

Dynamic Host Configuration Protocol

Felix Ruzzoli

Albert-Ludwigs-Universität Freiburg

11. Dezember 2007

- 1 **Motivation**
 - Wozu Autokonfiguration?
 - Neuerungen bei DHCP
- 2 **Geschichte des Bootstrappings**
- 3 **DHCP**
- 4 **Autokonfiguration unter IPv6**
- 5 **Abschluss**

Die Situation

- Große Rechnernetze

Die Situation

- Große Rechnernetze
- Mobile Internet Clients

Die Situation

- Große Rechnernetze
- Mobile Internet Clients
- häufige Änderungen an Konfigurationen

Das Problem

- Keine vorhandene Autokonfigurationslösung aus einem 'Guss'
 - BOOTP → nur statische Adressvergabe

Das Problem

- Keine vorhandene Autokonfigurationslösung aus einem 'Guss'
 - BOOTP → nur statische Adressvergabe
- Manuelle Konfiguration von Netzwerkclients zu aufwändig
 - bzw. nicht möglich → WLAN Zugänge

Das Problem

- Keine vorhandene Autokonfigurationslösung aus einem 'Guss'
 - BOOTP → nur statische Adressvergabe
- Manuelle Konfiguration von Netzwerkclients zu aufwändig
 - bzw. nicht möglich → WLAN Zugänge
- Keine dynamische Adressvergabe möglich

Die Lösung

Dynamic Host Configuration Protocol DHCP

- 1 **Motivation**
 - Wozu Autokonfiguration?
 - Neuerungen bei DHCP
- 2 **Geschichte des Bootstrappings**
- 3 **DHCP**
- 4 **Autokonfiguration unter IPv6**
- 5 **Abschluss**

Die Ansätze - Das *Lease*-Konzept

- Server verteilen sogenannte *Leases* an Clients
 - im Prinzip „verleihen“ der IP-Adresse

Die Ansätze - Das *Lease*-Konzept

- Server verteilen sogenannte *Leases* an Clients
 - im Prinzip „verleihen“ der IP-Adresse
- *Leasetime* von 0 bis ∞ sec.

Die Ansätze

- Alle Informationen für die Verbindung zum Internet müssen zur Verfügung stehen.

Die Ansätze

- Alle Informationen für die Verbindung zum Internet müssen zur Verfügung stehen.
 - IP-Adresse

Die Ansätze

- Alle Informationen für die Verbindung zum Internet müssen zur Verfügung stehen.
 - IP-Adresse
 - Subnetzmaske

Die Ansätze

- Alle Informationen für die Verbindung zum Internet müssen zur Verfügung stehen.
 - IP-Adresse
 - Subnetzmaske
 - default Gateway

- 1 Motivation
- 2 Geschichte des Bootstrappings
 - RARP
 - BOOTP
 - Andere Technologien
- 3 DHCP
- 4 Autokonfiguration unter IPv6
- 5 Abschluss

Reverse Address Resolution Protocol

- 1984 an der Stanford University entworfen

Reverse Address Resolution Protocol

- 1984 an der Stanford University entworfen
- Gegenstück zu ARP

Reverse Address Resolution Protocol

- 1984 an der Stanford University entworfen
- Gegenstück zu ARP
- ermitteln der Protokolladresse (IP) mittels der Hardwareadresse (MAC)

- 1 Motivation
- 2 Geschichte des Bootstrappings
 - RARP
 - BOOTP
 - Andere Technologien
- 3 DHCP
- 4 Autokonfiguration unter IPv6
- 5 Abschluss

BOOTstrap Protocol

- für Thin Clients gedacht

BOOTstrap Protocol

- für Thin Clients gedacht
- Zusammenspiel mit TFTP¹
 - ermöglicht booten vom Server

¹Trivial File Transfer Protocol

BOOTstrap Protocol

- für Thin Clients gedacht
- Zusammenspiel mit TFTP¹
 - ermöglicht booten vom Server

Aber: Leider immer noch statisch.

