

**Modul 9.1:
Autokonfiguration**

Lernziele:

Nach Durcharbeiten dieses Teilkapitels sollen Sie die Aufgabenstellung „Autokonfiguration“ erläutern und die beiden Konzepte SLAAC und DHCPv6 zur automatischen Konfiguration darstellen können.



Warum Autokonfiguration?

- Bei **jedem** netzwerkfähigen Gerät ist ein Minimum an Konfiguration notwendig. (Warum eigentlich?)
- Manuelle Konfiguration ist **umständlich**.
- Der Anwender hat oft nicht das notwendige **technische Vorwissen** für die Konfiguration seiner Systeme.
- Manuelle Konfiguration ist **fehleranfällig**.
- Manuelle Konfiguration wird bei zukünftigen Geräten und Anwendungen **kaum noch praktikabel** sein.
Beispiele: Fernwartung im Auto, Sensornetzwerke über IP (Rollladensteuerung), Management elektrischer Geräte (Kühlschrank, Waschmaschine, Trockner) durch die Energieerzeuger über IP.
- Automatische Konfiguration erspart dem Netzwerkadministrator eine Menge Arbeit.
- **Daher ist ein durchdachtes Konzept zur automatischen Konfiguration in jedem größeren Unternehmen unverzichtbar.**



Parameter zur Autokonfiguration

- Mithilfe einer automatischen Konfiguration werden einem System verschiedene Parameter übergeben, damit es optimal in das aktuelle Netz integriert ist.
- Bei einem Wechsel des Netzes sind die Parameter dynamisch auszutauschen. (Das ist oft ein kritischer Punkt.)
- Für die grundlegende Konfiguration eines IPv6-Systems sind folgende Parameter notwendig:
 - **globale IPv6-Adresse**
(die linklokale IPv6-Adresse ist meist nicht ausreichend, da sie nicht geroutet wird.)
 - **Default-Router**
 - sowie (meistens auch) **DNS-Server-Adresse**
- Daneben können noch weitere Parameter übergeben werden.
(Beispiele: SIP-Server-Adresse für Internettelefonie, NTP-Server-Adresse (NTP=Network Time Protocol) oder VLAN-ID)



Verfahren zur Autokonfiguration

- Bei **IPv4** stehen zwei Verfahren zur Autokonfiguration zur Verfügung:
 - Das Bootstrap Protocol (**BOOTP**, RFC 951) für die Konfiguration von Terminals und festplattenlosen Workstations
 - Das **Dynamic Host Configuration Protocol (DHCPv4)**, RFC 1531), um Clients eine Netzwerkkonfiguration zuzuweisen.
- Bei **IPv6** unterscheidet man grundsätzlich zwei Methoden zur Autokonfiguration:
 - **Stateless Address Autoconfiguration (SLAAC)**

„Stateless“ bedeutet, dass bei der Autokonfiguration kein „Status“ gespeichert wird. Für die Autokonfiguration wird also kein spezieller Server benötigt, was dieses Verfahren in der Administration besonders einfach macht.
 - **Stateless/Stateful Address Autoconfiguration mit DHCPv6**

Für dieses Verfahren wird ein DHCPv6-Server benötigt, der die Konfigurationsinformationen für die Systeme im Netz speichert und verwaltet. Hierdurch können weitere Informationen verteilt werden und der Netzadministrator hat bessere Möglichkeiten, das Netz zu steuern und zu überwachen. (Stateful = IPv6-Adresse wird an Client verteilt.)
- **Die beiden Verfahren SLAAC und DHCPv6 können auch kombiniert werden.**

Zum Beispiel SLAAC für globale IPv6-Adresse und Default Route, die weiteren Parameter über DHCPv6.



Ablauf Stateless Address Autoconfiguration (SLAAC)

- **Schritt 1: Erzeugung einer link-lokalen IPv6-Adresse.**

Diese hatte Form: FE80::<Interfacelidentifizier>).

- **Schritt 2: Aufspüren von Routern im lokalen Netz.**

Hierzu werden **Router-Solicitation-Nachrichten** an die link-lokale Multicast Adresse „All Router“ FF02:2 gesendet.

