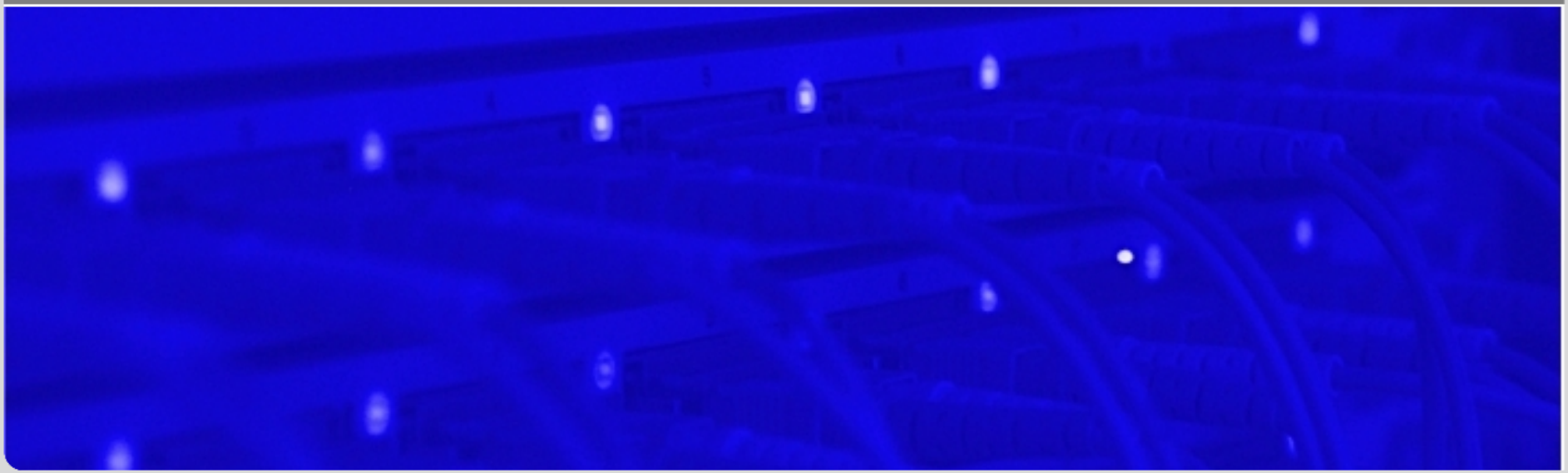
Karlsruher Institut für Technologie

TeleMatics

Next Generation Internet

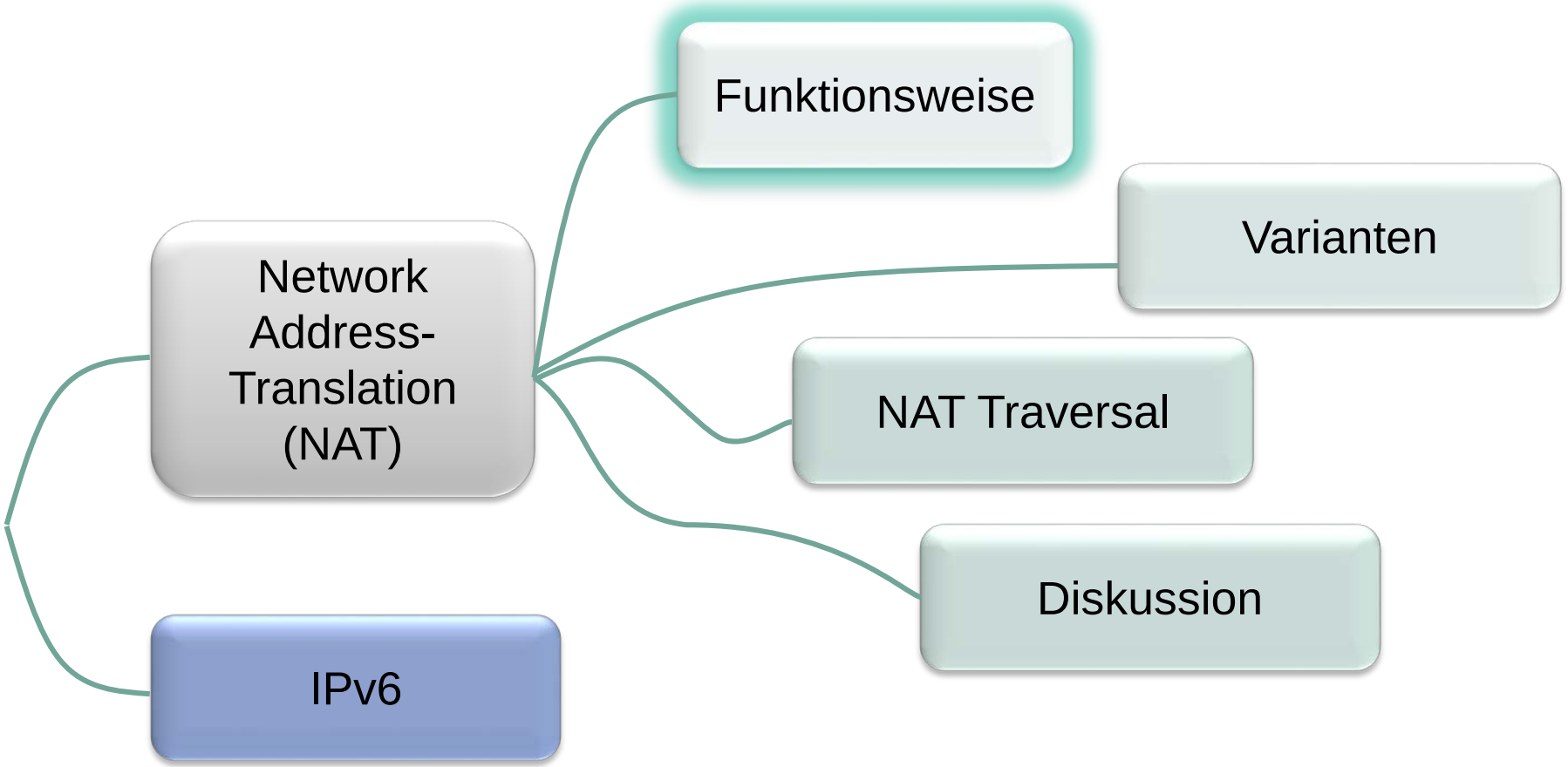
3. NAT & IPv6

INSTITUT FÜR TELEMATIK



Kapitelübersicht

Überblick


Funktionsweise
Varianten
Network Address-Translation (NAT)
NAT Traversal
Diskussion
IPv6

Network Address Translation

- NAT ist
  - keine Next Generation Technologie
  - aber als ein nahezu omnipräsentes Übel im heutigen Internet…
  - Anwendungen müssen teilweise darauf abgestimmt sein!
  - Carrier Grade NAT ist eingeführt!
- Teilkapitel gibt praktisches Beispiel
  - Verletzung der Architekturprinzipien aus Kapitel 2
    - NAT Geräte sind Middleboxes
  - Weitreichende Auswirkungen auf Protokolldesigns, komplexe Zusatzlösungen

Motivation und Funktionsweise

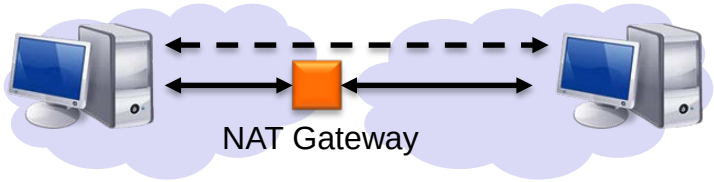
- Motivation
  - Bildung privater Netze („Intranets“) mit IPv4-Adressen aus einem der „privaten“ Adressräume: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16  [RFC1918]
  - Adressenknappheit: Zu wenig öffentliche IPv4-Adressen
  - Shared Address Space 100.64.0.0/10 für Carrier Grade NAT  [RFC6598]
- Funktionsweise
  - Umsetzen der IP-Adressen zwischen Adressräumen anhand bijektiver Abbildung zwischen Adressen
  - Umsetzung in Paketkopf und Nutzlast(!)
  - Versuch der transparenten Funktionsweise

Funktionsweise

- Binding: Bijektive Abbildung („Binding“) zwischen IP-Adressen (öffentlich ↔ privat)

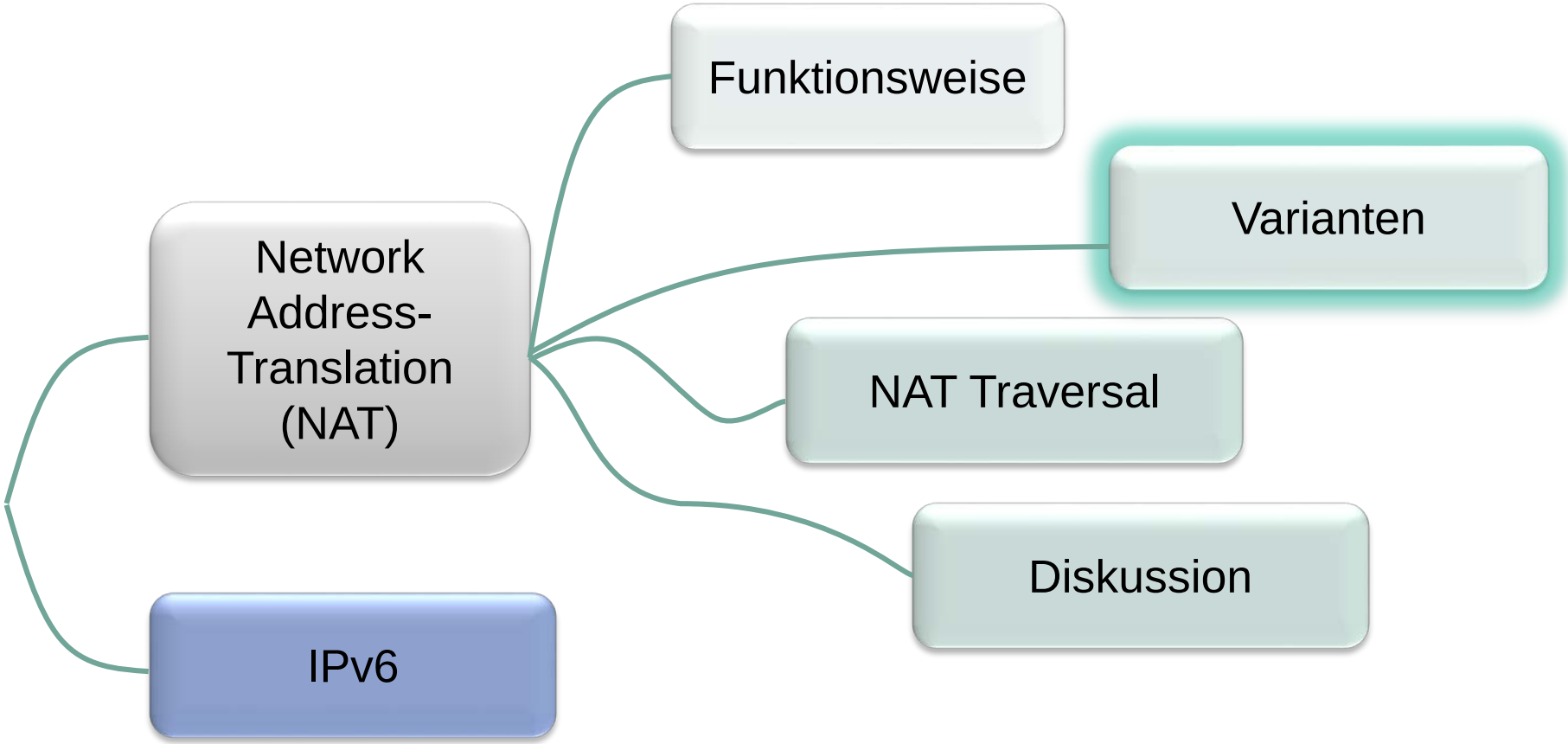
[RFC3022], ursprünglich [RFC1631] (1994)

- Umschreiben der Adressen im IP-Paket (u. ICMP-Paket) erfolgt „transparent“ durch NAT-Gateway



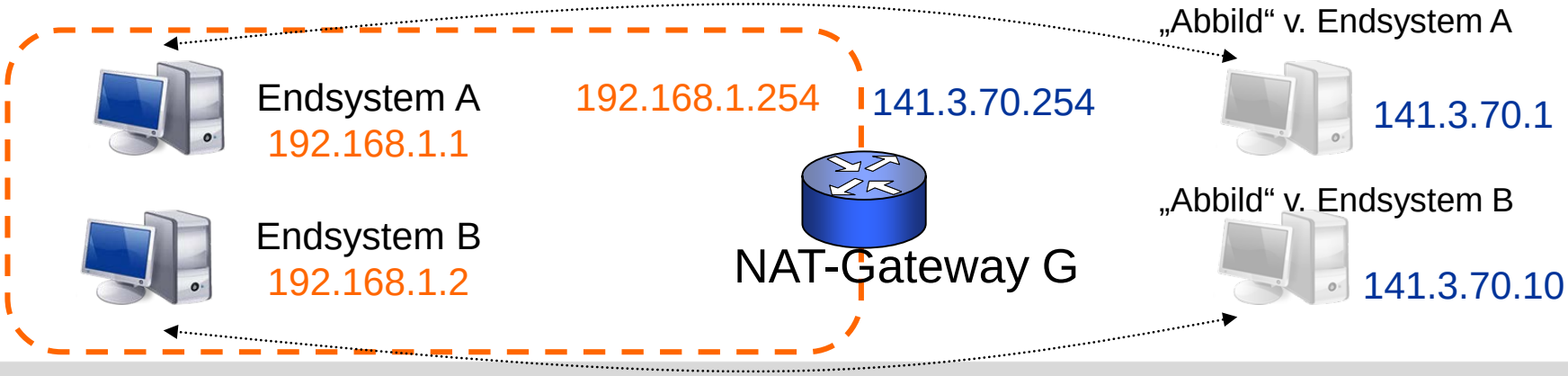
  - Erfordert Pool an öffentlichen Adressen
  - Anpassen der Prüfsummen (ggf. auch TCP/UDP) notwendig
  - Anwendungsabhängige Anpassung notwendig: wo stehen IP-Adressen in Nutzdaten?
    → Application Level Gateway (ALG)

Überblick


Funktionsweise
Varianten
Network Address-Translation (NAT)
NAT Traversal
Diskussion
IPv6

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

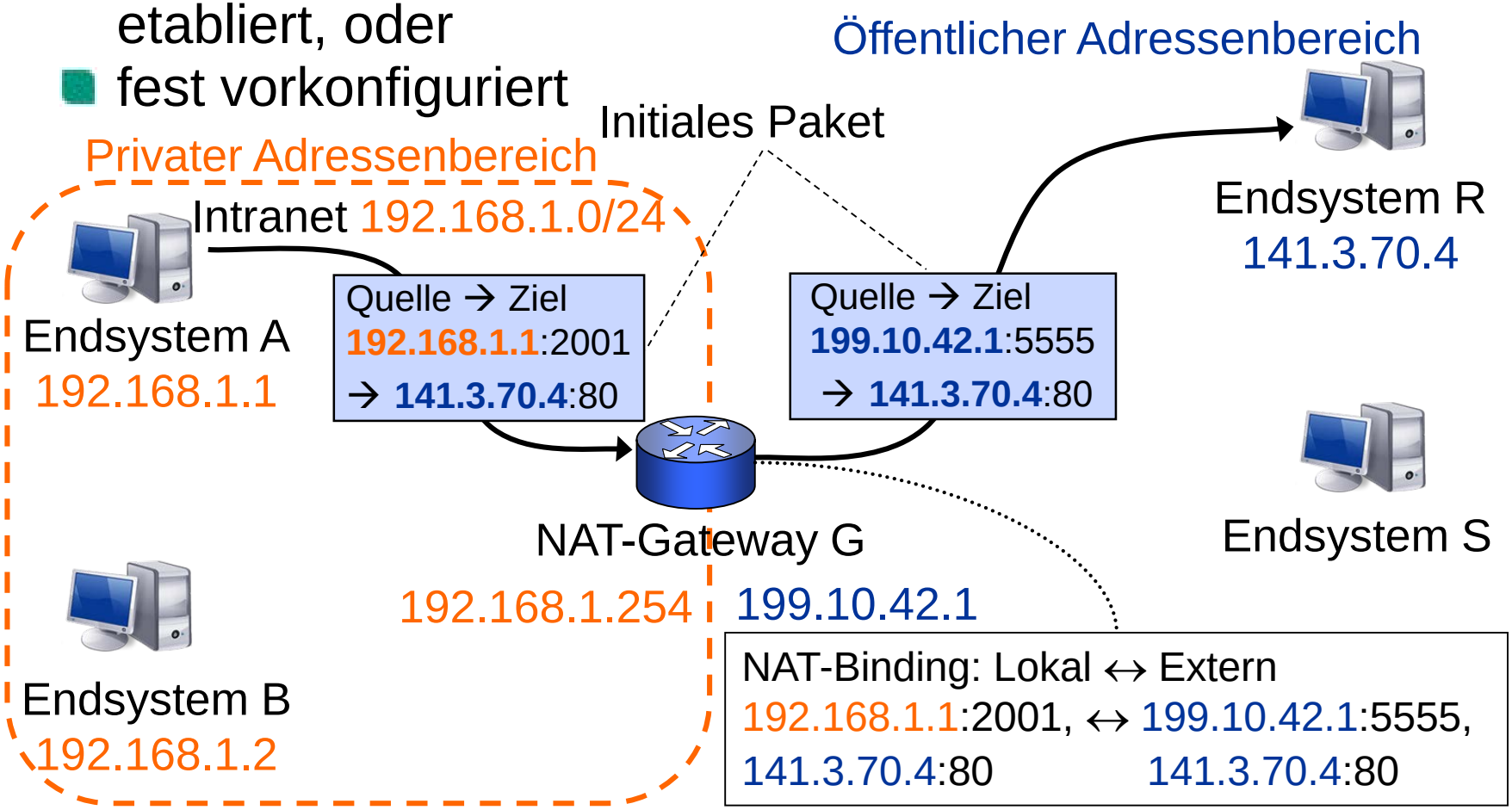


Network Address Port Translation

- 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
    - Nur bedingt möglich

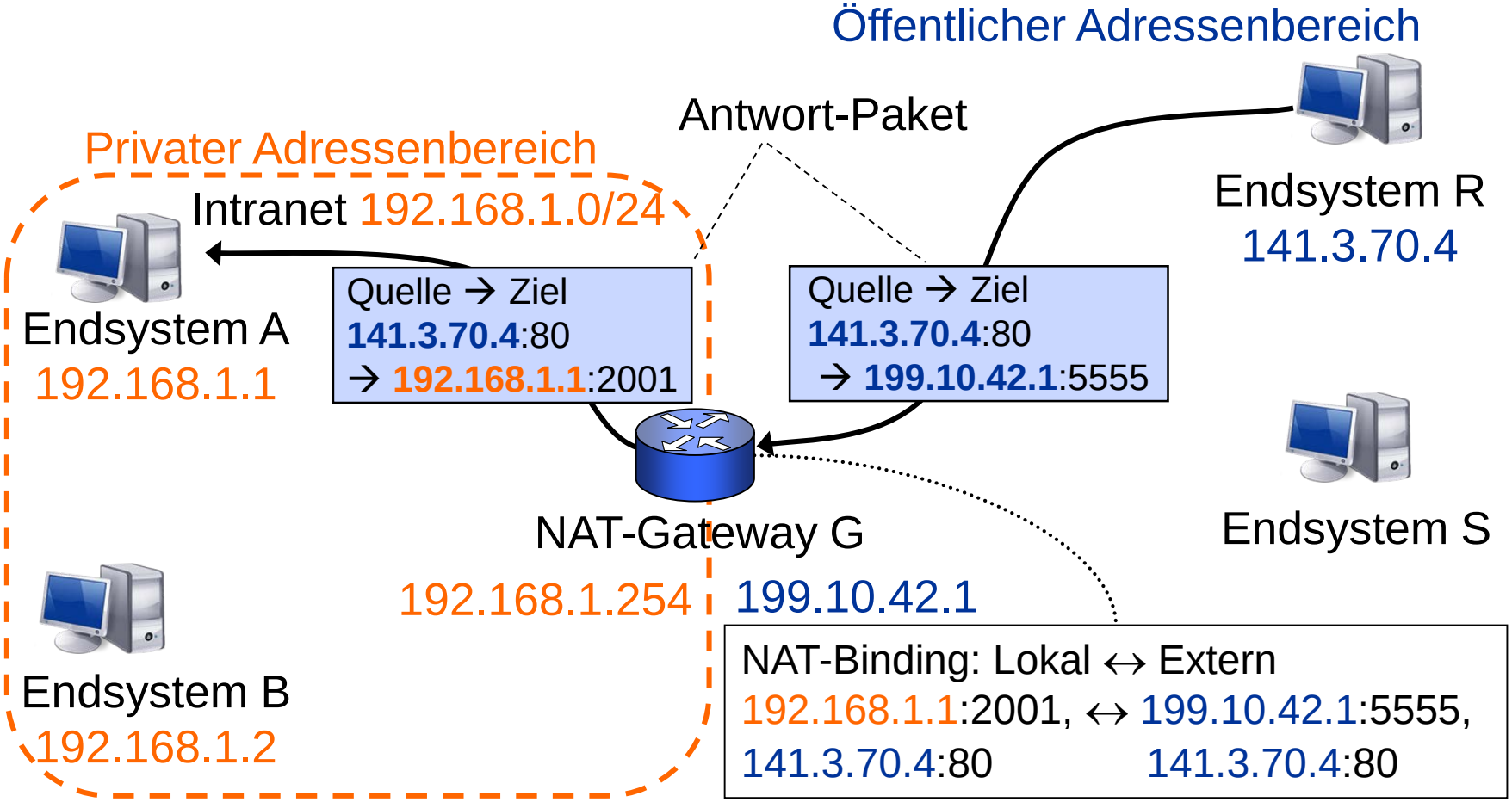
NAPT Beispiel (1)

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



NAPT Beispiel (2)

- Für Paket von extern muss ein Binding existieren!



