

Datenkommunikation Prof. Marke: Rendezvous Netzwerk-Dienste

Frank Markus Abbühl

Matrikelnummer 02202599

URL: <http://www.cs.fhm.edu/ifw99001/dako/semthem.html>

3. Dezember 2002

Inhaltsverzeichnis

1	Einleitung	2
1.1	Das Szenario	2
1.2	Geschichte und Politik	2
2	Zero-Konfiguration	3
2.1	Adressierung	3
2.2	Namensauflösung	4
2.3	Auffinden von Diensten	5
2.4	Reservierung von Multicast-Adressen	5
3	Rendezvous	6
3.1	Architektur	6
3.2	DNS-Struktur	7
3.3	Programmier-Schnittstellen	8
4	Zusammenfassung	9
A	Fragen und Antworten	9
A.1	Statisch, dynamisch, automatisch	9
A.2	Geschwätzigkeit	9
A.3	Namensdienste	10

1 Einleitung

Mit der zunehmenden Verbreitung mobiler Geräte und dem Eindringen von Netzwerktechnologie in die Haushalte steigt auch der Bedarf an einfach zu bedienenden und idealerweise selbstkonfigurierenden Geräten.

1.1 Das Szenario

Man stelle sich einmal folgende Situationen vor:

Ein Netzwerkdrucker wird ans lokale Netzwerk angeschlossen, sofort erscheint im Druck-Dialog ein neuer Eintrag mit den Eigenschaften des neuen Druckers. Die Anwender können sofort darauf drucken, ohne auch nur einen Treiber installieren zu müssen oder einen Rechner konfigurieren zu müssen. Mein Ferrari parkt in der Garage, von meinem Heimrechner aus kann ich die Musikdatenbank des eingebauten Ogg-Vorbis-Players mit meinen neuesten Erwerbungen füttern. Wenn ich dann mit meiner Freundin durch die Gegend düse und ihr ein Titel besonders gefällt, kann sie diesen einfach auf ihr Notebook herunterladen.

Der übernächste Generation an Videorecordern und mein nächster Toaster verfügen über ein RJ45 Buchse, wie sie standardmäßig neben jeder Steckdose in der Wand eingebaut sind. Im Fernsehprogramm aus dem Internet kann ich jede beliebige Sendung anwählen und der Videorecorder zeichnet sie auf, das funktioniert per Fernbedienung über den Fernseher, vom PC aus und via Mobiltelefon aus Südamerika.

1.2 Geschichte und Politik

Die Idee von selbst-konfigurierenden IP-basierten Netzwerken taucht bereits im Oktober 1989 im RFC 1122 auf, allerdings schlußfolgert der Autor: "We have not reached this ideal; in fact, we are not even close." [3]. In der Zwischenzeit haben viele Unternehmen eigene, proprietäre Protokolle entwickelt: Novell sein IPX, Apple Computer sein AppleTalk und Microsoft sein SMB/CIFS um nur einige zu nennen.

Frischer Wind kam in die Sache erst im Jahr 1997 bei einer Diskussion auf der net-thinkers Mailingliste, an der auch der damals noch an der Stanford Universität studierende Stuart Cheshire teilnahm, aus dessen Interview [5] diese Informationen stammen. Dieser trug seine Ideen Ende 1998 einigen Personen bei der IETF vor, wo im September 1999 die ZeroConf Arbeitsgruppe ins Leben gerufen wurde.

Stuart Cheshire, der mittlerweile als "Wizard without Portfolio" bei Apple Computer arbeitet, ist nach wie vor die treibende Kraft hinter ZeroConf und dessen Implementierung, Rendezvous. Aber auch Personen wie Bernard Aboba von Microsoft und Erik Guttman von Sun Microsystems sowie IBM

und AT&T sind in der Arbeitsgruppe vertreten, was nicht zuletzt eine wichtige Voraussetzung für die breite Akzeptanz neuer Standards darstellt. Mit Mac OS 10.2 (Codename Jaguar) liefert Apple bereits das erste ZeroConf-fähige Betriebssystem aus und veröffentlicht am 26. September den Quellcode von Rendezvous¹ unter einer Open Source Lizenz. Aber Apple steht nicht alleine da, zuerst haben Druckerhersteller wie Hewlett Packard, Epson und Lexmark ihre Unterstützung zugesagt, mittlerweile haben sich auch Canon, Philips, Sybase und Xerox angeschlossen.

