Quell-IP-Adresse des ausgehenden Interfaces
    - Unicast Ziel-IP-Adresse wird in NS als Target Address mitgeführt
    - eigene MAC-Adresse wird in Source Link-Layer-Address Option (SLLAO) des NS mitgeteilt (nicht notwendig für Unicast NS)
  - Ausgehende Pakete müssen bis zur Antwort zwischengespeichert werden
  - Neighbor Cache Eintrag bekommt Zustand INCOMPLETE

Adressauflösung – Details (2)

- NS empfangender Knoten
  - falls Quelladresse nicht „::“ (unspecified) und SLLAO vorhanden, Eintrag (→ INCOMPLETE) bzw. Aktualisierung (→ STALE) des Neighbor Cache Eintrags
  - Neighbor Advertisement wird zurückgeschickt
    - per Multicast an All-Nodes, wenn Quelladresse des NS unspezifiziert war (→ DAD), Solicited Flag= 0
    - sonst: per Unicast an Quelladresse des NS, Solicited Flag=1
    - TLLAO muss angehängt werden, wenn Zieladresse des NS Multicast-Adresse war
      - andernfalls Link-Layer-Adresse im Neighbor Cache des anfragenden Nachbarn bereits vorhanden (denn NS wurde ja empfangen)
    - Falls SLLAO in NS fehlt, muss Neighbor Discovery für Zuschicken des NA durchgeführt werden
    - Empfang eines Solicited NA (Solicited Flag=1) zeigt bidirektionale Erreichbarkeit an

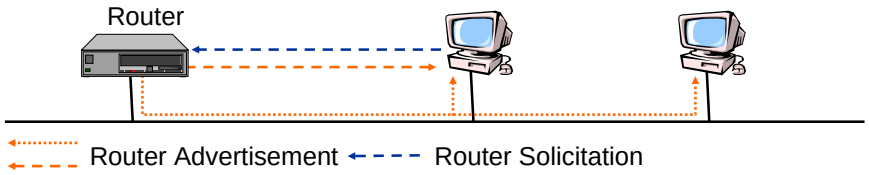
DAD – Details

- Prinzip: schicke NS von unspezifizierter Quelladresse („::“) mit Ziel der vorläufigen Adresse
- Join an All-Nodes Multicast Address
  - falls anderes System bereits mit gleicher Adresse konfiguriert
- Join für Solicited-Node Multicast Address
  - sollte zufällig verzögert werden → Stauvermeidung im Falle gleichzeitigen Wiederanfahrens
  - Erkennung falls zwei Knoten gleichzeitig DAD für gleiche Adresse durchführen
- NS an Solicited-Node Multicast Address der selbst konfigurierten neuen Adresse (Tentative Address) schicken, Absendeadresse ::/128
- Warten bis max. 3s auf NA, Wiederholung NS nach 1s
- Wenn NA oder NS für gleiche Adresse gehört wird, dann zieht sich Knoten zurück

Router-Erkennung (Router Discovery)

- Problem: Ermittlung eines Routers zum Senden von Paketen an Rechner außerhalb des eigenen Netzsegments (Off-Link Ziele)
- Lösung: Feststellen erreichbarer Router mit Router Discovery
  - Unsolicited RAs: Router senden „periodisch“ Router Advertisement-Nachrichten an die „All Nodes“-Adresse FF02::1.
  - Solicited RAs: Rechner können mittels Router Solicitation explizit ein Router Advertisement anfordern, welches dann zeitlich zufällig verzögert
    - per Multicast an All-Nodes gesendet wird (Regelfall)
    - oder per Unicast gesendet werden kann

Router-Erkennung (Router Discovery)