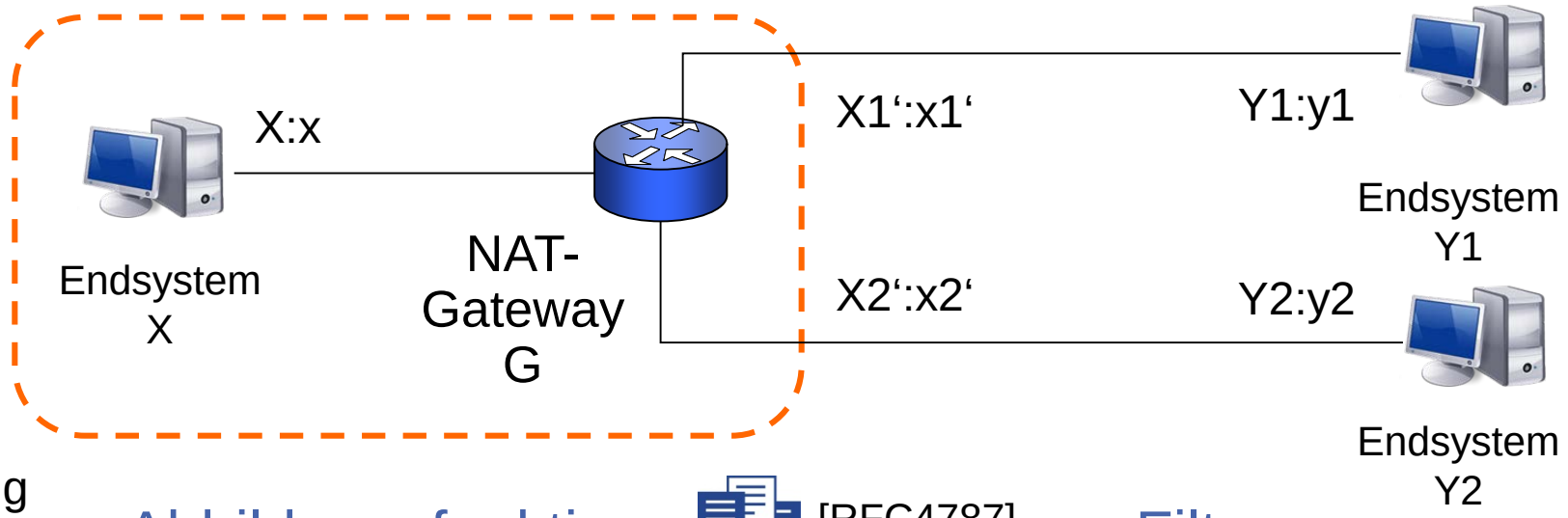
Probleme durch NAT Binding

- 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
    → erhöht Energiebedarf für mobile Geräte!
- 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 problematisch für zwei Endsysteme hinter verschiedenen NATs (z.B. Peer-to-Peer-Netze)  [RFC2663]
  - benötigte Hilfe durch Rendezvous-Knoten im öffentlichen Netz

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]

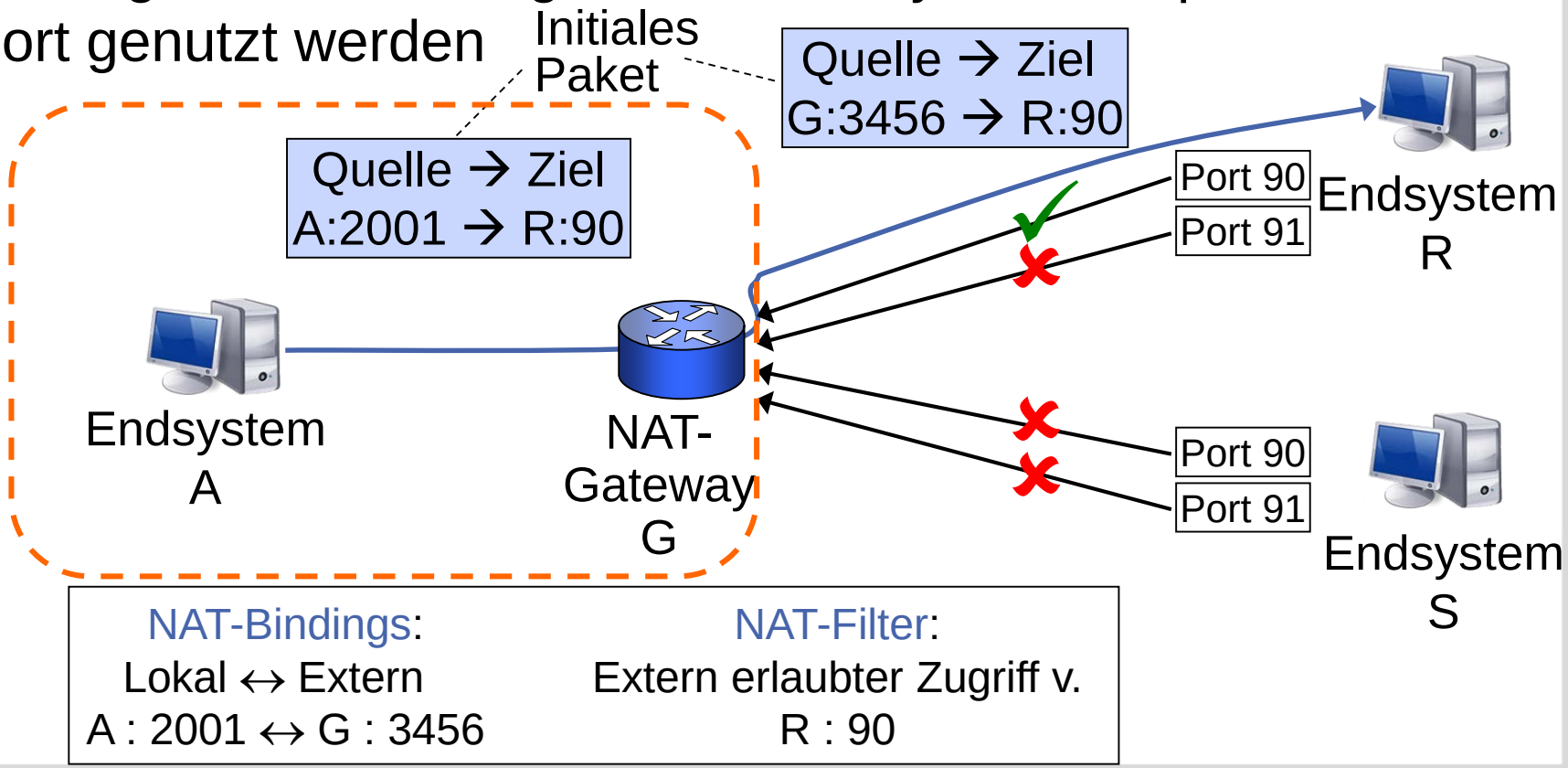
- Endsystem-unabhängig
- Adressabhängig
- Adress- und Port-abhängig

Filter

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

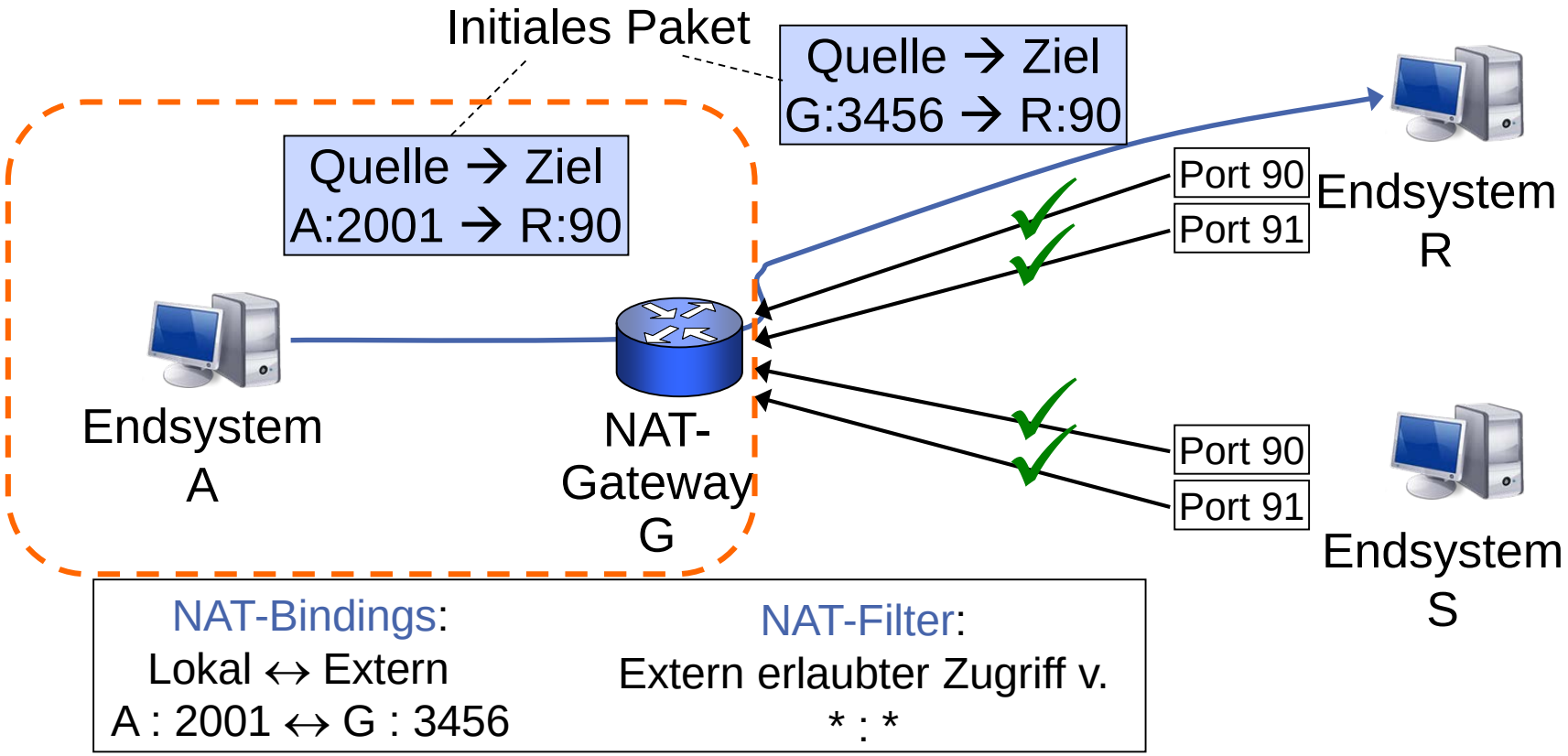
NAT-Varianten: Beispiel Symmetric NAT

- 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



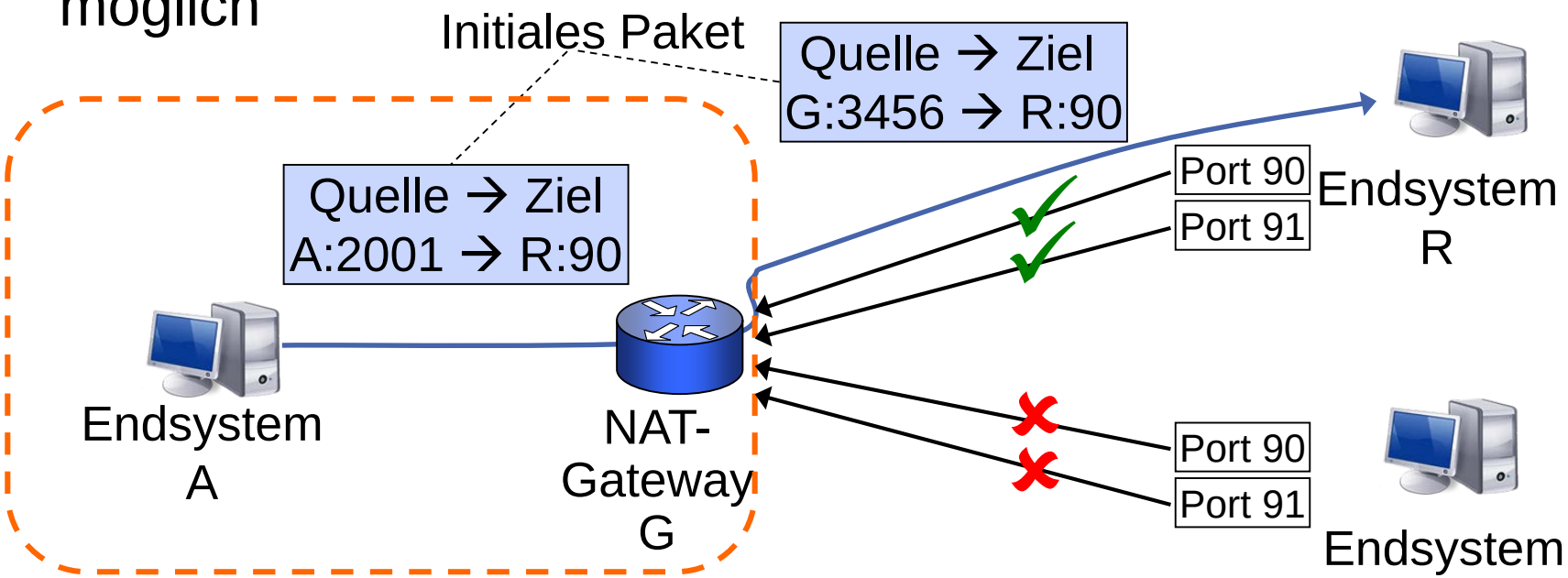
NAT-Variante: Full Cone

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



NAT-Variante: Restricted Cone

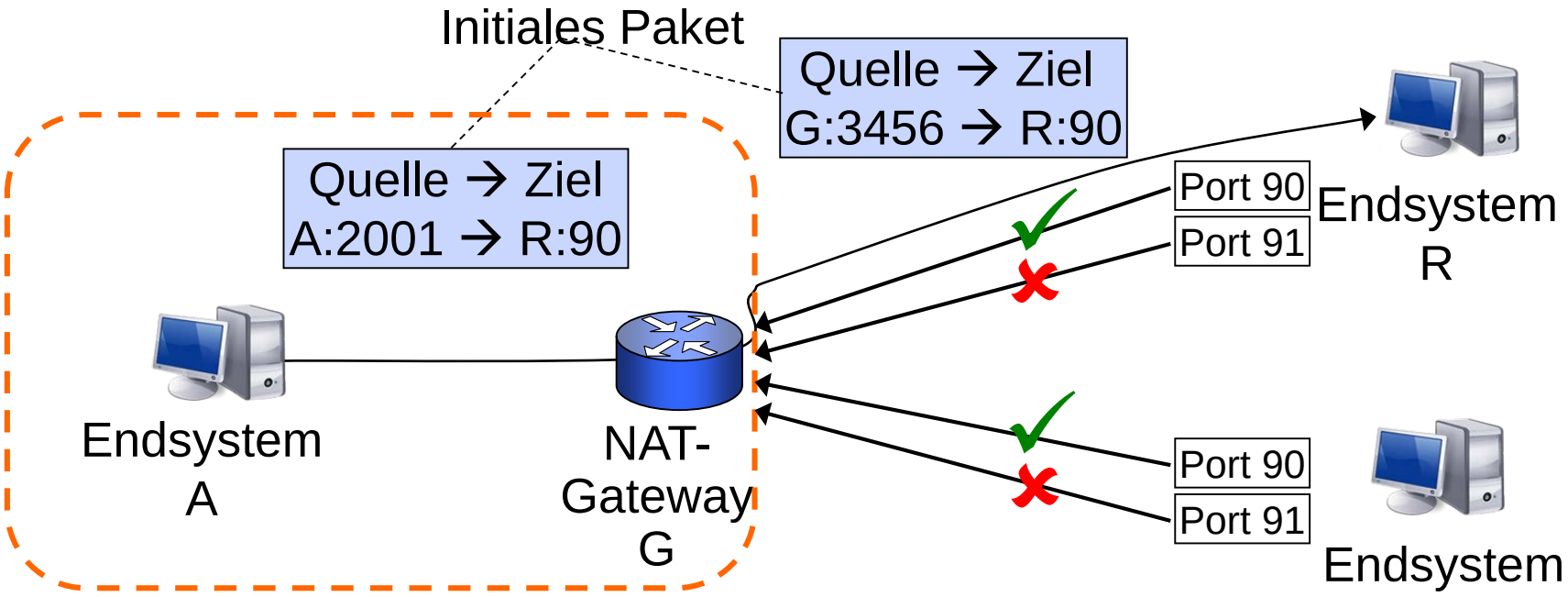
- Binding in Abhängigkeit von Quelladresse/-Port, eingehende Pakete eingeschränkt auf das gleiche Endsystem (Ziel des initialen Pakets), aber andere Ports möglich



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

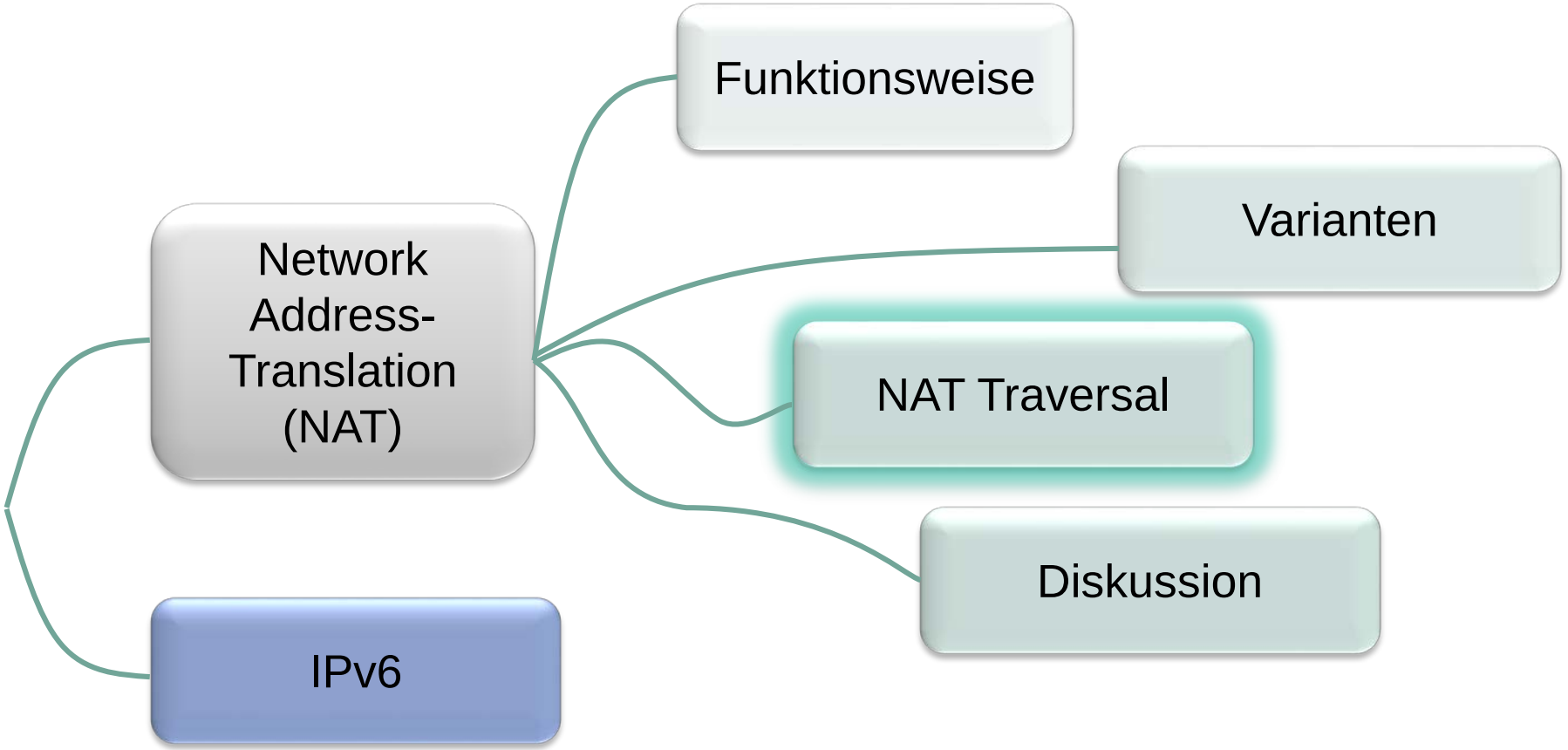
NAT-Variante: Port-Restricted Cone

- 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

Überblick


Funktionsweise
Varianten
Network Address-Translation (NAT)
NAT Traversal
Diskussion
IPv6