2 Zero-Konfiguration

Der aktuelle Vorschlag für die "Anforderungen für die automatische Konfiguration von IP Rechnern" [6] listet eine Reihe von Fähigkeiten auf, über die ein solches System verfügen muß: Automatische Konfiguration der Netzwerk-Schnittstellen, Auflösung von Hostnamen zu IP-Adressen, Reservierung von Multicast Adressen, sowie das Auffinden von Diensten. Zu jedem dieser Aspekte liegen bei der Internet Engineering Task Force (IETF) bereits Dokumente vor, die im folgenden kurz vorgestellt werden sollen, unter anderem auch in Hinblick auf wichtige Eigenschaften wie Skalierbarkeit, Integrationsfähigkeit und Sicherheit.

2.1 Adressierung

Prinzipiell lassen sich drei Arten unterscheiden, wie Netzwerk-Schnittstellen mit einer IP-Adresse versehen werden können: *Statisch* durch manuelle Vergabe nach den Maßgaben einer zentralen Instanz, *dynamisch* mit Hilfe eines DHCP Servers oder eben *automatisch*, indem die Netzwerkknoten untereinander darüber kommunizieren.

Zu diesem Zweck wurde bei der IANA der Adreßbereich 169.254.0.0/16 für sogenannte *link-local* Adressen registriert². Internet-Router dürfen solche Pakete, ebenso wie Pakete aus anderen reservierten Adreßbereichen, nicht weiterleiten.

Die Vergabe von link-local Adressen sollte stabil sein, so daß jedes Gerät möglichst lange seine einmal erhaltene Adresse behalten kann. Andererseits soll es sogar Netzwerk-Zusammenführungen³ verkraften und selbständig doppelt vorhandene Adressen erkennen und vermeiden, und das alles möglichst robust und zeitnah.

Amit Kucheria erklärt in [7] sehr anschaulich den Algorithmus, nach dem die Adreßvergabe funktioniert. Er umfaßt im Wesentlichen die folgenden Schritte:

¹Unter diesem Markennamen führt Apple seine Implementierung von ZeroConf

²ausgenommen 169.254.0.0/24 und 169.254.255.0/24, diese sind für zukünftige Verwendungen reserviert

³engl.: network joins

1. Der Host erzeugt eine *Zufallsadresse* aus dem link-local Adreßbereich, als Samen für den Pseudo-Zufallsgenerator wird die Hardware-Adresse der Netzwerkkarte (MAC) vorgeschlagen.
2. Anschließend stellt er per *ARP Sondierung* fest, ob die gewählte Adresse bereits vergeben ist und wählt gegebenenfalls eine Neue.
3. Per *ARP Ankündigung* via Broadcast teilt er den anderen Geräten im Erfolgsfall seine neue Adresse mit und füllt so deren ARP-Caches.
4. Zur zeitnahen *Konflikterkennung* reagiert der Host auf alle eingehenden ARP Pakete, die als Absender seine eigene link-local Adresse tragen und reagiert entsprechend.

Da das zugrundeliegende Adreßauflösungsprotokoll (ARP) in sich schon unsicher ist, kann auch das eben beschriebende, darauf aufbauende System keine umfassende Sicherheit bieten. Für einen Angreifer ist es ein Leichtes, durch manipulierte ARP Meldungen beliebig Verwirrung zu stiften. Um zumindest gegen gefälschte Pakete von außerhalb des lokalen Netzes gefeit zu sein, sollten ausschließlich Pakete mit einer Time to Live (TTL) von 255 akzeptiert und verschickt werden. Gegen interne Angreifer empfiehlt sich der Einsatz von netzwerkbasieren Einbruchserkennungssystemen.

2.2 Namensauflösung

Im vorangegangenen Abschnitt haben wir gesehen, wie Geräte automatisch mit einer IP-Adresse ausgestattet werden können, jedoch können Menschen anstatt mit Zahlen viel besser mit Namen umgehen. Die Auflösung von menschenlesbaren Namen zu IP-Adressen ist zweckmäßigerweise Aufgabe des *Domain Name Service* (DNS), einem Dienst, der in konfigurationsfreien Netzwerken nicht zwangsläufig zur Verfügung steht, da zu dessen Pflege ein menschlicher Administrator notwendig ist.

