Geschichte

Jeder Administrator, der mehr als drei Rechner in einem Netzwerk betreibt, ist froh, wenn er nicht jedes Gerät einzeln für das Netzwerk konfigurieren muss. Das Hinzufügen eines Rechners ist zwar nicht allzu mühsam, aber spätestens bei der ersten Änderung eines zentralen Parameters wie etwa des Defaultrouters spart man sich viele Wege im inzwischen wahrscheinlich gewachsenen Netzwerk, wenn die Konfiguration der Netzwerkeinstellungen zentral erfolgt. Bei Netzen von einigen Hundert oder gar Tausend Geräten ist an eine manuelle Konfiguration ohnehin nicht mehr zu denken.
Mit dem Bootstrap Protacol (BOOTP) wurde in den 80-er Jahren ein erstes Protokoll geschaffen, das die automatische Konfiguration der Adresse in einem LAN sicherstellte. Diesem folgte in den 90-er Jahren das weitaus flexiblere Dynamic Host Configuration Protocol (DHCP), welches es erlaubt,neben der Adresskonfiguration eine ganze Liste von Konfigurationsdaten über das Netzwerk an die Clients zu senden. DHCP ist ein Protokoll, um IP-Adressen in einem TCP/IP-Netzwerk zu verwalten und an die anfragenden Hosts zu verteilen. Mit DHCP ist jeder Netzwerk-Teilnehmer in der Lage sich selber automatisch zu konfigurieren.
DHCP ist ein Protokoll „im Hintergrund“. DHCP-Server, die den Dienst bereitstellen, findet man heute nicht nur auf Servern: Die meisten DSL-Router, die Internetprovider ihren Kunden für den Netzzugang anbieten, stellen IP-Adressen für die LAN-Seite des Netzwerks über DHCP bereit.

In der Frühzeit des Internet war es noch möglich, die wenigen Geräte IP-basierter LANs manuell zu konfigurieren. Doch schon in den 80-er Jahren wurde das Booten von Workstations über das Netzwerk populär, entweder um ganz ohne Harddisk zu laufen oder die Installation über das Netzwerk zu bekommen. Um nun nicht bei jedem Start die richtige IP-Adresse eingeben zu müssen, war ein entsprechendes Protokoll notwendig. Der Ansatz hiess Reverse ARP (RARP). Im Gegensatz zu ARP - dem Address Resolution Protocol zur Zuordnung von MAC-Adresse zu IP-Adresse wird hier die Frage ins Netz gestellt: „Welche IP hat die MAC-Adresse xy?“
Damit war es jedoch noch nicht getan, denn eine IP-Adresse allein liefert noch keine Angaben zum Routing, zur DNS-Auflösung und - für den Netzwerkboot ganz wichtig - zur Bezugsquelle für die Dateien. Also wurde der RFC-basierte Dienst Bootparam entwickelt. Um diesen Prozess und auch den Verwaltungsaufwand zu verringern (bei individueller Konfiguration müssen alle Hosts in /etc/bootparams gepflegt werden), entwickelten Bill Croft von der Stanford University und Iohn Gilmore von Sunl 1985 BOOTP. Damit konnte ein Gerät im LAN durch den Austausch von Nachrichten (Bootp Request/Bootp Reply) alle notwendigen Parameter abfragen.
Mit DHCP folgte der vorerst letzte Schritt in der Evolution des Protokolls, das die mangelnde Flexibilität von BOOTP bei stetig gewachsenen Anforderungen an ein solches Protokoll ausglich. DHCP ist abwärtskompatibel zu BOOTP, so dass auch Geräte, die nur das ältere Protokoll sprechen, von einem DHCP-Server bedient werden können.

Grundlagen

DHCPINFORM

Laut RFC dient diese Nachricht einem Client mit einer extern (z. B. Fest) konfigurierten Adresse dazu, weitere Konfigurationsparameter im Netz (etwa vom DNS-Server oder dem Defaultrouter) zu erfragen. Der Client schickt eine DHCPINFORM-Nachricht an den Server (oder per Broadcast, wenn der Server nicht bekannt ist) und trägt seine eigene IP-Adresse als Client-Adresse in das Paket ein. Der Server antwortet mit einem DHCPACK, das die angefragten Parameter enthält. Der Server kann die Anfrage auf Konsistenz in Bezug auf die von ihm verteilte Netzwerkkonfiguration prüfen, darf aber keine Lease Time zuweisen.
Im Web Proxy Autodiscovery Protocol (WPAD) werden im ersten Schritt DHCPINFORMs verwendet. Dabei wird an den DHCP-Server ein Inform mit der Option 252 (diese ist per Konvention, nicht per RFC vergeben) auf der Liste der angefragten Optionen gesendet. Als Antwort sollte der Client einen String mit einer URL für ein Proxy-Autokonfigurationsskript erhalten.

