Internet Control Message Protocol (ICMP)

- Zur Übertragung von Fehlermeldungen oder Informationsautausch auf Internet Layer
  - Time to live hat den Wert 0 erreicht
  - Host möchte testen ob ein anderer "up" ist
- Meldungen werden in IP Paketen gekapselt (wird zum Network Layer gezählt)
- Gebräuchliche Meldungstypen:

ICMP-Typ	Bedeutung (Fehler)	ICMP-Typ
3	Destination Unreachable	IP Paket kann vom Router nicht zugestellt werden..
4	Source Quench	Pufferspeicher des Routers voll, Pakete werden verworden, senderate soll gedrosselt v
5	Redirect	Hinweis das ein Paket direkt an den Zielhost gesendet werden kann.
11	Time Exceeded	Time to Live abgelaufen oder fragment. Paket kann nicht innerhalb nützlicher Zeit reassem
12	Parameter Problem	IP-Header enthält ungültige Parameter
	Bedeutung (Info)	
0	Echo Reply	Anwort auf Echo (Echo Reply), gleiche Daten wie Echo
8	Echo	Echo Request
13	Timestamp	Wie ein Echo, aber mit zusätzlicher Zeit. (32-Bit Wert, Millisekunden seit Mitternach
14	Timestand Reply	Timestamp Reply

- Destination Unreachable Codes:
  - 0 = net unreachable
  - 1 = host unreachable
  - 2 = protocol unreachable
  - 3 = port unreachable
  - 4 = fragmentation needed an DF set
  - 5 = source route failed

Trace Route Programm

Erlaubt den Weg zu einem Zielhost (oder fehlerhafter Router auf dem weg) zu finden.

- Man sendet UDP Datagramme an den Zielhost; wobei eine hohe Portnummer zufällig gewählt wird (default: 33434)
- Das erste Datagramm wird mit TLL=1 gesendet, der erste Router setzt TTL auf 0, verwirft das IP Paket und sendet eine Time Exceeded ICMP Message zurück, erster Router ist bekannt.
- Das gleiche mit TTL=2 und so weiter.

Um die Entfernung zu bestimmen, wird zugleich die Round-Trip Zeit gemessen.

IPv6

- 32-Bit Adressen zu kurz (IPv4)
- IPv6: 64-Bit Network- und 64-Bit Host-Nummer
- Header Format: ziemlich verändert, von 20 auf 40 Bytes gewachsen
- Mehrere Header: ein Header kann auf den nächsten zeigen (sog. extensions)
- Video-/Audiounterstützung: Flow-Label im Header

Transport Layer

- Stellt den Applikationen eine geeignete Ende-zu-Ende Qualität für Datenübertragung zu Verfügung.
  - UDP: gibt die Eigenschaften von IP fast unverändert weiter: Verbindungslos, Unzuverlässig
  - TCP: zusätzliche Funktionen: Verbindungsorientiert, zuverlässig.
- Bildet Schnittstelle zwischen Betriebssystem (Kernel Space) und Anwendungen (User Space).
- Der Zugriff auf die Funktionen des Transport Layers erfolgt via einer klar definierten Schnittstelle:
  - TCP/UDP Sockets (Unix/Linux/BSD)
  - WinSock (Windows)
- Kapselung:
  - Applikationsdaten erhalten einen TCP/UDP Header
  - Das Paket wird als User Datagram (UDP) oder Segment/TCP-Nachricht (TCP) bezeichnet
  - Ein Transport Layer Paket wird in ein IP Paket eingefügt

Multiplexing und Demultiplexing

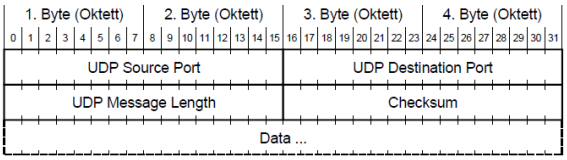
Identifikation eines Hosts über IP Adresse, Identifikation einer Applikation auf einem Host über Port Nummer.