¹Trivial File Transfer Protocol

- 1 Motivation
- 2 Geschichte des Bootstrappings
 - RARP
 - BOOTP
 - Andere Technologien
- 3 DHCP
- 4 Autokonfiguration unter IPv6
- 5 Abschluss

Die Alternativen

- RARP und DRARP

Die Alternativen

- RARP und DRARP
- TFTP

Die Alternativen

- RARP und DRARP
- TFTP
- ICMP

Die Alternativen

- RARP und DRARP
- TFTP
- ICMP
 - *ICMP mask request*

Die Alternativen

- RARP und DRARP
- TFTP
- ICMP
 - *ICMP mask request*
 - *ICMP router discovery*

Die Alternativen

- RARP und DRARP
- TFTP
- ICMP
 - *ICMP mask request*
 - *ICMP router discovery*
- BOOTP

Die Alternativen

- RARP und DRARP
- TFTP
- ICMP
 - *ICMP mask request*
 - *ICMP router discovery*
- BOOTP
- NIP - Network Information Protocol
 - Am MIT für dynamische verteilte IP-Adressvergabe verwendet

Die Alternativen

- RARP und DRARP
- TFTP
- ICMP
 - *ICMP mask request*
 - *ICMP router discovery*
- BOOTP
- NIP - Network Information Protocol
 - Am MIT für dynamische verteilte IP-Adressvergabe verwendet
- RLP - Resource Location Protocol

Die Alternativen

- RARP und DRARP
- TFTP
- ICMP
 - *ICMP mask request*
 - *ICMP router discovery*
- BOOTP
- NIP - Network Information Protocol
 - Am MIT für dynamische verteilte IP-Adressvergabe verwendet
- RLP - Resource Location Protocol
- Sun Microsystems Thin Clients
 - nutzen Kombination aus RARP, TFTP und RPC, genannt *bootparams*

1 Motivation

2 Geschichte des Bootstrappings

3 DHCP

- IP-Adresse per UDP
- UDP zur Kommunikation
- Format von DHCP Nachrichten
- Die DHCP Nachrichtentypen
- Zustände beim Ermitteln einer IP-Adresse
- Simulation eines Bootstrap Vorganges
- DHCP in der Praxis

4 Autokonfiguration unter IPv6

5 Abschluss

Der Trick..

- *Internet Protocol* setzt IP-Adresse zur Kommunikation voraus
 - diese ist aber nicht bekannt

Der Trick mit dem..

- *Internet Protocol* setzt IP-Adresse zur Kommunikation voraus
 - diese ist aber nicht bekannt

Lösung:

Der Trick mit dem Broadcasting..

- *Internet Protocol* setzt IP-Adresse zur Kommunikation voraus
 - diese ist aber nicht bekannt

Lösung: **Broadcasting**

Der Trick mit dem Broadcasting..

- *Internet Protocol* setzt IP-Adresse zur Kommunikation voraus
 - diese ist aber nicht bekannt

Lösung: **Broadcasting**

- IP-Standard definiert 255.255.255.255 als *limited broadcast* Adresse
 - IP-Software akzeptiert Broadcasts bevor sie selbst komplett Konfiguriert wurde

1 Motivation

2 Geschichte des Bootstrappings

3 DHCP

- IP-Adresse per UDP
- UDP zur Kommunikation
- Format von DHCP Nachrichten
- Die DHCP Nachrichtentypen
- Zustände beim Ermitteln einer IP-Adresse
- Simulation eines Bootstrap Vorganges
- DHCP in der Praxis

4 Autokonfiguration unter IPv6

5 Abschluss

UDP ist unzuverlässig

Gegenmaßnahmen:

- Verwendung von UDP Checksummen

UDP ist unzuverlässig

Gegenmaßnahmen:

- Verwendung von UDP Checksummen
- Setzen des *do not fragment* Bit

UDP ist unzuverlässig

Gegenmaßnahmen:

- Verwendung von UDP Checksummen
- Setzen des *do not fragment* Bit
- *timeout and retransmission*

1 Motivation

2 Geschichte des Bootstrappings

3 DHCP

- IP-Adresse per UDP
- UDP zur Kommunikation
- Format von DHCP Nachrichten
- Die DHCP Nachrichtentypen
- Zustände beim Ermitteln einer IP-Adresse
- Simulation eines Bootstrap Vorganges
- DHCP in der Praxis

4 Autokonfiguration unter IPv6

5 Abschluss

Die Aufteilung der Felder..

op (8)	htype (8)	hlen (8)	hops (8)
xid (32)			
secs (16)		flags (16)	
ciaddr (32)			
yiaddr (32)			
siaddr (32)			
giaddr (32)			
chaddr (128)			
sname (512)			
file (1024)			
options (var.)			

