TCP/IP

I – EINFÜHRUNG TCP/IP

DARPA: **Department of Defense** (**DoD**) **Advanced Research Projects Agency**
Geschichte: Versuchsreihe der Abteilung **Information Processing Techniques Office** (**IPTO**) des amerikanischen Verteidigungsministeriums (1968), **ARPANET** (Projektüberführung 1969), NCP, TELNET, FTP, IP, TCP, DNS, Teilung des Projekts und Entstehung von **MILNET** (1983)
TCP: **60er 70er Jahre**, Switching Methode, ungleiche Systeme verbinden, Standard Connectivity Dienstprogramme, skalierbares Client-Server-Gefüge, Windows Sockets API
Beteiligte Organisationen: Internet Society (**ISOC**), Internet Architecture Board (**IAB**)
IAB: kontrolliert folgende Ausschüsse: Internet Engineering Task Force **IETF**, Internet Assigned Numbers Authority **IANA** und Internet Research Task Force **IRTF**), veröffentlicht **vierteljährlich** den **IAB Official Protocol Standard**

RFCs: Standards, Dokumente, interner Arbeitsablauf im Internet, durch Übereinstimmung (allgemeiner Konsens) und nicht über Kommitee entwickelt, haben Klassifikation, ursprüngliche RFCs werden nie aktualisiert sondern mit neuer Nummer veröffentlicht,
ALLE STANDARDS SIND RFCs, NICHT ALLE RFCs SIND STANDARDS!
5 Klassifikationen nach Eignung zur Implementierung: **notwendig** (Implementation muß), **empfohlen** (wird angeraten), **wahlfrei** (optional), **eingeschränkte Verwendung** (ohne allgemeine Verwendung), **nicht empfehlenswert** (veraltet, nicht empfohlen)
Standard-Entwicklungsstufen: Vorgeschlagener Standard (stabil, geprüft), Entwurfsstandard (Semantik verständlich), Internet-Standard (hohe technische Entwicklungsstufe)
Drafts: sind Diskussionsunterlagen, keine RFCs
→ TCP/IP Standards werden als RFCs veröffentlicht
→ jeder kann ein RFC einreichen
→ RFCs können über Network Information Venters (NIC) bezogen werden

Dienstprogramme:
Dienstprogramme zur Dateiübertragung: FTP (in NT sowohl als Client als auch Server implementiert), TFTP (Trivial File Transfer Protocol), RCP (Remote Copy Protocol, Kopie auf UNIX-Hosts)
Dienstprogramme zur Remote-Ausführung: Telnet (Terminal-Emulation), RSH (Remote Shell), REXEC (Remote Execution)
Dienstprogramme zum Drucken: LPR (Line Printer Remote, Übertragung an Line Printing Daemon), LPQ (Line Printer Queue, Status einer Druckwarteschleife)
Dienstprogramme zur Diagnose: PING, IPCONFIG (verifiziert Konfiguration), FINGER, NSLOOKUP (überprüft Eingaben DNS-Datenbank), HOSTNAME (Echtheitsbestätigung), NETSTAT (Protokollstatistiken), NBTSTAT (überprüft Zustand NetBIOS, aktualisiert LMHOSTS-Cache, Festlegung registrierter Namen und Bereichs-ID), ROUTE (Routingtabelle), Tracert (verifiziert Route), ARP (Adress Resolution Protocol, zeigt MAC Adressen den Cache lokal ausgewerteter IP-Adressen an)

II – INSTALLATION UND KONFIGURATION TCP/IP

TCP-Dateien: Winnt\System32\Drivers und weiter \etc, HOSTS, LMHOSTS, NETWORKS, PROTOCOL, SERVICES
3 erforderliche WAN-Konfigurationsparameter: **IP**, **Subnet Mask**, **Standard-Gateway**
IP-Adresse: logisch **32bit**, Netzwerk-ID und Host-ID, Netzwerk ID identifiziert physische Mit-Netzwerkcomputer, Host-ID identifiziert Host im Netzwerk
Subnet Mask: dient zum Sperren eines Teils der IP-Adresse, Feststellung ob lokal oder remote
Standard-Gateway: TCP sendet Pakete für Remote-Netzwerke hier hin

2 DIENSTPROGRAMME ZUM TESTEN UND ÜBERPRÜFEN DER TCP/IP KONFIGURATION:

IPCONFIG: wurde eine doppelte Adresse konfiguriert, wird die IP-Adresse wie konfiguriert angezeigt, die Subnet-Mask lautet jedoch 0.0.0.0, Dienstprogramm WINIPCFG kann auch verwendet werden
PING: Diagnostizieren von Verbindungsfehlern, verwendet ICMP ECHO REQUESTS und ECHO REPLYS, Loopback-Treiber umgehen die Netzwerkkarte

Netzwerkmonitor: Überwachen und Sammeln von Netzwerkdaten für Analysezwecke, Sammel- und Anzeigefilter

III – TCP/IP ARCHITEKTUR

DoD Referenzmodell: entwickelte **DoD Protokoll Suite**, als das siebenschichtige ISO/OSI Referenzmodell noch nicht verfügbar war

TCP/IP Protokollfamilie: **4 Schichten**: **Anwendungs**-, **Transport**-, **Internet**-(Internetwork), **Netzwerkschnittstellenschicht** (Network Access)

1. **Netzwerkschnittstellenschicht**: Rahmenübertragung (**LAN-Technologien**: Ethernet, Token Ring, FDDI und **WAN Technologien**: Serielle Leitungen, Frame Relay, ATM); nicht detailliert festgeschrieben, so können höhere Schichten auf einer Vielzahl von Netzwerken eingesetzt werden.

2. **Internetschicht**: Pakete in Internetdatagramme einschließen, Leitwegalgorithmen, Einrichtung von Verbindungen und Adressen
   4 Internetprotokolle: **Internet Protocol IP** (Adressierung und Lenkung der Pakete zwischen Hosts, Schnittstellenspezifikation **NDIS**), **Address Resolution Protocol ARP** (Hardwareadressen von Hosts), **Reverse Address Resolution Protocol RARP**, **Internet Control Message Protocol ICMP** (Nachrichten, Fehler), **IGMP Internet Group Management Protocol** (Host-Gruppenmitgliedschaften lokalen Multicast Routern mitteilen)
   3 Routingprotokolle auf dieser Ebene: **Exterior Gateway Protocol EGP**, **Routing Information Protocol RIP**, **Open Shortest Path First OSPF**

3. **Transportschicht**: Methode für Datenaustausch und -übermittlung, Kommunikationssitzungen, zur Sicherung der Kommunikation nur **2 Protokolle**: **Transport Control Protocol TCP** (verbindungsorientierte zuverlässige Kommunikation), **User Datagram Protocol UDP** (verbindungslos, kleine Datenmengen ohne Empfangsbestätigung)
   - **Verbindungsorientiert**: Dienst, Kommunikation läuft in **3** definierten Phasen:
     - **Verbindungsaufbau**
     - **Datentransfer**
     - **Verbindungsabbau**
   - **Verbindungslos**: Kommunikation aber keine Verbindung wird aufgebaut

4. **Anwendungsschicht**: beinhaltet Programme, Dienstprogramme, API für den Endbenutzer
   - **2 Schnittstellen**: **Windows Sockets** (TCP/IP, IPX), **NetBIOS** (Benennungs, Nachrichtenübermittlungsdienste, TCP/IP und NetBEUI)

Protokolle für serielle Leitungen: **SLIP**, **PPP** (empfohlen)

ARP: Adreßauswertung IP zu Hardwareadresse, kennt nur Adressen im selben physischen Netz, über Rundsendung (Hardwareadresse des Ziels **000000000000**, da Anforderungspaket und MAC-Adresse zu diesem Zeitpunkt noch nicht bekannt, Speicherung als Eintrag im ARP-Cache)
Kommunikationsvoraussetzung: **Auflösung einer lokalen IP Adresse**, ARP-Anforderung, wenn lokal dann Cache-Überprüfung, keine Übereinstimmung → Rundsendung, jeder lokale Host

