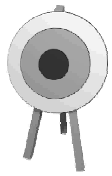
Streaming



Übertragen kontinuierlicher Medien
und Daten im Internet

Inhalt

- Kontinuierliche Datenströme
- Relevante Internetprotokolle
- Jitter- und Fehlerkontrolle

Kontinuierliche Datenströme

- Medien
  - Audio
  - Video
- sonstige kontinuierliche Daten
  - Verhaltenströme: z.B. Positionsänderungen
  - Meßdaten
  - Kontrolldaten (Fernsteuerung)

Eigenschaften von Medienströmen

- Strenge Zeitanforderungen:
  - Pakete, die zu spät ankommen, sind nutzlos
- Toleranz gegenüber Verlusten:
  - abhängig von Medium, Kodierung und menschlicher Wahrnehmung
- Periodizität:
  - z.B. feste Anzahl Bilder pro Sekunde (samples oder frames per second)

Kodierung von Samples

- Unabhänige Samples:
  - z.B. Motion-JPEG
- Differenzen zwischen Samples:
  - z.B. MPEG
  - anfällig gegenüber Verlust von Daten
- Hierarchisch (layered, scalable, sub-band coding)
  - layer 1: Leitsignal, genügt zur Wiedergabe mit geringer Qualität
  - layer k: verbessert Qualität (z.B. höhere frame size oder frame rate)

Beispiel: Probleme bei A/V-Konferenzen

- Verlust von Datenpaketen
- Multitasking in PCs, Workstations bietet keine Echtzeitgarantien
- Akustische und optische Probleme, teils wegen schlechter Hardware

Kontinuierliche Datenströme in verteilten, virtuellen Welten

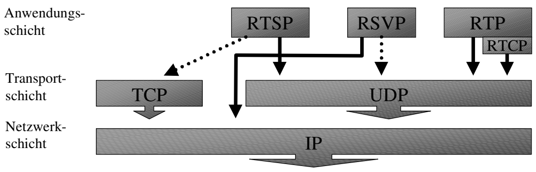
- Gesichter der Avatare = Projektion einer live Videoaufnahme auf das 3D-Objekt
- Audiochat mit 3D-Sound (Entfernung, Schallrichtung)
- Bewegungen des Avatars, etwa gesteuert durch unkonventionelle Eingabegeräte wie bei Peloton

Multicast-Transportdienste

	Zuverlässiges Multicast	**Echtzeit-Multicast**
Eine Quelle 1 zu N	Multicast-FTP Software-Updates	A/V-Konferenzen Internetradio, -kino
Mehrere Quellen N zu M	CSCW Verteiltes Rechnen	DIS Verteilte, virtuelle Welten

Relevante Internetprotokolle

- RSVP: Resource Reservation Protocol
- RTSP: Real-Time Streaming Protocol
- RTP: Real-Time Transport Protocol
  - RTCP: RTP Control Protocol



RSVP: Resource Reservation Protocol

- Erweitertes Multicasting
  - Jeder MRouter hat zusätzliche Datenstrukturen (priority queues)
  - Empfänger schickt Reservation-Nachricht zum Sender
  - An jedem MRouter-Knoten auf dem Weg zum Sender wird die angeforderte Bandbreite reserviert. Falls ein MRouter nicht mehr über genügend Bandbreite verfügt, schickt er Fehlermeldung an Empfänger.

RTSP: Real-Time Streaming Protocol

- Kontrolle mehrerer Sitzungen (Fernbedienung)
- Auswahl des Übertragungskanals (UDP, multicast UDP, TCP)
- Auswahl des Übertragungsmechanismus (z.B. RTP-basiert)

RTSP: Methoden

- DESCRIBE: lade Beschreibung einer Mediums vom Server
- ANNOUNCE: schicke Beschreibung eines Mediums
  - an Server zum Eintragen oder Ändern
  - an Client zum Ändern einer zuvor mit DESCRIBE abgefragten Beschreibung
- SETUP: Client schickt Liste akzeptierbarer Transportmechanismen, Server-Antwort enthält den ausgewählten Mechanismus
- PLAY: an Server, starte Übertragung
- PAUSE: unterbreche Übertragung, Fortsetzung mit PLAY
- TEARDOWN: beende Übertragung, gib Resourcen frei
- REDIRECT: Server teilt Client mit andere URI für dieses Medium zu verwenden
- RECORD: weist Server an Mediendaten aufzuzeichnen