..und ihre Bedeutung

- **op** - Ist der *Operation Code*, *request*(1) oder *reply*(2)

..und ihre Bedeutung

- **op** - Ist der *Operation Code*, *request*(1) oder *reply*(2)
- **htype, hlen** - Hardwaretyp des Netzwerks, Länge der Hardwareadressen (wie bei ARP)

..und ihre Bedeutung

- **op** - Ist der *Operation Code*, *request*(1) oder *reply*(2)
- **htype, hlen** - Hardwaretyp des Netzwerks, Länge der Hardwareadressen (wie bei ARP)
- **hops** - für Bootstrapping über Router hinweg

..und ihre Bedeutung

- **op** - Ist der *Operation Code*, *request*(1) oder *reply*(2)
- **htype, hlen** - Hardwaretyp des Netzwerks, Länge der Hardwareadressen (wie bei ARP)
- **hops** - für Bootstrapping über Router hinweg
- **xid** - Transaction ID

..und ihre Bedeutung

- **op** - Ist der *Operation Code*, *request*(1) oder *reply*(2)
- **htype, hlen** - Hardwaretyp des Netzwerks, Länge der Hardwareadressen (wie bei ARP)
- **hops** - für Bootstrapping über Router hinweg
- **xid** - Transaction ID
- **secs** - Sekunden seit Beginn der Adressakquisition des Clients

..und ihre Bedeutung

- **op** - Ist der *Operation Code*, *request*(1) oder *reply*(2)
- **htype, hlen** - Hardwaretyp des Netzwerks, Länge der Hardwareadressen (wie bei ARP)
- **hops** - für Bootstrapping über Router hinweg
- **xid** - Transaction ID
- **secs** - Sekunden seit Beginn der Adressakquisition des Clients
- **flags** - Siehe weiter unten

Das *flags* Feld

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|B|                MBZ                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

B: BROADCAST flag

MBZ: MUST BE ZERO (reserved for future use)

Die wichtigsten Felder

- **ciaddr** - Client IP Address (falls IP-Adresse schon bekannt)

Die wichtigsten Felder

- **ciaddr** - Client IP Address (falls IP-Adresse schon bekannt)
- **yiaddr** - Your IP Address (Die vom Server vergebene IP-Adresse in der Antwort)

Die wichtigsten Felder

- **ciaddr** - Client IP Address (falls IP-Adresse schon bekannt)
- **yiaddr** - Your IP Address (Die vom Server vergebene IP-Adresse in der Antwort)
- **siaddr** - Server IP Address (Die IP-Adresse des nächsten zu verwendenden DHCP Servers)

Die wichtigsten Felder

- **ciaddr** - Client IP Address (falls IP-Adresse schon bekannt)
- **yiaddr** - Your IP Address (Die vom Server vergebene IP-Adresse in der Antwort)
- **siaddr** - Server IP Address (Die IP-Adresse des nächsten zu verwendenden DHCP Servers)
- **giaddr** - Router IP Address (findet Verwendung beim Bootstrapping über Router hinweg)

Die wichtigsten Felder

- **ciaddr** - Client IP Address (falls IP-Adresse schon bekannt)
- **yiaddr** - Your IP Address (Die vom Server vergebene IP-Adresse in der Antwort)
- **siaddr** - Server IP Address (Die IP-Adresse des nächsten zu verwendenden DHCP Servers)
- **giaddr** - Router IP Address (findet Verwendung beim Bootstrapping über Router hinweg)
- **chaddr** - Client Hardware Address (MAC-Adresse des Clients)

Die wichtigsten Felder

- **ciaddr** - Client IP Address (falls IP-Adresse schon bekannt)
- **yiaddr** - Your IP Address (Die vom Server vergebene IP-Adresse in der Antwort)
- **siaddr** - Server IP Address (Die IP-Adresse des nächsten zu verwendenden DHCP Servers)
- **giaddr** - Router IP Address (findet Verwendung beim Bootstrapping über Router hinweg)
- **chaddr** - Client Hardware Address (MAC-Adresse des Clients)
- **sname** - Server Hostname

Die wichtigsten Felder

- **ciaddr** - Client IP Address (falls IP-Adresse schon bekannt)
- **yiaddr** - Your IP Address (Die vom Server vergebene IP-Adresse in der Antwort)
- **siaddr** - Server IP Address (Die IP-Adresse des nächsten zu verwendenden DHCP Servers)
- **giaddr** - Router IP Address (findet Verwendung beim Bootstrapping über Router hinweg)
- **chaddr** - Client Hardware Address (MAC-Adresse des Clients)
- **sname** - Server Hostname
- **file** - Name einer Bootdatei

1 Motivation

2 Geschichte des Bootstrappings

3 DHCP

- IP-Adresse per UDP
- UDP zur Kommunikation
- Format von DHCP Nachrichten
- Die DHCP Nachrichtentypen
- Zustände beim Ermitteln einer IP-Adresse
- Simulation eines Bootstrap Vorganges
- DHCP in der Praxis