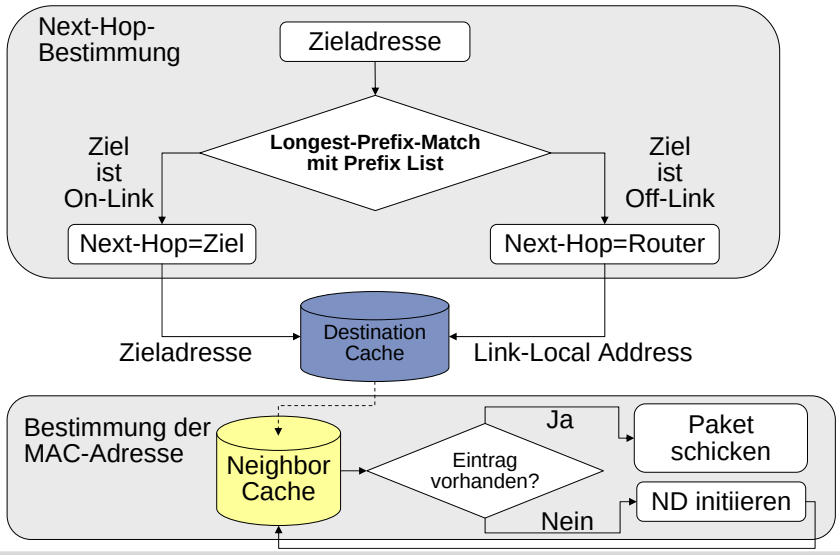


- Probleme:
  - Router-Adresse unbekannt: welche Zieladresse verwenden?
  - Schicht-2-Adresse des anfragenden Systems für Antwort notwendig
- Router merkt sich SLLA bereits in Neighbor Cache
- Router Discovery benötigt Multicast und Link-lokale Adresse
- Router Advertisement liefert u.a.
  - Router Adress(en) als Link-lokale Adresse, ggf. auch SLLA
  - Eintrag für Default Router List
  - Präfixe (On-link)
  - Konfigurationshinweis M/O Flags: Stateless oder Stateful (DHCP)

Präfix-Erkennung

- Problem: Absender eines IP-Paketes muss feststellen, ob sich der Zielrechner im eigenen Subnetz befindet (direktes Senden oder Senden über Router)
- Lösung: Entscheidung basierend auf dem Präfix des eigenen Subnetzes
  - Router-Advertisement-Nachrichten enthalten Präfix-Listen des Subnetzes
  - Vergleich der Zieladresse mit den Präfixen durch logische UND-Verknüpfung
  - Entspricht das Präfix der Zieladresse einem Präfix des Subnetzes, so wird das Paket direkt gesendet; ansonsten wird es an einen Router übermittelt

Sende-Algorithmus



Neighbor Unreachability Detection

- NUD wird nicht für Multicast-Adressen durchgeführt
- Nachbar erreichbar, falls Pakete als Bestätigung eingehen, dass er erreichbar ist (auf IP Ebene)
  → Zwei Möglichkeiten:
  - Hinweis über Transportprotokoll möglich (z.B. Empfangen von TCP ACKs): „Forward Progress“
  - Empfang einer NA-Nachricht auf eine NS-Nachricht (gezielt an Unicast-Adresse des Nachbarn → Unicast NS)
- Dadurch schnellere Überprüfung der Neighbor Cache Einträge möglich

Source Address Selection

- Schnittstellen können mehrere IPv6/IPv4-Adressen besitzen, welche sollte wann verwendet werden? [RFC3484]
- Einheitlicher Algorithmus wünschenswert
  - Policy Hook, um Standardverhalten abzuändern
    - Priorität und Label
  - Regeln zur Wahl der Quelladresse
    1. Bevorzuge gleiche Adresse wie Zieladresse
    2. Bevorzuge entsprechenden Scope (kleiner Scope bevorzugt)
    3. Vermeide abgelehnte Adressen
    4. Bevorzuge Heimadressen
    5. Bevorzuge die Adresse des ausgehenden Interfaces
    6. Bevorzuge passendes Label
    7. Bevorzuge öffentliche Adressen
    8. Bevorzuge längstes passendes Präfix

Secure Neighbor Discovery (SEND)

- Großes Problem bei IPv4
  - ARP Spoofing und ARP Cache Poisoning im gleichen IP-Subnetz → weitreichende Man-in-the-Middle-Angriffe möglich

→ Secure Neighbor Discovery (SEND) [RFC3971]

- Sicherheitsmechanismen
  - Anforderungen: ohne IPSec, vorkonfigurierte Adressen, PKI, vertrauenswürdige Server auskommen
  - → Cryptographically Generated Adresses (CGA) [RFC3972]
  - Interface-ID mittels Hash-Funktion (derzeit noch SHA-1) aus öffentlichem Schlüssel, Zufallszahl, weiteren Parametern gebildet
  - Zugehörigkeit einer CGA zu öffentlichem Schlüssel kann geprüft werden (kein Fälschen von IP-Adressen möglich)
  - Nachrichten können mit privatem Schlüssel signiert werden

SEND

Sicherung des Neighbor Discovery

- Verzicht auf PKI
  → Zugehörigkeit öffentlicher Schlüssels zu Identität des Endsystems nicht prüfbar
  → keine Zugriffskontrolle möglich, aber Maskerade-Angriffe werden verhindert
- SEND-Lösung: Optionen für Neighbor/Router-Discovery-Nachrichten, die RSA-Signatur tragen

Sicherung des Router Discovery

- Router Discovery ermöglicht einfachen Man-in-the-Middle oder DoS-Angriff → Absicherung notwendig
- Identität von Routern kann anhand von Zertifikaten und Zertifizierungspfaden überprüft werden.
  - Überprüfung zurückgezogener Zertifikate unklar