NAT Traversal

- „Transparentes“ Umsetzen durch NAT GW funktioniert nicht zuverlässig
  - Bei verschlüsselten Nutzdaten
  - Wenn das Protokoll weiterentwickelt wird, das ALG aber nicht
- Idee
  - Endsysteme bzw. Anwendungen übernehmen Umsetzung in Payload
  - Müssen dafür aber die Abbildung kennen
  - Erkennung von NAT Gateways und Lernen der Abbildung → NAT Discovery

NAT-Discovery: STUN

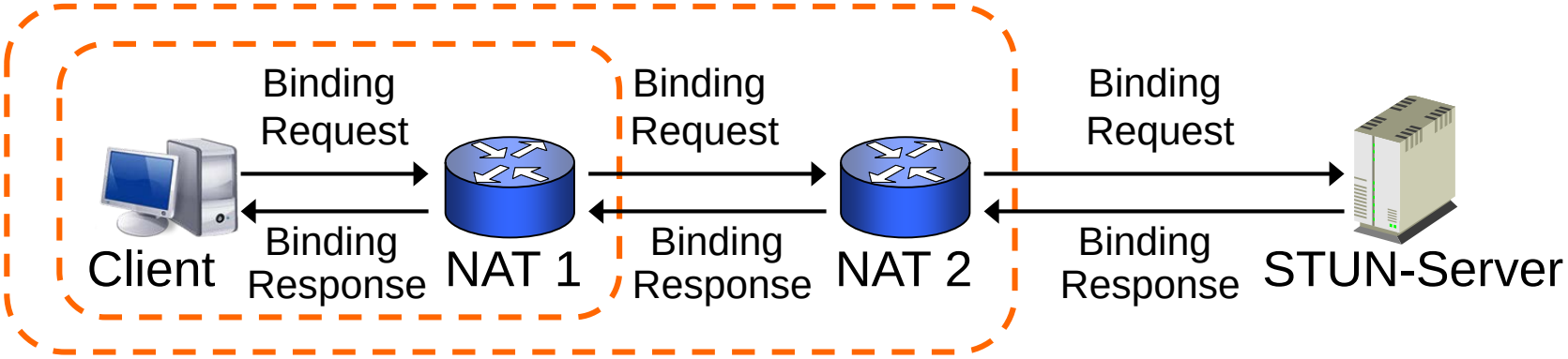
- STUN – Session Traversal Utilities for NAT
  - Client/Server-Protokoll


[RFC5389]

    - 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

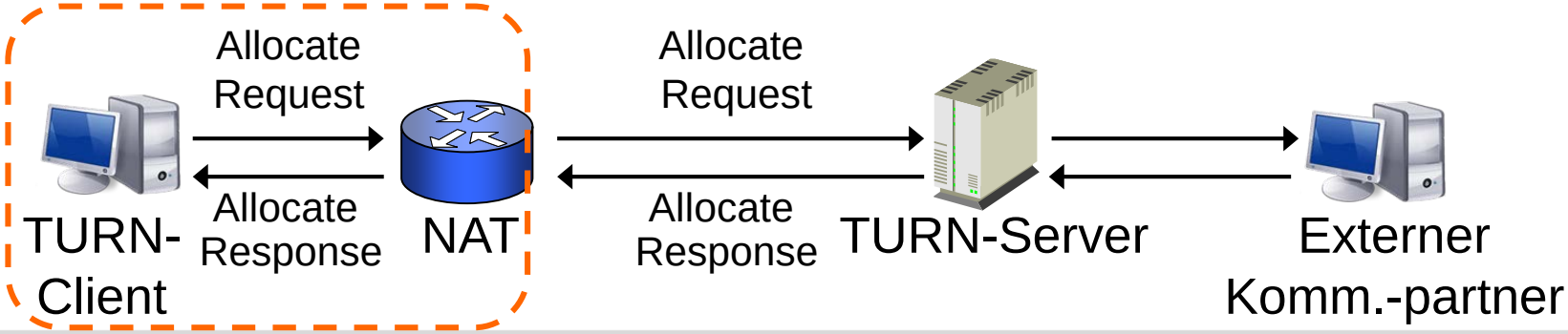


Relay NAT

- Problem: STUN hilft nicht bei symmetrischen NATs
  - STUN-Erweiterung (früher eigenständiges Protokoll TURN – Traversal Using Relays around NAT)
    - gleiche Syntax wie STUN, definiert 10 neue Nachrichten

[RFC5766]

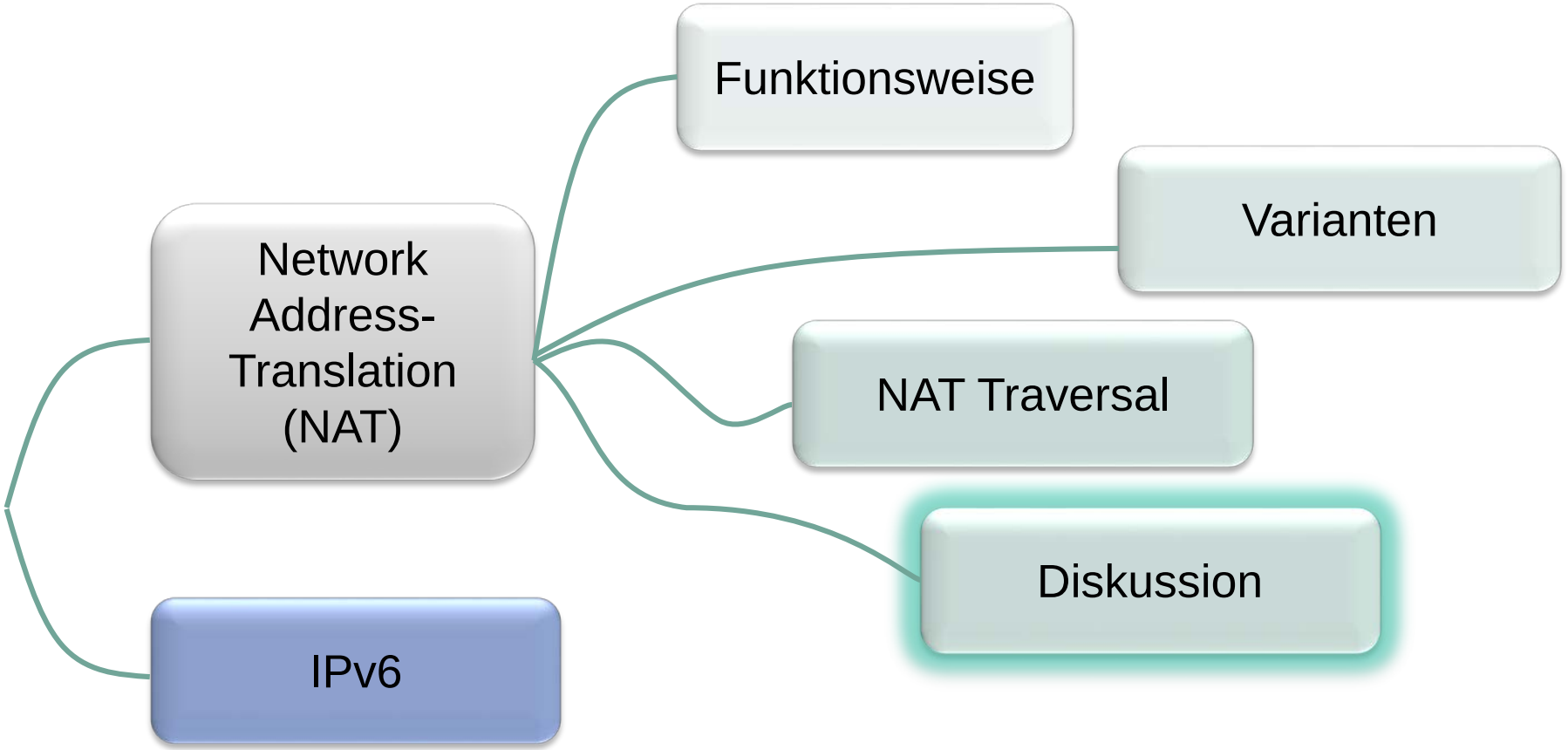
  - nutzt Server mit öffentlichen Adressen im Netz, der als Relay für Medienströme dient (UDP und TCP)
    - Initiale Kapselung von Daten notwendig (Send Indication)
      - Overhead von 44 Bytes
      - Umschalten auf Betrieb ohne Kapselung erfordert Magic Cookie in UDP-Paketen zur Erkennung der Kontrollpakete
    - Erlaubt Client nicht, Server hinter NAT zu betreiben



PCP – Port Control Protocol

- Protokoll, um explizit Mappings zu etablieren
- PCP Client in Anwendung/Endsystem, PCP-Server in NAT-Gateway/Firewall  [RFC6887]
- Für verschiedenste Middleboxes geeignet, u.a.
  - NA(P)T-Gateways
  - Carrier Grade NATs
  - Firewalls
- Koordination insbesondere notwendig für Shared IP Addresses

Überblick


Funktionsweise
Varianten
Network Address-Translation (NAT)
NAT Traversal
Diskussion
IPv6

NAT – Vorteile/Nachteile

- Vorteile
  - schafft Abhilfe bei Adressraumknappheit
  - vermeidet Umnummerieren des Netzwerks bei ISP Wechsel
- Nachteile
  - ist aber kein Sicherheitsmechanismus!
  - Verringerte Leistung, erhöhter Energiebedarf
  - 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

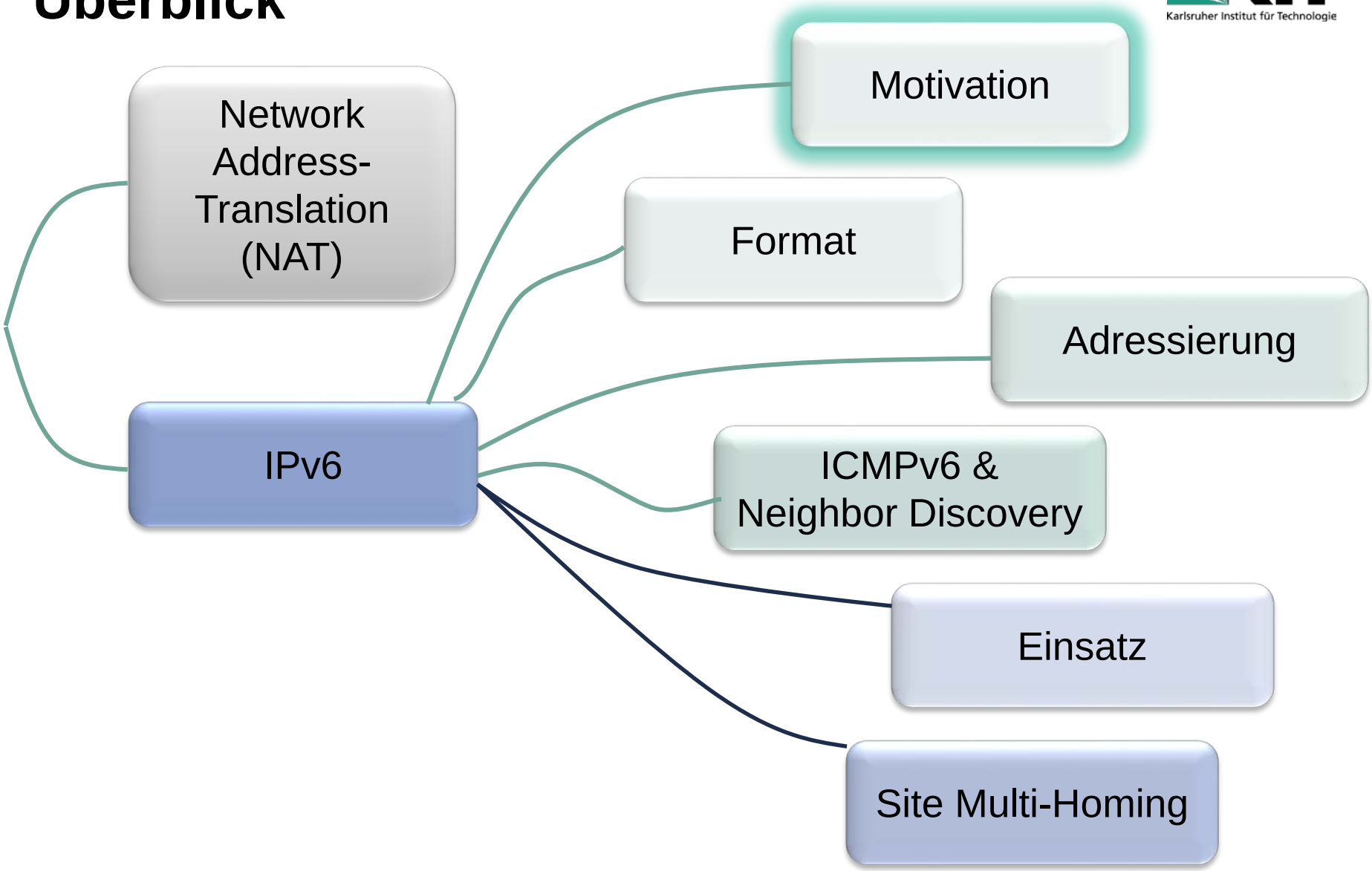
NAT – Weitere praktische Probleme

- 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! → korrupte Daten möglich
  - 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 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
- Auch NATs brauchen öffentliche Adressen!

Überblick


Network Address-Translation (NAT)
IPv6
Motivation
Format
Adressierung
ICMPv6 & Neighbor Discovery
Einsatz
Site Multi-Homing

Motivation für IPv6

- **Anwachsen des Internets**
  - IP-Adressraum erschöpft
  - mehr Internet-fähige Geräte, Kleinstgeräte, Sensoren „Internet der Dinge“ → Adressenbedarf 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

Das IPv4-Internet ist…

immer noch ein “Experiment”:  [LiCe11]

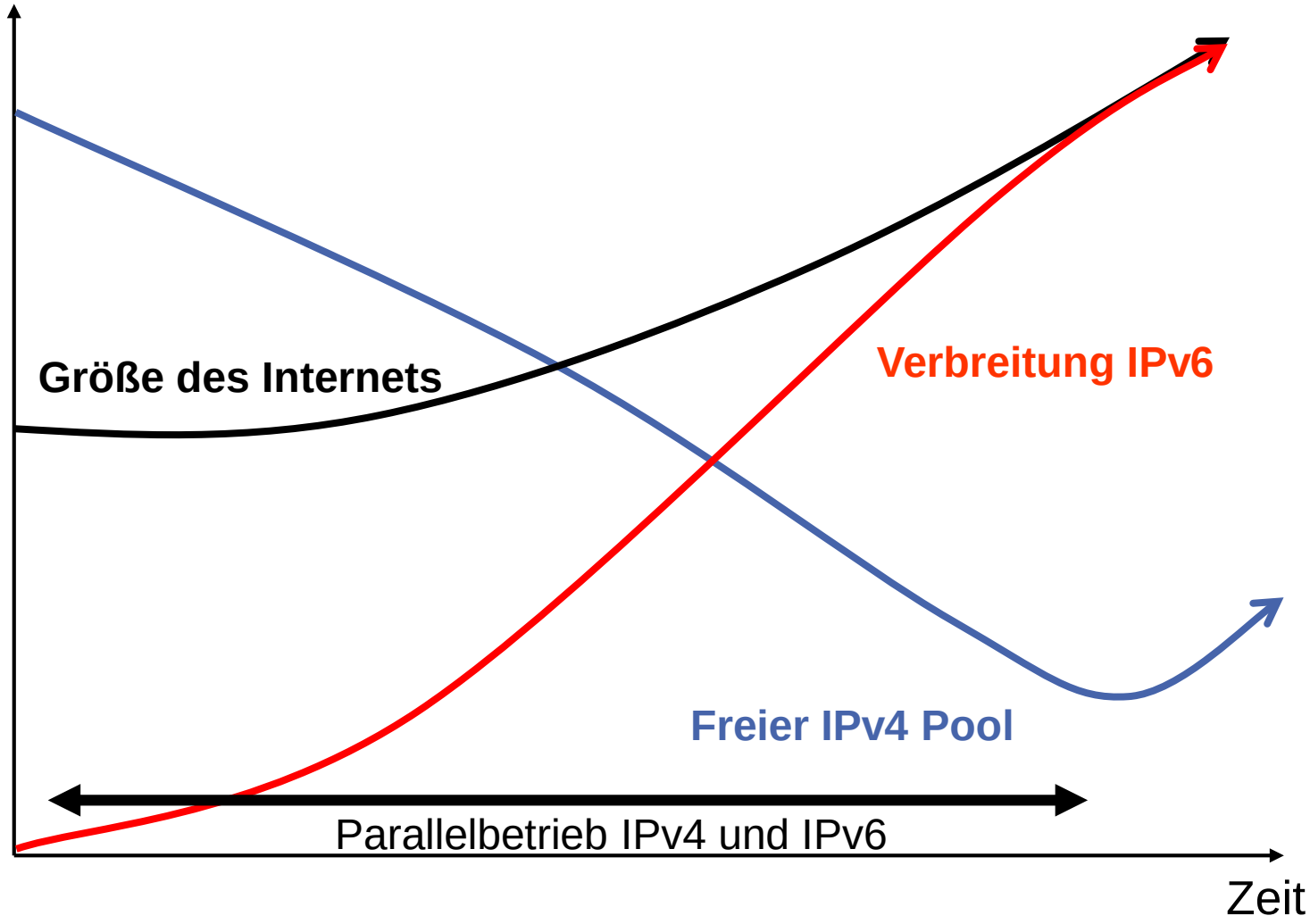
Vint Cerf: “A debate among the engineers and scientists working on the Internet ran for nearly a year without a firm conclusion. Some suggested 32-bit addresses (8 bits of network, 24 bits of host), some said 128 bits, and others wanted variable-length addresses. The last choice was rejected by programmers who didn't want to fiddle around finding the fields of an Internet packet. The 128-bit choice seemed excessive for an experiment that involved only a few networks to begin with. By this time, the research effort had reached its fourth iteration (the IP layer protocol was called IPv4), and as program manager, I felt a need to get on with live testing and final design of TCP and IP. In lieu of consensus, I chose 32 bits of address. I thought 4.3 billion potential addresses would be adequate for conducting the experiments to prove the technology. If it worked, then we could go back and design the production version. Of course, it is now 2011, and the experiment never exactly ended.”