Relaying

Die Broadcasts der Clients erstrecken sich immer nur über ein LAN-Segment. Wenn man nicht in jedes Subnetz einen eigenen DHCP-Server stellen möchte, bietet es sich an, einen DHCP Relay Agent auf einem dazwischen liegenden Router einzurichten. Somit kann ein DHCP-Server Clients in verschiedenen Netzen bedienen und für verschiedene Netze auch verschiedene Konfigurationen zuweisen.
Damit der Server aber nun weiss, aus welchem Netz die Anfrage kommt, muss dies bereits im Discover, das ja eigentlich noch gar keine IP-Adressen enthält, kenntlich gemacht werden. Dazu füllt der Relay Agent das giaddr-Feld im DHCP-Paket mit der IP-Adresse des Interface aus, auf dem die Anfrage empfangen wurde. So kann der DHCP-Server das Subnetz zuordnen und die entsprechenden Parameter zuweisen.

DHCP-Refresh
In der DHCP-ACK-Nachricht ist die Lease-Time (Leihdauer) angegeben, die aussagt, wie lange der Client die zugewiesene IP-Konfiguration verwenden darf. Nach der Hälfte der Lease-Time muss der standardkonforme Client einen erneuten DHCP-REQUEST sendet. In der Regel wird der DHCP-Server ein DHCP-ACK mit identischen Daten und einer aktualisierten Lease-Time schicken. Damit gilt die Nutzung der IP-Adresse als verlängert.
Aber was ist, wenn der DHCP-Server nicht antwortet und somit die aktuelle IP-Konfiguration nicht bestätigt/verlängert wird. Beispielsweise, weil der DHCP-Server ausgefallen ist oder vom Netz genommen wurde. In diesem Fall wird der Client die IP-Konfiguration ohne Einschränkungen weiter verwenden, bis die Lease-Time endgültig abgelaufen ist. Allerdings wird er vor dem Ablauf noch mal versuchen, eine Verlängerung der IP-Konfiguration von diesem DHCP-Server zu erhalten.
Wenn dieser DHCP-Server nicht mehr erreichbar ist, weil vielleicht inzwischen ein anderer DHCP-Server zuständig ist, dann wird der Client noch vor dem endgültigen Ablauf der Lease-Time mit einem erneuten DHCP-DISCOVER versuchen, eine Adresszuweisung von einem anderen DHCP-Server zu erhalten.

DHCP-Not Acknowledged
Sollte der DHCP-Server keine Adressen mehr zur Verfügung haben oder während des Vorgangs ein anderer Client diese Adresse zugesagt bekommen haben, sendet der DHCP-Server ein DHCPNAK (DHCP-Not Acknowledged).

Optionen
Um etwas mehr Struktur in den potentiellen Wildwuchs an Herstellererweiterungen zu bringen, wurde für DHCP die sogenannte Vendor Class eingeführt. Dabei handelt es sich um eine Zwischenstufe, unterhalb derer Optionen umdefiniert werden können. Der Vendor Space selbst ist ein String, z. B. SUNW, der von Sun Workstations bei einem Netzwerkboot per DHCP verwendet wird.
Wenn ein Client nach Optionen verlangt, die auf dem Server nicht konfiguriert sind, muss der Server die Frage nach diesen Optionen ignorieren und dem Client eine Antwort mit allen bekannten Werten senden. Dabei darf der Server eine Meldung ausgeben, dass eine Frage nach etwas Unbekanntem gestellt wurde. Der Client soll dennoch versuchen fortzufahren, selbst wenn nicht alle Fragen beantwortet wurden - auf die Gefahr hin, dass dann nicht unbedingt alle Funktionalitäten zur Verfügung stehen. Auch hier kann der Client eine Meldung ausgeben, dass eine angefragte Option nicht beantwortet wurde.