DNS Discovery

- Autokonfiguration wichtiger Vorteil von IPv6, es fehlen aber weitere Angaben wie z.B. Adressen der DNS-Server
- Drei verschiedene Ansätze: [RFC4339]
  - Wohlbekannte (Anycast-) Adressen
  - Erweiterung der Autokonfiguration durch neue Router Advertisement Option „DNS Server“ (je Server eine Option) [RFC5006]
  - Stateless DHCPv6 [RFC3736]
- Möglicherweise auch alle drei Methoden kombinierbar

3.4 Übergang IPv4 → IPv6

- Einsatz von Tunneln
- IPv6 Pakete können über reine IPv4-Netze durch Tunnel übertragen werden
  - IPv6-Paket wird in IPv4-Paket eingepackt
  - Zusatzaufwand (IPv4-Kopf + Ein-/Auspacken) verringert Leistung
- Tunnelvarianten
  - Konfigurierte Punkt-zu-Punkt-Tunnel
    - bildeten Grundlage für das ehemalige „6bone“-Overlay-Netz
  - Automatische Tunnel
    - Direkter automatisch etablierter Tunnel zwischen zwei Dual-Stack-Hosts
    - Lokal: „6over4“ – IPv4 als „virtueller“ Link-Layer
    - Internet: „6to4“ – Verbindung von v6-Netzen über v4-Netze
    - Teredo: UDP-basierter Tunnel, funktioniert auch hinter NATs

Übergang IPv4 → IPv6

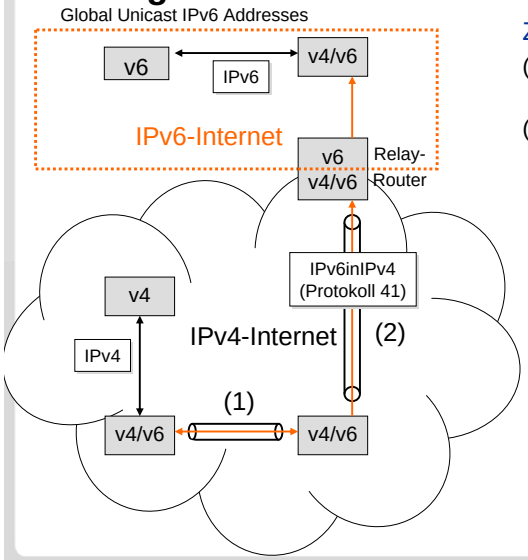
- Grundlegende Übergangsmechanismen [RFC4213]
  - Dual IP Layer (Dual-Stack): IPv6-fähige Knoten besitzen eine IPv4 und (mind.) eine IPv6-Adresse
    - Entscheidung über Verwendung von IPv4 oder IPv6 wird durch DNS bzw. Anwendung getroffen
    - Existiert eine IPv6-Adresse? AAAA Resource Record
  - Konfigurierte Tunnel (IPv6 in IPv4)
    - Router-to-Router
    - Host-to-Router
    - Host-to-Host
    - Router-to-Host
- Ergänzung durch NAT64/DNS64

Beispiel: DNS-Anfrage nach AAAA-Records

```
> dig www.tm.uka.de AAAA

; <<>> DiG 9.2.1 <<>> www.tm.uka.de AAAA
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9013
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 1

;; QUESTION SECTION:
;www.tm.uka.de.                 IN      AAAA

;; ANSWER SECTION:
tm.uka.de.              86400   IN      DNAME   tm.uni-karlsruhe.de.
www.tm.uka.de.          0       IN      CNAME   www.tm.uni-karlsruhe.de.
www.tm.uni-karlsruhe.de. 86400  IN      AAAA    2001:638:204::42

;; AUTHORITY SECTION:
tm.uni-karlsruhe.de.    86400   IN      NS      deneb.dfn.de.
tm.uni-karlsruhe.de.    86400   IN      NS      iraun1.ira.uni-karlsruhe.de.
tm.uni-karlsruhe.de.    86400   IN      NS      iraun2.ira.uni-karlsruhe.de.
tm.uni-karlsruhe.de.    86400   IN      NS      netserv.rz.uni-karlsruhe.de.

;; ADDITIONAL SECTION:
iraun1.ira.uni-karlsruhe.de. 82871 IN   A       141.3.10.90

;; Query time: 10 msec
;; SERVER: 2001:638:204::11#53(2001:638:204::11)
;; WHEN: Fri Apr 16 15:13:58 2004
;; MSG SIZE  rcvd: 249
```

DNS-Anfrage via IPv6!