überprüft, keine Übereinstimmung → Ignore, Übereinstimmung → ARP-Antwort und Aktualisierung des Cache
Auflösung IP Adresse Remote Host: Überprüfung lokale Routing-Tabelle, keine Zuordnung → Standardgateway Rundsendung und Datenpaket an weiterleitenden Router, ICMP Rückantwort durch Ziel-Host an Ursprung
ARP-Cache: dynamische (automatisch) und statische (Löschung bei Neustart oder arp-d oder falsche Angabe) Einträge, dauerhafter Eintrag lokale Rundsendungshardwareadresse **FFFFFFFFFFFF**, Lebensdauer Eintrag 10 Minuten, bei Nichtverwendung 2 Minuten, bei maximaler Kapazität Löschung des ältesten Eintrag, Registrierung **ARPCacheLife** zufügen entschärt Vorgaben
ARP-Paketstruktur: Hardware-Typ, Protokolltyp, Länge Hardware Adresse, Länge Protokolladresse, Operation (Opcode), HW-ADR Absender, PROT-ADR Absender, HW-ADR Ziel, PROT-ADR Ziel
Arp –g: ARP-Cache anzeigen
Arp –s xxx.xxx.xxx.xxx Hardware-Adresse: In Cache einfügen
Arp –a: zuerst PING eingeben, dann Erhalt der phys. Adresse der Netzwerkkarte 2. Computer

ICMP: synonym **Internet Control Message Protocol**, kleines ergänzendes Protokoll auf der IP-Schicht um Fehlermeldungen an höhere Schichten weiterzugeben (**z.B. Ziel nicht erreichbar, Routerumleitung, Echoanfrage**), unzuverlässige IP-Datagramme, **ICMP Source Quench** Meldung von Router an Host um Übertragungsgeschwindigkeit zu reduzieren (synonym **Quelldrosselung**), NT als Router kann nicht
ICMP-Paketstruktur: Typ, Code, Prüfsumme, typenabhängige Daten

IGMP: unzuverlässige IP-Datagramme, Weitergabe an Router, zeigt Multicasting-fähigen welche Host-Gruppen sich im Netzwerk befinden
IGMP-Paketstruktur: Version, Typ, Unbenutzt, Prüfsumme, Gruppenadresse (Host Membership Report)

IP: synonym **Internet Protocol**, **verbindungsloses** Protokoll (übermittelt **TPDUs**: **Transport Protocol Data Units**), Definition in **RFC 791** und **MIL-STD 1777**, Adressierung und Steuerung, keine Empfangsbestätigung erforderlich da TCP verantwortlich, Einfügung in Vorspann von IP-Ursprung-ADR, Ziel-ADR, Protokoll, CRC, TTL → Standard **128 Sekunden**, kann TPDU in kleinere Teile fragmentieren (falls Nachricht zu groß für das Netzwerk)

IP-Paketstruktur: **Version** (**4 Bit**, enthält IP Versionsnummer), **IHL** (**4 Bit**, IP-Vorspannlänge), **Dienste-Typ** (**8 Bit**, besteht aus **3** Prioritätsbits: **D-Bit**: fordert niedrige Wartezeiten, **T-Bit**: fordert hohen Durchsatz, **R-Bit**: fordert hohe Zuverlässigkeit, **2** nicht verwendete Bits, maximal **2** Anforderungen können gleichzeitig eingestellt werden), **Gesamtlänge** (Länge Vorspann + Daten), **Kennzeichnung** (Identifikation des ganzen Datagramms, nicht Fragmente), **Fragmentierungs-Flag** (zeigt ob Fragmentierung vorhanden und ob das aktuelle das letzte ist), **Fragment-Offset** (Einordnung des Fragments im Datagramm), **TTL** (synonym **Time to live**, wie lang Datagramm im Internet bleiben darf), **Protokoll** (Name des Weiterverarbeitungsprotokoll nach IP, z.B. hat **TCP** die **Protokollnummer 6**), **CRC** (Vorspannprüfsumme, bei jedem neuen TTL Wert wird der CRC neu berechnet), **Ursprungsadresse**, **Zieladresse**, **Optionen und Auffüllen mit Nullen** (zusätzl. Dienste oder Füllwerte), **Daten**

IP auf dem Router/Gateway: **TTL minus 1sec**, bei **0** Verwerfung, bei Fragmentierung neuer Vorspann mit Flag, Fragment ID und Fragment-Offset (Ablageinfo), neue CRC Berechnung

Bezeichnung von Dateneinheiten:

- **TPDU**: Dateneinheit der TCP Schicht
- **Paket**: Bezeichnung der IP Schicht
- **Datenblock**: Benennung in den unteren Schichten

TCP: synonym **Transmission Control Protocol**, zuverlässiger, **verbindungsorientierter** Zustelldienst, arbeitet im **Voll-Duplex-Modus**, **Datenflußkontrolle**, Übertragung in Segmenten, innerhalb bestimmter Dauer **ACK-Signal** zur Empfangsbestätigung (ACK on: Feld Bestätigungsnummer enthält Folgenummer des als nächstes zu empfangenden Bytes), bleibt es aus Neusendung

TCP-Paketstruktur: **Quellanschluß** (gibt das **Quell-ULP Upper-Layer-Protocol** an), **Zielanschluß** (Ziel-ULP), **Folgenummer** (zur Datenstrom-Segmentierung), **Bestätigungsnummer**, **Daten-Offset** (**4 Bit**, gibt anzahl der **32 Bit** Wörter im Vorspann am, zeigt wo Daten relativ zum Vorspann beginnen), **Reserviert** (**0**, da für künftige Verwendung reserviert), **Flag-Feld** (**6 Bits**, **URG**: Dringlichkeitsanzeiger gültig, **ACK**: Bestätigungsnummer gültig, **PSH**: **Push-Flag** zur dringenden Übertragung, **RST**: setzt bei Fehler Transportverbindung zurück, **SYN**: wird von Sender und Empfänger im jeweils ersten Paket gesetzt – Handshakestart zum Aufbau einer virtuellen Verbindung, **FIN**: keine weiteren Daten, Verbindung kann freigegeben werden), **Fenster** (**2** Puffer, Sendefenster und Empfangsfenster, Fenstergröße wird durch Anzahl ausstehender Pakete bestimmt, sog. **gleitende Fenster**), **Prüfsumme** (Identitätsvergleich zweimal berechneter Prüfsummen bei Sender und Empfänger, erzeugt **ACK** oder **NACK**: d.h. fehlendes ACK, siehe **PAR**), **Dringlichkeitsanzeiger** (wichtiger Datenhinweis für ULP), **Optionen** (Mindestlänge = Mindestfüllwert **32 Bit** oder **Vielfaches** (Modulstruktur)), **Daten**

PAR: synonym **Positive Acknowledgment with Retransmission**, eine Methode: NACK wird nicht explizit gesendet, das vermindert Overhead im Netzwerk durch Reduzierung des Verkehrs
→ **Netzwerkoverhead wird durch Bestätigung des Empfängers verursacht**

Anschlüsse: Protokoll-Anschlußnummer, beliebige Zahl zwischen **0 und 65536**, bekannte Anwendungen Server-seitig besitzen von **IANA** zugewiesene Anschlußnummern, Client-seitig dynamisch durch OS, Anschlüsse und Sockets bestehen aus einem zusammenhängenden Nummernsatz
Sockets: Endpunkt Netzwerkkommunikation, **3** Angaben: IP-ADR, Dienstart, Anschluß
TCP-Anschlüsse: **unter 256** häufig verwendete definierte Anschlüsse, z.B. **21** für FTP

TCP-Dreier-Handshake: TCP-Sitzung Einleitung und Beendigung
1. Segmentsendung, **SYN-Flag on** (→ dann steht im Feld Folgenummer die erste Folgenummer, die zur Übertragung verwendet wird)
2. Bestätigung, SYN-Flag on, Folgenummer Startbyte
3. Zurücksendung mit bestätigter Folge- und Bestätigungsnummer

TCP-Schiebefenster: Datenpufferung zwischen **2** Hosts mit Schiebefenster, jeder Host **2** (Empfang /Senden), Größen passen sich gegenseitig an (Client und Server), Größe des Fensters gibt Datenvolumen an

- **Gleitende Fenster**: Puffer können für jedes aus dem Puffer entfernte Paket mit einer kleineren Folgenummer ein weiteres Paket mit einer höheren Folgenummer aufnehmen
- **Senedefenster < Empfangsfenster**: damit kein Überlauf entsteht

UDP: **verbindungsloser Datagrammdienst**, schneller aber weniger zuverlässig als TCP, arbeitet ohne ACK Bestätigungen, Verwendung für kleine Datenmengen z.B. NetBIOS Namensdienst oder SNMP
UDP-Anschlüsse: Funktion einer Multiplex-Nachrichten-Warteschlange, d.h. Nachrichten können gleichzeitig empfangen werden, Achtung Nummernanschlüsse können von der Zahl wie TCP identisch sein, sind aber UDP-Nummern!