RTP: Real-Time Transport Protocol

RFC 1889

- Übertragung von Daten mit Echtzeitanforderungen an mehrere Teilnehmer
- Separate Ströme für jedes Medium
  - Auswahl durch Empfänger
  - Synchronisation beim Empfänger
- RTP: Verpacken und Transportieren der Daten
- zusätzlich: RTCP = RTP Control Protocol
  - Überwachung (Monitoring) der Übertragungsqualität (quality of service QoS) und
  - primitive Sitzungskontrolle: wer tritt einer Sitzung bei, wer verläßt sie

RTP: Pakete

- Header:
  - payload type (7 Bit)
  - sequence number (16 Bit)
  - timestamp (32 Bit)
  - SSRC = synchronization source ID (32 Bit)
  - CSRC list = contributing sources (je 32 Bit)
- Payload = Daten des Medienstroms
  - etwa Audio- oder Videoframes
- Erweiterbar/Anpaßbar
  - definiere payload type
  - definiere Format für payload

RTCP: Pakettypen

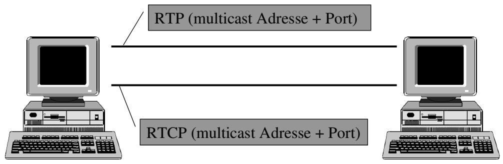
- Pakettypen:
  - SR = sender report
  - RR = receiver report
  - SDES = source description
  - BYE: beende Sitzung
  - APP: anwendungsspezifisch
- Receiver Reports:
  - SSRC
  - Verlust (prozentual)
  - Verlust (gesamt)
  - höchste empfangene Sequenznummer
  - 'interarrival jitter'
  - letzer SR Zeitstempel
  - Verzögerung seit letztem SR Zeitstempel

daraus berechenbar:

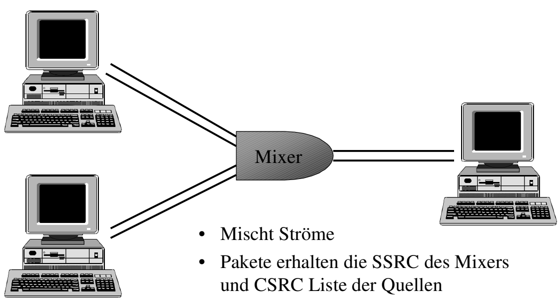
- Transportverzögerung
- Paketverlustrate
- mittlere payload Datenrate
- mittlere Paketrate

- Sender kann Übertragungsmode ändern
- Empfänger kann Probleme erkennen
- Administratoren können Netz evaluieren

RTP/RTCP: einfachstes Szenario

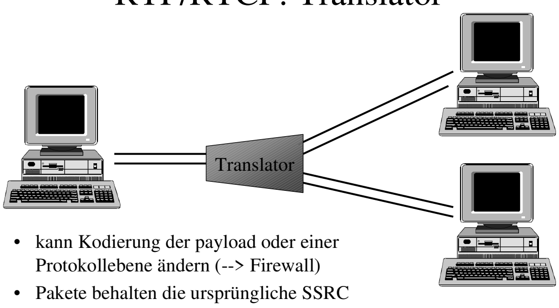


RTP/RTCP: Mixer



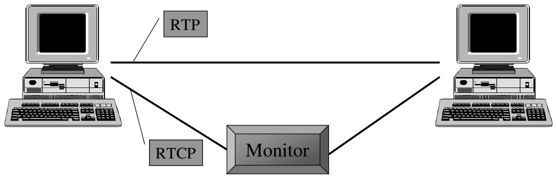
- Mischt Ströme
- Pakete erhalten die SSRC des Mixers und CSRC Liste der Quellen
- Beispiel: mische mehrere Audioströme zur Übertragung über eine Verbindung mit niedrigerer Bandbreite

RTP/RTCP: Translator



- kann Kodierung der payload oder einer Protokollebene ändern (--> Firewall)
- Pakete behalten die ursprüngliche SSRC
- Beispiel: Replikator = ausgehende Verbindungen über Unicast

RTP/RTCP: Monitor



- schätzt die Übertragungsqualität ab, Fehlerdiagnose durch und erstellt Statistiken für Administratoren
- Eigenständige Anwendung
- Teil der übertragenden Anwendungen und versetzt diese in die Lage, Übertragungsmodalitäten anzupassen.