- Multiplexen: Mehrere Kommunikationsbeziehungen zwischen Applikation werden mittels Port Nummern eindeutig bezeichnet.
- Demultiplexen: Verteilen der eingehenden Daten mittels der Port Nummern auf die Applikationen (Es wird zuerst das Type-Feld (ARP/IP/RARP) ausgewertet. Ists IP Type, so wird zwischen ICMP, IGMP, TCP und UDP unterschieden. Aufgrund der Portnummer im TCP- oder UDP Header können die Daten einer Applikation zugeordnet werden.

User Datagram Protocol (UDP)

Dient dem Multiplexen und Demultiplexen der Datagramme auf die Applikation. Verbindungslos und unzuverlässig

Header



- Source Port (16 Bits): Identifiziert sendende Appl. (0 wenn nichts zurückkommen soll)
- Destination Port (16 Bits): Identifiziert Appl. des Empfängers
- Message Length (16 Bits): Länge des UDP Datagramms inkl. Header (in Bytes) (Max. 65535 Bytes)
- Checksum: Prüfsumme über Pseudo-Header, UDP Header und Daten.

Transmission Control Protocol (TCP)

Soll unzuverlässiges IP erweitern um zuverlässigen Datentransport zwischen Applikationen. Netze ROuter und Zielhost sollen nicht überlastet werden.

- Verbindungsorientiert (End-zu-End Dienst, Verbindung ist virtuell: wird nur durch Software hergestellt)
- Zuverlässiger Verbindungsaufbau (beide Endpunkte müssen bestätigen)
- Verbindungsaufbau über 3-Way-Handshake
  - Anfrage Client: Seq=100, Ack=0, SYN
  - Bestätigung Server: Seq=200, Ack=101, SYN/ACK
  - Bestätigung Client: Seq=101, Ack=201, ACK

- Hohe Zuverlässigkeit (Richtige Reihenfolge der Daten ohne Datenverlust)
- Vollduplexübertragung
- Stream-Schnittstelle
- Eleganter Verbindungsabbau (Mit 4 Nachrichten; Zustellung aller Daten auch dann, half Close)
- Übertragung gekapselt in IP Paket (Router leiten weiter, IP-Modul des Empfänger liefert es an das TCP-Modul weiter)
- Umlaufverzögerung (Round Trip Delay) wird laufend gemessen und Wartezeit bis Retransmission entsprechend eingestellt.

Sliding Window

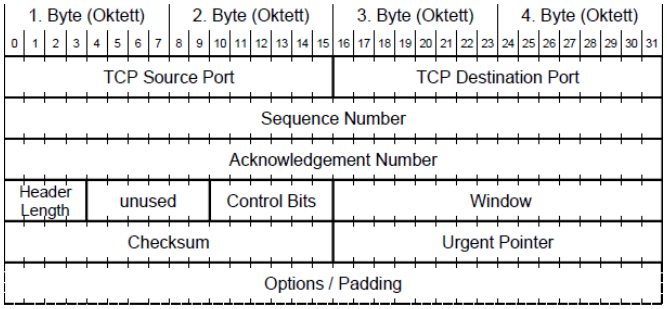
- Sender und Empfänger einigen sich auf fixe Fenstergrösse
- Fenstergrösse = Maximale Paketmenge die ohne bestätigung gesendet werden darf
- Sender speichert jedes Paket bis zur bestätigung
- Die Bestätigung enhält Anzahl offene Bytes im Fenster (Bei 0 gibts später eine erneute bestätigung, das wieder Platz frei ist)

Congestion Control

Überlastungsüberwachung (des Netzwerks), Congestion Window wird vom Sender selbst ermittelt. Das kleinere der beiden Fenster (Congestion oder Sliding) ist ausschlaggebend.

- Slow Start: Algorithmus zum ermitteln der Congestion Window grösse.
- Beginnt mit Maximum Segment Size (MSS=1460 Bytes), bei Bestätigung wirds verdoppelt
- Ab einer bestimmten Schwelle (Threshold, Initial 64 KB) nimmt das Fenster nur noch um 1 MSS zu
- Bei einem Timout wird die Schwelle auf 1/2 des Congestion Window und Congestion Window auf 1 MSS gesetzt.

TCP-Header



- Source/Destination Port (je 16 Bits): Sender- und Empfängerport (bezeichnet Applikation auf Serverseite)
- Sequence Number (32 Bits):