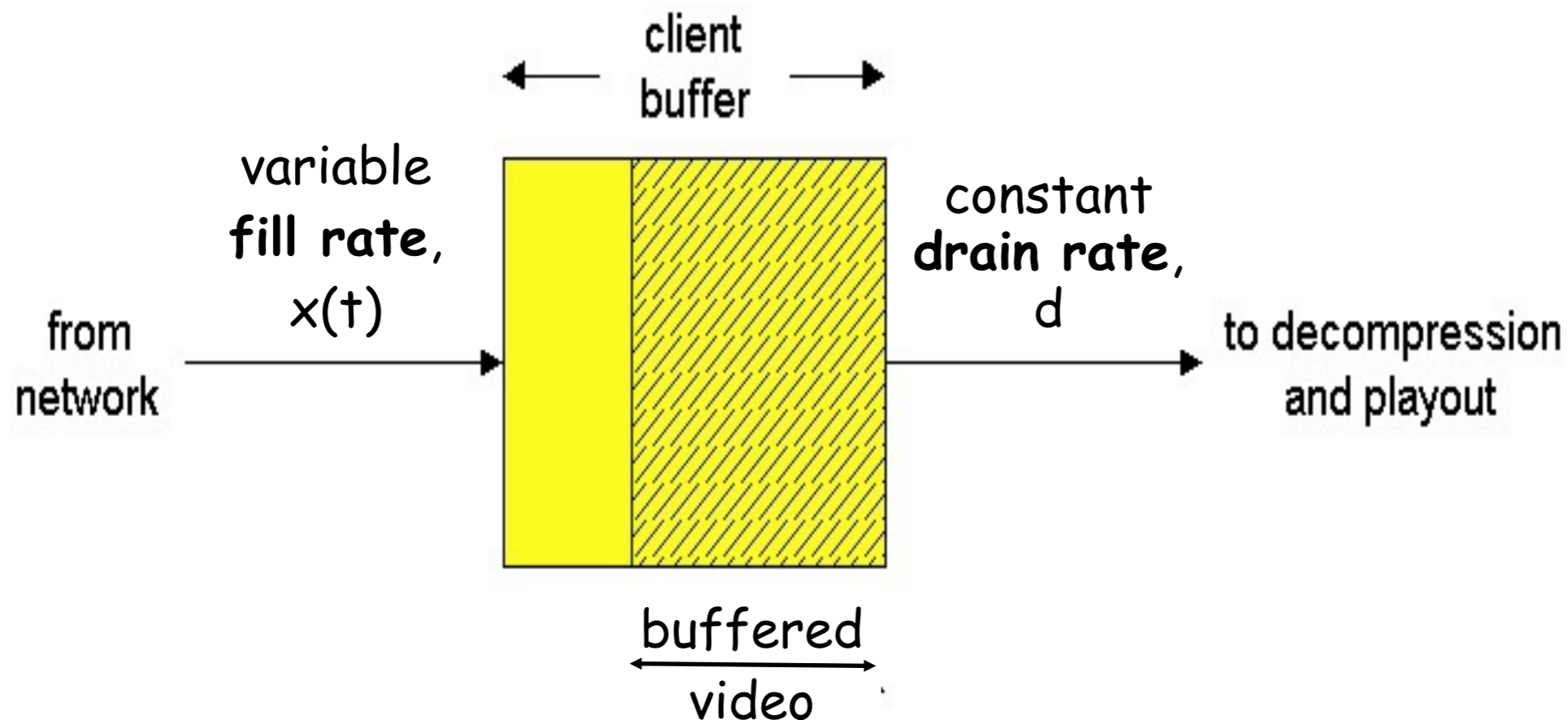


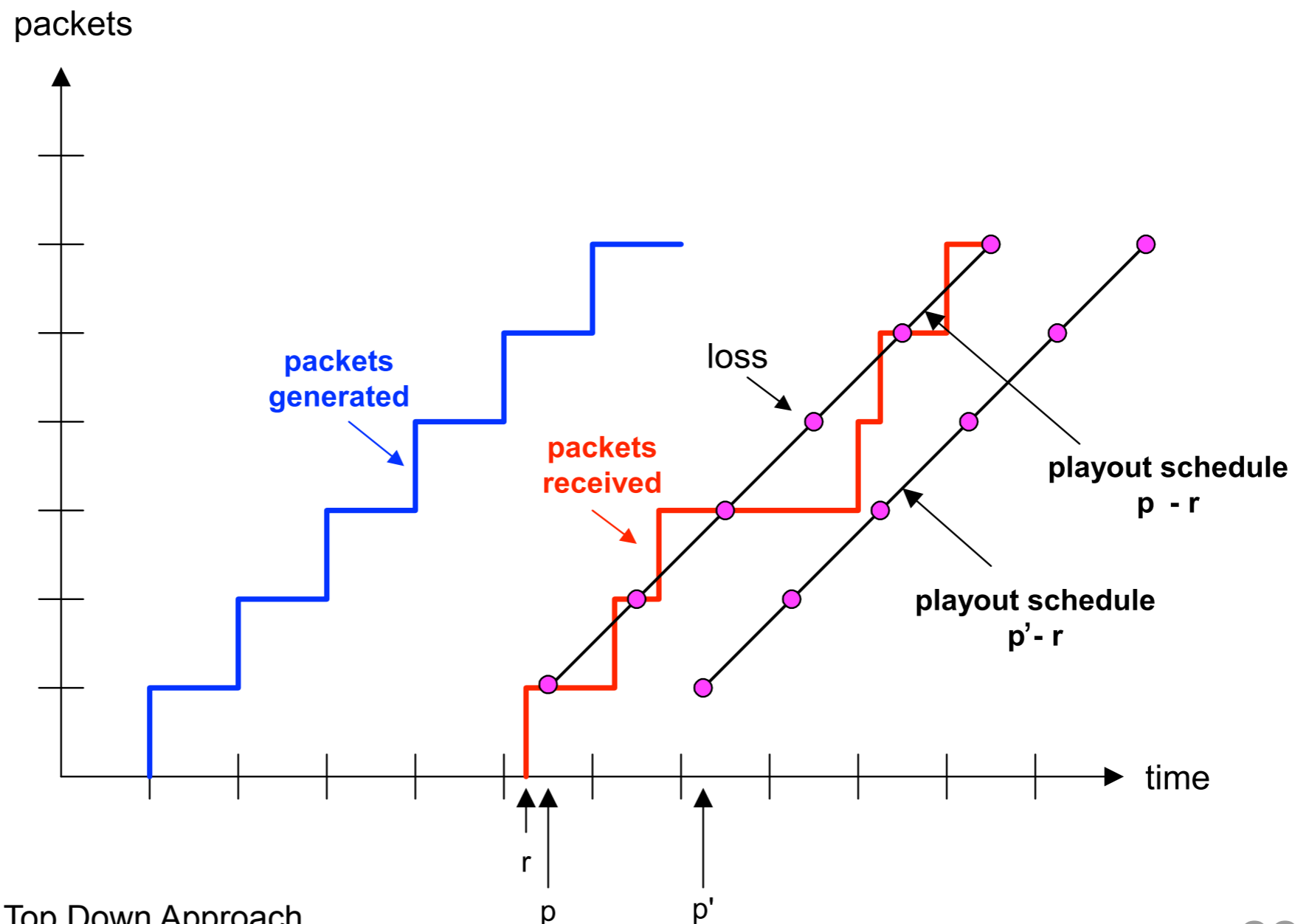
- Puffer auf der Client-Seite
 - Abspielverzögerung kompensiert Netzwerk-Delay und Delay-Jitter



- Puffer auf der Client-Seite
 - Abspielverzögerung kompensiert Netzwerk-Delay und Delay-Jitter