- **Schritt 3: Router verteilen IPv6-Präfix**

Router senden **Router-Advertisement-Nachrichten** mit einem geeigneten Präfix an die link-lokale Multicast Adresse „All Nodes“ FF02:1.

Für jedes Präfix erzeugt der Client eine globale IPv6-Adresse als Kombination aus Präfix und Interface-Identifizier. Beispiel:

Interface Identifizier des Clients:

a667:06ff:fe7d:6082

Erhaltener Adresspräfix:

2001:0dc8:0100:f101::0/64

Globale IPv6-Adresse:

2001:0dc8:0100:f101:a667:06ff:fe7d:6082

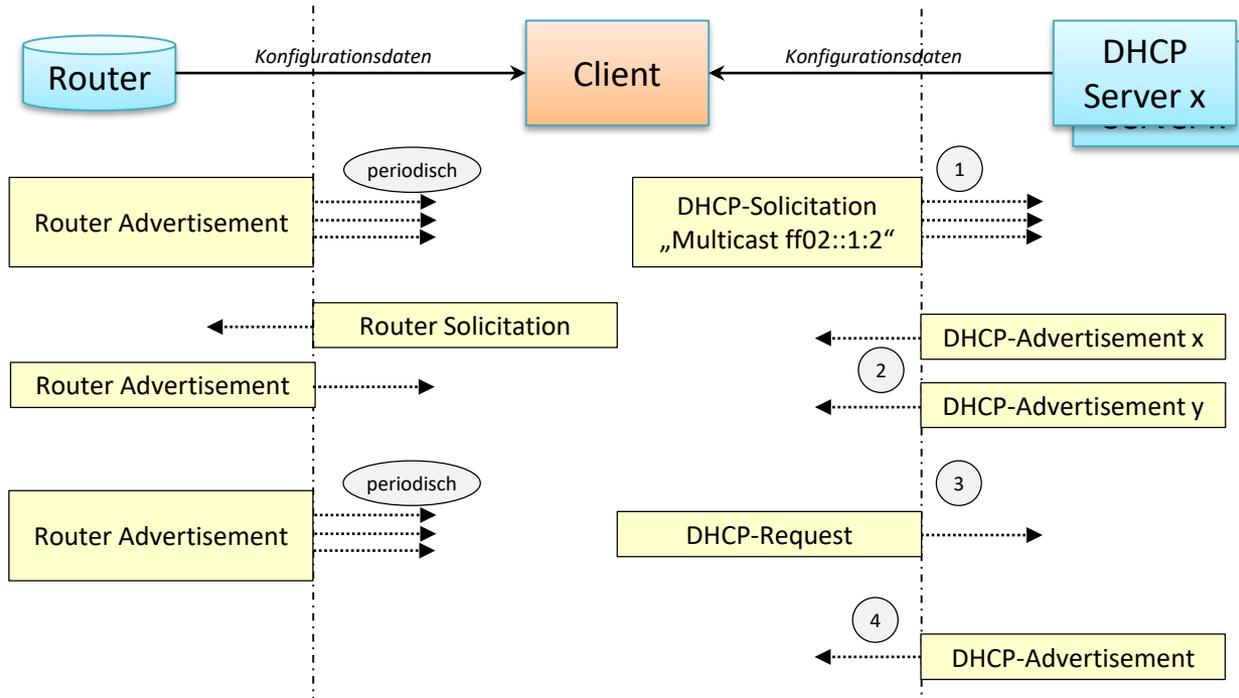


Anmerkungen zu Stateless Address Autoconfiguration (SLAAC)