In Abwesenheit eines konventionellen DNS Servers – oder zu dessen Ergänzung – steht in ZeroConf-Netzwerken Multicast DNS (mDNS) zur Verfügung. Die Funktionsweise wird in [10] beschrieben. Namensanfragen für die Domäne *local.arpa.*, welche fest vorgegeben ist für den link-local Adreßbereich 169.254.0.0/16, werden fix an die Multicast-Adresse 224.0.0.251 gerichtet. Jedes Gerät, das mit einem Multicast DNS Responder ausgestattet ist, kann nun an diese Adresse gerichtete Anfragen beantworten.

Um die Netzlast gering zu halten, sind jedoch einige Vorkehrungen nötig. Ein Responder darf nur antworten, wenn er dafür sorgt, daß Duplikate vermieden werden, indem er vom Anfrager mitgeschickte Informationen auswertet und durch eine zufällige Zeitverzögerung Mehrfachantworten unterbindet, indem er während der Wartezeit andere Antworten analysiert.

Beim Start muß der Responder erstens durch seinerseitige Multicast Anfragen die Konsistenz seiner eigenen Informationen überprüfen und zweitens

durch proaktive Antworten den anderen Respondern die Möglichkeit geben, eventuelle Konflikte zu erkennen.

Multicast DNS-Anfragen können auch zur Auflösung nicht-lokaler Hostnamen verwendet werden, sofern mindestens ein Responder Zugang zu einem DNS-Server besitzt. Allerdings erhöht dies auch die Gefahr, daß Adressen gefälscht werden, was die Verwendung von DNSSEC nahelegt, wenn Authentizität eine Rolle spielt.

2.3 Auffinden von Diensten

So angenehm es sein mag, bestimmte Rechner anhand eines leicht zu merkenden Namens anstatt einer kryptischen Ziffernfolge zu identifizieren, so wenig interessiert es den gemein Anwender, auf welchem Gerät ein bestimmter Dienst angeboten wird. Vielmehr interessieren sie die zur Verfügung stehenden Dienste selbst.

In herkömmlichen IP-Netzwerken muß man wissen, unter welcher Adresse und auf welchem Port ein bestimmter Dienst zur Verfügung steht. Erleichtert wird dies ein wenig durch einige Konventionen, die sich im Laufe der Zeit eingebürgert haben. So sind beispielsweise die meisten HTTP-Server auf Rechnern mit dem Namen *www* und unter Port Nummer 80 erreichbar. Ohne dieses Wissen ist es jedoch schwer, verfügbare Dienste zu finden.

Die Stärke von bisherigen proprietären Protokollen wie AppleTalk oder SMB besteht darin, genau diese Lücke zu schließen, verfügbare Ressourcen wie Dateiserver und Drucker stehen einfach über die "Netzwerkumgebung" oder im "Finder" zur Verfügung.

Doch es sind keine proprietären Erweiterungen notwendig, um dieses Ziel zu erreichen. ZeroConf macht sich wiederum den *Domain Name Service* (DNS) zu nutze, da dieser alle Anforderungen erfüllt:

- Der DNS stellt bereits einen zentralen Verzeichnisdienst zur Verfügung, alle benötigten Informationen können in DNS SRV-Records abgelegt werden.
- Zur Registrierung von Diensten bietet das DNS Protokoll die *DNS Dynamic Update* Erweiterung.
- DNS verfügt bereits über ein bewährtes Sicherheitsmodell namens DNSSEC.
- Das Protokoll zum Abfragen von Informationen ist DNS selbst.

2.4 Reservierung von Multicast-Adressen

Zur Vermeidung von Konflikten benötigen viele Anwendungen eine eigene, eindeutige Multicast-Adresse. Das *ZeroConf multicast address allocation protocol* (ZMAAP) erlaubt die Reservierung eindeutiger Multicast-Adressen,

verhindert die Vergabe bereits verwendeter Adressen und hilft bei der Konflikterkennung.

Da ZMAAP kein zentraler Bestandteil von Rendezvous ist, müssen die Details diesmal im Dunkeln bleiben, die genauen Anforderungen sowie Lösungsansätze beschreibt A. Williams in [6].