→ IPv6 ist die Produktivvariante des Internet Protokolls…

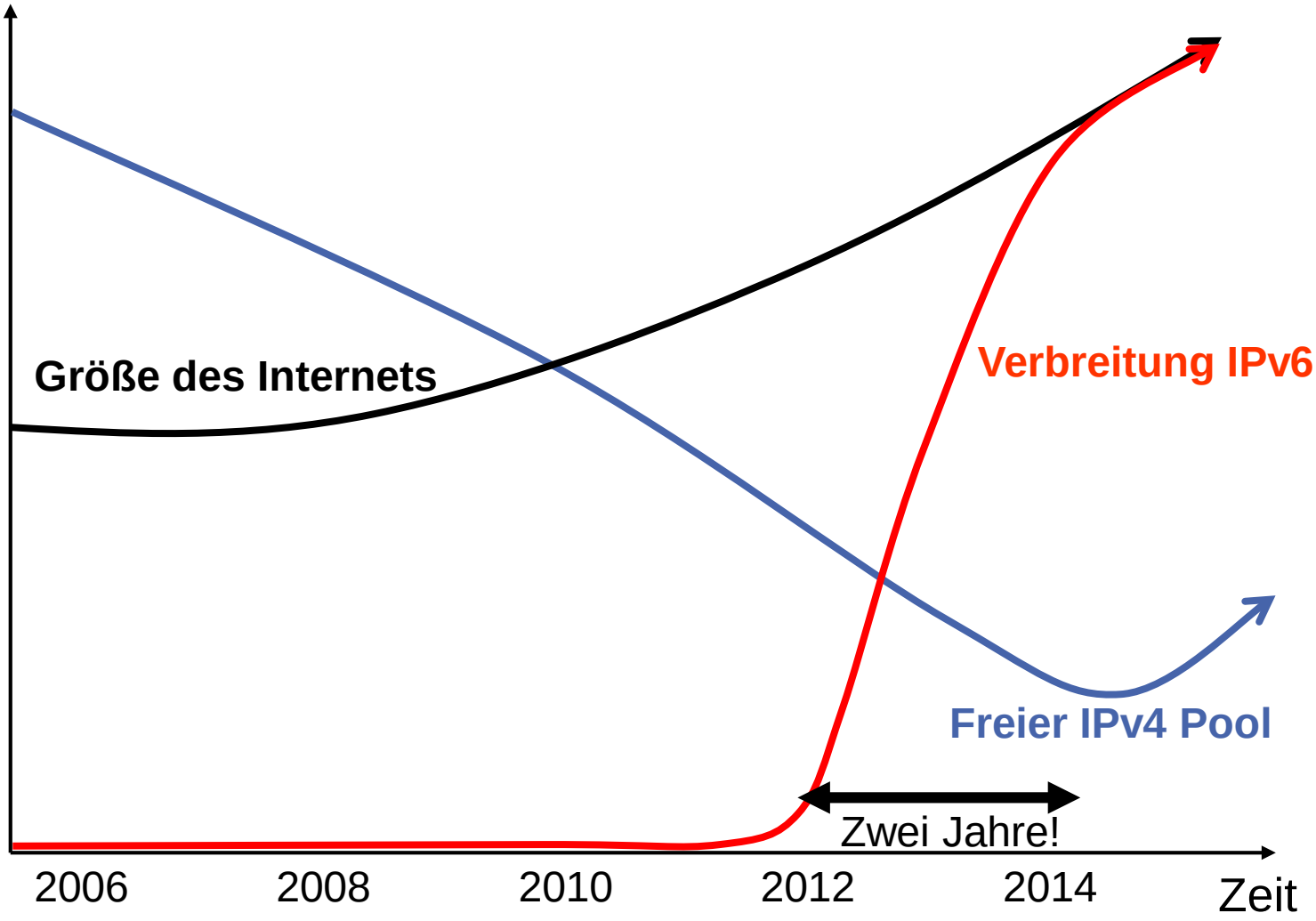
Adressenknappheit?

- Adressen werden von der IANA an Regional Internet Registries (RIRs) in Form von /8-Blöcken vergeben
  → Dieser Pool ist am 01.02.2011 erschöpft gewesen!
  - Prognose IETF 1994: gehen zwischen 2005–2011 aus
  - Führte zur Entwicklung von IPv6
- Wie lange sind denn noch IPv4-Adressen verfügbar?
  - APNIC sind April 2011 bereits die IPv4-Adressen ausgegangen, RIPE im September 2012!
  - Es war unklar wie lange die Pools der anderen RIRs noch reichen
    → September 2015 waren fast alle Adressen vergeben
    (bis auf AFRINIC)

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 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

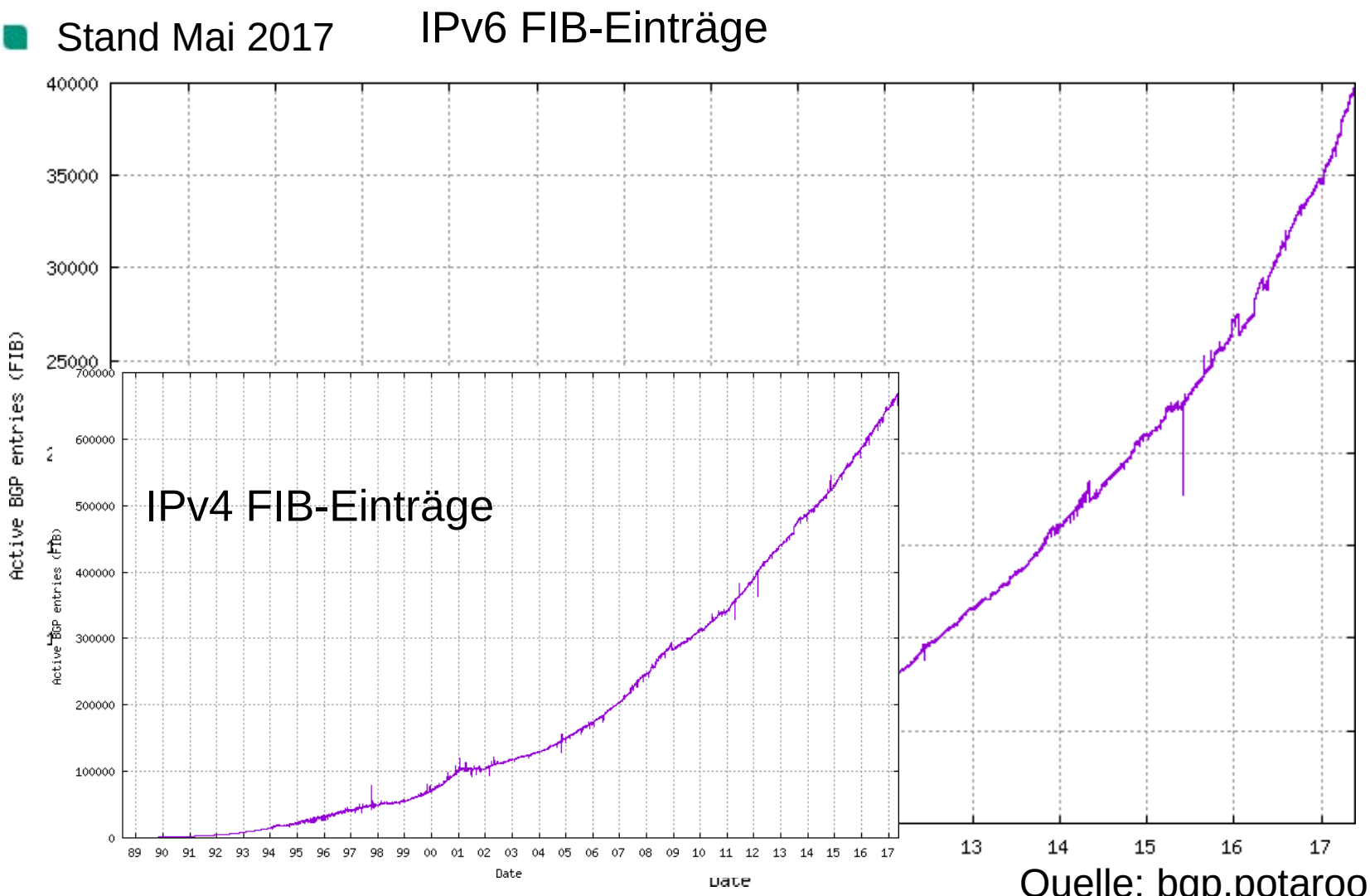
Weitere NAT-Lösungen...

für längere Co-Existenz benötigt, derzeit in Umsetzung:

- NAT64 v6/v4-Übersetzung NAT64/DNS64
  - Zugriff von IPv6only-Systemen auf nur in IPv4 verfügbare 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, Erreichbarkeit?
- „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...

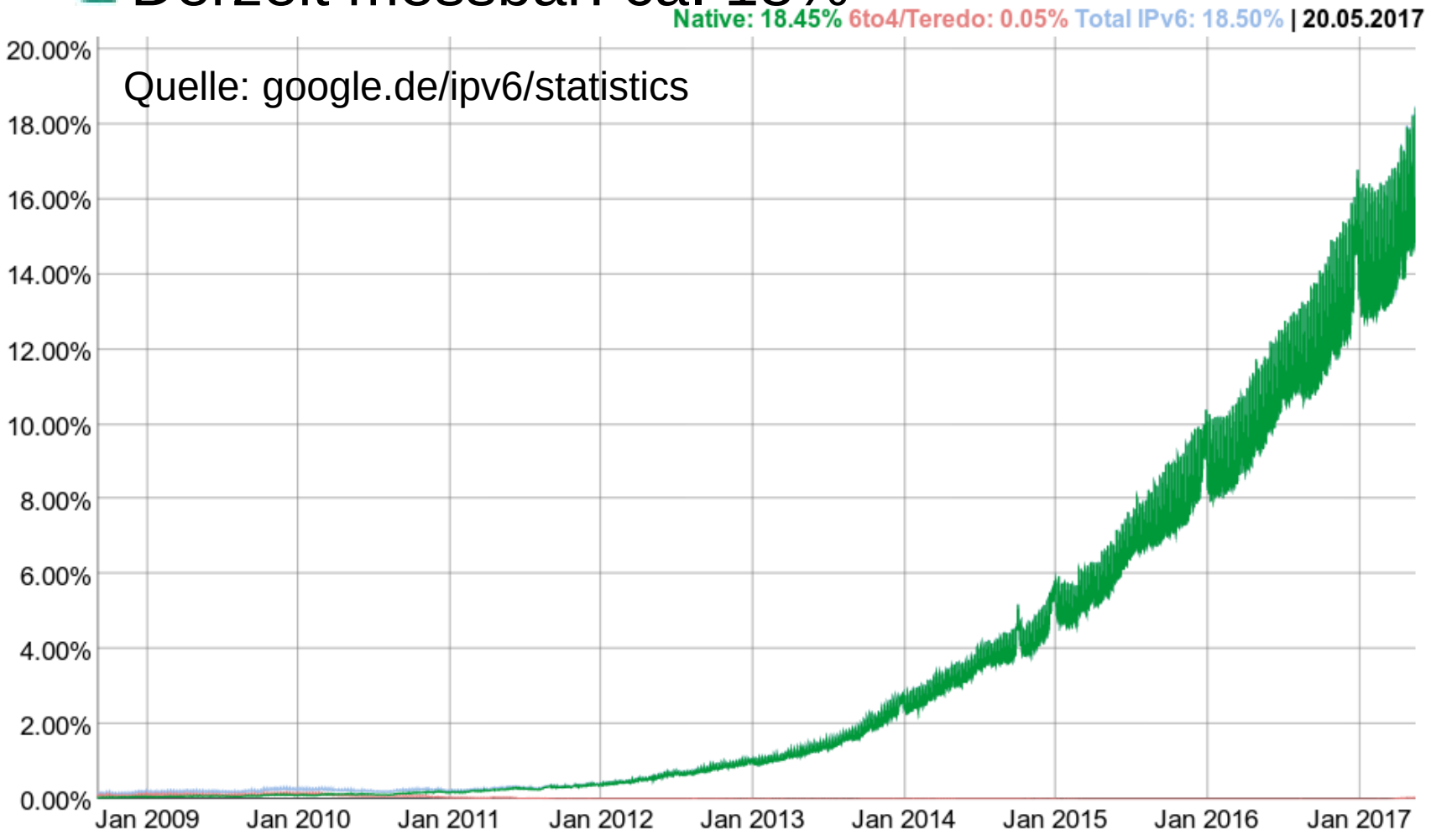
- Stand Mai 2017



Quelle: bgp.potaroo.net

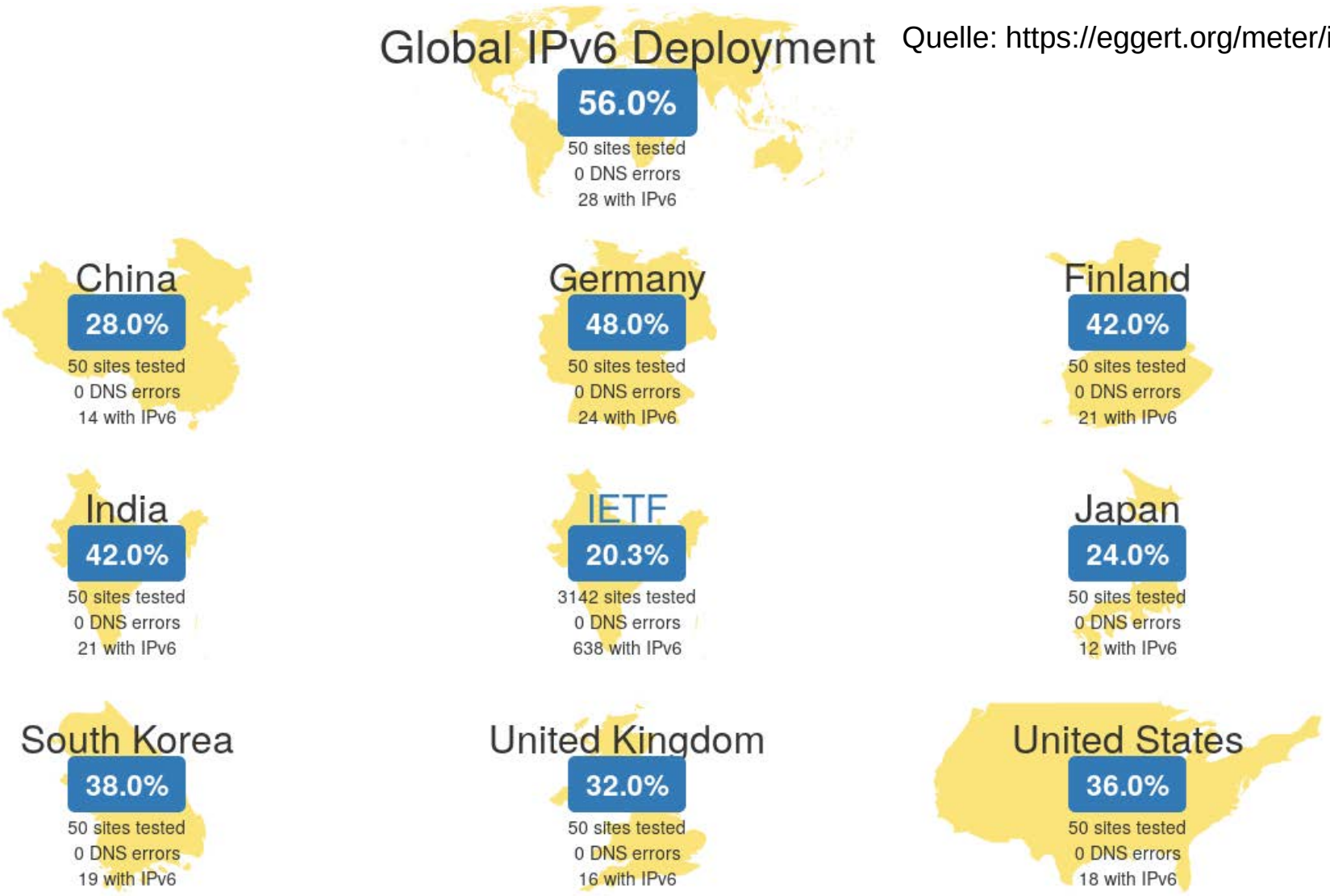
IPv6-Nutzung

- Derzeit messbar: ca. 18%



IPv6-Nutzung Top Web Sites

Quelle: https://eggert.org/meter/ipv6



Kurze Geschichte von IPv6

- 1991:
  - Gründung der Arbeitsgruppe ROAD (Routing and Addressing)
  - Mailingliste „Big-Internet“

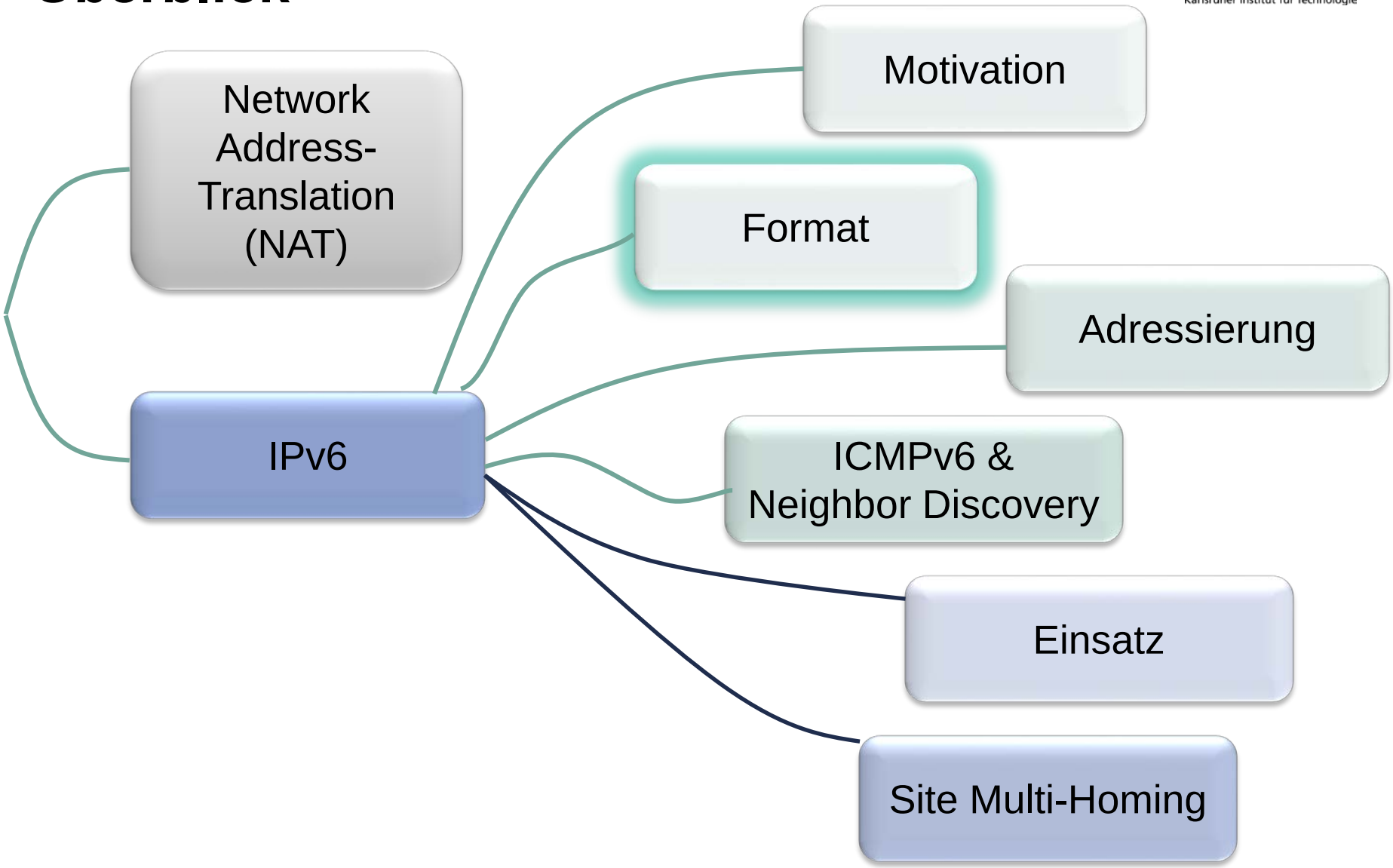
[SAMI06]

- 1992: Vorschläge, u.a.
  - The „P“ Internet Protocol (PIP)
  - TCP and UDP with Bigger Addresses (TUBA)
  - Simple Internet Protocol (SIP)

[Brad06]

- 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 „Full-Standard“ in wenigen Wochen erwartet)
- Sept 2007: IPv6 WG geschlossen, 6MAN WG gegründet

Überblick


Network Address-Translation (NAT)
IPv6
Motivation
Format
Adressierung
ICMPv6 & Neighbor Discovery
Einsatz
Site Multi-Homing

Neuerungen in IPv6 (1)