Übergang IPv4 → IPv6 – Dual Stack Strategie

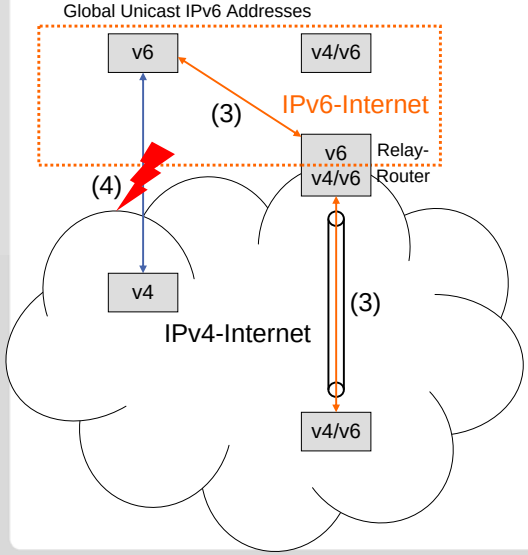


Zwischen v4/v6 Dual Stack Hosts

(1) Bidirektionaler konfigurierter IPv6-in-IPv4 Tunnel (Host-Host)

(2) Konfigurierte Default Route zu einem Relay Router (statischer Tunnel, Host-Router, Router-Host)

Übergang IPv4 → IPv6 – v6-only



v4/v6-Host mit v6-only Host

(3) Hinrichtung wie voriger Fall v4→GU, d.h. konfigurierte Default Route zu einem Relay Router (statischer Tunnel)

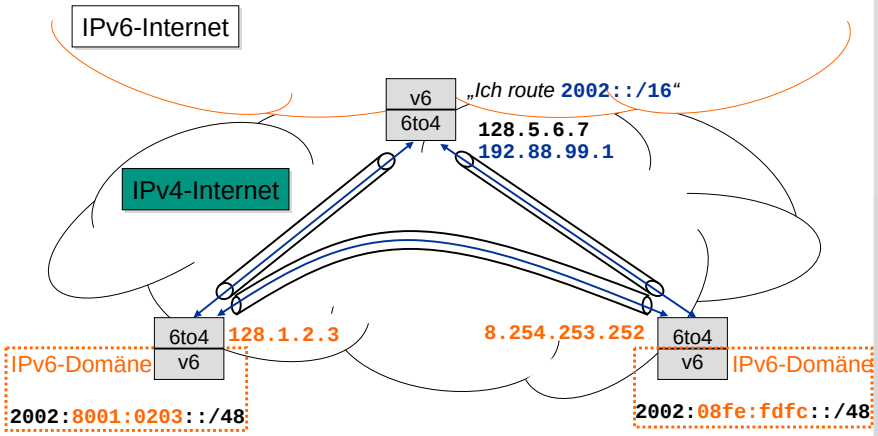
v4-only mit v6-only

(4) Keine direkte Kommunikation möglich! Nur Indirekt mittels IPv4/IPv6-Gateways → NAT-PT

Übergangsstrategien – 6to4

- „6to4“ – Verbinden von IPv6 Domänen über IPv4-Wolken [RFC3056]
  - IPv6-Domäne besitzt mind. eine weltweit eindeutige IPv4-Unicastadresse
    - 48-Bit Domänen-Präfix: 2002::/16 + IPv4-Adresse
    - Grenzrouter tunnelt sämtliche IPv6-Pakete mit Präfix 2002::/16 über das externe IPv4-Netz (Internet)
    - (mehrere) Relay Router zum nativen IPv6-Netz erreichbar über ihre jeweilige IP-Unicastadresse oder über Anycastadresse 192.88.99.1 bzw. 2002:c058:6301:: [RFC3068]
  - IPv4-Adresse des Relay-Routers wird im Netzwerkteil codiert
    - Relay-Router kann aus Netzwerkadresse IPv4-Unicast-Adresse des Zielrouters extrahieren → Ziel-Tunnelendpunkt
  - Zahlreiche Sicherheitsaspekte zu beachten! [RFC3964]

6to4 Beispiel



Änderungen bei Anwendungen

- Änderungen ziemlich gering: Anwendungen müssen längere Adressen verwenden → neue Adressfamilie AF_INET6 sowie neue Adressstrukturen notwendig (**`in6_addr`**, **`sockaddr_in6`**).
  - Zusätzliche Felder (Class Field, Flow Label) sowie Interfaces müssen gesetzt werden können (Erweiterung der Socket API)
  - Zur Auflösung des logischen Namens statt **`gethostbyname()`** nun **`getaddrinfo()`** (in anderer Richtung: **`getnameinfo()`**) verwenden → Funktionen unterstützen IPv6 und IPv4 [RFC3493]
  - Statt **`inet_ntoa()`** nun **`inet_ntop()`** zur textuellen Darstellung
  - HowTo [Castro09]
- Betriebssysteme: Linux, MacOS X >=10.2, Vista/Windows7
  - Win-XP kommt auch mit IPv6: zum Aktivieren „ipv6 install“ in Kommandozeile eingeben
- Viele wichtige Anwendungen funktionieren bereits mit IPv6

Änderungen im Routing

Die längeren IPv6-Adressen müssen auch durch die Routingprotokolle unterstützt werden

- Inter-Domain-Routing [RFC2858]
  - MBGP – Multiprotocol BGP kann u.a. IPv6-Adressen transportieren
- Intra-Domain-Routing [RFC2740]
  - OSPFv3 ist für IPv6 entwickelt worden
  - IS-IS kann ebenfalls mit kleiner Änderung IPv6 Routen tragen

Wieso bisher kaum IPv6?