4 Autokonfiguration unter IPv6

5 Abschluss

8 verschiedene Nachrichten

- **DHCPDISCOVER (1)**
Client Broadcast
- **DHCPOFFER (2)**
Server an Client, Antwort auf DHCPDISCOVER, Konfigurationsparameter werden angeboten
- **DHCPREQUEST (3)**
Client an Server, Anfrage der angebotenen Parameter, Bestätigung der Korrektheit einer vorher zugewiesenen Adresse oder verlängern der *Leasetime*
- **DHCPDECLINE (4)**
Client teilt dem Server mit, dass die angebotene Adresse bereits verwendet wird

8 verschiedene Nachrichten

- **DHCPACK (5)**
Server an Client, die Konfigurationsparameter werden zugeteilt
- **DHCPNACK (6)**
Server an Client, die verwendete Netzwerkadresse ist nicht länger gültig
- **DHCPRELEASE (7)**
Client an Server, Abbruch des *Lease*
- **DHCPINFORM (8)**
Client fragt Server nach lokalen Konfigurationsparametern

1 Motivation

2 Geschichte des Bootstrappings

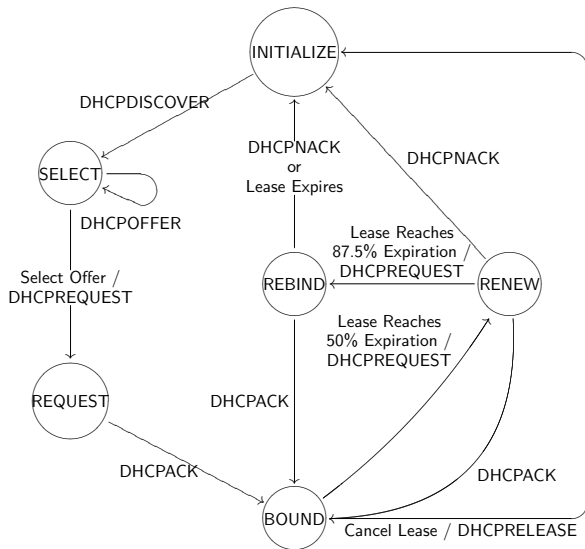
3 DHCP

- IP-Adresse per UDP
- UDP zur Kommunikation
- Format von DHCP Nachrichten
- Die DHCP Nachrichtentypen
- Zustände beim Ermitteln einer IP-Adresse
- Simulation eines Bootstrap Vorganges
- DHCP in der Praxis