IV – IP ZUWEISUNG

IP-Adresse: Position eines Systems innerhalb eines Netzwerks, eindeutig, einheitliches Format (binäre oder Punkt-Dezimal-Schreibweise), **2** Teile Netzwerk- und Host-ID, unterstützte Hosts gesamt **3.720.314.628**
Netzwerk-ID: physisches Netzwerk, alle darin enthaltenen Systeme haben die gleiche ID
Host-ID: Arbeitsstation, Server, Router
Adreßklassen: 5 Adreßklassen sind definiert, MS TCP/IP unterstützt 3 (A,B,C), Angabe welche Bits für Netzwerk- und Host-ID verwendet werden
Klasse A: große Anzahl Hosts, höherwertige Bit 0, Aufteilung 1Netzwerk-ID 3 Host-ID, 126 Netzwerke mit bis zu 17 Millionen Hosts pro Netzwerk
Klasse B: zwei höherwertige Bits mit Kombination 1 0, Aufteilung 2 NID 2 HID, 16384 Netzwerke mit 65000 je
Klasse C: kleine LANS, Kombination 1 1 0, 3 NID 1 HID, 2 Millionen Netzwerke mit je 254 Hosts
Klasse D: Multicast-Gruppen, 0 oder 1 oder keine Hosts, Kombination 1 1 1 0, keine Netzwerk- oder Host-Bits, Weiterleitung an ausgewählten Teil für Multicast-Adressen registrierte Hosts, MS unterstützt Klasse D für Anwendungen
Klasse E: experimentell für zukünftigen Einsatz, höherwertige Bits 1 1 1 1
Adressierungsrichtlinien: NID nicht 127 da Loopback, NID und HID nicht alle 1 da als Rundsendungsadresse gedeutet, NID und HID auch nicht 0 da sonst nur „dieses“ Netzwerk angesprochen, HID innerhalb der lokalen NID eindeutig, jedes LAN und dessen Hosts eindeutige NID, wenn Router verbunden pro WAN-Verbindung eigene NID
Subnet-Mask: Bestimmung ob Host lokal oder remote, jeder Host muß haben, entweder Standard-Subnet-Mask oder benutzerdefiniert für unterteilte Subnets
Standard-Subnet-Mask: nicht unterteilt, abhängig von Adreßklasse, NIDs alle 1 Bit eingestellt (equal 255 Oktett), HIDs alle 0 Bit (equal 0 Oktett)
AND-Vergleich: IP-interner Prozeß zum Festlegen des Paketziels lokal/remote, Vergleich IP-Adresse und Subnet Mask für Paketursprung und Ziel, Bitvergleich, beide 1 dann Ergebnis 1 sonst 0
IP6.0-Adressierung: **IPv6**, früher **IPng** (IP next generation), gegen zukünftige Erschöpfung des Adressraumes, 16 Oktette, 8 durch Punkte getrennte Oktettpaare in Hexadezimal, völlig neue Paketstruktur, 128 Bit Quell- und Zieladressen, 4x größer als IPv4, gewünschte Bandbreite für zeitabhängige Dienste, leichte Unterstützung-Hinzufügung für neue Hardware- und Anwendungstechnologien

V - SUBNETTING

Subnet: unterschiedliche Netzwerk- oder subnet-ID verwenden, Verwendung für Unternehmen
Vorteile: Variation unterschiedlicher Technologien (Ethernet/Token Ring), Begrenzungüberschreitung wie max. Hosts pro Segment, Verbesserung Netzwerkleistung
Wechselbeziehung zwischen Subnet-Zahl und Host-Zahl: je mehr vom einen, desto weniger vom anderen
Definieren der Subnet-Mask:

1. Anzahl phys. Segmente festlegen, Konvertierung Binär-Format
2. Anzahl der benötigten Bits für Segmente als höhere Bits mit 1 darstellen, Ermittlung der Subnet-Mask

Subnetting bei mehreren Oktetten: Verwendung mehr als 8 Bit, Verwendung eines privaten Netzwerks da im Intranet
Subnet-ID:

Supernetting: NIDs werden ausgeliehen und als HID getarnt, Verwendung von Classless Inter-Domain Routing CIDR → Zusammenfassung von zusätzlichen Einträgen zu einem, effizienteres Routing

VI – Implementieren des IP-Routings

Routing: Pfadauswahl zum Senden von Paketen, synonym „Gateways“, Routing-Entscheidung über Routing-Tabelle (die Entscheidungskriterien heißen **Routing-Metrics**)

Routing-Tabelle: Einträge mit IP-Adressen von Router-Schnittstellen zu anderen Netzwerken
Ablauf: lokal/remote Bestimmung, Route zum Remote-Host in Tabelle?, keine Route dann über eigene Standard-Gateway-Adresse an Router, Router sucht? Kein Pfad dann Standard-Gateway-Adresse des Routers, weiter über Hops bis Ziel-Host
Erkennung ausgefallener Gateways: TCP sendet solange an konfiguriertes Standard-Gateway bis Bestätigung, wenn Hälfte von TcpMaxDataRetransmission überschritten und mehrere Gateways konfiguriert → TCP fordert Wechsel zum nächsten Gateway in der Liste
Statisches Routing: manuelle Erstellung und Aktualisierung, keine Informationsweitergabe an andere Router, keine Pfad-Aktualisierung automatisch, zu allen bekannten Netzwerken müssen für jeden Router statische Routing-Tabelleneinträge konfiguriert werden: NID Adresse des anzusteuernden Netzwerkes und IP der Netzwerkschnittstelle

Erstellen einer Routing Tabelle:

Route add (Netzwerk) mask (SubnetMask) (Gateway)	Hinzufügen (Muß neu bei Boot)
Route –p …	Ständige Route (Registry-save)
Route print	Anzeige der Standardeinträge
Route delete (netzwerk) (Gateway)	Löscht Route
Route change (Netzwerk) (Gateway)	Änderung
Route –f	Löscht alle Routen

Statischer Eintrag:

Netzwerkadresse	NID des Zielnetzwerkes, bei Verwendung Netzwerkname aus **Datei Networks** entnehmen
Subnet Mask	der Netzwerkadresse
Gateway-Adresse	IP und Hostname der Schnittstelle zum Zielnetzwerk, bei Verwendung Hostname aus **Datei Hosts** entnehmen

→ Dateilokation in \systemroot\system32\drivers\etc

Standardeinträge Routing-Tabelle:

0.0.0.0	Standardroute für nicht angegebene Netze
Subnet-Broadcast	im lokalen Subnet verwendete Adresse
Netzwerk-Broadcast	im Netzwerkverbund verwendete Adresse
Lokales Loopback	
Lokales Netzwerk	an Hosts
Lokaler Host	Verweis auf lokale Loopback-Adresse

Dynamisches Routing: **RIP** (Routing Information Protocol), **OSPF** (Open Shortest Path First), automatische Routing-Änderung und Info-Austausch zwischen Routern, Standard-Gateway-Adresse muß konfiguriert werden, sonst keine weitere Konfiguration
NT als Router: „Mehrfach vernetzter Computer“
RIP: Versendung der RIP-Nachrichten über UDP-Anschluß **520**, Austausch von NID und Entfernung (Hop-Count Feld, Maximum **15**, danach unerreichbar, automatische Auswahl der Route mit niedrigster Hop-Zahl), Standardaktualisierungsintervall **30 sec** **auf MAC-Ebene Rundsendung**, sodann Hinzufügen neuer Routen, wenn bestehende Route weniger Hops dann Austausch, Versand über Rundsendung
RIP-Probleme: Anwendung nur bei kleinen Netzwerkverbund und keine WANs, wenn viele Einträge dann Netzwerkbelastung, **Problem der langsamen Konvergenz**: **3** Minuten Zeitüberschreitungswert, bei Nichtverfügbarkeit Rundsendung
Integrieren von statischem und dynamischem Routing: Routing Tabellen statisch/dynamisch muß jeweils statische Route hinzugefügt werden
Implementation des NT-IP-Routers: Netzwerkkarten und Treiber installieren oder mehrere IP-Adressen auf einer Netzwerkkarte, IP und Subnet Mask hinzufügen, Netzwerk-TCPIP-Eigenschaften-Routing-**IP-Forwarding** aktivieren, dann unter Netzwerk-Dienste **IPRIP-Dienst** hinzufügen oder statische Routen eingeben
TRACERT: Überprüfung ob Routing über Router ordentlich ausgeführt wird, Feststellung an welcher Stelle das Routing fehlgeschlagen ist, Erkennung langsamer Router über Reaktionszeit-Ausgabe, Anwendung tracert www.microsoft.com