- Über das **Router-Advertisement** erfährt der Client die Adresse eines Routers im Netz, den er als **Default-Gateway** benutzen kann, sowie ein **Präfix**.
Damit hatte er alles, um auf IP-Ebene kommunizieren zu können.
- Die **IPv6- Autokonfiguration per SLAAC ist in vielen Situationen nicht ausreichend**, denn es werden beispielsweise keine Adressen für DNS- oder NTP-Server erzeugt.
- Durch **Setzen des *Managed-Flags*** in Router-Advertisements kann der Client angewiesen werden, sich weitere Konfigurationsinformationen über DHCPv6 zu besorgen.
- In RFC6106 wird beschrieben, wie **per SLAAC auch DNS-Konfigurationsinformationen** verteilt werden können.
- Die mit SLAAC erzeugte globale IPv6-Adresse enthält die MAC-Adresse und ist damit weltweit über einen längeren Zeitraum hinweg eindeutig. Um die Privatsphäre zu schützen, wurde eine Erweiterung mit dem Namen „**Privacy Extension**“ eingeführt. Hierbei wird aus einer Zufallszahl ein sich periodisch ändernder Interface-Identifizierer generiert.



Andere Darstellung: Die IPv6-Autokonfiguration-Protokolle verwenden Multicast statt Broadcast





DHCPv6 - Zusatzinformationen

- **DHCPv6-Server** verwalten die Netzwerkkonfiguration und stellen diese den Clients auf Nachfrage zur Verfügung.
- **Client/Server-System:**
 - **DHCPv6-Client (UDP-Port 546):** Stellt Anfrage nach Parametern an Server und empfängt die Parameter von DHCPv6-Server
 - **DHCPv6-Server (UDP-Port 547):** kennt alle notwendigen Konfigurationsparameter und stellt sie den anfragenden Clients zur Verfügung
 - Die well-known Multicast-Adresse für DHCPv6 ist ff02::1:2
- DHCPv6 war in den letzten zehn Jahren ein Protokoll mit sehr hoher Änderungsrate innerhalb der IPv6-Familie. Es wurde ständig an weiteren Optionen gearbeitet.
- **DHCPv6 ist jetzt stabil.** Es wird beschrieben in RFC PS 8415 (Nov 2018). (PS = Proposed Standard)
- DHCPv6 kann als Ergänzung zur Stateless Address Autoconfiguration (SLAAC) eingesetzt werden.



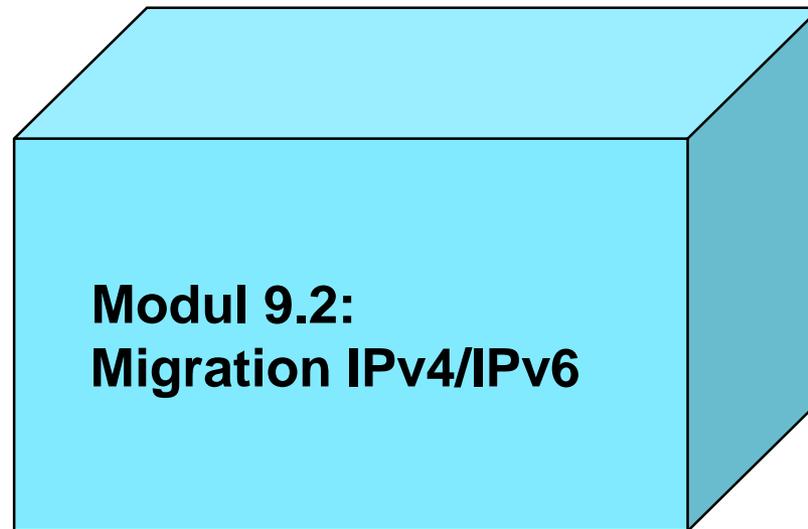
DHCPv6-Unterstützung in gängigen Betriebssystemen