- Relativ späte Unterstützung durch Router- und Betriebssystemhersteller
  - Größter Hersteller Cisco unterstützt IPv6 erst seit Mitte 2001!
  - Inzwischen viele Endsysteme IPv6-fähig
- Wenig neues
  - Viele Vorzüge von IPv6 mittlerweile auch für IPv4 implementiert
    - Autokonfiguration (DHCP, allerdings aufwändiger)
    - Authentifizierung und Verschlüsselung
    - Multicastadressen m. impliziter Reichweite 239.0.0.0/24
- Operationale Kosten steigen (Personal, Training, ...)
- Bisher kaum Nachfrage nach IPv6-Unterstützung
  - fehlende IPv6-fähige Software (IPv6-fähige Anwendungen fehlen teilweise noch)
  - einige Provider bereits IPv6-fähig, aber kaum Kundennachfrage

Weitere Probleme...

- IPv6 ohne Abwärtskompatibilität entwickelt
  - IPv4 kein Spezialfall von IPv6
- Keine sinnvollen Übergangsstrategien entwickelt
- Keine ausreichende Hardware-Unterstützung
  - Performance-Nachteile, u.a. bei ACLs
  - Kein so umfangreiches, ausgereiftes Test-Equipment
- Management-Software und Tools häufig nur für IPv4 verfügbar
- Autokonfiguration von Adressen in einigen Fällen unbrauchbar
  - Kontrolle der Adressen aus Sicherheitssicht (Logging usw.) wünschenswert → DHCP

IPv6Forum

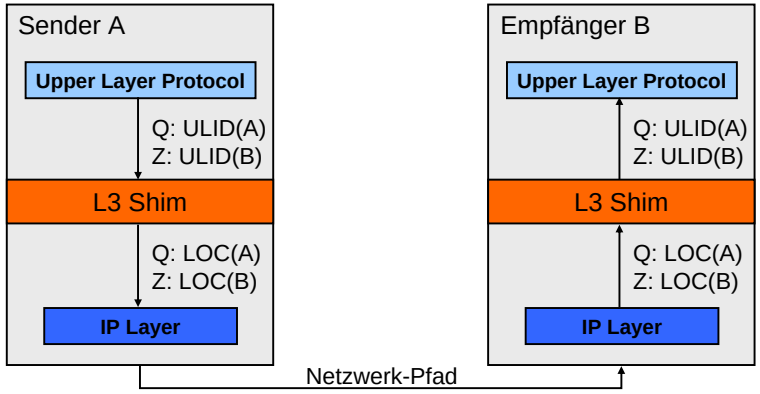
- IPv6Forum
  - Konsortium aus Hardware-Hersteller für das Internet sowie Forschungs- und Lehreinrichtungen
  - Ziel: Verbreitung der Bekanntheit und den praktischen Einsatz von IPv6 fördern
  - Zahlreiche Hinweise zu IPv6 unter http://www.ipv6forum.org/
  - Ähnliche Ziele national: Deutscher IPv6 Rat http://www.ipv6council.de/

3.5 Site Multi-Homing – Problematik

- Mehrere IPv6-Adressen kein Problem, aber
  - Anwendungen müssen Ausfall überleben
    - TCP-Verbindungen überleben keinen Adressenwechsel
    - Verschiebt Problematik in Anwendungen
  - Problem mit Ingress-Filtern (verwerfen topologisch falsch adressierte Pakete)
  - Erfordert richtige Auswahl der Adresse auf Senderseite (Source Address Selection)
  - Skalierbares Site Multi-Homing für IPv6? (Anforderungen in [RFC3582])
  - Identifier/Locator-Problematik spielt auch hier eine Rolle

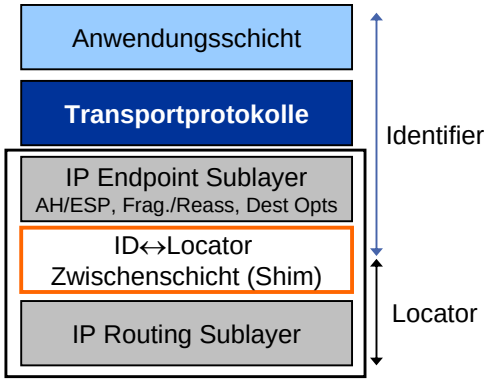
Site Multi-Homing: Grobkonzept

- Transportprotokolle und Anwendungen sehen nur den ULID (Upper Layer Identifier)
- Shim (Zwischenschicht) setzt ULID in Locator (LOC) aus Locator Set um
- Für das Transportprotokoll und die Anwendungen ändert sich nichts