VII – DHCP

DHCP: Dynamic Host Configuration Protocol, automatische IP-Adressen Zuweisung, Überwindung von Konfigurationsbeschränkung, Erweiterung des **BOOTP**-Protokolls
BOOTP: Clientstart ohne Festplatte und automatische Konfiguration von TCP/IP, Unterstützung im Service Pack 2, RFC 1532

Gegenüberstellung manuelle/automatische Konfiguration:

TCP/IP manuell: Gefahr der Auswahl einer ungültigen IP-Adresse, Kommunikationsprobleme aufgrund falscher Subnet-Mask/Gateway möglich, Verwaltungsaufwand durch Verschiebung in Subnets
TCP/IP über DHCP: alle erforderlichen Konfigurationsinfos automatisch

DHCP-Funktionsweise: **4** Phasen, Prozeß wird für jede Netzwerkkarte getrennt durchgeführt, UDP-Kommunikation über Anschlüsse 67, 68
PHASE1: **IP-Lease-Anforderung** (Rundsendung an alle DHCP-Server, Quelladresse 0.0.0.0, Zieladresse 255.255.255.255, sogenannte **DHCPDISCOVER**-Nachricht → Hardware-Adresse und Computer-Name des Clients
→ **kein DHCP-Server verfügbar?** Nach **1 sec** weitere Angebote **in 9, 13, 16sec** Intervallen, dann alle **5 Minuten**
PHASE2: **IP-Lease-Angebot** (von allen DHCP-Servern **DHCPOFFER**-Nachricht, Angebot von IP-Adresse, Subnet-Mask, Lease-Dauer und Server-ID, gleichzeitig Reservierung der IP-Adresse)
PHASE3: **IP-Lease-Auswahl** (Auswahl aus erstem erhaltenen Angebot und **DHCPREQUEST**-Nachricht als Info-Rundsendung, andere Server ziehen Angebote zurück)
PHASE4: **IP-Lease-Bestätigung** (Server-seitig **DHCPACK**-Nachricht, Client-seitig Inititalisierung und Bindung von TCP/IP und Eintrag in Registry unter H_L_M\system\CCS\Services\Netzwerkkarte\Parameters\TCPIP)
→ **nicht erfolgreich? DHCPNACK**-Nachricht wird rundgesendet

IP-Lease-Erneuerung: nach **50%** der Lease-Dauer über **DHCPREQUEST** an Server, Server nicht erreichbar kann der Client noch bis Ende der Dauer ohne Erneuerung weitermachen, bei Neustart Versuch dieselbe IP-ADR zu leasen über DHCPREQUEST mit Angabe der letzten IP
Nachfolgende Erneuerungsversuche: nach **87,5%** Lease-Dauer erneuter Versuch zu beliebigem Server, **keine Neuanmeldung**: Netzwerkfehler über Anwendung, die über ungültigen TCPIP-Protokollstapel kommunizieren möchte
IPCONFIG: Überprüfen IP-Konfiguration, Optionen und Lease-Dauer erneuern/aufheben

Ipconfig	IP-Adresse, Subnet-Mask, Gateway
Ipconfig /all	etc.
Ipconfig /renew	Aktualisieren der Lease
Ipconfig /release	Freigeben einer Lease über DHCPRELEASE-Nachricht bei Wechsel in Anderes Netzwerk, beim Herunterfahren kein automatisches Release

Implementieren mehrerer DHCP-Server: für jedes Subnet eigener Bereich (Folge von IP-Adressen), mehrere Bereiche für jedes Subnet (75% verfügbarer IP-Adressen des lokalen Subnets, Bereich für Remote-Subnet 25% aber Router als DHCP-Relay-Agent dann wichtig)

DHCP-Anforderungen:

1. Router, die Subnets mit DHCP-Servern und Clients verbinden, müssen als BOOTP-Relay-Agenten fungieren
2. DHCP-Server-Dienst konfigurieren, wenn nicht Schritt 1 dann in jedem Subnet
3. DHCP-Server ist mit statischer IP, Subnet Mask, Gateway etc konfiguriert
4. DHCP-Bereich definieren
5. DHCP-Client muß DHCP-fähig sein (NT4, W95, 3.11 mit TCPIP32, MS Network Client 3 für DOS und TCPIP-Treiber im Real-Modus, LAN Manager 2.2c, kein OS2)

Adresse der Netzwerkkarte: IPCONFIG /ALL, **NET CONFIG SERVER**, über **VERWALTUNG-DIAGNOSE-NETZWERK-TRANSPORTE**

Installation:

1. **Dienste-Hinzufügen-MicrosoftDHCPServer-Reboot**

2. **DHCP-Bereich**: Verwaltung-DHCP-Manager-DHCP-Server-Lokaler Computer-Bereich-Erstellen, DHCP-Pool mit eindeutigen einmal vorhandenen IP-Adressen konfigurieren, auch ausgeschlossenen Adressbereich, Beschränkung der Lease-Dauer (Tage,Stunden,Minuten), **AKTIVIERUNG** durch Ja-Bestätigung (**gelbe Glühbirne** neben IP-Adresse als Aktivierungsbestätigung)
3. **DHCP-Bereichsoptionen:Bereich**: GLOBAL (für alle Clients in allen Subnets dieselben Konfigurationsinfos), BEREICH (für im Bereich enthaltene Adressen, z.B. Gateway für Subnets, prior zu GLOBAL), CLIENT (reservierter DHCP-Adreß-Lease, prior zu vorigen)

MS-fähige Optionen: **003 Router** (gibt IP an, lokal definiertes Standard-Gateway wird aber bevorzugt), **006 DNS-Server**, **0046 WINS/NBT-Knotentyp** (NetBIOS over TCP/IP Namensauswertung, **B-Knoten** (Broadcast), **P-Knoten** (Peer), **M-Knoten** (Mixed), **H-Knoten** (Hybrid)), **044 WINS/NBNS-Server** (überschreibende WINS-Server-IP), **047 NetBIOS-Bereichs-ID** (Kommunikation nur mit anderen NetBIOS Host mit gleicher Bereichs-ID)
→ nicht Microsoft-DHCP-fähige Clients können jede konfigurierte Option empfangen und verwenden
Konfiguration einer Client-Reservierung: z.B. bei Servern mit nicht WINS-aktivierten Clients, da LMHOSTS-Datei für Remote Netzwerk ohne Änderung, PING-ARP-DHCPmanager-LOKALERcomputer-GLÜHBIRNE-RESERVIERUNG HINZUFÜGEN-IP-UID-CLIENTname-Hinzufügen-Schließen
DHCP-RELAY-AGENT: Verwendung zusammen mit stat. Oder dyn. Router, Weitergabe von DHCP-Nachrichten in verschiedene Netzwerke an DHCP-Server, fängt Rundsendungen ab
INSTALLATION: Netzwerk-Dienste-Hinzufügen-Dialogfeld Auswahl Netzwerkdienst-DHCP Relay Agent-OK-NT Setup-weiter-unbeaufsichtigte Installation-Ja-TCP Properties-DHCP Relay hinzufügen-IP Adresse-OK-BOOT
Deaktivierung DHCP-Relay-Agenten: Dienste-DHCP Relay Agent-Startart-Deaktiviert-OK-BOOT
DHCP-Datenbank: Sicherung alle **60 min**, Beschädigung → BAK Herstellung
Sicherung: \systemroot\system32\dhcp\backup\jet , **BackupInterval** in Minuten und Neustart des Dienstes unter **HLM\CCS\Services\DHCPServer\Parameters\BackupInterval**
Wiederherstellung: automatisch (bei Neustart des Dienstes) oder manuell (**RestoreFlag** auf **1** setzen, nach Wiederherstellung automatisch **0**) oder manuell (Inhalt **System32\Dhcp\Backup\Jet** in **System32\Dhcp** kopieren und Server-Dienst Neustart)
Dateien: dhcp.mdb, dhcp.tmp, jet.log und jet*.log (Transaktionsprotokolle), system.mdb (Strukturinformation)
Komprimieren: automatisch, bei Umstieg von NT351 und Datenbank >**30MB** erforderlich, Vorgang über Verzeichniswechesel \systemroot\system32\dhcp\ und Ausführung **Jetpack** Dienstprogramm nach Dienststop, Syntax **jetpack dhcp.mdb *temporärer_Name*.mdb**, Dienststart