Gebräuchliche Optionen
Im Folgenden werden von den vielen laut RFC 2132 verabschiedeten Optionen diejenigen vorgestellt, die tatsächlich in der Praxis Anwendung finden.

Router	gibt den oder die Defaultrouter in einem Netz an.
Domain Name Server	der Name der DNS-Domain, in der sich der Client befindet. Dieser wird auch zur Auflösung von unqualifizierten Hostnamen benutzt.
Log Server	ein Syslog-Server, an den der Client seine Logfiles senden soll
LPR Server	ein Druckserver, der über das LPD-Protokoll angesprochen werden kann.
Domain Name	der Name der DNS-Domain, in der sich der Client befindet. Dieser wird auch zur Auflösung von unqualifizierten Hostnamen benutzt
Perform Router Discover	legt fest, ob der Client nach der Methode aus RFC 1256 selbst nach Routern im Netz suchen soll (Ja/Nein)
Host Name	der eigene Hostname als nullterminierter String
NTP Server	eine Liste möglicher NTP-Server, nach ihrer Wichtigkeit geordnet.
Subnet Mask	die Subnetz-Maske, die in dem Netz gelten soll.

ISC-DHCP-Server

Der am weitesten verbreitete DHCP-Server ist der DHCPD des Internet Software Consortium (ISC). Er läuft auf fast allen Betriebssystemen. Das ISC zeichnet sich auch für den Nameserver BIND verantwortlich. Windows-Server-Betriebssysteme bringen einen DHCP-Dienst mit. Netzwerkdevices wie Router oder Hardware-Firewalls besitzen häufig ebenfalls einen DHCP-Server-Dienst.

Dateien
Folgende Dateien und Verzeichnisse sind für einen funktionierenden DHCP-Daemon notwendig:
dhcpd.conf → Konfigurationsdatei, in der alle Parameter vermerkt sind
/var/lib/dhcp → oder ein ähnliches Verzeichnis, in das der DHCP-Daemon schreiben kann. Wenn dieser mit einer eigenen Benutzerkennung läuft, muss das Verzeichnis für diese Kennung schreibbar sein.
dcpd.leases → (im schreibbaren Verzeichnis) enthält die zugeteilten Adressen mit einem Vermerk, wann die Zuteilungen beginnen und enden, in welchem Zustand sie sich gerade befinden, welche MAC-Adresse sie zuletzt zugeteilt bekommen haben und gegebenenfalls die Hostnamen der Clients. In einer hochverfiigbaren Konfiguration werden auch die Zustände der HA-Partner hier vermerkt.
Vor dem ersten Start des Daemon muss die Datei dhcpd.leases existieren und schreibbar sein, andernfalls gibt der Daemon eine Fehlermeldung aus. Mittels touch dhcp.leases im richtigen Verzeichnis kann sie angelegt werden.

Aufbau von dhcd.conf
Der wesentliche Inhalt der Datei sind Parameter für die Clients im Netz. Dabei können Optionen hierarchisch angegeben werden. entweder für alle Anfragen, für ein Subnetz, einen Adresspool in einem Subnetz oder letztlich für jeden Host individuell. Optionen werden vererbt, so dass die jeweils spezielleren Optionen die allgemeineren überschreiben. Um dies an einem Beispiel zu illustrieren, sei folgende Situation gegeben:

- DNS-Server für alle Clients ist 192.168.1.1
- Defaultrouter für alle Clients ist 192.168.1254
- DNS-Domain für alle Clients ist tsbe.ch
- das Subnetz 192.168.1.16-192.168.1.31 hat den Defaultrouter 192.168.1.30
- der Host 192.168.1.25 soll die DNS-Domain tsbe.net haben

Wenn also der Host 192.168.1.25 nach seiner Konfiguration fragt, so bekommt er neben seiner IP-Adresse den Defaultrouter 192.168.1.30 und die Domain tsbe.net zugewiesen.
Zusätzlich kann es Parameter geben, die nicht die Clients betreffen. Dies sind Angaben für dynamische DNS-Updates (welche Zone anwelchem DNS-Server mit welcher Authentisierung) und die Hochverfügbarkeit (für welchen Pool wird mit welchem anderen Server gearbeitet).
Eine einfache dhcpd. conf für das Netzwerk 192.168.l.0/24 soll den Aufbau verdeutlichen:

```
Option domain-name "tsbe.de“;
Option domain—name—servers 192.168.1.l;
Option routers 192.168.1.254;
ddns-update-style none;

subnet 192.168.1.0 netmask 255.255.255.0 {
      range 192.168.1.5 192.168.1.100;
}
```