Site Multi-Homing – Zwischenschicht „Shim“

- Platzierung: Oberhalb IP-Routing unterhalb IP-Endpunkt-Funktionalität (Reassemblierung, IPsec), gleich wie Extension Header



Shim6 Protokoll – Ablauf [RFC5533]

- Das Shim6-Protokoll verwendet IPv6-Erweiterungsköpfe
- Initialer Kontakt
  - Anwendung in System A möchte Paket an B senden
  - vorerst noch kein Shim6-Einsatz (ULID Paar wird bestimmt)
- 4-Wege-Handshake: Austausch Locator Set (Partner muss auch Shim6 können)
  - Kontext für Locator Set wird etabliert
  - Kontext wird über Garbage Collection deaktiviert
- Kommunikation wird normal weitergeführt, durch Einsatz von ULIDs als Locator Paar sogar ohne Zusatzaufwand
- Erreichbarkeitstests werden ständig durchgeführt
- Fehlerfall: neues funktionierendes Locator-Paar aushandeln und auf dieses umschalten
  - Nach Umschalten tragen Datenpakete Shim6 Extension Header (8-Byte) mit Context-Tag (47 Bits)

Shim6 – Diskussion

- Lösung im Einklang mit Ende-zu-Ende-Prinzip
  - keine Infrastrukturänderung erforderlich
- Lösung funktioniert nur, wenn beide Endsysteme Shim6-fähig sind
- Wenig Akzeptanz bei Providern
  - Endsysteme entscheiden über Verkehrsfluss bzw. –lasten → passt nicht zu Traffic Engineering
  - Keine Möglichkeit für Provider Netz als Ganzes zu steuern
- Neue Lösungen für das Routing-System gesucht, basierend auf ID-/Locator Split

3.6 Aufgaben

- 3.1 Nennen Sie die Unterschiede zwischen IPv4 und IPv6
- 3.2 Welche (Leistungs-)Vorteile bringt IPv6 für das Routing?
- 3.3 Ein Rechner mit Ethernet-Interface habe die MAC-Adresse 00:90:27:72:0B:48.
  - a) Wie lautet seine Link-Local-Adresse?
  - b) Auf welche anderen IPv6-Adressen antwortet er noch?
  - c) Welche weiteren MAC-Adressen sind hierzu nötig?
- 3.4 Weshalb macht NAT für IPv6 keinen Sinn?
- 3.5 Was sind/waren Hinderungsgründe für die Einführung von IPv6?

3.7 Literatur (1)

[Brad06] Scott Bradner, „A history of the IETF IPng effort“, http://www.sobco.com/ipng/

[Bush07] Randy Bush: “IPv6 Transition & Operational Reality”, NANOG-41 / Albuquerque, 2007.10.16, http://rip.psg.com/~randy/071016.v6-op-reality.pdf

[Castro09] Eva M. Castro: Porting applications to IPv6 HowTo, http://gsyc.escet.urjc.es/~eva/IPv6-web/ipv6.html, 2009

[draft-shim6] E. Nordmark und M. Bagnulo. Shim 6: Level 3 multihoming shim protocol, Februar 2009, http://tools.ietf.org/wg/shim6/draft-ietf-shim6-proto

[Dura06] Alain Durand: Managing 100+ Million IP Addresses, Presentation at NANOG 37, San Jose, Juni 2006, http://www.nanog.org/mtg-0606/durand.html

[Hain05] Tony Hain, „A Pragmatic Report on IPv4 Address Space Consumption “, Cisco IPJ, Volume 8, Nr. 3, September 2005 http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html

[Huit97] Christian Huitema: „IPv6 – The New Internet Protocol“, 2nd ed., Prentice Hall, 1997.