VIII – NetBIOS over TCP/IP

NetBIOS: Definition von 2 Entitäten (**SCHNITTSTELLE** auf Sitzungsebene und **PROTOKOLL** für Sitzungsverwaltung/Datentransport)
NetBIOS-Namen: eindeutige **16byte** Adresse, entweder exklusiver Name oder Gruppenname (gleichzeitige Sendung an mehrere Computer)
Prozeßbeispiel für Namen: NT-Server-Dienst, **15 Zeichen** bestehender Computername und **16. Hexadezimales Zeichen**
Registrierung von Namen: bei Start NetBIOS-Namensregistrierungsanforderung, wenn Name schon vergeben dann negative Antwort als **Initialisierungsfehler**
Ermittlung von Namen: über lokale Rundsendung oder NetBIOS-Namensserver, NetBIOS-Namensabfrageanforderung
Freigabe von Namen: bei Beendigung Anwendung oder Dienst, dann Verwendung freigegeben
Segmentieren NetBIOS-Namen mit Bereichen: Isolation von Bereichen, keine Leistungssteigerung aber Verringerung der vom Host akzeptierten Pakete, angehängte Zeichenkette an NetBIOS-Namen (keine Eindeutigkeit bei mehreren Bereichen notwendig),

NetBIOS over TCP/IP Kommunikation nur bei übereinstimmender **Bereichs-ID**, Konfiguration der Berichs-ID über TCP/IP-Properties-Register WINS

NetBIOS-Namensauswertung: NetBIOS-Name zu IP-Adresse, bevor IP in Hardware-Adresse umgewandelt werden kann, zahlreiche Auswertungsmethoden
MS-Auswertungsmethoden: Cache-NameServer(NBNS-WINS)-3x lokale Rundsendung-LMHOSTS-HOSTS (wenn DNS für Windows-Auflösung aktiviert ist)-DNS (Intervalle 5,10,20,40 sec, dann Fehlermeldung)
Rundsendungsauswertung: zuerst Cache-Suche ohne Rundsendung → nicht vorhanden dann Rundsendung → Suche in NetBIOS Tabellen der empfangenden Computer → ARP um Quellhost zu identifizieren und dann Versendung der Namensabfrageantwort
Rundsendungsgrenzen: Vorsicht, bei Routern Weiterleitung von Broadcasts i.d.R. deaktiviert
Name-Server-Auswertung: nach Cache-Durchsuchung, default **3** Versuche vom Client um WINS-Server zu ermitteln → dann Versuch **sekundärer WINS-Server**, jedoch nicht wenn **primärer WINS-Server** Adreßzuordnung verneint → zuletzt wieder via ARP

Knoten zur Namensauswertung:
B-Knoten: Rundsendung, Probleme Netzwerkauslastung und Router
P-Knoten: Peer-to-peer, NBNS, z.B. WINS, Problem Konfiguration der Clients mit NBNS-IP und Betriebsbereitschaft des Servers
M-Knoten: Kombi B und P, gemischt
H-Knoten: Hybrid, Kombi P und B → also nur andere Reihenfolge
Erweiterte MS B-Knoten: Verwendung LMHOSTS, **#PRE** Einträge in LMHOSTS werden in Cache Speicher geladen → Rundsendung wenn immer noch nicht da → dann tats. LMHOSTS
Konfiguration von Knotentypen: **HLM\CCS\Services\Netbt\Parameters**, default ohne WINS erweitert B, default mit WINS H

NBTSTAT:
Nbtstat –n: Auflistung registrierter Namen
Nbtstat –c: Anzeige Cache
Nbtstat –R: Cache wird mit #PRE Einträge aus LMHOSTS neu geladen

LMHOSTS: statische ASCII-Datei zur Auswertung NetBIOS zu IP, Position **\systemroot\System32\Drivers\Etc** (hier auch Beispieldatei **Lmhosts.sam**)
Vordefienierte Schlüsselwörter: # gezeichnet, # in älteren NetBIOS over TCP/IP als Kommentar betrachtet

#PRE	zuerst zu ladende permanente Einträge in Cache
#DOM:domänenname	vereinfacht Domain-Aktivitäten, z.B. Anmeldebest. Über Router
#NOFNR	Vermeidung ausgerichteter Namensabfragen bei alten LAN-MGR-UNIX
#BEGIN_ALTERNATE	redundante Liste alternativer Positionen LMHOST-Dateien
#END_ALTERNATE	
#INCLUDE	Speicherung und Suche in unabhängiger Datei, z.B. zentral gespeichert
#MH	Mehrere Einträge für mehrfach vernetzten Computer

Namensauswertungsprobleme: falsche Einträge in Datei, ungültige IP-Adressen, IP-Duplikate, fehlende Einträge

IX – WINS

WINS: Verzicht auf Rundsendungen bei Auswertung Comp.namen zu IP-Adr, dynamische Datenbank, erweiterter **NBNS**, direkte Requests und Answers von Client-Server, wenn WINS down kann immer noch broadcast verschickt werden
Vorgang: Client registriert sich beim Start, Anforderungen gehen direkt an WINS-Server, Zuordnungen gehen direkt zurück, Kommunikation über UDP-Anschluß **137**
Namensregistrierung: Client besitzt IP des primären und optional des sekundären WINS-Server, dort Registrierung in Datenbank, Registrierungszeitspanne ist TTL (Time To Live)

Namenserneuerung: nur temporäre Registrierung, erstmals Versuch Leaseverlängerung nach Ablauf **1/8** der TTL, dann in **2 Min** Abständen bis **½** der TTL, dann Versuch Registrierung beim sekundären WINS-Server mit gleichem Procedere, dann wieder Wechsel zum primären …
Namensfreigabe: Client ist selbst für Lease-Verwaltung zuständig, Sendung einer Freigabemeldung bei Shutdown z.B., Namensfreigabeantwort mit TTL **0**
Auftreten von doppelten Namen: Bei bereits vorhandenem Namen wird der registrierte Namensbesitzer angezweifelt (**3** Überprüfungsanforderungen in **500 Millisec** Intervall), bei Multihomed-Computern Überprüfung aller IP Adressen
WINS-Server nicht verfügbar: **3** Versuche über **ARP** primären WINS zu finden, dann sekundär, dann broadcast
Standardbetriebsart: WINS-Client (**H-Knoten**), also über Namensserver zuerst, dann Rundsendung
Implementierung des WINS-Dienstes: im Netzwerkverbund nur **1** WINS-Server nötig da gerichtete Datagramme, weiterleitbar über Router, **2** WINS-Server sind **fehlertolerantes Sicherungssystem**
Empfehlungen für WINS-Server: Server kann **pro Minute 1500** Namensregistrierungen und **4500** Namensabfragen, **WINS-Server + Sicherungsserver** für **10000** Clients ausreichend, pro **Prozessor Aufrüstung 25%** mehr Performance, Beschleunigung der Namensregistrierung durch **Deaktivierung der Protokollierung** von Datenbankänderung (bei Systemabsturz jedoch Verlust der letzten Aktualisierungen)
WINS-Voraussetzungen: muß mindestens 1x auf Server vorhanden sein (muß kein Domänencontroller sein), Konfiguration mit IP-Adresse Subnet Mask und Standardgateway (DHCP auch denkbar aber statische Zuweisung besser)
Konfiguration WINS-Server: **Installation**, **statische Zuordnung** für Clients ohne WINS damit Clients in Remote-Netzwerken auch mitkommunizieren können, **WINS-Proxy-Agenten** zur Ermöglichung Namensauswertung Clients ohne WINS, **WINS-Unterstützung auf DHCP-Server**
Clients ohne WINS: statische Zuordnung von IP/NetBIOS-Name (über **Verwaltung-WINSmanager-Zuordnungen-statische Zuordnungen**) und für DHCP-Clients IP reservieren → LMHOSTS so nicht erforderlich
Konfigurationsparameter statischer Zuordnung: Computername (ohne WINS), IP (ohne WINS), Typ (**EINZELN** → eindeutige IP, **GRUPPE** → Rundsendung Namenspakete da kein Eintrag der Ips in WINS-Datenbank, **DOMÄNENNAME** → Zuordnung NetBIOS Namen zu Adressen, max. 25 Adressen und danach Überschreibung einer replizierten Adresse oder der ältesten Registrierung, **INTERNET-GRUPPE** → Gruppierung von Ressourcen, **MEHRFACH VERNETZT** → bis 25. Adressen)
WINS-PROXY-AGENT: bei nicht-WINS-fähigen Clients, er ermöglicht Namensauswertung indem er Rundsendungen und Namensregistrierungen abhört und an WINS-Server weiterleitet, Konfiguration über **\HLM\System\CCS\Services\NetBT\Parameters** und Parameter **EnableProxy** auf **1**, bei Auswertungsversuch zuerst Überprüfung des Cache, muß in jedem Subnet vorhanden sein es sei denn Router können broadcasts weiterleiten (UDP-Anschlüsse **137/138** aktiviert), **maximal 2x** im Subnet, PROXY-AGENT ist ein WINS-CLIENT und KEIN SERVER!
Datenbank-Replikation zwischen WINS-Servern: **Push/Pull Partner**, **PUSH** schickt Meldung über Datenbankänderung an PULL (**PUSH soll sein Server mit schneller Netzwerk-anbindung**), **PULL** fordert neue Datenbankeinträge (Kopien) über Versionsnummernvergleich an (**PULL bei langsamen Verbindungen und verschiedenen Standorten**), Datenbank wird also nicht komplett sondern nur neuer Teil repliziert
4 Replikationsstartarten: Systemstart, festgelegter Zeitraum, Schwellenwert für Anzahl von Registrierungen und Änderungen, Erzwingen im WINS-Manager über Replikationspartner (**JETZT REPLIZIEREN** → Replikationsanforderung wird in Warteschlange aufgenommen)
Intervalleinstellung: über **Replikationsoptionen-Pullpartner konfigurieren**, bei **PUSH-partner AKTUALISIERUNGSZÄHLER**, **PUSH MIT AUSBREITUNG** bewirkt ausgewählte WINS-Server der weiteren Verbreitung der Push-Benacjrichtigungen
Automatische WINS-Replikationspartner: bei unterstütztem Multicasting, Auffindung anderer WINS-Server durch Senden von Multicasting Datagrammen an **224.0.1.24 (sie sind dann automatisch PUSH und PULL)**, default alle **40min**, Zeitspanne zwischen Pullreplikationen alle **2**