3 Rendezvous

Die bisher vorgestellten Lösungsansätze für die automatische Zuweisung von link-local Adressen zur Konfiguration der Netzwerk-Schnittstellen auch ohne DHCP-Server, die Namensauflösung mittels Multicast-DNS und das Auffinden von angebotenen Netzwerk-Diensten erscheinen zwar greifbar nahe, sind jedoch nur von theoretischer Natur. Es existiert jedoch auch schon eine erste Implementierung, die im Folgenden vorgestellt werden soll.

Der Name *Rendezvous* ist Apples Markenzeichen für die eigene Umsetzung des zukünftigen IETF ZeroConf Standards, dessen theoretische Grundlagen bereits kurz aufgezeigt wurden. Es ist nun an der Zeit, einen genaueren Blick auf die Design-Entscheidungen und die Architektur selbst sowie auf die Verwendung des Domain Name Service (DNS) zu werfen und die verschiedenen Ebenen der Programmier-Schnittstellen zu betrachten.

3.1 Architektur

Die automatische Zuweisung von link-local IP Adressen ist im Mac OS X bereits auf Betriebssystemebene implementiert. Netzwerkschnittstellen befolgen bereits beim Systemstart das oben beschriebene Protokoll. Im Gegensatz zu älteren Versionen von Mac OS oder auch Microsoft Windows werden link-local Adressen beim Vorhandensein eines DHCP-Servers nicht verworfen, sondern parallel zur zugewiesenen IP-Adresse beibehalten⁴.

Multicast-DNS zur Auflösung von Namen zu IP-Adressen ist als eigener Dienst implementiert, dem mDNSResponder, der auf jedem Rendezvous-fähigen Gerät laufen muß. Er ist vor allem darauf ausgelegt, die von AppleTalk und Konsorten bekannte Geschwätzigkeit in Grenzen zu halten, eine der größten Schwächen solcher Protokolle. Diese wird durch Verwendung von Caching, Unterdrückung von doppelten Antworten und zurückhalten-der Ankündigung von Diensten erreicht.

Zur Illustration hier noch ein anschauliches Beispiel, was alles beim Einschalten eines hypothetischen IP-fähigen Videorecorders vonstatten gehen könnte:

1. Der Videorecorder wählt zufällig die IP-Adresse 169.254.96.212.

⁴Diese Vorgehensweise entspricht auch dem IPv6 Standard, bei dem link-local Adressen standardmäßig vorgesehen sind.

2. Er schickt eine ARP-Anfrage heraus, erhält aber keine Antwort, die IP-Adresse scheint also noch nicht vergeben zu sein.
3. Nun verkündet er per ARP Broadcast seine gewählte IP-Adresse.
4. Der Multicast DNS Responder wird gestartet.
5. Der (erfundene) Videodienst wird gestartet und bindet sich an Port 4410.
6. Er sieht nach, ob bereits eine Instanz mit dem Namen *BlockBuster* vom Typ *_video._tcp.local* existiert. Wenn ja, versucht er es mit einem anderen Namen, zum Beispiel *BlockBuster2*.
7. Ist das erledigt, wird der Dienst unter dem Namen *BlockBuster._video._tcp.local* per Multicast-DNS veröffentlicht. Diese Ankündigung wiederholt er nach einer, zwei, vier, acht, ... Sekunden, bis zu einem Maximum von 4096 Sekunden.

3.2 DNS-Struktur

Zur Repräsentation von link-local Adressen verwendet Rendezvous das Domänensuffix *local*. Es handelt sich dabei nicht um eine Domäne im eigentlichen Sinne, im Gegensatz dazu müssen Namen in der *local*-Domäne nicht weltweit eindeutig sein. Es spricht nichts dagegen, daß zwei Rechner *marax.local* heißen, solange sich diese nicht im selben Netzwerk befinden. Sollten doch einmal Namenskonflikte auftreten – was in der Praxis äußerst selten vorkommt – wird der Anwender nach einem alternativen Namen gefragt.