4 Autokonfiguration unter IPv6

5 Abschluss

Endlicher Automat



1 Motivation

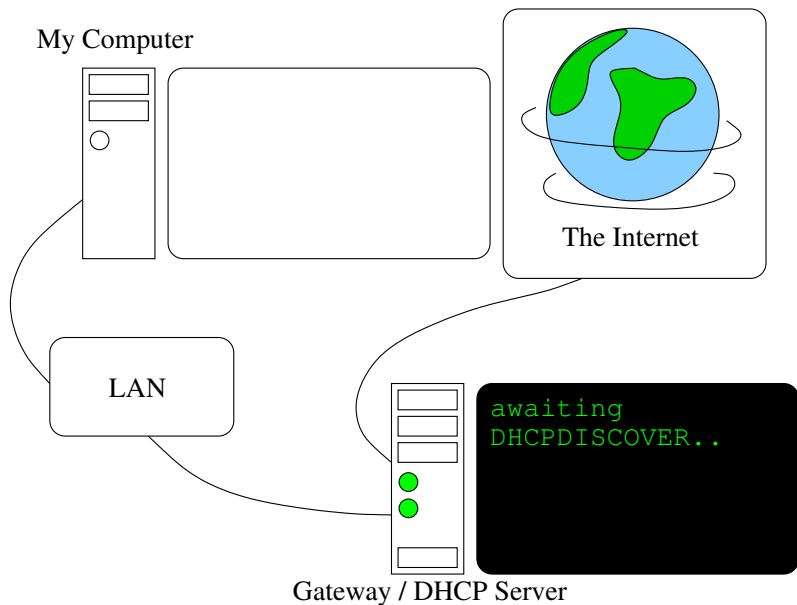
2 Geschichte des Bootstrappings

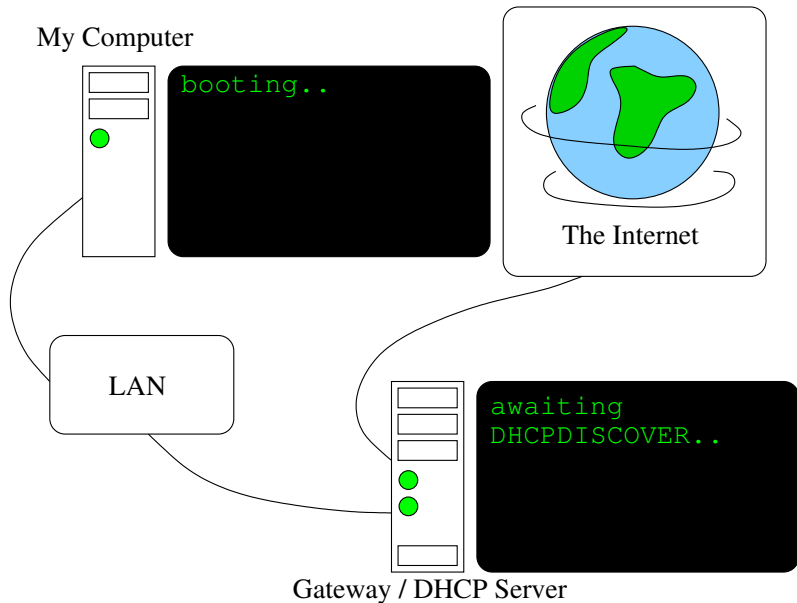
3 DHCP

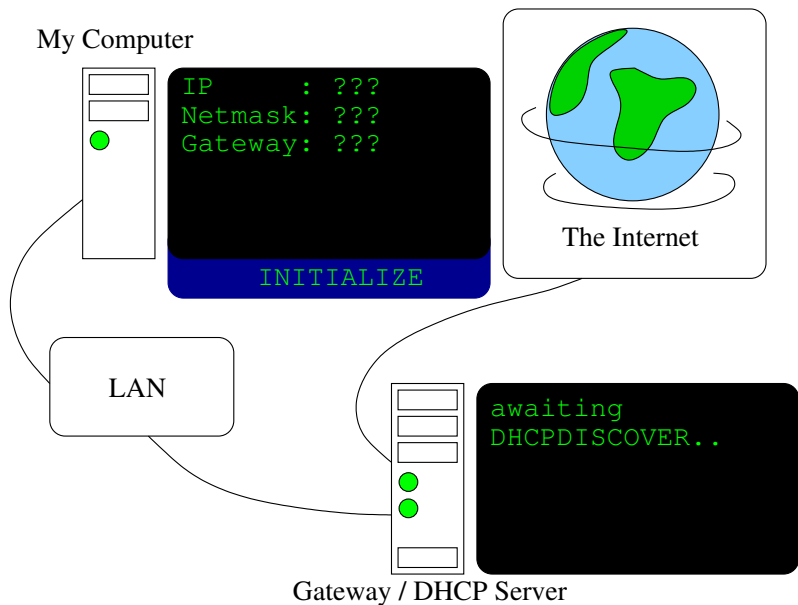
- IP-Adresse per UDP
- UDP zur Kommunikation
- Format von DHCP Nachrichten
- Die DHCP Nachrichtentypen
- Zustände beim Ermitteln einer IP-Adresse
- Simulation eines Bootstrap Vorganges
- DHCP in der Praxis