Stunden, wenn Router unterstützen kein Multicasting dann nur im lokalen Netz, WINS-Partnerschaften sind defualt deaktiviert, manuelle Deaktivierung über Registry-Key **UseSelfFndPnrs** auf **0** und **McastIntvl** auf **hohen Wert**

WINS-Datenbank: nbtstat -R (löscht Cache) → WINS-Dienst starten → Verwaltung-WINS Manager → Zuordnungen-Datenbank anzeigen → nur von ausgewählten Besitzern anzeigen / Sortierreihenfolge / Filter setzen
Zuordnungselemente: COMPUTER (eindeutiger Name), COMPUTERSTAFFEL (Gruppe, Internetgruppe, mehrfach vernetzt), NetBIOS Name, IP, **A** oder **S** (aktiv dynamisch oder statisch, **X** unter **A** heißt Entfernung aus der Datenbank), Ablaufdatum, Versions-ID (eindeutige Hexadezimalzahl)
Aufräumprozeß starten: freigegebene Einträge und woanders registrierte entfernen
Intervalldauer unter WINS-Server Konfiguration: **Erneuerungsintervall** (default **144h**), **Alterungsintervall** (Zeit zwischen Freigabe und Veralterung, default **144h**), **Alterungszeitüberschreitung** (Zeit zwischen Veralterung und Entfernung, min **24h**, default **144h**), **Überprüfungsintervall** (Aktivitätsüberprüfung und Namensbesitz, default **576h** ~ **24 Tage**)
Anfangsreplikation: Pull Parameter default aktiviert um beim Bootup Einträge anzufordern, beim Push um Datenbankstatus zu informieren
Winscl.exe: dynamische Einträge können aus Datenbank gelöscht werden
Erweiterte Konfiguration: **Migration Ein/Aus** (statische Datensätze des Typ Einzeln oder Mehrfach Vernetzt gelten als dynamisch wenn ein Konflikt auftritt, so können sie sofern nicht mehr gültig überschrieben werden)
Datenbank-Sicherung: nach Festlegung des Sicherungsverzeichnisses automatisch alle **24h**, Registrierungseinträge für WINS-Server unter **HLM\System\CCS\Services\WINS** auch regelmäßig sichern (Schlüssel speichern)
Wiederherstellung: Diensthalt und Dienstneustart (Computer stellt bei Beschädigung Sicherung automatisch wieder her) oder via WINS Manager (**Zuordnungen-Lokale Datenbank wiederherstellen**)
WINS-Datenbankdateien: Wins.mdb, Winstmp.mdb, j50.log, j50.chk
Komprimieren: eigentlich automatisch, sonst jetpack

X – Durchsuchen des IP-Netzwerkverbunds und Domänenfunktionen

Suchdiensttypen: **Hauptsuchdienst** (Hauptliste verfügbarer Server innerhalb Domäne oder Arbeitsgruppe), **Sicherungssuchdienst** (Kopie der Suchliste Hauptsuchdienst), **Domänenhauptsuchdienst** (Synchronisation der Suchlisten sämtlicher Hauptsuchdienste der Domäne in Remote-Netzwerken, nur PDC kann sein)
Zusammenstellungsprozeß: durch Hauptsuchdienst (Liste der Server innerhalb Domäne oder Arbeitsgruppe und Liste der anderen Domänen oder Arbeitsgruppen
Verteilungsprozeß: Auslösung durch Ankündigung des Hauptsuchdienst (regelmäßige Pakete mit Ankündigungen) oder Kopieren der Suchliste vom Hauptsuchdienst zum Sicherungssuchdienst
Suchanfragen des Clients: Kontakt Hauptsuchdienst des Ziel – Hauptsuchdienst schickt Liste der **3** Sicherungssuchdienste zurück – Client fordert Netzwerkressourcenliste von diesen an – Sicherungs-Server antworten mit Liste der Server – Client wählt Server aus und erhält Ressourcenliste
Durchsuchen: NetBIOS-Rundsendungen (daher wichtig WINS oder LMHOSTS verwenden wegen Routern: ermöglicht Durchsuchen der Domänenaktivität über Subnets)
IP-Router-Lösung: Router ist für Weiterleitung von Broadcasts konfiguriert, nicht empfohlen da nicht 100% Konfliktfrei und Verminderung der Netzwerkleistung
WINNT-Lösung: WINS oder LMHOSTS
Durchsuchen mit LMHOSTS: **HAUPTSUCHDIENST**: **130.20.7.80 <Domänenhauptsuchdienst> #PRE #DOM:<Domänenname>**,

DOMÄNENHAUPTSUCHDIENST (muß Einträge der Hauptsuchdienste in Remote-Subnets besitzen)

Domänenfunktionen in einem IP-Netzwerkverbund:
Rundsendungen bei: Durchsuchen, Domänenanmeldung, Änderung Kennwort, Replikation der Benutzerdatenbank durch Domain-Controller
LMHOSTS: nach #DOM Einträgen wird gesucht, diese sollten beim Client auch alle Remote-DomCon enthalten, so Anmeldung auch wenn lokale PDC/BDC offline, wenn keine lokalen DomCon ist #DOM zum Anmelden erforderlich, Nicht-WINS-PDC muß #DOM Einträge für seine BDCs haben

XI – Host-Namensauswertung

Namensschemata: unterschiedliche Namensschemata bei **NT** (NetBIOS-Name erforderlich um MS Netzwerkbefehle zu benutzen) und **UNIX** (benötigen lediglich IP-Adresse)
Host-Name: vom Administrator zugewiesener Alias, Host-Name muß nicht NetBIOS-Name entsprechen, kann beliebigen **256**-Zeichenkette entsprechen, Host kann mehrere Hostnamen besitzen, anstelle IP Adresse, Dienstprogramm **HOSTNAME**
Host-Namensauswertung: Prozedur → **Hostname-IP Adresse-Hardware Adresse**, verschiedene Methoden sind konfigurierbar
Standardmethoden: **Lokaler Hostname**, **HOSTS-Datei**, **DNS-Server**
MS-Methoden: **NBNS**, **Lokale Rundsendung**, **LMHOSTS-Datei**
LMHOSTS-Datei: wird ausschließlich für Remote-Hosts verwendet
HOSTS-Datei: Hostnamen lokal und remote werden IP-Adressen zugeordnet

1. **Schritt HOSTS-Auswertungsvorgang**: Verwendung Hostname → bin ichs selber? → HOST-Datei wenn nicht übereinstimmend → dort gefunden dann Wandlung in IP oder Suche in anderen Auswertungsmethoden
2. **Schritt IP to MAC**: Zielhost lokal dann ARP durchsucht ARP-Cache oder Zielhost broadcastet, Zielhost remote dann ermittelt ARP Router

DNS: zentrale Online-Datenbank, FQDN zu IP Umwandlung, UNIX-Umgebungen

1. **Schritt**: DNS sucht in Datenbank→ keine Antwort dann in Intervallen 5,10,20,40,5,10,20 sec Wiederholung → keine anderen Auswertungsmethoden mit Erfolg, dann bricht Prozeß ab
2. **Schritt**: s.o.