Mit dieser kurzen Datei werden die Adressen von 192.168.1.5-192.168.1.100 an DHCP-Clients verteilt. Die erste Zeile teilt jedem Client als Defaultdomain tsbe.ch zu. Zeile 2 setzt den DNS-Server auf 192.168.1.1. Die dritte Zeile setzt den Defaultrouter auf 192.168.1254. Sinnvollerweise packt man diese Option in den Subnetzblock, da jedes Subnetz einen anderen Defaultrouter hat. In diesem einfachen Beispiel kann die Angabe aber auch im allgemeinen Block stehen bleiben. Zeile 4 enthält eine Anweisung zur Steuerung dynamischer DNS-Updates, mit denen der vom Client mitgeschickte Name im Nameserver eingetragen wird. Mit der Zeile 4 abgeschaltet. Ohne Angabe würde der DHCP-Server eine Fehlermeldung produzieren. Schliesslich folgt die Subnetzdefinition. Für diesen Netzblock (in dem auch unser DHCP—Server liegt) gilt die Range-Definition, aus der dann Adressen zugeteilt werden.

Syntax
An dem einfachen Beispiel wird die Syntax der dhcpd.conf deutlich. Anweisungen werden immer mit einem Semikolon beendet, die Ausnahme sind Blockdefinitionen, die in geschweiften Klammern stehen. Alle DHCP-Optionen, die den Clients zugeteilt werden. haben die Form:

```
option optionsname wert;
```

Ist der Wert ein String wie etwa der Domainname, so wird dieser in doppelte Anführungszeichen gesetzt. Zahlen und IP-Adressen benötigen keine.

Netzblöcke und Subnetze
Wenn ein DHCP-Server mehrere Subnetze bedient. stehen die subnetzspezifischen Angaben innerhalb von Subnetzblöcken in der dhcpd . conf. Mindestens der Defaultrouter und der Bereich. aus dem DHCP-Adressen vergeben werden, unterscheiden sich pro Subnetz. Ein Subnetzblock ist wie folgt aufgebaut:

```
subnet netzadresse netzmaske {
      Optionen für diesen Bereich
}
```

Der ISC-DHCPD braucht für jedes angeschlossene Netz und jedes Netz, für das er Adressen zuteilen soll, eine solche Definition. Auch wenn alle weiteren Parameter allgemeingültig sind. weil nur ein Netz bedient werden soll, wie im vorangegangenen Beispiel, ist dies zwingend notwendig. Soll zum Beispiel ein DHCP-Server für das Netz 10.1.1.0/ 24 zuständig sein, das hinter einem Router liegt, aber der Server selbst im Netz 192.168.1.0/24 stehen. so muss auch für das Netz 192.168.l.0/24 eine Subnetzdefinition vorhanden sein. Diese kann leer sein. wenn in diesem Netz keine Anfragen bearbeitet werden sollen. Aber ohne diese Definition für die unmittelbar angeschlossenen Netze verweigert der DHCP-Daemon den Start.
Generell spart sich der Administrator Mühe, wenn er netzspezifische Parameter wie den Defaultrouter immer in subnet-Blöcke stellt, da dann weniger umkonfiguriert werden muss, wenn später einmal ein zweites Netz hinzukommt.
Der Parameter range gibt eine Start- und Endadresse an. Aus diesem Bereich werden dann die Clients bedient. Der Bereich muss in das Subnetz passen. Sollten in demselben LAN-Segment mehrere verschiedene IP-Netze betrieben werden, für die der DHCP—Server Adressen verteilen soll. so müssen die subnet-Blöcke in einen shared-netvork-Block eingebettet werden.