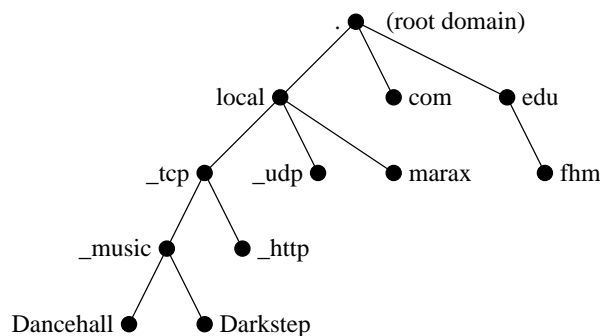


Abbildung 1:

Neben den Hostnamen werden im DNS auch Informationen über die angebotenen Dienste gespeichert, es werden dabei die bei der Internet Assigned Numbers Authority (IANA) registrierten Protokoll- und Portnamen verwendet. Zur Unterscheidung von "echten" Domänen-Namen wird diesen ein

Unterstrich vorangestellt. Der Vorteil dieser Konvention liegt nun darin, daß Dienste nicht mehr unbedingt mit einem wohlbekanntem Port assoziiert sein müssen, es genügt die Kenntnis des Namens. Statt an Port 21 könnte der FTP Server sich auch an Port 31337 binden.

Alle Daten, die ein Client benötigt⁵, werden in SRV-Datensätzen gespeichert.

```
DebianMirror._ftp._tcp._local. 60 IN SRV 0 0 21 marax.local.
```

Um in der Lage zu sein, alle vorhandenen Instanzen zu einem bestimmten Dienst zu finden, zum Beispiel alle verfügbaren Musikboxen, wird durch PTR-Datensätze eine weitere Indirektionsstufe eingeführt:

```
_music._tcp._local. 28800 PTR Dancehall._music._tcp._local.  
_music._tcp._local. 28800 PTR Darkstep._music._tcp._local.
```

Sind für den Betrieb weitere, Dienstspezifische Informationen nötig, können diese in TXT-Datensätzen untergebracht werden. Viele Druckserver erwarten zum Beispiel zusätzlich die Angabe einer Warteschlange.

Für die Kodierung der Namen von einzelnen Instanzen sieht Rendezvous UTF-8 kodierte Zeichenketten vor, da diese Namen normalerweise nicht eingetippt, sondern aus einer Liste ausgewählt werden können. Statt *Debian-Mirror* könnte der Name des FTP Server auch aus griechischen Hieroglyphen bestehen. Ob das eine so gute Idee ist, wird sich herausstellen.

3.3 Programmier-Schnittstellen

Rendezvous bringt auch eine vielschichtige Schnittstelle für Anwendungsprogrammierer mit. Die folgende Illustration zeigt die verschiedenen Abstraktionsebenen.

Foundation Framework API	NSNetService
Core Services Framework API	CFNetService
Low-Level Mach Port API	DNSServiceDiscovery
Multicast DNS Responder	

Tabelle 1: Schichten der Rendezvous API

Auf der untersten Ebene befindet sich der Multicast DNS Responder, die Interaktion mit Rendezvous kann daher direkt über das DNS-Protokoll stattfinden, allerdings müssen die Anwendungsprogramme dann selbst dafür sorgen, die ZeroConf Protokolle einzuhalten.

⁵Hostname und Portnummer, in Zukunft wohl auch Priorität und Gewichtung für einen Lastausgleich

Mit Hilfe der `DNSServiceDiscovery` Klassen läßt sich bereits auf einer abstrakteren Ebene in Sinne von "Service Browsern" und "Services" programmieren. Da diese Klassen Teil des freien Darwin Projektes sind, steht der Quellcode als Open Source Software unter [1] zur Verfügung.

Einen höheren Grad der Integration in die MacIntosh Programmierumgebung Cocoa bieten die Klassen `NSNetService` und `NSNetServiceServiceBrowser`. Deren Methoden unterstützen sowohl das Bekanntmachen als auch das Auffinden von Diensten sowie die Namensauflösung.

4 Zusammenfassung

In dieser Arbeit wurde versucht, einen groben Überblick über Rendezvous, Apples Implementierung des bald zu erwartenden ZeroConf Standards der Internet Engineering Task Force (IETF) zu geben. Die Verwendung von existierenden und erprobten Technologien wie link-local Adressierung und dem Domain Name Service bieten ein solides Fundament für die Zukunft, so daß auch renommierte Unternehmen ihre Unterstützung anekündigt und teilweise auch schon in Produkte umgesetzt haben. Nicht zuletzt die Tatsache, daß Apple Rendezvous als Open Source Software öffentlich zugänglich gemacht hat, gibt Hoffnung, das diese innovative Technologie bald eine weite Verbreitung findet. Wer weiß, vielleicht wird hinter dunklen Vorhängen schon an einer Implementierung für BSD oder Linux gebastelt.