MS-Methoden für Namensauswertung: [bei voller Konfiguration mit LMHOSTS und NBNS]: Überprüfung ob eigene IP (dann keine Netzwerkaktivität) → HOSTS-Datei (muß im lokalen System sein) → DNS-Server (evtl. Intervallregelung) → lokaler NetBIOS-Namens-Cache → 3 Versuche NBNS kontaktieren → 3 Rundsendungen im lokalen Netzwerk → lokale LMHOSTS-Datei → alles fehlgeschlagen? Letzte Möglichkeit zu kommunizieren ist indem ich die IP-Adresse des Zielhosts kenne
HOSTS-Datei: statisch, ist mit der UNIX-Datei kompatibel, Eintrag besteht aus IP-Adresse, die einem oder mehreren Host-Namen entspricht, default-Eintrag **lokaler_host**, **\systemroot\System32\Drivers\Etc**, Eintrag max. **255** Zeichen, nicht case sensitive

XII – DNS

DNS: verteilte Datenbank, hierarchische Client/Server-Namensstruktur, Ausführung auf Anwendungsschicht, benutzt UDP und TCP
3 Komponeneten: Resolver, Namens-Server, Domänennamensraum
Resolver: DNS-Client, leitet Anfragen an Namens-Server weiter, häufig in Anwendung integriert oder wird im Host-Computer als Bibliotheks-Routine ausgeführt
Namens-Server: sind in Gruppen auf verschiedenen Ebenen zusammengefaßt (sog. Domänen)
Domänennamensraum: Baumstruktur

1. **Domäne der Stammebene**: obere Hierarchiestufe, verwendet Null-Bezeichnung, mit Hilfe eines Punktes (**.**) kann auf diese Domäne verwiesen werden
2. **Domänen der oberen Hierarchieebene**: com, edu, gov … (können Domänen der unteren Hierarchieebene und hosts umfassen)
3. **Domänen der unteren Hierarchieebene**: Hosts und Teildomänen

Autoritätszone: Teil des Domänennamensraum, für den ein bestimmter Namens-Server zuständig ist, dieser speichert Adreßzuordnungen für die Zone, umfaßt mindestens 1 Domäne, Verweis auf diese als „**Stammdomäne**“, um Verwaltung der Zonen-datei auf verschiedene Gruppen zu ermöglichen, effizientere Datenreplikation, Zonen sind um bestimmten **Domänenknoten** angeordnet, dieser heißt **Stammdomäne der Zone**
Primäre Namens-Server: Zonendaten über lokale Dateien, Änderungen
Sekundäre Namens-Server: erhält Daten von weiteren Namens-Servern im Netzwerk, der für diese Zone autorisiert ist, **Zonentransfer** (Weiterleiten der Zoneninfos)
→ Infos über Zonen in seperaten Dateien, Server kann also primär für bestimmte Zonen und sekundär für andere sein
3 Gründe für sekundäre Namensserver: REDUNDANZ, SCHNELLERER ZUGRIFF FÜR REMOTE-STANDORTE, REDUZIERUNG DER DATENLAST
Master-Namensserver: Quelle, aus der ein sekundärer Namensserver in der DNS-Hierarchie die Zoneninformationen erhält
Nur-Cache-Server: Funktion nur Abfragen durchführen, Antworten speichern, Ergebnisse zurückgeben, verfügt nicht über Domänenautorität (Zonendaten werden nicht lokal gespeichert), keine Weiterleitung von Zonentransfer !!!

Namensauswertung:
3 **Abfragetypen**: **REKURSIV** (Server wird aufgefordert Erfolg oder Mißererfolg zu melden, kein Verweis), **ITERATIV** (Rückgabe der am ehesten geeigneten Antwort, ausgewerteter Name oder Verweis), **INVERS** (keine Entsprechung im DNS-Namensraum → also Suche in allen Domänen, spezielle Domäne **in-add.arpa**, Verwaltung untergeordneter Elemente wird in dieser Domäne Orgnisationen übertragen)
in-add.arpa: Spezielle Ressourceneinträge, **Pointer-Einträge** (**PTR**),
z.B. für 51.200.55.157.in-add.arpa um Host-Name für 157.55.200.51 zu ermitteln
Zwischenspeicher: Administrator legt TTL fest

4 **Konfigurationsdateien für typischen DNS-Server**: Datenbankdatei, Reverse-Lookup-Datei, Cache-Datei, Boot-Datei
Datenbankdatei: [zone.dns, z.B. microsoft.com.dns] wird zwischen Master-Namensserver und sekundären Namens-Servern repliziert
Ressourceneinträge: **SOA** (State of Authority, allgemeine Parameter DNS-Zone, erster Eintrag), **NS** (zusätzliche Namensserver), **HOST** (**A**-Eintrag für Zuordnungen), **CNAME** (Canonical Name, IP hat mehrere Hosts, Zuweisen von Aliasnamen)
SOA-Zeichendefinition: @ (dieser Server), **IN** (Internet-Eintrag), Hostname ohne Punkt am Ende wird Stammdomäne angehängt, Email @ wird durch Punkt ersetzt, Zeilenumbrüche in Klammern setzen
Reverse-Lookup-Datei: **z.y.x.w.in-addr.arpa**, wichtig für Sicherheitskonfigurationen
Cache-Datei: **cache.dns**, Einträge des Stamm-Domänen-Servers, grundsätzlich auf allen Namensservern identisch, muß vorhanden sein, für Abfragen außerhalb der Zone und autorisierten Domäne, für INTRANETS ohne INTERNET muß diese Datei abgeändert werden
Boot-Datei: Überwachung des Startvorgangs des DNS-Server (Berkeley Internet Name Daemon BIND-spezifische Implementierung ohne Benutzung des NT DNS-Managers
Boot-Datei-Befehle: directory (weitere angegebene Daten), cache (Cache-Datei), primary (für Namensserver autorisierte Domäne und Datenbankdatei), secondary (IP-Adressen von Master-Servern)

XIII – Implementieren von DNS

NSLOOKUP: primäres Diagnosetool für DNS, ermöglicht Interaktion mit DNS-Server, 2 Modi (interaktiv (~ mehrere Infos) und nicht interaktiv (~ einzige Info))
2 **Verwaltungsmöglichkeiten des DNS-Servers**: DNS-Manager oder manuelle Bearbeitung der DNS-Konfigurationsdateien
Konfiguration der Server-Eigenschaften: default nur-Zwischenspeicher, Schnittstellen, Forwarder, Boot-Methode

Manuelle Konfiguration: **\systemroot\System32\Dns**

Integration von DNS und WINS:
DNS: statisch, manuelle Aktualisierung, hierarchisches Modell, Aufteilung in Zonen, Datenbankdatei kann WINS-Eintrag enthalten und mit WINS so kooperieren
WINS: dynamische Registrierung, einfacher Namensraum mit Replikation
Aktivieren von WINS-Lookup: über **DNS-Manager-Eigenschaften der Zone-Register WINS Lookup-WINS Auswertung verwenden-IP WINS-Server**
WINS Reverse-Lookup: **WINS-R-Eintrag** im Zonenstamm
WINS-TTL: Zeitlimit für Zwischenspeicher default 10 Minuten