[Hust10] Geoff Huston, „IPv4 Address Report“, Mai 2010, http://ipv4.potaroo.net/

Literatur (2)

[Hust06] Geoff Huston, „IPv6 – Extinction, Evolution or Revolution? “, ISP Column, Januar 2006, http://www.potaroo.net/ispcol/2006-01/ipv6revolution.html

[Phif00] L. Phifer: „The Trouble with NAT“, The Internet Protocol Journal, Vol 3, No.4, Dec 2000, Cisco, kostenlos erhältlich unter http://www.cisco.com/ipj

[RFC 1631] K. Egevang und P. Francis. The IP Network Address Translator (NAT). RFC 1631 (Informational), Mai 1994. Obsoleted by RFC 3022. http://www.ietf.org/rfc/rfc1631.txt

[RFC 1918] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot und E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), Februar 1996. http://www.ietf.org/rfc/rfc1918.txt

[RFC 2365] D. Meyer. Administratively Scoped IP Multicast. RFC 2365 (Best Current Practice), Juli 1998. http://www.ietf.org/rfc/rfc2365.txt

[RFC 2460] S. Deering und R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), Dezember 1998. http://www.ietf.org/rfc/rfc2460.txt

[RFC 2464] M. Crawford. Transmission of IPv6 Packets over Ethernet Networks. RFC 2464 (Proposed Standard), Dezember 1998. http://www.ietf.org/rfc/rfc2464.txt

Literatur (3)

[RFC 2529] B. Carpenter und C. Jung. Transmission of IPv6 over IPv4 Domains without Explicit Tunnels. RFC 2529 (Proposed Standard), März 1999. http://www.ietf.org/rfc/rfc2529.txt

[RFC 2663] P. Srisuresh und M. Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663 (Informational), August 1999. http://www.ietf.org/rfc/rfc2663.txt

[RFC 2740] R. Coltun, D. Ferguson und J. Moy. OSPF for IPv6. RFC 2740 (Proposed Standard), Dezember 1999. http://www.ietf.org/rfc/rfc2740.txt

[RFC 2766] G. Tsirtsis und P. Srisuresh. Network Address Translation – Protocol Translation (NAT-PT). RFC 2766 (Proposed Standard), Februar 2000. Updated by RFC 3152. http://www.ietf.org/rfc/rfc2766.txt

[RFC 2858] T. Bates, Y. Rekhter, R. Chandra und D. Katz. Multiprotocol Extensions for BGP-4. RFC 2858 (Proposed Standard), Juni 2000. http://www.ietf.org/rfc/rfc2858.txt

[RFC 3022] P. Srisuresh und K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022 (Informational), Januar 2001. http://www.ietf.org/rfc/rfc3022.txt

[RFC 3027] M. Holdrege und P. Srisuresh. Protocol Complications with the IP Network Address Translator. RFC 3027 (Informational), Januar 2001.

Literatur (4)

[RFC 3056] B. Carpenter und K. Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056 (Proposed Standard), Februar 2001. http://www.ietf.org/rfc/rfc3056.txt

[RFC 3068] C. Huitema. An Anycast Prefix for 6to4 Relay Routers. RFC 3068 (Proposed Standard), Juni 2001. http://www.ietf.org/rfc/rfc3068.txt

[RFC 3175] F. Baker, C. Iturralde, F. Le Faucheur und B. Davie. Aggregation of RSVP for IPv4 and IPv6 Reservations. RFC 3175 (Proposed Standard), September 2001. http://www.ietf.org/rfc/rfc3175.txt

[RFC 3177] IAB und IESG. IAB/IESG Recommendations on IPv6 Address Allocations to Sites. RFC 3177 (Informational), September 2001. http://www.ietf.org/rfc/rfc3177.txt

[RFC 3194] A. Durand und C. Huitema. The H-Density Ratio for Address Assignment Efficiency An Update on the H ratio. RFC 3194 (Informational), November 2001. http://www.ietf.org/rfc/rfc3194.txt

[RFC 3306] B. Haberman und D. Thaler. Unicast-Prefix-based IPv6 Multicast Addresses. RFC 3306 (Proposed Standard), August 2002. Updated by RFCs 3956, 4489. http://www.ietf.org/rfc/rfc3306.txt

Literatur (5)

[RFC 3484] R. Draves. Default Address Selection for Internet Protocol version 6 (IPv6). RFC 3484 (Proposed Standard), Februar 2003. http://www.ietf.org/rfc/rfc3484.txt

[RFC 3493] R. Gilligan, S. Thomson, J. Bound, J. McCann undW. Stevens. Basic Socket Interface Extensions for IPv6. RFC 3493 (Informational), Februar 2003. http://www.ietf.org/rfc/rfc3493.txt

[RFC 3582] J. Abley, B. Black und V. Gill. Goals for IPv6 Site-Multihoming Architectures. RFC 3582 (Informational), August 2003. http://www.ietf.org/rfc/rfc3582.txt

[RFC 3587] R. Hinden, S. Deering und E. Nordmark. IPv6 Global Unicast Address Format. RFC 3587 (Informational), August 2003. http://www.ietf.org/rfc/rfc3587.txt

[RFC 3697] J. Rajahalme, A. Conta, B. Carpenter und S. Deering. IPv6 Flow Label Specification. RFC 3697 (Proposed Standard), März 2004. http://www.ietf.org/rfc/rfc3697.txt

Literatur (6)