Fehlerbehandlung bei Multicasting

- Fehler = Paketverlust
- Fehlerkorrektur (Error Recovery)
  - Redundanz
  - erneute Übertragung

Daten	# 1'en
0 0	1
0 1	0
1 0	1
1 1	0

- Fehlervertuschen (Error Concealment)
  - Interpolation (z.B. bei Videobildern)
  - Approximation

$\Big\}$ hängt vom Medium ab !

Forward Error Control

- `k` Datenpakete und `h` Paritätspakete
- `h/k` = over-code (Kodierungsoverhead)
- wenn nicht mehr als `h` Pakete der `h+k` Pakete verloren gehen, kann Empfänger die `k` Pakete rekonstruieren.
- Hybride Verfahren:
  - Sende `k` Pakete. Bei Verlust sende `h` Paritätspakete.
  - Sende `k+h` Pakete. Falls Verlust >`h` Pakete schicke `k` Pakete erneut.

Interarrival Jitter

- Mittlere Abweichung der Zeitspannen zwischen den Absendezeitpunkten und den zugehörigen Empfangszeitpunkten zweier Pakete.
- Sei S_i der Sendezeitpunkt des i-ten Pakets und R_i dessen Empfangszeitpunkt

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

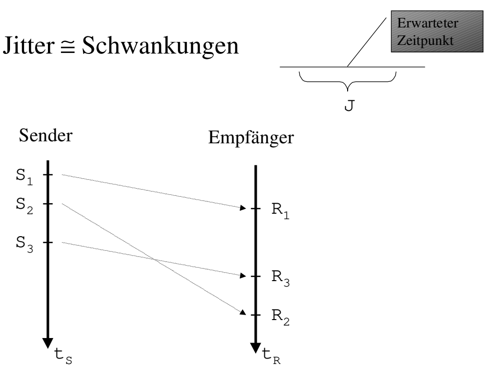
$$J = \frac{\sum_{i=2}^{n} |D(i-1,i)|}{n}$$

$$\tilde{J}(i) = \tilde{J}(i-1) + \frac{|D(i-1,i)| - \tilde{J}(i-1)}{16}$$

Abschätzung zur Übertragungszeit (in RTP)

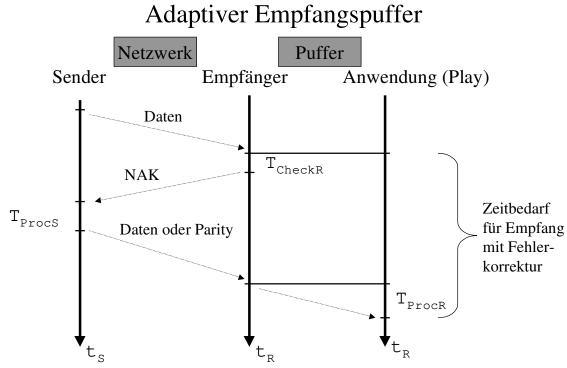
Interarrival Jitter

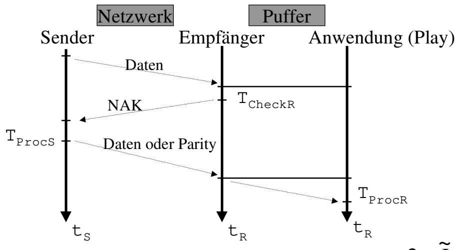
- Jitter ≅ Schwankungen



Kombinierte Jitter- und Fehlerkontrolle

Adaptiver Empfangspuffer





$$T_{Buffer} = T_{CheckR} + T_{NAK} + T_{ProcS} + T_{Parity} + T_{ProcR} + \frac{3 * \tilde{J}}{2}$$

Min. Anzahl Pakete im Puffer: $\frac{T_{Puffer}}{T_{Sample}}$ mit T_{Sample} Sendeintervall

T_{CheckR}, T_{ProcS} und T_{ProcR} sind fix, alle anderen können anhand der Timestamps geschätzt.