XIV – Connectivity in heterogenen Umgebungen

Voraussetzung für Kontaktaufnahme mit Remote-OS: Transporttreiber-Connectivity (TCP/IP, NBF, IPX), SMB-Connectivity (Dateifreigabeprotokoll)
Remote-Host zu NT: z.B. UNIX → **NFS** (transparenter Zugriff, von **Sun Microsystems** entwickelt, verwendet UDP), **FTP**, **SMB-basierte Clients**
Unterstützte Dateisysteme von NFS-Servern: NTFS, FAT, CDFS, HPFS
REXEC: Echtheitsbestätigung auf der Grundlage von Benutzername und Kennwort, Syntax **rexec tcpiphost Befehl**, wird nach Ausführung beendet
RSH: Befehlsausführung auf Servern mit RSH-Daemon, Benutzername muß in der .rhosts Datei auf dem UNIX-Host konfiguriert sein, kein Kennwort, Syntax **rsh Unixhost Befehl**
Telnet: VT100 /VT52 oder TTY-Emulation, Telnet-Server muß konfiguriert sein, ebenfalls Benutzerkonto
RCP: Keine Anmeldung aber Eintrag in .rhosts, benötigt Privilegien für Remote-Ausführung, kein Kennwort, Dateikopierbefehl, Syntax **rcp Host1.Benutzer1:Quelle Host2.Benutzer2:Ziel**
FTP: Übertragung von binären oder Textdateien, FTP-Server geschützt oder über anonyme Verbindung konfigurierbar, Syntax **ftp [Optionen] Host Befehl**
FTP-Befehle: binary (Änderung Datenübertragungstyp zu binär), get (auf lokalen host kopieren), put (hinkopieren), ! (Eingabeaufforderung temporär), quit or bye
TFTP: Trivial File Transfer Protocol, verbindungsloser Dienst über UDP, keine Unterstützung für Echtheitsbestätigung, benötigt weltweit eingerichtete Lese/Schreibberechtigungen auf Remote-System, MS hat nur Client-Software, Syntax **tftp –i get file-one file-two**
Webbrowser: Client im WWW, HTTP-Protokoll, jedes Objekt in einem HTTP-Dokument erfordert eine eigene Verbindung, unterstützen mehrere Datenübertragungsprotokolle wie **FTP**, **Gopher**, **HTTP**, **NNTP**
Druckerdienstprogramme:
Clientdienstprogramme: **LPQ**, **LPD**, kommunizieren mit dem **LPD** (Line Printer Daemon) auf dem Server
LPD: TCP/IP Druckserver, Dienst unter NT (**LPDSVC**), jeder Computer kann so Druckaufträge an NT schicken
Druckdienst-Aufruf: über Systemsteuerung, Eingabeaufforderung, Server-Manager, Registry **HLM\System\CCS\Services\LPDSVC\Parameters**
LPR: Druckanwendung auf dem Client, NT-Client kann Druckaufträge an beliebigen LPD-Server schicken, Druckumgebung Druckmanager oder Befehlszeilenprogramm mit
Syntax **lpr-Sip_Adresse –Pdruckername Dateiname**
LPQ: nach Übermittlung von Druckaufträgen Überprüfung Druckstatus,
Syntax **lpq –Sip_Adresse –PDruckername –l**
Konfigurieren des Druckmanagers mit LPR-Druckmonitor: TCP/IP Druckunterstützung und LPR-kompatiblen Drucker hinzufügen, benötigt LPR-Anschluß (Druckertyp), IP und Druckername
NT als Druckgateway: Empfang von Druckaufträgen von MS-Clients und Weiterleitung an TCP/IP basierten Drucker, Client benötigt kein LPD/TCP/IP, NT kann aber auch von beliebigen LPR-Clients empfangen und an jeden Systemdrucker weiterleiten
Druckerinstallation: Dienste-MS TCP/IP Druckdienst-Neustart, Neuer Drucker-Anschluß hinzufügen-LPR Port …

XV – Implementieren der MS SNMP-Dienste

SNMP: Statusinformationen zwischen verschiedenen Hosts überwachen und austauschen, Teil der TCP/IP Familie, sendet auf Anfrage oder aufgrund bestimmten Ereignis Infos
Unterstützte Plattformen: NT, LAN-Manager-Server, Router und Gateways, Minicomputer und Mainframes, Terminal-Server, Netzwerk-Hubs
SNMP-Verwaltungssystem: holt oder erhält Infos von Agenten, jeder Computer der SNMP-Verwaltungs-Software ausführt
3 Verwaltunsoperationen: **get** (Anforderung bestimmter Wert, z.B. Festplattenspeicher), **get-next** (nächster Wert, Umkehrung von speziellen Objekttabellen), **set** (Wertänderung)
SNMP-Agent: jeder Computer, der SNMP-Agenten-Software ausführt (Dienst ~ Agent), normalerweise Server oder Router
Trap: von Agenten eingeleitete Operation, z.B. Kennwortverstoß
MS-SNMP-Dienst: Verarbeitung von Statusinfos, Information, Installation und Einsatz auf jedem NT-Rechner, Aktivieren von Datenquellen zur Leistungsüberwachung mit Hilfe des **Systemmonitors**
SNMP-Architektur: über Windows-Sockets-API, UDP-Anschluß 161, DLLs zur Unterstützung von **MIBs** (Management Information Bases, **Win32-SNMP-Verwaltungs-API** für Drittanbieter-Entwicklung
Unterstützte MIBs: **Internet MIB II** (Definition **171** Objekte Fehler- und Konfigurationsanalyse), **LAN Manager MIB II** (**90** Objekte mit Statistik-, Freigabe-, Sitzungs-, Benutzer- und Anmeldeinfos), **DHCP MIB** (dhcpmib.dll wird automatisch bei Serverinstall installiert, **14** Objekte), **WINS MIB** (winsmib.dll, **70** Objekte)
Hierarchischer Namensbaum: für MIB-Objekte, Objekte haben globale eindeutige Namen, Objektbezeichnungen werden durch Punkte getrennt, Unternehmen erhalten eigene Definitionsrechte
SNMP-Communities: Gruppe von Hosts, die SNMP-Dienst ausführen, zugewiesener Community-Name, SNMP-Agent kann Mitglied bei mehreren Communities sein, nur Agenten und Manager der gleichen Community können kommunizieren, default-Name **Public**
Informationssammlung: SNMP-Verwaltungssystem schickt Anforderung an Agent-Überprüfung Community-Weiterleitung an DLL-DLL gibt Infos an Agenten-SNMP-Paket an SNMP-Manager
Installation SNMP-Dienst: Parameter-Festlegung **Send-Trap** (Community-Name) und **Trap-Destination** (Name/IP-Adresse Host)
SNMP-Sicherheitsdienst: für Agenten, Echtheitsbestätigungs-Trap (optional), angenommene Communities, von jedem Host SNMP-Pakete annehmen, nur von diesen Hosts SNMP-Pakete annehmen
Konfiguration SNMP-Agentendienst: SNMP Eigenschaften-Register Agent-Kontakt/Standort/Dienst auswählen
Dienstetypen: **physisch** (NT verwaltet physische Geräte), **Sicherung/Subnet** (Bridgeverwaltung), **Internet** (NT ist IP-Gateway/Router), **Ende-zu-Ende** (IP-Host, default gewählt), **Anwendung** (TCP/IP Anwendungen, default gewählt)
Fehlererkennung beim SNMP-Dienst: Dokumentation im Systemprotokoll der Ereignisanzeige
SNMPUTIL-Dienstprogramm: Überprüfung ob SNMP fehlerfrei für Kommunikation mit Verwaltungssystem arbeitet, Syntax **snmputil Befehl Agent Community Objektbezeichner**

XVI – Fehlerbehebung bei MS TCP/IP

Problemursachen: Konfiguration (Host/Dienstinitialisierung), IP-Adressierung (Verbindung), Subnetting (Ping)…
Dienstprogramme: PING (Konfigurationstest), ARP (ungültige Einträge im ARP-Speicher), NETSTAT (Protokollstatistiken), NBTSTAT (Aktualisieren LMHOST-Speicher), IPCONFIG (Konfiguration inkl. DHCP und WINS), TRACERT (Routenüberprüfung), ROUTE (Routing-Tabelle), NSLOOKUP (DNS), SNMP (statistische Infos an Management System), Ereignisprotokoll, Systemmonitor (Leistung und Engpaß), Netzwerkmonitor (Sammeln von Daten und Analyse), Registrierungseditor
Richtlinien zur Fehlerbehebung: **von unterer zu oberer Ebene der Protokollfamilie**, **PING Test** (Loopback-eigene IP-Standard Gateway-Remote Host), **Sitzungsaufbau mit Host** (NET

USE-Ziel Host NetBIOS?-Bereichs-ID identisch?-Verwendung richtiger NetBIOS Name?-WINS/LMHOSTS haben richtigen Eintrag?) → **Windows Sockets Sitzungsaufbau** (Daemon konfiguriert?-Berechtigungen auf Host?-HOSTS/DNS haben gültigen Eintrag?)