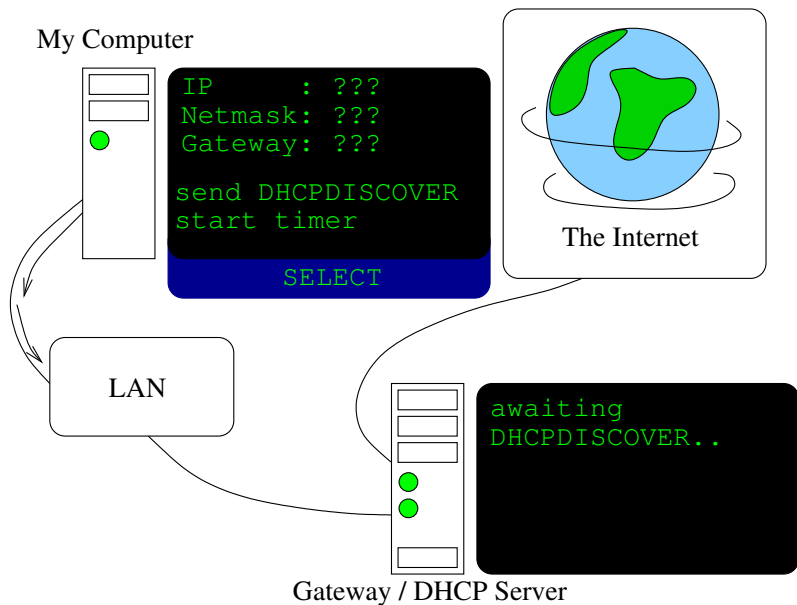
4 Autokonfiguration unter IPv6

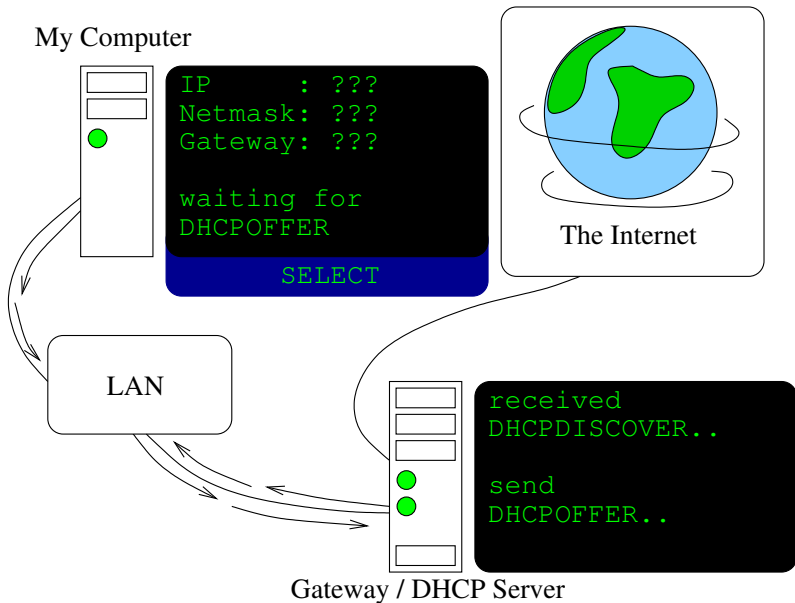
5 Abschluss

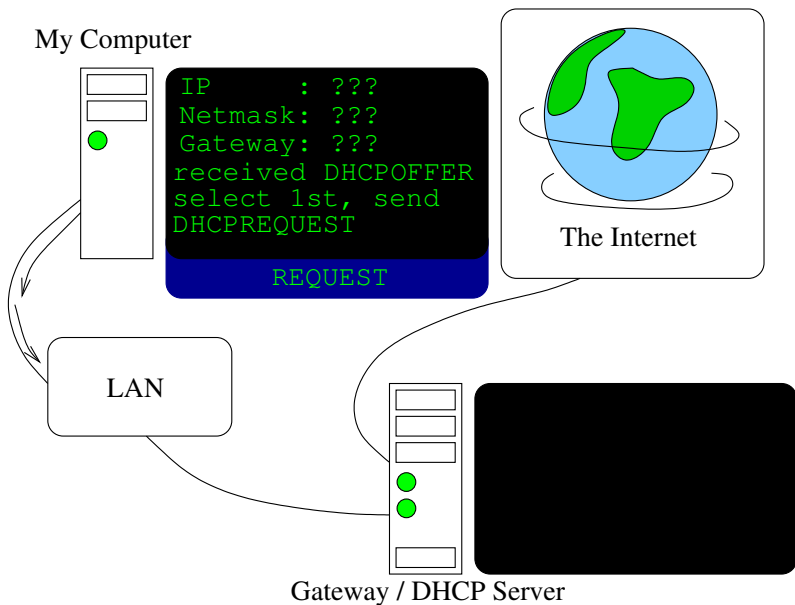


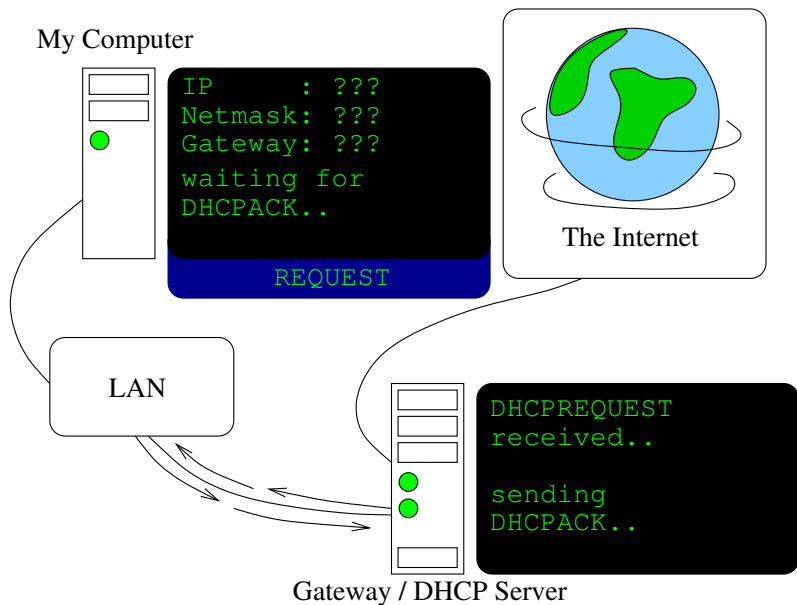


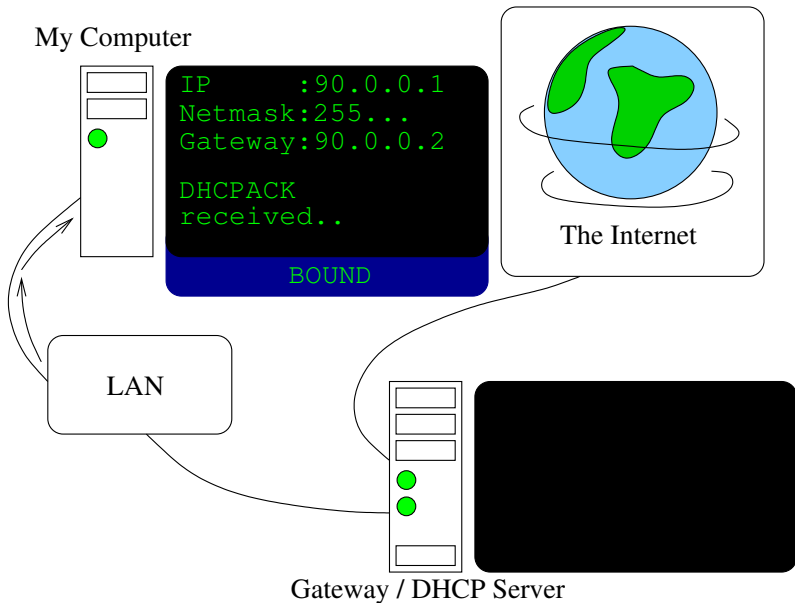


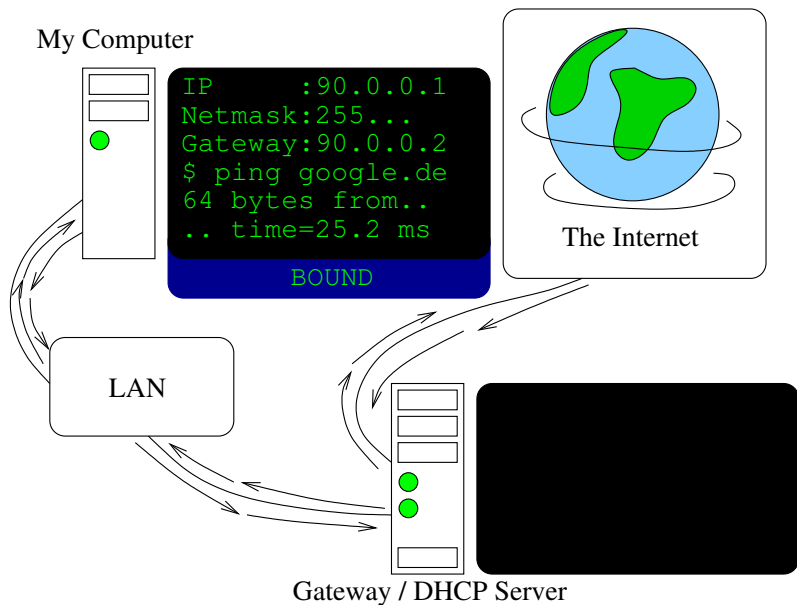












1 Motivation

2 Geschichte des Bootstrappings

3 DHCP

- IP-Adresse per UDP
- UDP zur Kommunikation
- Format von DHCP Nachrichten
- Die DHCP Nachrichtentypen
- Zustände beim Ermitteln einer IP-Adresse
- Simulation eines Bootstrap Vorganges
- DHCP in der Praxis

4 Autokonfiguration unter IPv6

5 Abschluss

Serverseitige Implementierungen

Wo findet man DHCP Server?

- bei vielen Routern On-Board

Serverseitige Implementierungen

Wo findet man DHCP Server?