Grundlegender Dienst immer noch der gleiche wie bei IPv4, aber…

- Erweiterte Adressierung
  - Erhöhung der Adresslänge von 32 Bit auf 128 Bit
    - ca. 3,4 * 10^{38} Adressen
  - Neue Adresstypen (Anycast, Unique Local Addresses, usw.)
- Schnelle Bearbeitung in Routern durch vereinfachtes Paketformat
- Einfachere Klassifikation von Paketen in Datenströme (mit Hilfe des Flow Labels) [RFC6437]

Neuerungen in IPv6 (2)

- 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
- Automatische Systemkonfiguration [RFC4862] (Stateless Address Autoconfiguration – SLAAC)
  - Zustandslos: Präfix über Router Advertisements plus Interface-ID (selbst erzeugt)
  - Zustandsbehaftet: traditionelles DHCP (Dynamic Host Configuration Protocol)

Neuerungen in IPv6 (3)

- Bessere Unterstützung mobiler Systeme (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 (4)

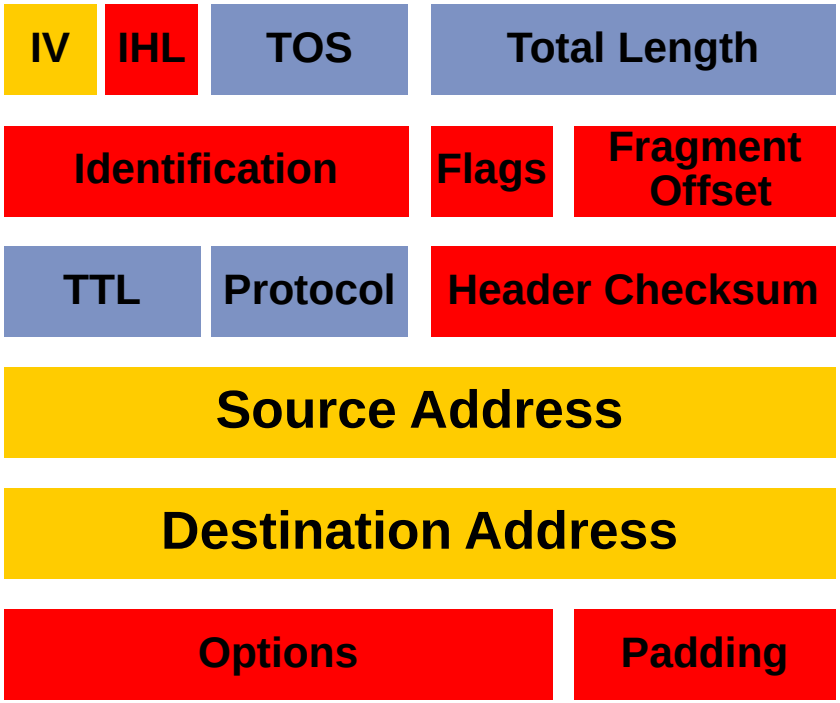
- Berücksichtigung von Sicherheitsaspekten
  - Zwingende Unterstützung von Sicherheitsmechanismen für alle IPv6-Knoten! (aber kein zwingender Einsatz)
  - Unterstützung von Authentifizierung und Datenintegrität
    - Authentifizierung/Integritätssicherung: Authentication Header (AH)-Erweiterungskopf
    - 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 (5)

- Vereinfachung der Administration bzgl. Adressen
  - Vermeidung der Renummerierung von Subnetzen bei Änderung des ISP, z.B. wenn /48-Präfixe vergeben
  - Multi-Homing: Verwendung mehrerer Adressen gleichzeitig, aber skalierbar →Shim6
- 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

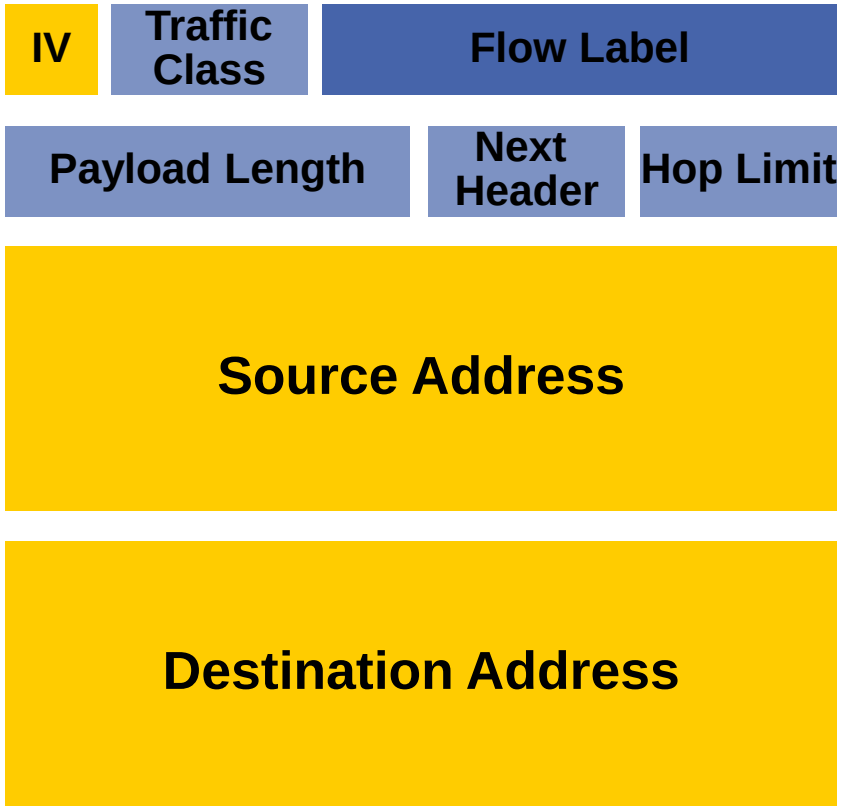
Vergleich Kopffelder IPv6 und IPv4

IPv4-Protokollkopf



- Identische Felder in IPv4 und IPv6
- Felder nur in IPv4
- Geänderte Felder in IPv6
- Neue Felder in IPv6

IPv6-Protokollkopf



IV: IP Version
IHL: IP Header Length
TOS: Type of Service
TTL: Time to Live

Änderungen Paketformat (1)

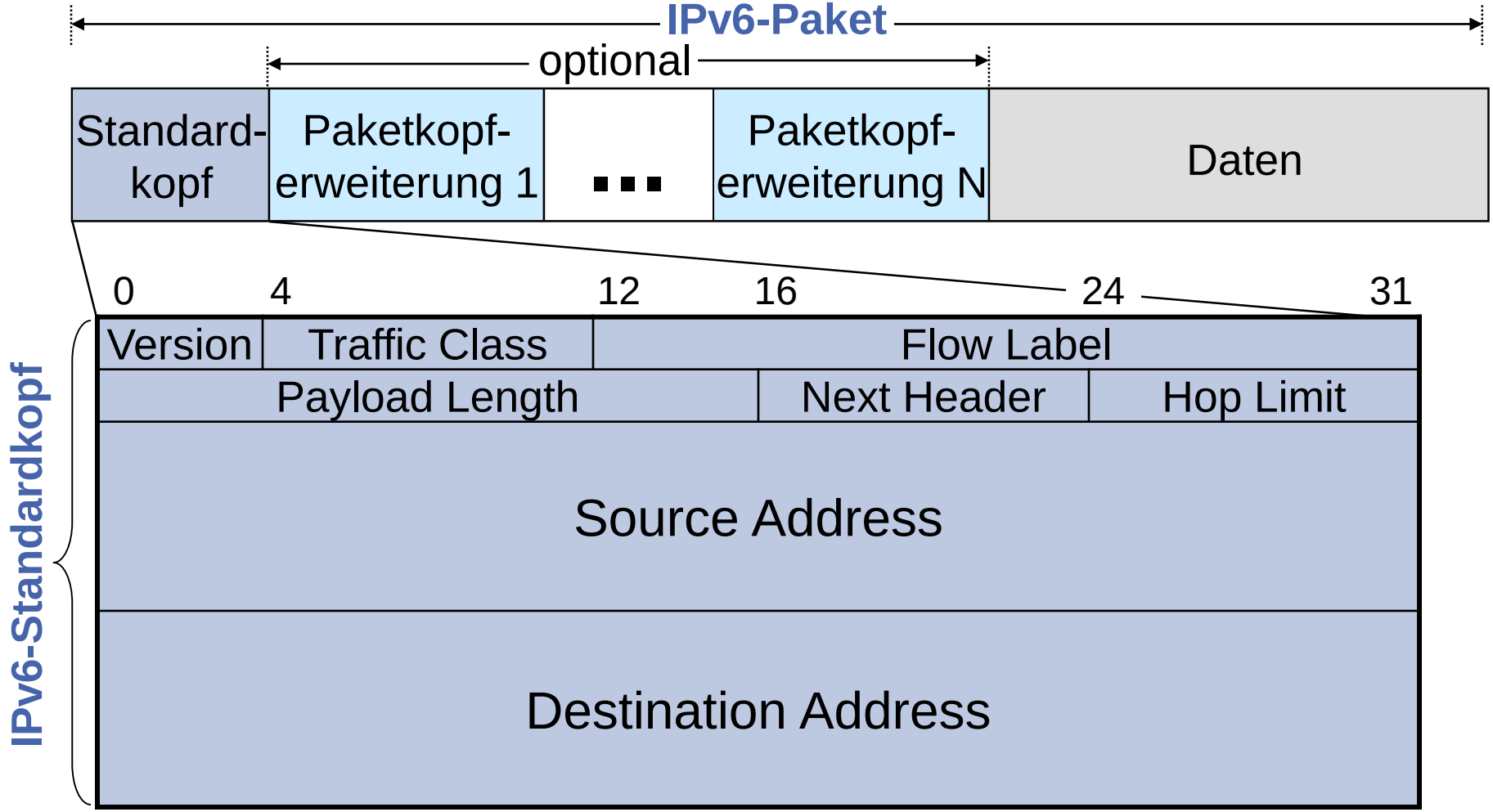
- Standard-Paketkopf mit fester Länge und nur 8 Feldern (13 bei IPv4)
- Keine (Kopf-)Prüfsumme → UDP-Prüfsumme jetzt zwingend
- Keine Hop-by-Hop-Fragmentierung
  → Nur Ende-zu-Ende-Fragmentierung plus Path-MTU-Discovery
- IPv6-Spezifikation erfordert Link-MTU >= 1280 Bytes (IPv4: 68 Bytes)
- Unterstützung von Ressourcenreservierung/ Verkehrslenkung (Flow Label und Traffic Class)
  - Quelle setzt Flow Label, bleibt unverändert

[RFC6437]

- Kette von optionalen Paketkopferweiterungen

IPv6-Paketformat

[RFC2460]



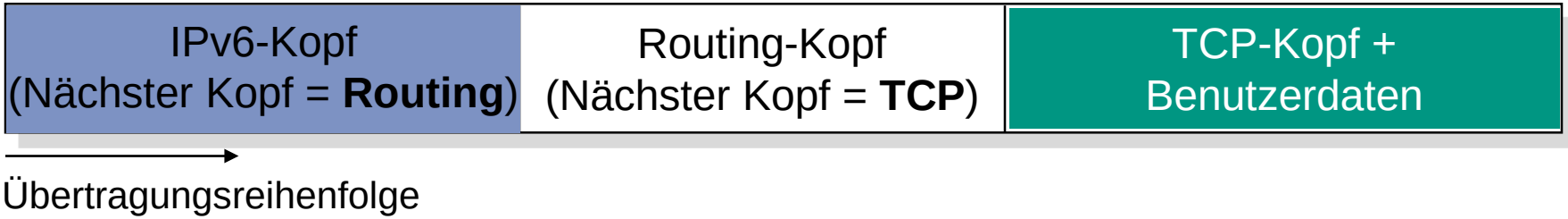
Änderungen Paketformat (2)

- Verschieben von Optionen in flexible Paketkopferweiterungen
  - Kopferweiterungen sind an 64-Bit-Grenzen ausgerichtet
  - sollten zumindest einheitlichen Anfang haben, zwecks Überspringen
  - Einteilung in Erweiterungsköpfe für
    - Zwischensysteme
    - Endsysteme


[RFC6564]

IPv6-Paketkopferweiterungen

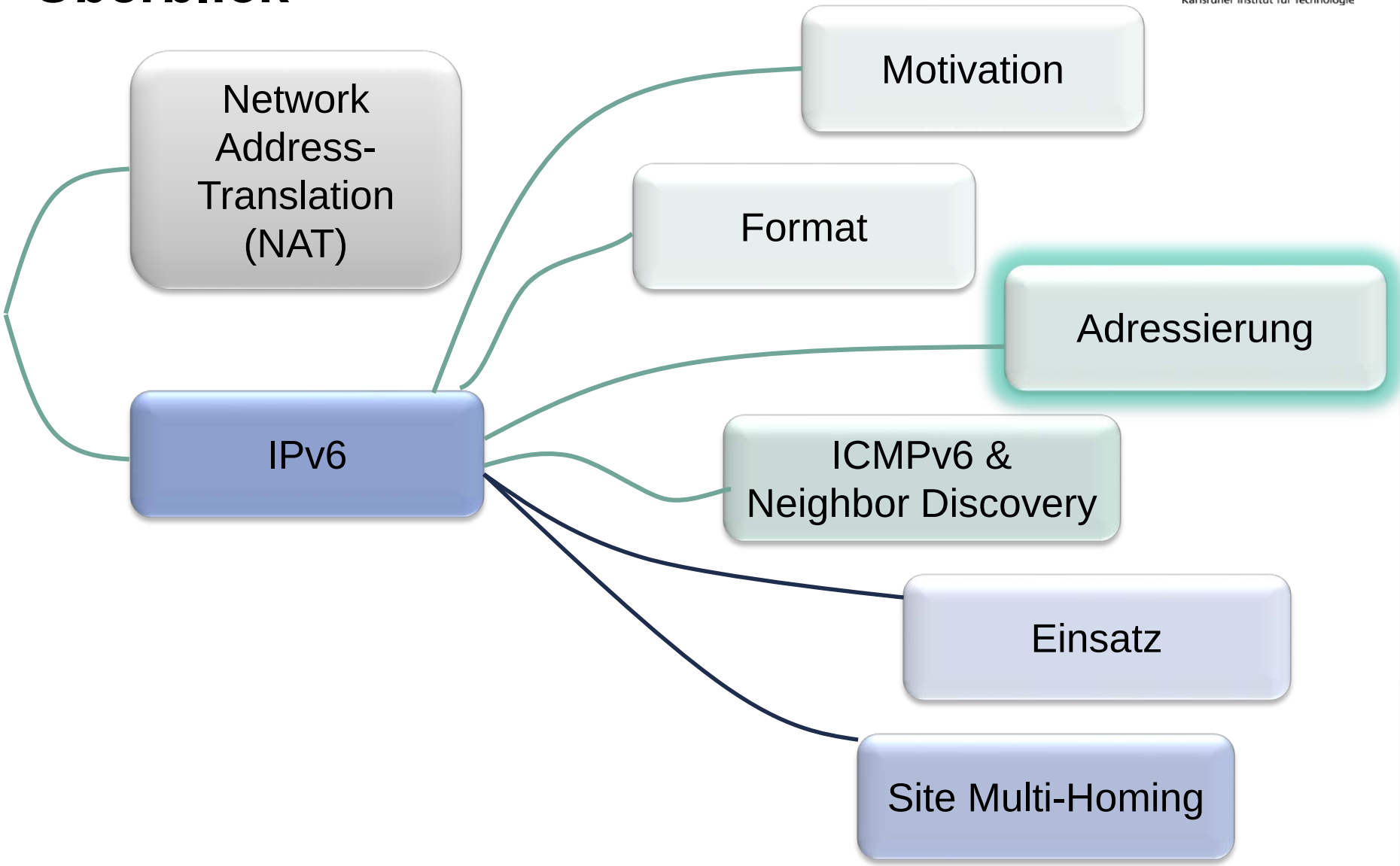
- Feld „Nächster Kopf“ (Next Header): Typ der nachfolgenden Paketkopferweiterung
  - Beispiel:



- 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)
6. Fragmentierung (Fragment Header)
7. Authentifizierung/Integritätssicherung (AH – Authentication Header)
8. Verschlüsselung (ESP – Encapsulating Security Payload)
9. Optionen für endgültiges Ziel (Destination Options (2)) höhere Schicht, z. B. TCP oder UDP

Überblick


Network Address-Translation (NAT)
IPv6
Motivation
Format
Adressierung
ICMPv6 & Neighbor Discovery
Einsatz
Site Multi-Homing

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 empfohlen


[RFC5952]

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
  - 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

[RFC4291]

- IPv6-Adressen sind Netzwerkschnittstellen (Interfaces) und nicht Netzknoten zugeordnet
  - Schnittstelle kann mehrere IPv6-Adressen besitzen
- 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-Embedded IPv6 Address: 64:ff9b::192.0.2.33/96
    zur algorithmischen Übersetzung von IPv4 in IPv6-Adressen

Scoped Address Architecture

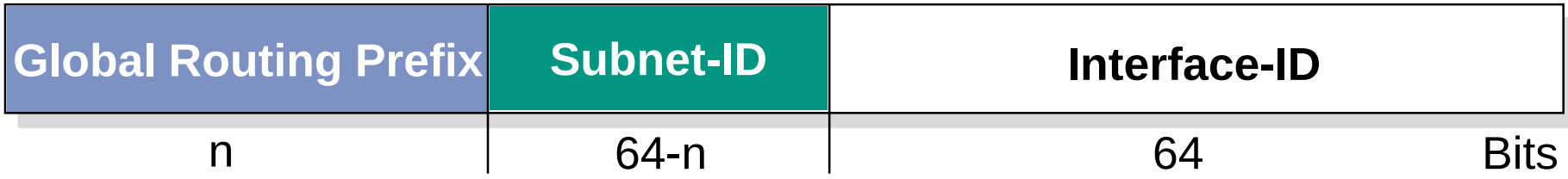

[RFC4007]

- Bereichsbegrenzung (Scope) bei IPv6 fester Bestandteil der Architektur
- 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

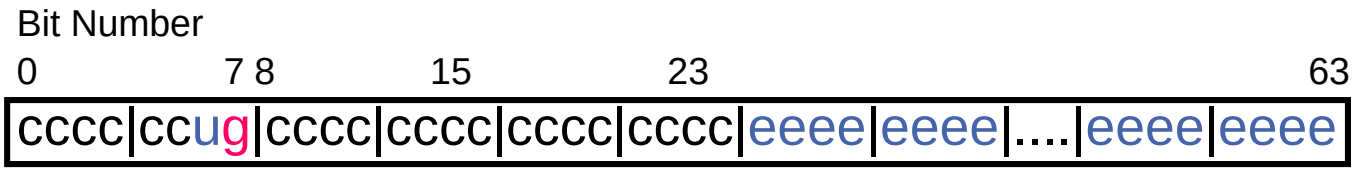
- Struktur [RFC3587]



  - 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 manuell, zufällig oder 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

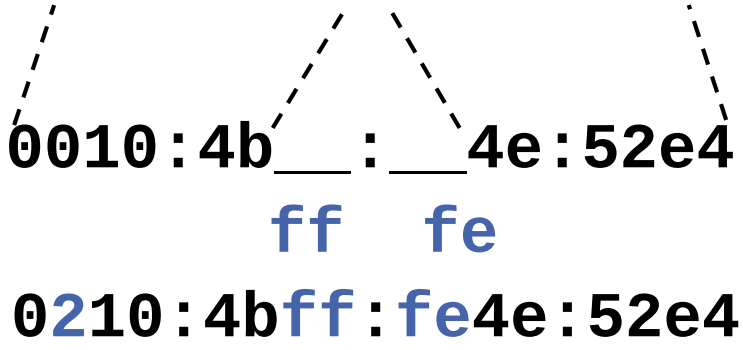
- Ziel: Eindeutigkeit im Subnetz für automatisches Generieren der IP-Adresse (Auto-Konf.) [RFC7136]
- IID weist keine interne Struktur auf!
- Interface kann mehrere IIDs gleichzeitig besitzen
- IID kann manuell oder per DHCP zugewiesen sowie zufällig, kryptographisch bzw. algorithmisch generiert werden
- Ursprüngliche Idee: verwende global eindeutige IDs, basierend auf MAC-Adresse – erleichtert Multi-Homing
  - Grundlage bilden global eindeutige IEEE EUI-64-Adressen:



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

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)



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

Schutz der Privatsphäre (1)