Pools
Wenn innerhalb eines Subnetzes unterschiedliche Parameter für verschiedene Gruppen von Hosts gelten sollen. kann als Hierarchiestufe innerhalb eines Subnetzes ein Pool dienen. Um das Ganze an einem Beispiel zu erläutern, erweitern wir dhcpd. conf, so dass nur noch zehn unbekannte Hosts (ohne Angabe einer Hostdeklaration) eine Adresse zugeteilt bekommen. Der Rest steht für registrierte Hosts zur Verfügung.

```
Option domain—name "tsbe.de';
option domain—name-servers 192.168.1.1;
ddns-update-style none;

subnet 192.168.1.0 netmask 255.255.255.0 {
      option routers 192.168.1.254;
      pool {
            range 192.168.1.5 192.168.1.14;
            allow unknown-clients;
      }
      pool {
            range 192.168.1.15 192.168.1.100;
            deny unknown-clients;
      }
}
```

Die Unterteilung erfolgt aufgrund des Rangs. Hier werden nur zehn Adressen an Hosts vergeben, die nicht explizit in der Konfigurationsdatei deklariert wurden. allow unknown-clients bzw. deny unknown-clients erlaubt bzw. verbietet unbekannten Hosts die Zuteilung einer IP-Adresse. Stünde eine dieser Anweisungen auf der äusseren Ebene, so würde sie für die gesamte Konfiguration gelten. könnte aber in einem subnet. pool oder host überschrieben werden.

Hosts
Die feinste Abstufung ist schliesslich der einzelne Host. Die Zuordnung, welcher Server/Client welchem Eintrag entspricht, erfolgt entweder über die MAC-Adresse oder einen DHCP-Client-Identifier, den der Client mitsenden muss. Dabei hat der Identifier die höhere Priorität.
Das folgende Beispiel enthält zwei Hostdefinitionen. Der erste Host identifiziert sich über die Hardwareadresse, der zweite über einen Identifier:

```
host hostl {
      hardware ethernet 00:aa:bb:11:22:33;
      fixed—address 192.168.1.8;
}
host host2 {
      dhcp-client—identifier "host2";
      fixed—address 192.168.1.9;
}
```

DNS-Anbindung

Haben die Clients über den DHCP-Server eine IP-Adresse bezogen, sind sie zunächst einmal nur unter dieser ansprechbar. Um die Clients auch mit einem im ganzen Netzwerk bekannten Namen anzusprechen, lässt sich der DHCP-Server auch mit einem Nameserver koppeln. Nach der Zuweisung einer Adresse an einen Client werden die Adresse und ein Hostname auf dem DNS-Server eingetragen. Der verwendete Hostname kann entweder vom Client gesendet oder vom Server vorgegeben werden.
Ohne die Option ddns-update-style ist eine DNS-Anbindung nicht funktionsfähig. Die Anweisung akzeptiert eines von drei Argumenten:

`none:` dynamische DNS-Updates vollständig abschalten.
`ad-hoc:` Ad-Hoc-Schema für die Updates verwenden.
`interim:` Interim-Schema für die Updates verwenden.

Ad-hoc und Interim implementieren die Updates kompatibel zu RFC 2136 für allgemeine DNS-Updates (nicht nur DHCP-Server schreiben auf einen DNS—Server), so dass sie mit jedem DNS-Server arbeiten sollten, der diesen RFC implementiert.

Ad-Hoc
Das Ad-Hoc-Schema ist eine Eigenentwicklung der ISC-Entwickler. Es funktioniert aber nicht in einem High-Availability-Szenario, da es nicht vorsieht, dass zwei DHCP-Server Updates ausführen können. Für einen DNS-Eintrag benötigt der DHCP-Server einen Fully Qualified Domain Name (FQDN) des Clients. Diesen bestimmt der Server im Ad-l-Ioc-Schema in zwei Phasen:
zunächst den Hostnamen, dann den Domainnamen. Zur Bestimmung des Hostnamens wird zunächst überprüft, ob die Option ddns-hostname6 für diesen Client gesetzt wurde. Wenn ja, wird diese angewendet, wenn nein, wird überprüft, ob im DHCP-Client-Identifier ein FQDN mitgeschickt wurde. Ist dies auch nicht Fall, prüft der Server, ob der Client eine Hostname-Option mitgeschickt hat. Ist immer noch kein Name gefunden, so ist die letzte Möglichkeit eine host-Deklaration für diesen Client. Ist eine vorhanden, wird diese angewendet. War keine der Methoden zur Hostnamen-Ermittlung erfolgreich, wird kein DNS-Update ausgeführt.
Wenn nun ein Host- und ein Domainname ermittelt wurden, kann ein Update stattfinden. Der DHCP—Server versucht zunächst, einen A-Record anzulegen. Erhält er als Antwort des Nameservers einen Fehlercode, dass es diesen A-Record bereits gibt, so findet der Update nicht statt. da ja sonst ein DHCP-Client die Rolle eines Servers mit fester Adresse übernehmen könnte - das würde Man-in-the-Middle-Angriffe sehr einfach machen. War das Anlegen des A-Records erfolgreich, so wird auch der PTR oder Reverse Record angelegt.