[RFC 3701] R. Fink und R. Hinden. 6bone (IPv6 Testing Address Allocation) Phaseout. RFC 3701 (Informational), März 2004. http://www.ietf.org/rfc/rfc3701.txt

[RFC 3715] B. Aboba und W. Dixon. IPsec-Network Address Translation (NAT) Compatibility Requirements. RFC 3715 (Informational), März 2004. http://www.ietf.org/rfc/rfc3715.txt

[RFC 3736] R. Droms. Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6. RFC 3736 (Proposed Standard), April 2004. http://www.ietf.org/rfc/rfc3736.txt

[RFC 3879] C. Huitema und B. Carpenter. Deprecating Site Local Addresses. RFC 3879 (Proposed Standard), September 2004. http://www.ietf.org/rfc/rfc3879.txt

[RFC 3956] P. Savola und B. Haberman. Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address. RFC 3956 (Proposed Standard), November 2004. http://www.ietf.org/rfc/rfc3956.txt

[RFC 3964] P. Savola und C. Patel. Security Considerations for 6to4. RFC 3964 (Informational), Dezember 2004. http://www.ietf.org/rfc/rfc3964.txt

[RFC 3971] J. Arkko, J. Kempf, B. Zill und P. Nikander. SEcure Neighbor Discovery (SEND). RFC 3971 (Proposed Standard), März 2005. http://www.ietf.org/rfc/rfc3971.txt

Literatur (7)

[RFC 3972] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard), März 2005. http://www.ietf.org/rfc/rfc3972.txt

[RFC 3986] T. Berners-Lee, R. Fielding und L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (Standard), Januar 2005. http://www.ietf.org/rfc/rfc3986.txt

[RFC 4007] S. Deering, B. Haberman, T. Jinmei, E. Nordmark und B. Zill. IPv6 Scoped Address Architecture. RFC 4007 (Proposed Standard), März 2005. http://www.ietf.org/rfc/rfc4007.txt

[RFC 4193] R. Hinden und B. Haberman. Unique Local IPv6 Unicast Addresses. RFC 4193 (Proposed Standard), Oktober 2005. http://www.ietf.org/rfc/rfc4193.txt

[RFC 4213] E. Nordmark und R. Gilligan. Basic Transition Mechanisms for IPv6 Hosts and Routers. RFC 4213 (Proposed Standard), Oktober 2005. http://www.ietf.org/rfc/rfc4213.txt

[RFC 4291] R. Hinden und S. Deering. IP Version 6 Addressing Architecture. RFC 4291 (Draft Standard), Februar 2006. http://www.ietf.org/rfc/rfc4291.txt

[RFC 4339] J. Jeong. IPv6 Host Configuration of DNS Server Information Approaches. RFC 4339 (Informational), Februar 2006. http://www.ietf.org/rfc/rfc4339.txt

Literatur (8)

[RFC 4692] G. Huston. Considerations on the IPv6 Host Density Metric. RFC 4692 (Informational), Oktober 2006. http://www.ietf.org/rfc/rfc4692.txt

[RFC 4787] F. Audet und C. Jennings. Network Address Translation (NAT) Behavioral Requirements for Unicast UDP. RFC 4787 (Best Current Practice), Januar 2007.http://www.ietf.org/rfc/rfc4787.txt.

[RFC 4861] T. Narten, E. Nordmark, W. Simpson und H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard), September 2007. URL: http://www.ietf.org/rfc/rfc4861.txt

[RFC 4862] S. Thomson, T. Narten und T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862 (Draft Standard), September 2007. URL: http://www.ietf.org/rfc/rfc4862.txt

[RFC 4941] T. Narten, R. Draves, S. Krishnan. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. September 2007. http://www.ietf.org/rfc/rfc4941.txt

[RFC 5006] J. Jeong, S. Park, L. Beloeil und S. Madanapalli. IPv6 Router Advertisement Option for DNS Configuration. RFC 5006 (Experimental), September 2007. http://www.ietf.org/rfc/rfc5006.txt

Literatur (9)

[RFC 5389] J. Rosenberg, R. Mahy, P. Matthews, und D. Wing. Session Traversal Utilities for NAT (STUN). RFC 5389 (Proposed Standard), October 2008. http://www.ietf.org/rfc/rfc3489.txt

[RFC 5533] E. Nordmark und M. Bagnulo. Shim6: Level 3 Multihoming Shim Protocol for IPv6. RFC 5533 (Proposed Standard), Juni 2009. http://www.ietf.org/rfc/rfc5533.txt

[RFC 5766] R. Mahy, P. Matthews und J. Rosenberg. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). RFC 5766 (Proposed Standard), April 2010. http://www.ietf.org/rfc/rfc5766.txt

[Sami06] Rahmat M. Samik-Ibrahim, „the LONG and windy ROAD”, http://rms46.vlsm.org/1/42.html