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

Schutz der Privatsphäre (2)

- Problem mit RFC 4941
  - Häufig wechselnde Adressen – nur für ausgehende Kommunikation
  - Stabile EUI-64-basierte Adresse bleibt für Server-Funktionen erhalten
  - Dennoch Korrelation möglich
- Stable Privacy-Enhanced Addresses  [RFC7217]
  - Zusätzlich zu temporären RFC4941-Adressen
  - Algorithmische Methode
  - Gleiche, stabile IID innerhalb des gleichen Subnetzes
  - IID wechselt, wenn Subnetz gewechselt wird
  - Müssen trotz Neustart des Knotens stabil bleiben

Link-Local Unicast-Adressen

- Jedes IPv6-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]

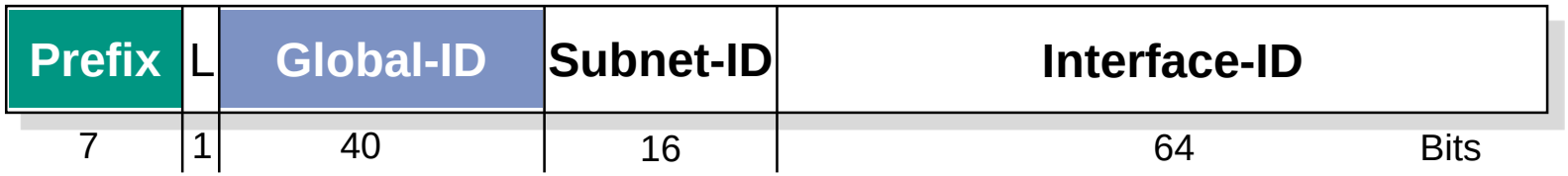
- Site-Locals wurden inzwischen zurückgezogen
- Probleme im Wesentlichen begründet durch
  - 1. Mehrdeutigkeit/Nicht-Eindeutigkeit der Adressen
  - 2. Unscharfe Definition einer „Site“
- Kein Einsatz von Site-Locals!
- Bessere Lösung: ULAs, siehe nächste Folie

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

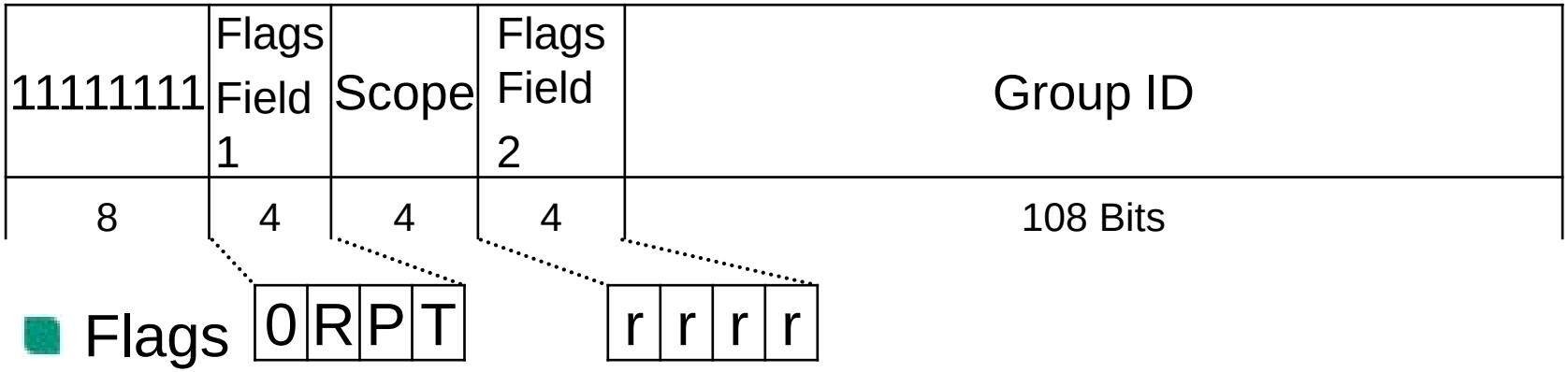
- Aufbau:



  - 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, RFC7371]



  - 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:  [RFC7346]

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

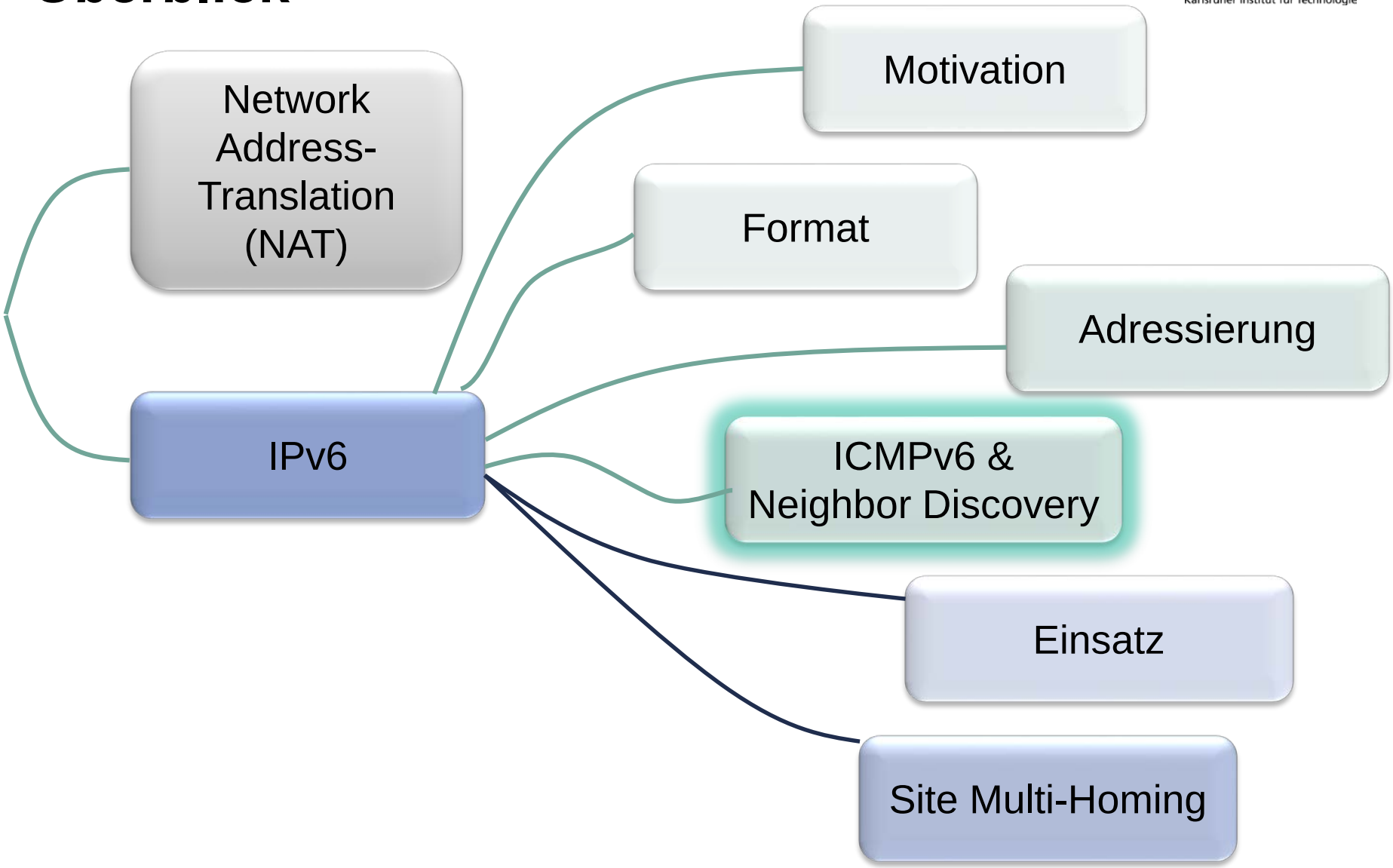
Anycast-Adressen

- 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

Überblick


Network Address-Translation (NAT)
IPv6
Motivation
Format
Adressierung
ICMPv6 & Neighbor Discovery
Einsatz
Site Multi-Homing

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

Neighbor Discovery – Mechanismen

1. Auffinden von Routern und Konfigurationsparametern (Router Discovery)
2. 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

3. Auffinden von On-link-Systemen (Adressauflösung)
4. Erkennung von nicht mehr erreichbaren Nachbarn (Neighbor Unreachability Detection)
5. Erkennung von Adresskonflikten (Duplicate Address Detection)

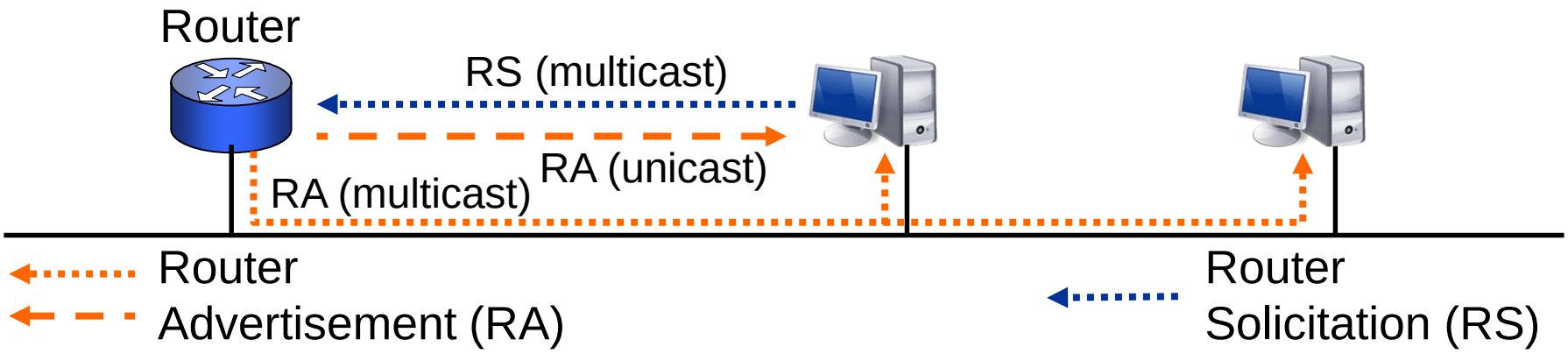
Neighbor Discovery – Nachrichten

- Router Solicitation
  - Anforderung eines Router Advertisement für Schnittstellenkonfiguration
- Router Advertisement
  - Periodisch vom Router versendet oder auf Anforderung durch Router Solicitation
  - Ermöglicht Router-, Prefix- und Parameter-Discovery
- Neighbor Solicitation
  - Adressauflösung Onlink-IPv6-Adresse → Link-Layer-Adresse
  - Erkennung von nicht mehr erreichbaren Nachbarn
  - Duplicate Address Detection
- Neighbor Advertisement
  - Antwortnachricht auf Neighbor Solicitation
- Redirect – Hinweis (Router → Host) auf bessere Route

Router-Erkennung (Router Discovery)

- Problem: Ermittlung eines Routers zum Senden von Paketen an Rechner außerhalb des eigenen Netzsegments (Off-Link Ziele)
  - Router-Adresse unbekannt: welche Zieladresse verwenden?
  - Schicht-2-Adresse des anfragenden Systems für Antwort notwendig
- 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 RA anfordern, welches dann zeitlich zufällig verzögert
    - per Multicast an All-Nodes gesendet wird (Regelfall)
    - oder per Unicast gesendet werden kann
  - Router Discovery benötigt Multicast und Link-lokale Adresse
  - Optimierung: anfragendes System schickt Link-Layer-Adresse in RS gleich mit, Router merkt sich diese für Antwort

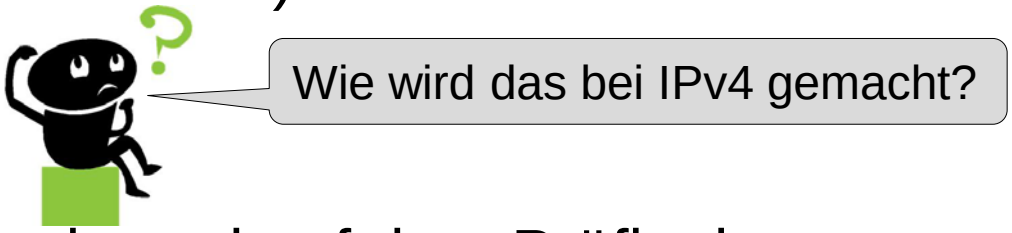
Autokonfiguration



- RA enthält
  - Router-IP-Adresse (Link-local) und ggf. Link-Layer-Adresse
    → Eintrag in die Default Router List, Link-Layer Adresse im Neighbor Cache
  - Konfigurationshinweis für Adressenkonfiguration (Stateless Address Autoconfiguration und/oder DHCPv6)
  - Präfix-Liste für die Konfiguration globaler Adressen und Onlink-Bestimmung
  - Weitere Parameter, z.B. MTU-Größe, Gültigkeitsdauern d. Präfixe

Präfix-Erkennung (Prefix Discovery)

- Problem: Absender eines IP-Paketes muss feststellen, ob sich der Zielrechner im eigenen Subnetz befindet (direktes Senden oder Senden über Router)



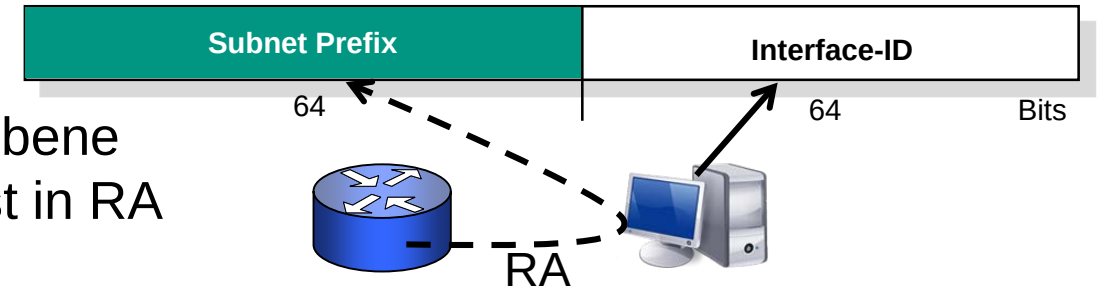
- Lösung: Entscheidung basierend auf dem Präfix des eigenen Subnetzes
  - Router-Advertisement-Nachrichten enthalten Präfix-Listen des Subnetzes
  - Vergleich der Zieladresse mit den Präfixen durch logische UND-Verknüpfung
  - Entspricht das Präfix der Zieladresse einem Präfix des Subnetzes, so wird das Paket direkt gesendet (On-link Ziel); ansonsten wird es an einen Router übermittelt (Off-Link Ziel)

Adressenkonfiguration

- Adresskonfiguration Alternativen
  - Stateless
    - Stateless Address Autoconfiguration (SLAAC)
    - Für jedes Präfix in Präfixliste generiere Adresse durch Kombination aus Präfix und Interface-ID
    - Duplicate Address Detection (DAD) notwendig
  - Stateful
    - per DHCPv6 zugewiesene Adresse
    - Duplicate Address Detection trotzdem notwendig
  - RA enthält M/O-Flags als Hinweis
    - M= „Managed Address Configuration“ → vollständige Adressenkonfiguration via DHCPv6
    - O= „Other Configuration“ → zusätzliche Konfiguration via DHCPv6, z.B. DNS-Server-Adressen

Zustandslose Adressautokonfiguration

- Stateless Address Autoconfiguration vereinfacht das Anschließen von IPv6-Hosts
- 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 aus Prefix List in RA



- Erfordert minimale Konfiguration der Router (falls überhaupt)

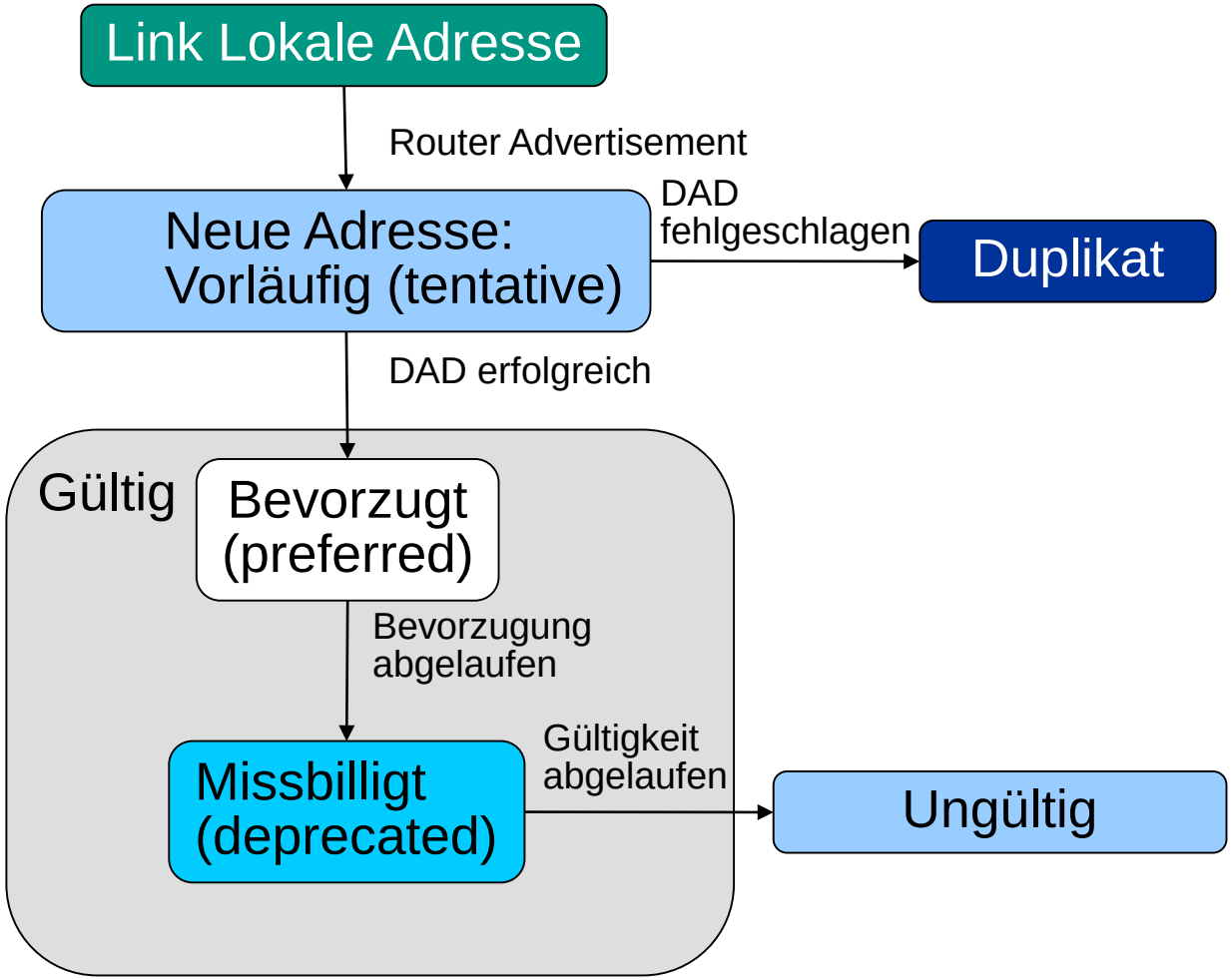
Adressautokonfiguration

- 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
    - Wird per RA verteilt

Autokonfiguration der Link-lokalen Adresse

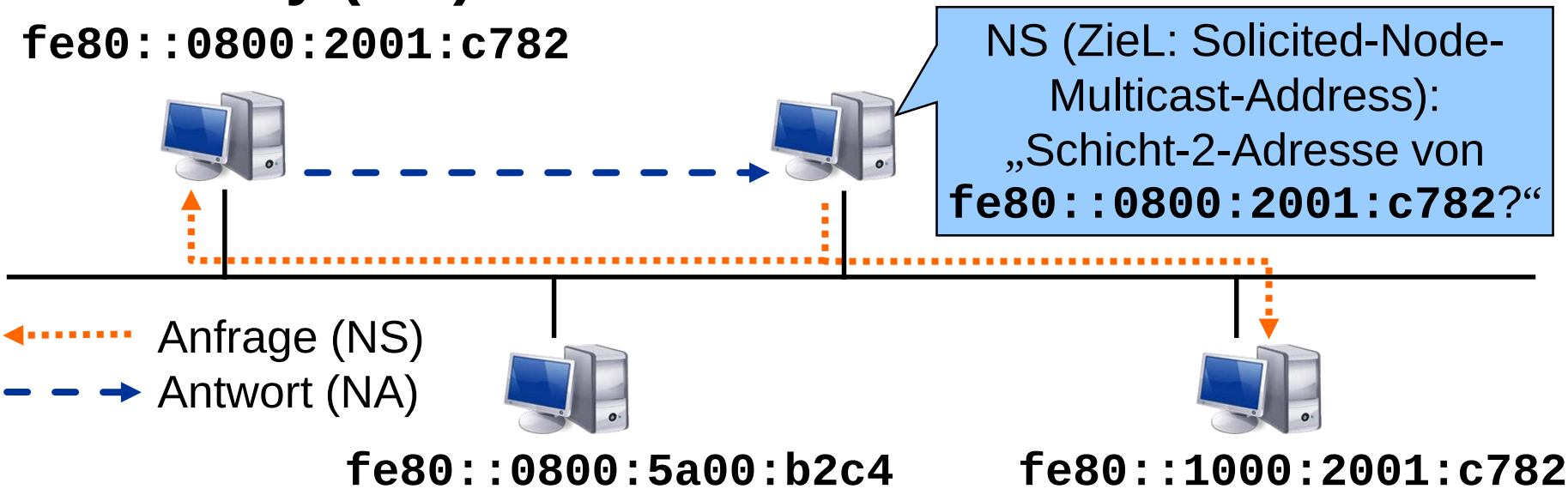
- Erzeugen der Interface-ID aus MAC-Adresse
- Ergänzen des Präfix fe80::0 mit IID
- ergibt zusammen eine vorläufige (tentative) Link-lokale Adresse
- Durchführen der Duplicate Address Detection
  - Anfrage nach Link-Layer-Adresse zu eigener IP-Adresse
  - Erhalt von Antwort: IP-Adresse bereits in Benutzung!
  - Ausbleiben einer Antwort: IP-Adresse eindeutig

Adressenkonfiguration



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

Adressauflösung durch Neighbor Discovery (ND)



- Problem: Ermittlung der Schicht-2-Adresse (=Link-Layer-Adresse) zu einer (On-Link) IPv6-Unicast-Adresse
- Lösung: Anfrage mittels Neighbor Solicitation (NS)
  - Rechner hören auf ihre Solicited-Node Multicast Address
    im Bsp.: ff02::1:ff01:c782 bzw. ff02::1:ff00:b2c4
  - NS an die Solicited Nodes Address des Zielrechners – kein(!) Broadcast
  - Zielrechner übermittelt seine Schicht-2-Adresse an den anfragenden Rechner in Neighbor Advertisement (NA) (Unicast)

Verwendung von Multicast

- Einsatz von Multicast für
  - Router Discovery
    - All routers ff02::2
    - All nodes multicast address ff02::1 (ohne MLD)
  - Neighbor Discovery
    - Solicited-Node Multicast Address
  - Duplicate Address Detection
    - 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...