Wenn eine Lease abläuft oder der Client sich mittels DHCPRELEASE abmeldet, werden der A-Record und der PTR-Record wieder entfernt.

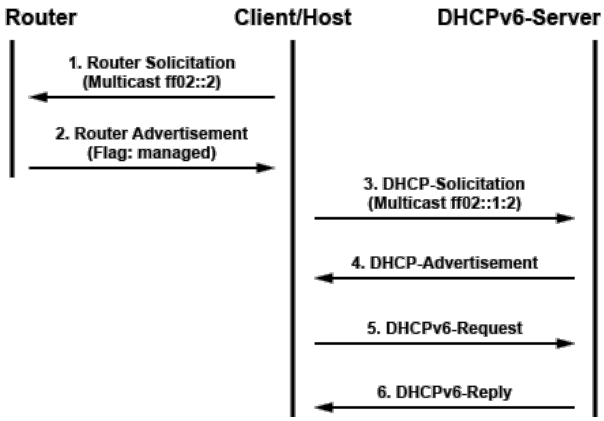
Interim
Es werden nicht unbedingt A- und PTR-Record vom Server eingetragen, wenn der Client in seiner Anfrage eine FQDN-Option mitschickt. Diese Option enthält ein Flag. das bestimmt, ob der Client seinen A-Record selbst updaten möchte. Ist es gesetzt, kann der DHCP-Server konfiguriert werden, sich an die Vorgabe des Clients zu halten. Dies geschieht mit der Anweisung `allow client-updates`. Das Gegenstück heisst

`ignore client-updates`. Wenn der Server die Updates der Clients erlaubt und der Client einen FQDN in der FQDN-Option mitschickt, wird dieser verwendet, um den PTR-Record einzutragen. Den A-Record trägt der Client selbst ein.

Security
Wer in einer Nameserverkonfiguration dynamische Updates erlaubt sollte sicherstellen, dass es nicht zu unerwünschten Änderungen kommt. Im DNS-Teil wird darauf eingegangen, wie ein Nameserver zu konfigurieren ist, damit dies mit einem vertretbaren Mass an Sicherheit zu bewerkstelligen ist.
Um Updates abzusichern. können geheime Schlüssel eingesetzt werden (dnssec-keygen), die zusammen mit einem Hash-Algorithmus verifizieren, dass der Sender des Updates berechtigt ist. dieses durchzuführen.

Verbindungsaufbau IPv6

Bei IPv6 benötigt die IP-Konfiguration eigentlich keinen DHCP-Dienst. Dafür gibt es die Stateless Address Autoconfiguration (SLAAC). Mit SLAAC kann sich ein IPv6-Host vollautomatisch konfigurieren und die notwendigen IP-Konfiguration besorgen. In der Praxis sind nicht alle Betriebssysteme dazu in der Lage. So können manche den DNS-Server auf diese Weise nicht entgegennehmen (RDNSS-Option). DHCPv6 ist im Prinzip das einzige Verfahren, welches diese und weitere Angaben innerhalb der IPv6-Autokonfiguration ergänzen kann. Um wie bei IPv4 mit DHCPv4 die gleichen Funktionalitäten für IPv6 zu ermöglichen, wurde DHCPv6 definiert.
Weil bei DHCPv6 die IP-Konfiguration zentral vergeben und gespeichert wird, spricht man von "Stateful" Address Configuration.
In der Praxis sieht die IP-Autokonfiguration häufig so aus: Per Router-Advertisement werden die IP-Grundparameter verteilt und mit DHCPv6 alles weitere. Die Autokonfiguration bleibt dabei "stateless". Anders sieht es aus, wenn auch DHCPv6 der Präfix verteilt wird. Dann ist die Autokonfiguration "stateful".
Hinweis: Auch bei einer "stateful" IPv6-Konfiguration muss das Router-Advertisement aktiviert sein. Nur so kann jeder Host seine link-lokale IPv6-Adresse erzeugen. Dann allerdings ohne den globalen Netzpräfix für die globale IPv6-Adresse. Der globale Präfix ist in diesem Fall im DHCPv6-Server hinterlegt (stateful). Der generiert daraus die 128 Bit lange IPv6-Adresse für die anfragenden Clients. Welchen Hostanteil (Interface Identifier) der Server aus dem möglichen Adressraum verwendet, hängt dabei von der Implementierung ab.