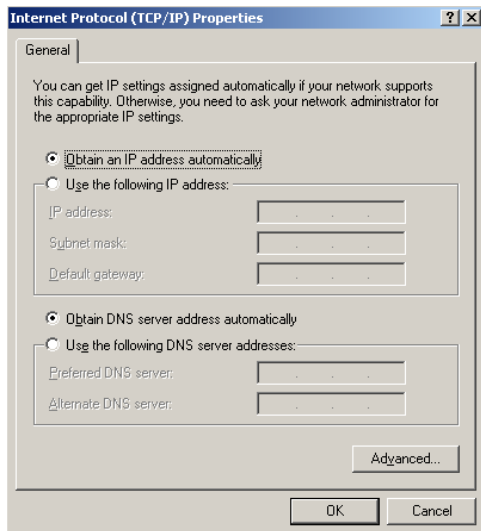
- bei vielen Routern On-Board
- MS Windows Server Betriebssysteme

Serverseitige Implementierungen

Wo findet man DHCP Server?

- bei vielen Routern On-Board
- MS Windows Server Betriebssysteme
- Linux Server, z.B. *dnsmasq* oder *dhcpcd*

Clientseitiges DHCP unter Windows..



..und unter Linux

```
memmaker@mmx23/~ >$ sudo dhclient eth0
Internet Systems Consortium DHCP Client V3.0.5
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/00:0b:6a:83:71:3c
Sending on   LPF/eth0/00:0b:6a:83:71:3c
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.1.77
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.77
bound to 192.168.1.23 -- renewal in 3117 seconds.
memmaker@mmx23/~ >$
```


- 1 Motivation
- 2 Geschichte des Bootstrappings
- 3 DHCP
- 4 **Autokonfiguration unter IPv6**
 - DHCPv6
 - Stateless Address Autoconfiguration
- 5 Abschluss

Dynamic Host Configuration Protocol für IPv6

- Nutzt UDP Pakete

Dynamic Host Configuration Protocol für IPv6

- Nutzt UDP Pakete
- Mehr Konfigurationsmöglichkeiten

Dynamic Host Configuration Protocol für IPv6

- Nutzt UDP Pakete
- Mehr Konfigurationsmöglichkeiten
- *Aber:* Wie DHCP angepasst für IPv6

Dynamic Host Configuration Protocol für IPv6

- Nutzt UDP Pakete
- Mehr Konfigurationsmöglichkeiten
- *Aber:* Wie DHCP angepasst für IPv6

Vorteil: Viel Kontrolle

Dynamic Host Configuration Protocol für IPv6

- Nutzt UDP Pakete
- Mehr Konfigurationsmöglichkeiten
- *Aber:* Wie DHCP angepasst für IPv6

Vorteil: Viel Kontrolle

Nachteil: Server notwendig

- 1 Motivation
- 2 Geschichte des Bootstrappings
- 3 DHCP
- 4 **Autokonfiguration unter IPv6**
 - DHCPv6
 - Stateless Address Autoconfiguration
- 5 Abschluss

Stateless Address Autoconfiguration

- In IPv6 „eingebaut“

Stateless Address Autoconfiguration

- In IPv6 „eingebaut“
- Hosts ermitteln selbständig lokale IP-Adressen
 - MAC Adresse + *prefix* = eindeutige IP-Adresse

Stateless Address Autoconfiguration

- In IPv6 „eingebaut“
- Hosts ermitteln selbständig lokale IP-Adressen
 - MAC Adresse + *prefix* = eindeutige IP-Adresse
 - Ohne Router, nur lokal gültige IP-Adresse

Stateless Address Autoconfiguration

- In IPv6 „eingebaut“
- Hosts ermitteln selbständig lokale IP-Adressen
 - MAC Adresse + *prefix* = eindeutige IP-Adresse
 - Ohne Router, nur lokal gültige IP-Adresse

Vorteil: Kein Server notwendig

Stateless Address Autoconfiguration

- In IPv6 „eingebaut“
- Hosts ermitteln selbständig lokale IP-Adressen
 - MAC Adresse + *prefix* = eindeutige IP-Adresse
 - Ohne Router, nur lokal gültige IP-Adresse

Vorteil: Kein Server notwendig

Nachteil: Wenig Kontrolle

ENDE

Vielen Dank für die Aufmerksamkeit..

Primärquellen

- Douglas E. Comer, *Internetworking with TCP/IP*
- RFCs
 - 903 - RARP
 - 951 - BOOTP
 - 1122 und 1123 - Requirements for Internet Hosts
 - 1531, 2131 - DHCP
 - 1971 und 4862 - IPv6 Stateless Address Autoconfiguration
 - 3315 - DHCPv6