Multicast-Adressen → Link Layer

- Abbildung von einer IP-Multicast-Adresse auf Link-Layer-Adresse?
  - Nutzung von Link-Layer-Gruppenadressen
- 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!

Konzeptionelle Datenstrukturen

- Es gibt vier verschiedene konzeptionelle Caches in jedem Endsystem:
  - 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
    - 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

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 – 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 ND für Zuschicken des NA durchgeführt werden
    - Empfang eines Solicited NA (Solicited Flag=1) zeigt bidirektionale Erreichbarkeit an

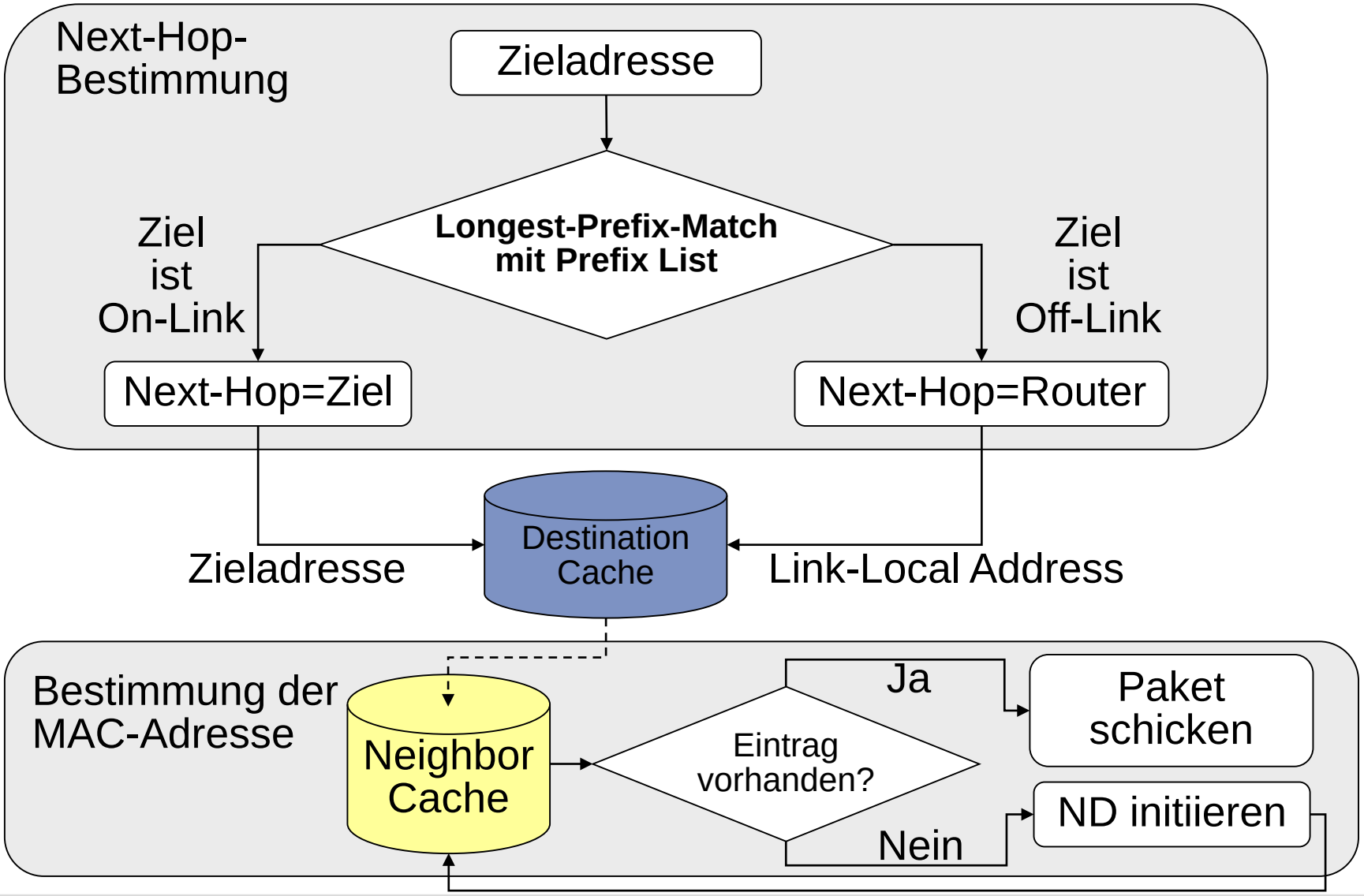
Adressauflösung – Subnet Anycast

- Anycast im lokalen Subnetz funktioniert direkt über Neighbor Discovery
- Falls eine per NS angefragte Adresse als Anycast-Adresse konfiguriert ist
  - NA wird zufällig verzögert
  - NA mit Override-Flag=0, so dass erste Antwort im Neighbor Cache nicht von nachfolgenden Antworten überschrieben wird
  - das zuerst antwortende System wird kontaktiert

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 oder Empfang eines RA
  - 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

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

Duplicate Address Detection

- Für welche Adressen wird keine DAD ausgeführt?
  - Link local Unicast
  - Global Unicast
  - Unique Local Addresses
  - Multicast
  - Anycast
- Pingo http://pingo.upb.de/8577

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

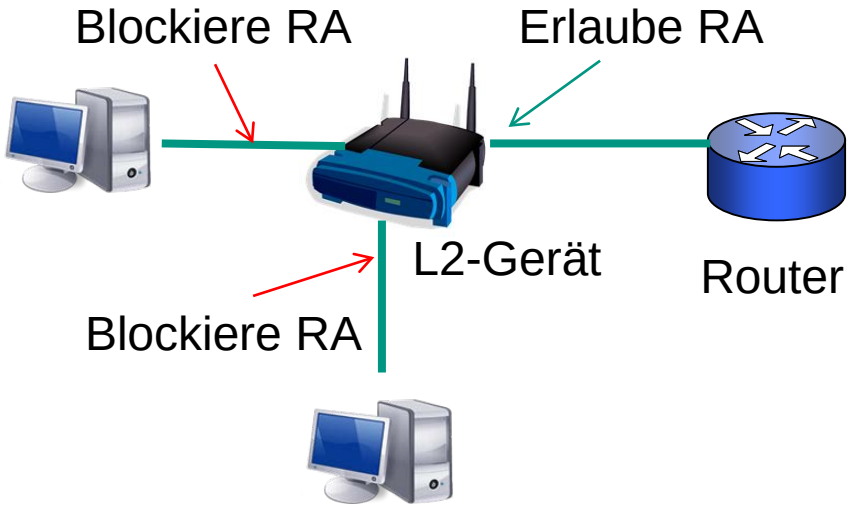
- Router Discovery ermöglicht einfachen Man-in-the-Middle oder DoS-Angriff → Absicherung notwendig
- Identität von Routern kann anhand von Zertifikaten und Zertifizierungspfaden überprüft werden.
  - Überprüfung zurückgezogener Zertifikate unklar

Alternative „RA Guard“

- SEND nicht verbreitet

[RFC6105]

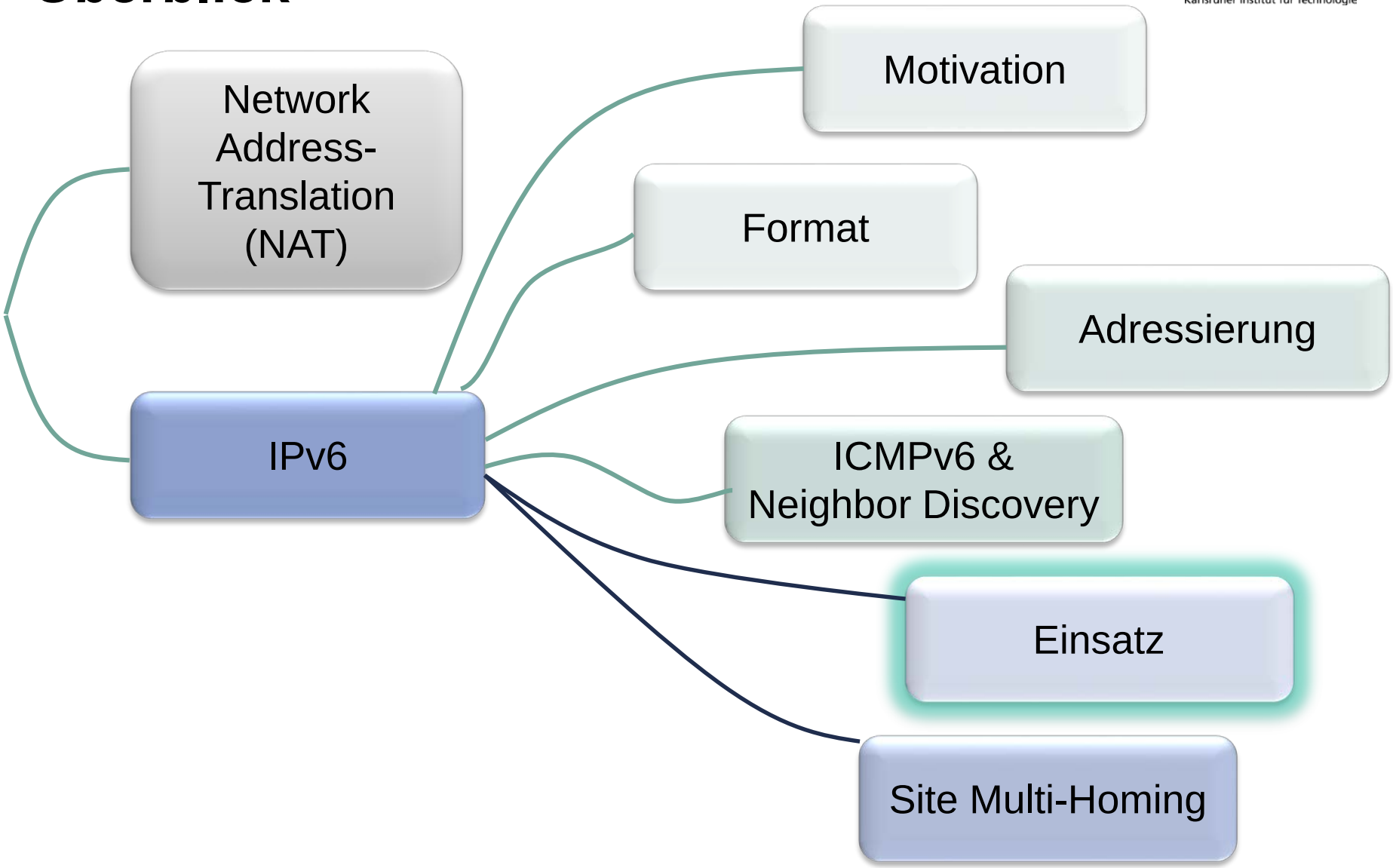
- Leichtgewichtiger Mechanismus RA Guard
  - Auch als Ergänzung zu SEND
- Idee
  - Blockiere RA auf Ports an denen kein Router angeschlossen ist



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)  [RFC8106]
  - Stateless DHCPv6  [RFC3736]
- Möglicherweise auch alle drei Methoden kombinierbar

Überblick


Network Address-Translation (NAT)
IPv6
Motivation
Format
Adressierung
ICMPv6 & Neighbor Discovery
Einsatz
Site Multi-Homing

Übergang IPv4 → IPv6

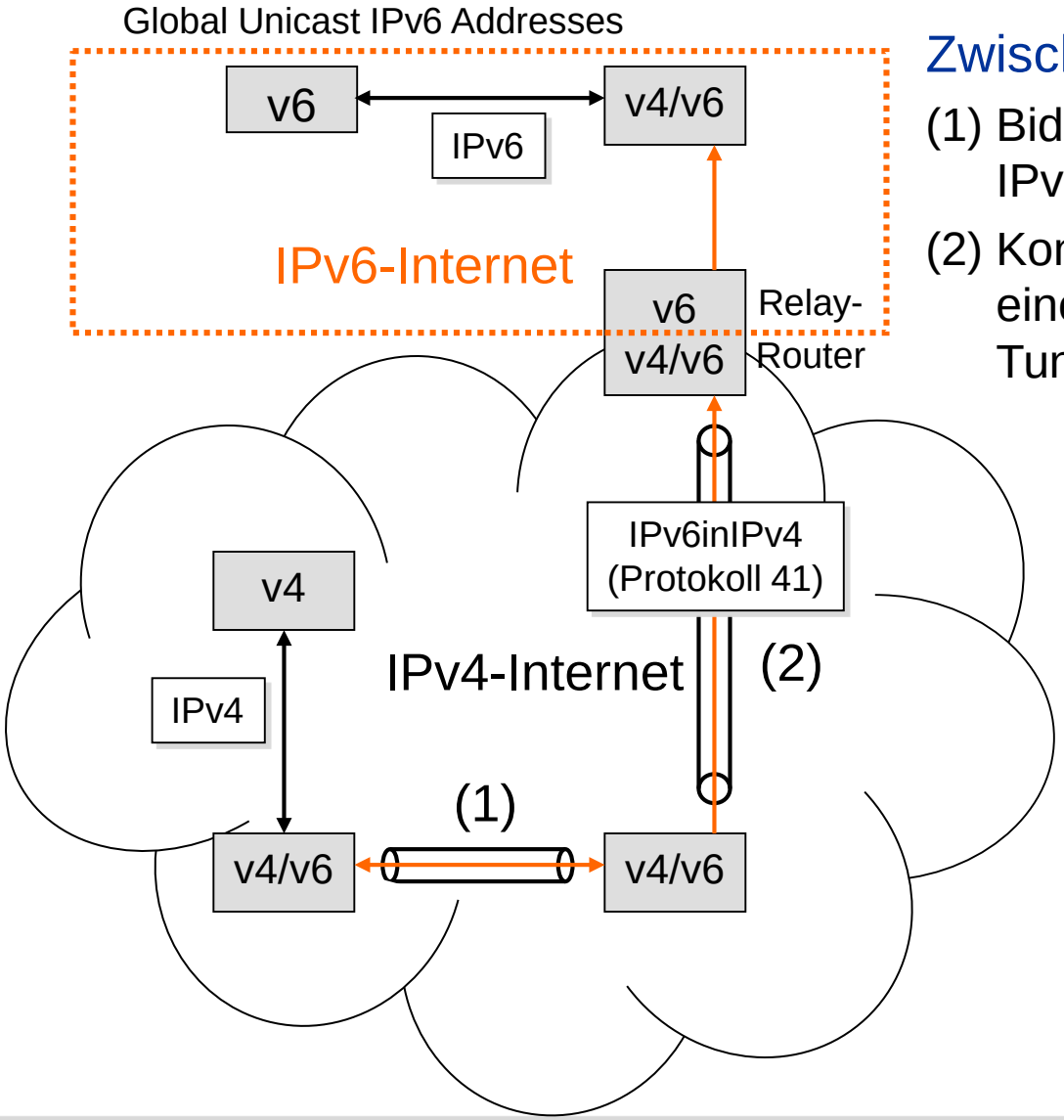
- Grundlegende Übergangsmechanismen  [RFC4213]
  - Dual IP Layer (Dual-Stack): IPv6-fähige Knoten besitzen eine IPv4 und (mind.) eine IPv6-Adresse
    - Entscheidung über Verwendung von IPv4 oder IPv6 wird durch DNS bzw. Anwendung getroffen
    - Existiert eine IPv6-Adresse? AAAA Resource Record
  - Tunnelmechanismen
    - IPv6-Paket wird in IPv4-Paket eingepackt
    - Zusatzaufwand (IPv4-Kopf + Ein-/Auspacken) verringert Leistung
    - Konfigurierte Tunnel
    - Automatische Tunnel: 6to4, Teredo
- Ergänzung durch NAT64/DNS64

Beispiel: DNS-Anfrage nach AAAA-Records

```
>dig AAAA www.tm.kit.edu

; <<>> DiG 9.7.0-P1 <<>> AAAA www.tm.kit.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37233
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 6, ADDITIONAL: 0

;; QUESTION SECTION:
;www.tm.kit.edu.                        IN      AAAA

;; ANSWER SECTION:
www.tm.kit.edu.         68285   IN      AAAA    2a00:1398:2:4000::42

;; AUTHORITY SECTION:
edu.                    126084  IN      NS      g.edu-servers.net.
edu.                    126084  IN      NS      d.edu-servers.net.
edu.                    126084  IN      NS      l.edu-servers.net.
edu.                    126084  IN      NS      a.edu-servers.net.
edu.                    126084  IN      NS      c.edu-servers.net.
edu.                    126084  IN      NS      f.edu-servers.net.

;; Query time: 0 msec
;; SERVER: 2a00:1398:2:4000::11#53(2a00:1398:2:4000::11)
;; WHEN: Mon Feb 27 16:08:34 2012
;; MSG SIZE  rcvd: 171
```

DNS-Anfrage via IPv6!