Der eigentliche Ablauf von DHCPv6 erfolgt in 4 Schritten (Schritt 3 bis 6). In der folgenden Beschreibung ist die vorhergehende Autokonfiguration der globalen IPv6-Adresse über SLAAC berücksichtigt (Schritt 1 und 2). Wobei das Router Advertisement (Schritt 2) der Router Solicitation (Schritt 1) zuvorkommen kann.

1. Der Client bzw. Host sendet eine Router Solicitation (RS) an die Multicast-Adresse "ff02::2". Damit bittet der Client um einen Präfix für die globale IPv6-Adresse.
2. Der zuständige Router antwortet mit einem Router Advertisement (RA). Die Nachricht enthält die MTU (Größe der IP-Pakete) und den globalen Präfix für das Netzwerk (stateless) oder das Flag "managed" (stateful).
3. Dann sendet der Client eine DHCP-Solicitation-Nachricht an die DHCPv6-Multicast-Adresse "ff02::1:2" (alle DHCPv6-Server).
4. Die erreichbaren DHCPv6-Server antworten mit einer DHCP-Advertisement-Nachricht, die die Parameter (DNS-Server, NTP-Server etc.) zur Vervollständigung der IP-Konfiguration enthält (bei stateful auch den globalen Präfix).

5. Der Client wählt eine IP-Konfiguration aus und fordert sie beim jeweiligen DHCPv6-Server mit einem DHCPv6-Request explizit an.
6. Der DHCPv6-Server speichert die IP-Konfiguration mit der Client-ID (Stateful Address Configuration) und bestätigt dem Client die IP-Konfiguration per DHCPv6-Reply. Alle anderen DHCPv6-Server, die keine Anforderung des Clients erhalten haben, geben ihre angebotene IPv6-Adresse wieder frei.

Der Client konfiguriert sich nach Erhalt der Bestätigung und kann anschließend im Netz mit einer globalen IPv6-Adresse kommunizieren.

DHCPv6 vs. Router Advertisment
Ob man DHCPv6 ODER Router Advertisement für die IPv6-Autokonfiguration verwendet, darüber kann man gar nicht entscheiden. Grundsätzlich kommt man ohne Router Advertisements nicht aus. Auch wenn man keine "stateless" IPv6-Adressen haben möchte, sind Router Advertisements nötig. Der zuständige Router verteilt seine Router Advertisements mit dem Flag "managed", woran ein Host erkennt, dass er sich die globale IPv6-Adresse per DHCPv6 holen muss. Das heisst, ohne RA ist kein DHCPv6 möglich.
DHCPv6 hat noch einen weiteren Makel, weshalb RAs unverzichtbar sind. Die ursprüngliche Spezifikation von DHCPv6 sah die Konfiguration eines Standard-Gateways nicht vor. Hierzu gibt es nur den Vorschlag "DHCPv6 Route Options", womit sich die Adresse des Default-Gateways über DHCPv6 konfigurieren lässt, wie es bei DHCPv4 üblich ist. "DHCPv6 Route Options" würde es erlauben, DHCPv6 ohne Router-Advertisements zu betreiben. Allerdings ist es fraglich, ob das jemals so zum Einsatz kommen wird.
Deshalb gilt, dass jede IPv6-Konfiguration mit DHCPv6 auch immer parallel mit Router Advertisements erfolgen muss.