| | Windows XP | Windows 7 | Windows 8 | Ubuntu 10.04 | Ubuntu 11.10 | Ubuntu 12.04 | Mac OS X 10.7/10.8 | iOS 5.1.1 | Android 4.03 |
|---|---------------------------|-----------------------------------|-----------------------------------|---------------------------|--------------|--------------|--------------------|-------------|---------------------------|
| IPv6-Support | Installation erforderlich | Ja | Ja | Ja | Ja | Ja | Ja | Ja | Ja |
| IP-Adresse / Default-Route über RA | Ja / Ja | Ja / Ja | Ja / Ja | Ja / Ja | Ja / Ja | Ja / Ja | Ja / Ja | Ja / Ja | geräte-abhängig / Nein |
| DNS-Adresse über RA (RDNSS-Option) | Nein | externer Source-Code erforderlich | externer Source-Code erforderlich | Ja | Ja | Ja | Ja | Ja | Nein |
| Privacy-Extension | per Default | per Default | per Default | manuell | manuell | per Default | per Default | per Default | manuell |
| DHCPv6-Client Support | Installation erforderlich | Ja | Ja | Installation erforderlich | Ja | Ja | Ja | Ja | Nein |
| DHCPv6-Client über O-/M-Flag steuerbar | Nein | Ja | Ja | Nein | Ja | Ja | Ja | Ja | - |

Stand 2012

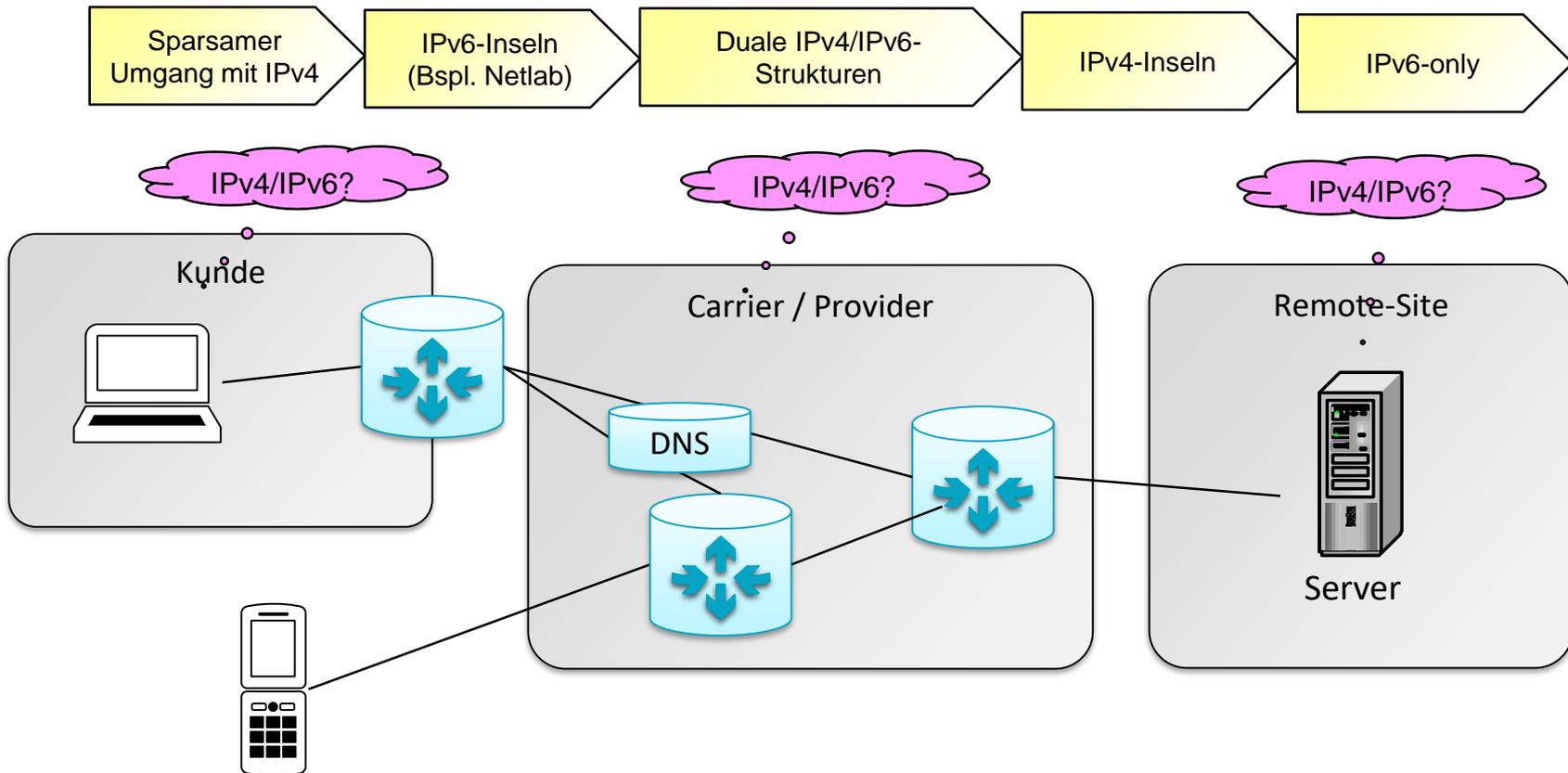
Nachlesen in:

Christian Schneider, Martina Kannen, Martin Leischner: *Bringdienst - IPv6-Autokonfiguration für Clients*, iX-Magazin für professionelle Informationstechnik, 11/2012 (Seite 72)



Problemstellung

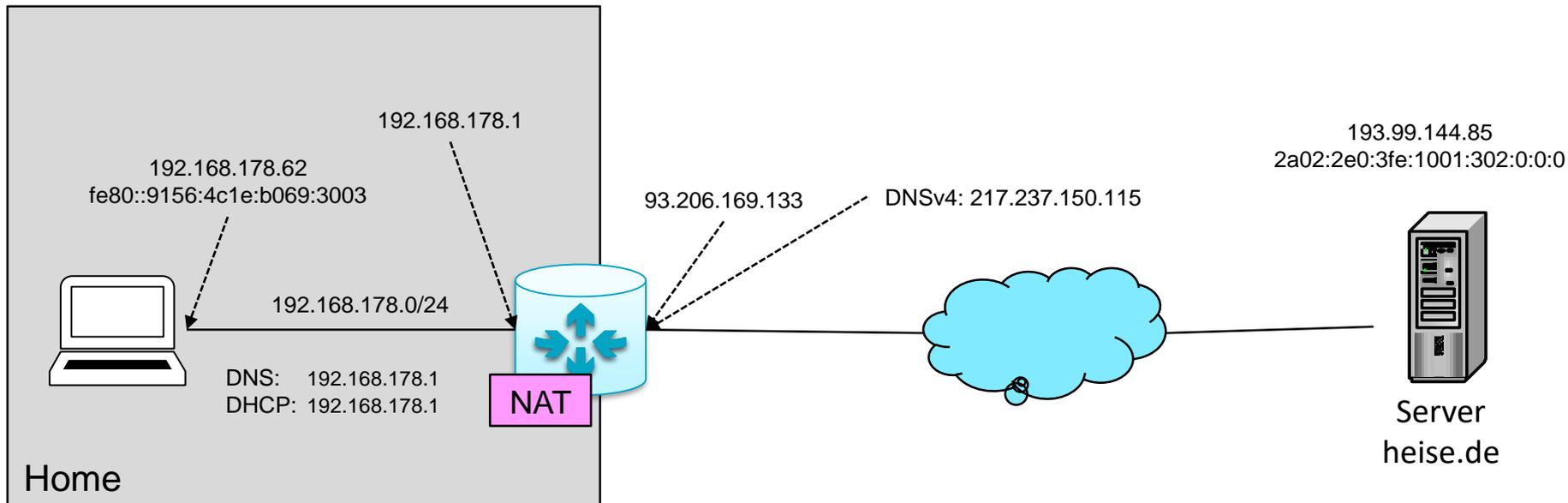
Hat sich herumgesprachen: die IPv4-Adressen gehen langsam zur Neige.





Home-Lösung : Sparen von IPv4-Adressen mit NAT

- Network address translation (NAT)
- Network address and port translation (NAPT) gemäß RFC 2663 (notwendig für mehrere Clients)
- Zusatznutzen Anonymität



Home-Lösung: Trotz IPv4-NAT in's IPv6-Netz

IPv6-Tunnel benötigt neben der Clientsoftware

- Tunnelserver (der die IPv6-Adressen, ... bereitstellt)
- Relay, das den Verkehr wieder entkapselt

Tunnelprovider: Teredo (Microsoft), SixXS,