A Fragen und Antworten

A.1 Statisch, dynamisch, automatisch

Frage: Erläutern sie den Unterschied zwischen statischer, dynamischer und automatischer Zuweisung von IP-Adressen. Gehen sie insbesondere auf die Schritte ein, die für die automatische Konfiguration einer Netzwerk-Schnittstelle nötig sind.

Antwort: Statische Adressen werden manuell vom Systemadministrator vergeben. Für die dynamische Zuweisung wird ein DHCP Server eingesetzt, der allerdings auch von einem Administrator aufgesetzt werden muß. Bei einer automatischen Adressvergabe ist keine menschliche Interaktion notwendig. Dazu wählt der Host eine zufällige Adresse aus und überprüft durch ARP-Sondierung, ob diese Adresse noch frei ist. Ist dies der Fall, informiert er die anderen Rechner durch einen ARP Broadcast, andernfalls wählt er eine neue Adresse.

A.2 Geschwätzigkeit

Frage: Durch welche Maßnahmen kann ein Multicast DNS-Responder die Netzlast reduzieren? Erklären sie eine davon im Detail.

Antwort: Durch extensives Caching, Unterdrückung von mehrfachen Antworten und durch sparsames Offerieren von Diensten. Caching: Der mDNS-Responder kann alle Ankündigungen, Anfragen und die darauffolgenden Antworten, die ja über Multicast übermittelt werden, mithören und zwischenspeichern, so daß er keine eigenen Anfragen mehr abschicken muß. Mehrfachantworten: Auch wenn der Responder die Antwort im eigenen Cache vorrätig hat, verzögert er diese um eine zufällige Zeitspanne im Millisekundenbereich. Hat in dieser Zeitspanne bereits jemand anders richtig geantwortet, wird die Antwort komplett unterdrückt. Sparsames Offerieren: Nach dem Start von Diensten wird die Zeitspanne zwischen zwei Multicasts jeweils verdoppelt, bei langlebigen Diensten bis zu einem Maximum von 4096 Sekunden, also knapp über einer Stunde.

A.3 Namensdienste

Frage: Beschreiben sie anhand einer Zeichnung, die Rendezvous den Domain Name Service (DNS) zur dezentralen Speicherung von Hostnamen und Diensten nutzt!

Antwort: Zur Representation von link-local Adressen dient das *local*.-Domänensuffix. Hostnamen sind direkt unterhalb von *local*. zu finden. Zum Auffinden von Hostnamen und Portnummern angebotener Dienste werden SRV-Datensätze verwendet, Für Instanz-Auflistungen PTR-Datensätze und für dienstspezifische Zusatzinformationen TXT-Datensätze.

Literatur

- [1] Apple: "Rendezvous" Homepage:
<http://developer.apple.com/macosx/rendezvous/>
- [2] Homepage der ZeroConf Working Group: <http://www.zeroconf.org/>
- [3] R. Braden: "Requirements for Internet Hosts – Communication Layers", IETF, Oktober 1989
- [4] Jared White: "Rendezvous: It's Like a Backstage Pass to the Future", The Idea Basket, 1. Juli 2002
- [5] Jared White: "On Rendezvous, TiVo, and Parliamentary Titles, An Exclusive Interview with Stuart Cheshire", The Idea Basket, 18. Juli 2002
- [6] A. Williams: "Requirements for Automatic Configuration of IP Hosts", IETF, 19. September 2002
- [7] Amit Kucheria: "Implementation of IETF ZeroConf WG draft titled Dynamic Configuration of IPv4 Link-Local Addresses", 15. November 2001

- [8] Erik Guttman: "Autoconfiguration for IP Networking: Enabling Local Communication", IEEE Internet Computing, May/June 2001
- [9] Stuart Cheshire, Bernard Aboba, Erik Guttman: "Dynamic Configuration of IPv4 Link-Local Addresses", IETF, 23. August 2002
- [10] Stuart Cheshire: "Performing DNS queries via IP Multicast", IETF, 13 Juli 2001
- [11] Stuart Cheshire: "Discovering Named Instances of Abstract Services using DNS", IETF Draft, 13. Juli 2001