Funktionsweise für IPv4

Im einfachsten Fall hat ein Rechner eine physische Verbindung zum lokalen Netzwerk, verfügt aber selbst weder über eine IP-Adresse noch eine IP-Konfiguration. Daher schickt er eine DHCP-Anfrage in einem Paket an die Broadcastadresse im Netzwerk - auf IP-Ebene lautet diese 255.255.255.255.
Auf der darunterliegenden Ebene (z. B. Ethernet) wird ebenfalls die entsprechende Broadcastadresse verwendet (z. B. FF:FF:FF:FF:FF:FF) ebenso wie für andere Hardwareschichten. Damit ist sichergestellt, dass alle mit dem Netz verbundenen Rechner die Anfrage erhalten und der Server sie beantworten kann. Das Paket mit der Anfrage wird als **UDP-Paket an Port 67** gesendet, also dieselbe Portnummer, die auch BOOTP-Server verwenden.

Die erste Nachricht ist vom Typ **DHCPDISCOVER**. Der Client erfragt so ohne Vorgabe einer eigenen IP-Adresse eine Netzwerkkonfiguration. Die Menge der Informationen, die ein Client erwartet, ist variabel. Verschiedene Betriebssysteme etwa verlangen unterschiedliche Listen von Optionen, die sie für den Netzbetrieb benötigen. Die Abfrage folgender Angaben ist üblich:

- Eigene IP-Adresse / Netzmaske
- Defaultroute
- DNS-Server

Nach weiteren Optionen fragt der Client bei Bedarf: Windows-Systeme beispielsweise nach NetBios Servern, Sun Install Clients nach dem Fileserver und dem Dateinamen zum Booten.

Dem Client können nun mehrere Server antworten. Diese Antworten heissen **DHCPOFFER**. Der Client kann abwarten, ob mehr als eine Antwort ankommt, und dann aufgrund der angebotenen Parameter entscheiden, welche „Offer“ er annimmt. Um nun die Adresse tatsächlich zugewiesen zu bekommen, schickt der Client eine Nachricht vom Typ **DHCPREQUEST**. Das Paket wird daraufhin nicht an die Broadcastadresse gesendet, sondern an den Server, dessen Parameter der Client akzeptiert hat. In diesem Request ist auch das Feld der angefragten IP-Adresse nicht leer, sondern enthält die Adresse aus dem Angebot des Servers. Schliesslich quittiert der Server die Adresszuweisung mit einer Nachricht vom Typ **DHCPACK** (Acknowledged), in der die tatsächlich übermittelten und gespeicherten Parameter sicherheitshalber nochmals mitgeschickt werden. Bei Erhalt der ACK-Nachricht sollte der Client überprüfen, ob die Hardwareadresse im Paket stimmt, und mittels ARP Requests testen, ob die zugewiesene Adresse nicht doch schon im Netz existiert. Wenn dies der Fall ist, teilt der Client dem Server den Misserfolg in einem **DHCPDECLINE** mit. Verweigert der Server die Zuteilung (als Antwort auf das Request-Paket), schickt er ein **DHCPNAK** (Not Acknowledged) statt eines **ACK**. Bleiben die Anfragen des Client unbeantwortet, probiert er es mehrere Male von Neuem mit einer **DHCPDISCOVER**-Nachricht.

Eine zugewiesene IP-Adresse kann der Client beim Herunterfahren mit einer **DHCPRELEASE**-Nachricht an den Server wieder freigeben. Damit steht sie im Pool wieder zur Verfügung, was die Flexibilität in kleinen Netzen erhöht, wo nur wenige Adressen zur Verfügung stehen. Ob dies geschieht, hängt von der Implementierung des Clients ab. Im RFC wird dieses Verhalten mit „should“ empfohlen: Der Client kann so verfahren, muss es aber nicht, wenn er dem Standard entsprechen will.
Jede zugewiesene Adresse ist mit einer Gültigkeitsdauer (lease Time) versehen, die von der Konfiguration des DHCP-Servers abhängt. Ist diese überschritten, findet die Verhandlung von Neuem, aber verkürzt, statt: Der Client schickt einen **DHCPREQUEST** mit den zuletzt gültigen Parametern an den DHCP-Server, um nach Möglichkeit die alte IP-Adresse zu behalten. Auch bei einem Neustart kann ein Client die Verhandlung mit den Angaben der letzten Verbindung im Request beginnen. Ist das Netz dasselbe und die Adresse noch nicht wieder vergeben, kann der Server per **ACK** einfach bestätigen. Andernfalls sendet er ein **NAK**, und der Client fängt mit dem **DHCPDISCOVER** von vorne an.