Übergang IPv4 → IPv6 – Dual Stack Strategie

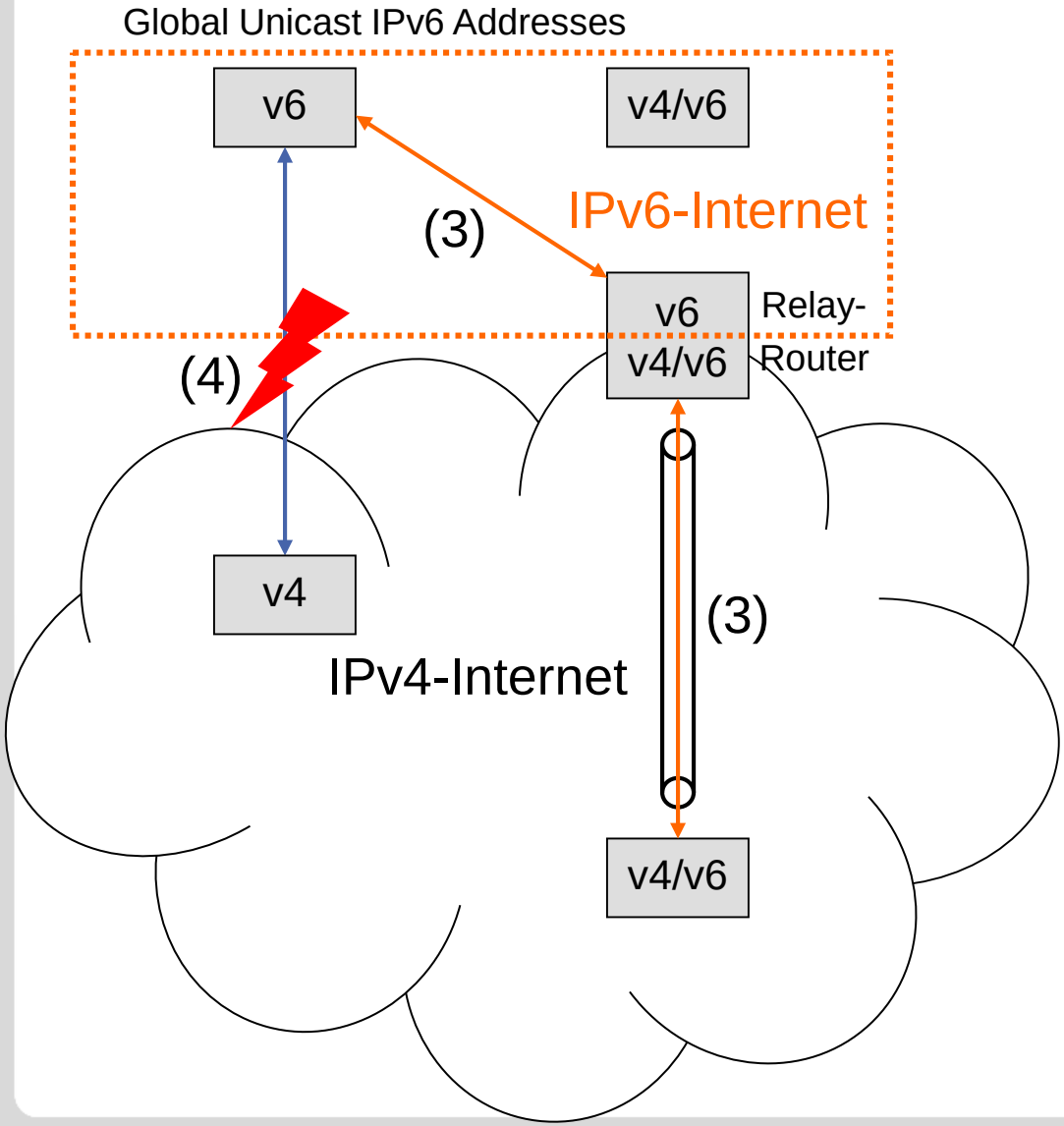


Zwischen v4/v6 Dual Stack Hosts

(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)

v6-only mit v4-only

(4) Keine direkte Kommunikation möglich! Nur Indirekt mittels IPv4/IPv6-Gateways → NAT64

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, Windows7-10
- Die meisten Anwendungen funktionieren bereits mit IPv6

Änderungen im Routing

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

- Inter-Domain-Routing  [RFC4760]
  - MBGP – Multiprotocol BGP kann u.a. IPv6-Adressen transportieren
- Intra-Domain-Routing  [RFC5340]
  - OSPFv3 ist für IPv6 entwickelt worden
  - IS-IS kann ebenfalls mit kleiner Änderung IPv6 Routen tragen

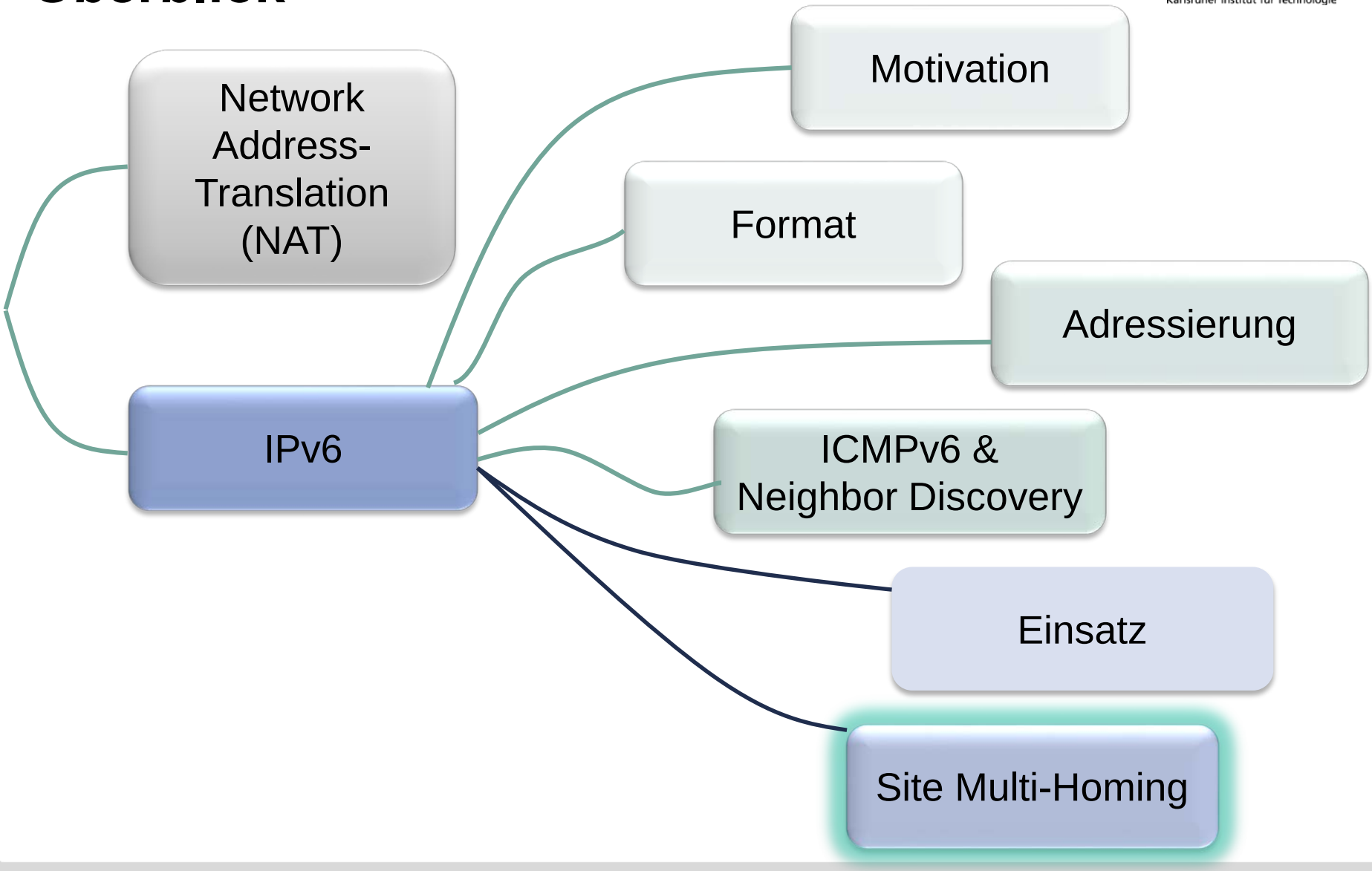
Wieso bisher kaum IPv6?

- Relativ späte Unterstützung durch Router- und Betriebssystemhersteller
  - Halbherzige Implementierungen in Software…
  - 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

Überblick

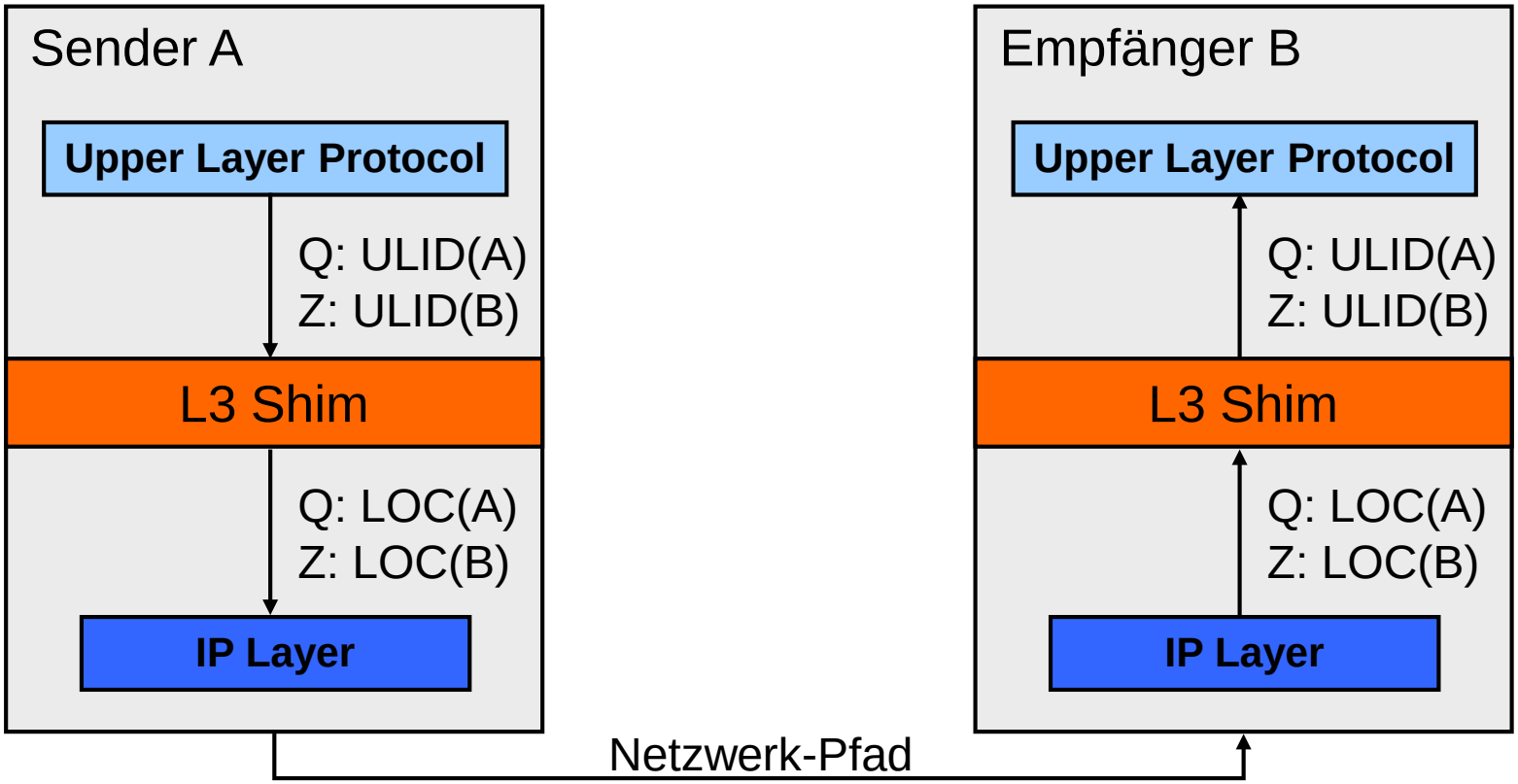

Network Address-Translation (NAT)
IPv6
Motivation
Format
Adressierung
ICMPv6 & Neighbor Discovery
Einsatz
Site Multi-Homing

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

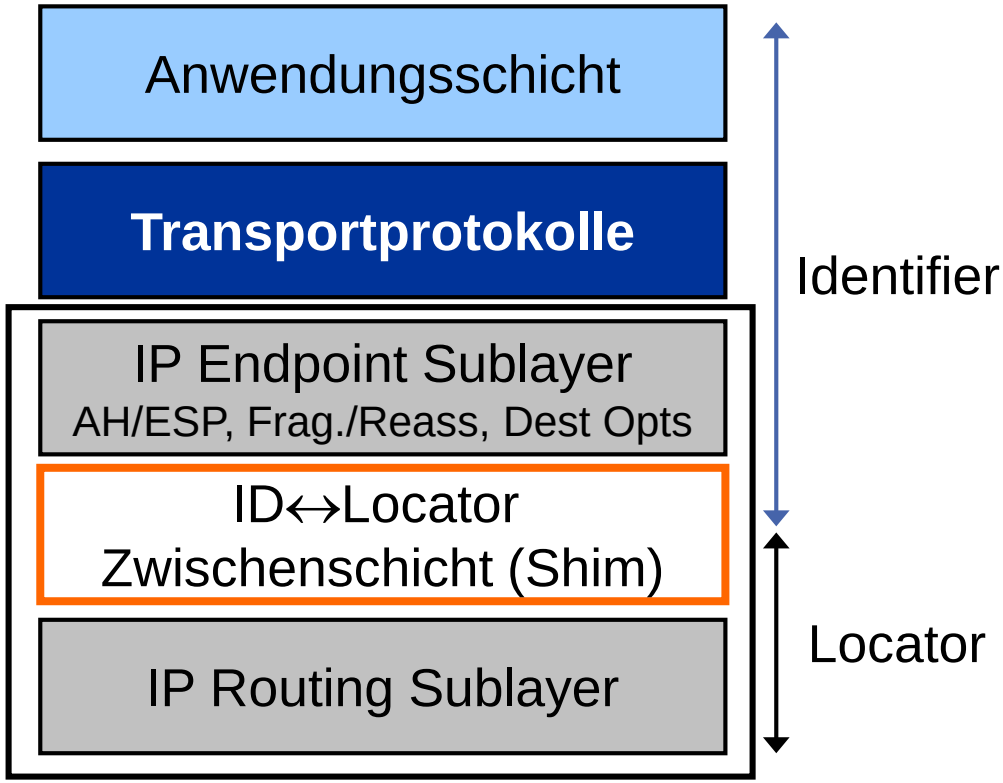
Site Multi-Homing: Grobkonzept

- 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

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?

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

[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

[LiCe11] Thomas A. Limoncelli, Vinton G. Cerf: Successful Strategies for IPv6 Rollouts. Really. Communications of the ACM Vol. 54 No. 4, Pages 44-48 DOI 10.1145/1924421.1924438, http://cacm.acm.org/magazines/2011/4/106582-successful-strategies-for-ipv6-rollouts-really/fulltext

Video: http://www.youtube.com/v/mZo69JQoLb8#13m1s#!flashvars#playerMode=embedded

[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

Literatur (3)

[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

[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 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

Literatur (4)

[RFC 3027] M. Holdrege und P. Srisuresh. Protocol Complications with the IP Network Address Translator. RFC 3027 (Informational), Januar 2001. http://www.ietf.org/rfc/rfc3027.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, 7371. http://www.ietf.org/rfc/rfc3306.txt

[RFC 3493] R. Gilligan, S. Thomson, J. Bound, J. McCann und W. 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

Literatur (5)

[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. Updated by RFCs 6494, 6495, 6980. URL: http://www.ietf.org/rfc/rfc3971.txt

[RFC 3972] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard), März 2005. Updated by RFCs 4581, 4982. URL: http://www.ietf.org/rfc/rfc3972.txt

Literatur (6)

[RFC 3986] T. Berners-Lee, R. Fielding und L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (INTERNET STANDARD), Januar 2005. Updated by RFCs 6874, 7320. URL: 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. Updated by RFC 7346. URL: 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. URL: 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. URL: 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. Updated by RFCs 5952, 6052, 7136, 7346, 7371, 8064. URL: http://www.ietf.org/rfc/rfc4291.txt

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

Literatur (7)

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

[RFC 4760] T. Bates, R. Chandra, D. Katz und Y. Rekhter. Multiprotocol Extensions for BGP-4. RFC 4760 (Draft Standard), Januar 2007. Updated by RFC 7606. URL: http://www.ietf.org/rfc/rfc4760.txt

[RFC 4787] F. Audet und C. Jennings. Network Address Translation (NAT) Behavioral Requirements for Unicast UDP. RFC 4787 (Best Current Practice), Januar 2007. Updated by RFCs 6888, 7857. URL: 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. Updated by RFCs 5942, 6980, 7048, 7527, 7559, 8028. 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. Updated by RFC 7527. URL: http://www.ietf.org/rfc/rfc4862.txt

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

Literatur (8)

[RFC 5340] R. Coltun, D. Ferguson, J. Moy und A. Lindem. OSPF for IPv6. RFC 5340 (Proposed Standard), Juli 2008. Updated by RFCs 6845, 6860, 7503. URL: http://www.ietf.org/rfc/rfc5340.txt

[RFC 5389] J. Rosenberg, R. Mahy, P. Matthews und D. Wing. Session Traversal Utilities for NAT (STUN). RFC 5389 (Proposed Standard), Oktober 2008. Updated by RFC 7350. URL: http://www.ietf.org/rfc/rfc5389.txt

[RFC 5533] E. Nordmark und M. Bagnulo. Shim6: Level 3 Multihoming Shim Protocol for IPv6. RFC 5533 (Proposed Standard), Juni 2009. URL: 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. Updated by RFC 8155. URL: http://www.ietf.org/rfc/rfc5766.txt

[RFC 5952] S. Kawamura und M. Kawashima. A Recommendation for IPv6 Address Text Representation. RFC 5952 (Proposed Standard), August 2010. URL: http://www.ietf.org/rfc/rfc5952.txt

[RFC 5969] W. Townsley und O. Troan. IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification. RFC 5969 (Proposed Standard), August 2010. URL: http://www.ietf.org/rfc/rfc5969.txt

Literatur (9)

[RFC 6104] T. Chown und S. Venaas. Rogue IPv6 Router Advertisement Problem Statement. RFC 6104 (Informational), Februar 2011. URL: http://www.ietf.org/rfc/rfc6104.txt

[RFC 6105] E. Levy-Abegnoli, G. Van de Velde, C. Popoviciu und J. Mohacsi. IPv6 Router Advertisement Guard. RFC 6105 (Informational), Februar 2011. Updated by RFC 7113. URL: http://www.ietf.org/rfc/rfc6105.txt

[RFC 6177] T. Narten, G. Huston und L. Roberts. IPv6 Address Assignment to End Sites. RFC 6177 (Best Current Practice), März 2011. URL: http://www.ietf.org/rfc/rfc6177.txt

[RFC 6437] S. Amante, B. Carpenter, S. Jiang und J. Rajahalme. IPv6 Flow Label Specification. RFC 6437 (Proposed Standard), November 2011. URL: http://www.ietf.org/rfc/rfc6437.txt

[RFC 6564] S. Krishnan, J. Woodyatt, E. Kline, J. Hoagland und M. Bhatia. A Uniform Format for IPv6 Extension Headers. RFC 6564 (Proposed Standard), Internet Engineering Task Force, April 2012. URL: http://www.ietf.org/rfc/rfc6564.txt

Literatur (10)

[RFC 6598] J. Weil, V. Kuarsingh, C. Donley, C. Liljenstolpe und M. Azinger. IANA-Reserved IPv4 Prefix for Shared Address Space. RFC 6598 (Best Current Practice), Internet Engineering Task Force, April 2012. URL: http://www.ietf.org/rfc/rfc6598.txt

[RFC 6887] D. Wing, S. Cheshire, M. Boucadair, R. Penno und P. Selkirk. Port Control Protocol (PCP). RFC 6887 (Proposed Standard), Internet Engineering Task Force, April 2013. Updated by RFC 7488. URL: http://www.ietf.org/rfc/rfc6887.txt

[RFC 7136] B. Carpenter und S. Jiang. Significance of IPv6 Interface Identifiers. RFC 7136 (Proposed Standard), Internet Engineering Task Force, Februar 2014. URL: http://www.ietf.org/rfc/rfc7136.txt

[RFC 7217] F. Gont. A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC). RFC 7217 (Proposed Standard), Internet Engineering Task Force, April 2014. URL: http://www.ietf.org/rfc/rfc7217.txt

Literatur (11)

[RFC 7371] M. Boucadair und S. Venaas. Updates to the IPv6 Multicast Addressing Architecture. RFC 7371 (Proposed Standard), Internet Engineering Task Force, September 2014. URL: http://www.ietf.org/rfc/rfc7371.txt.

[RFC 8106] J. Jeong, S. Park, L. Beloeil und S. Madanapalli. IPv6 Router Advertisement Options for DNS Configuration. RFC 8106 (Proposed Standard), März 2017. URL: http://www.ietf.org/rfc/rfc8106.txt

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