Fixed Playout Delay

- Sender erzeugt regelmäßig Pakete
- erstes Paket kommt zum Zeitpunkt r
- erster Abspiel-Zeitpunkt: p
- zweiter Abspiel-Zeitpunkt: p'

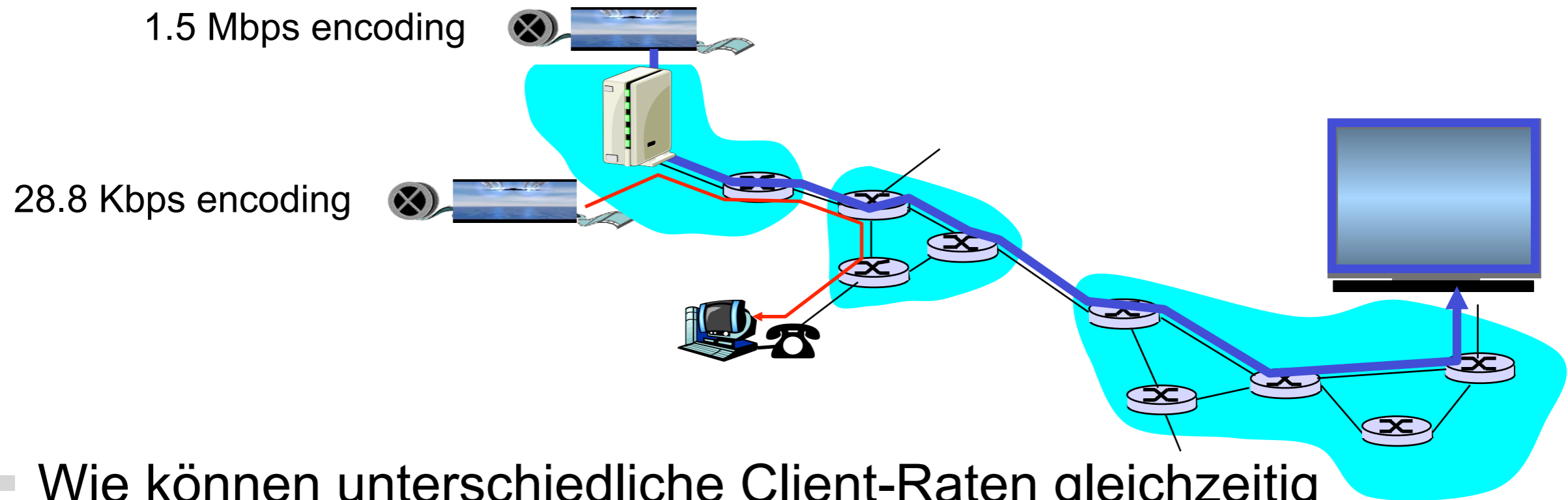


■ UDP

- Server sendet an Client angepasste Datenrate
- unabhängig von der Netzwerklast
 - Senderate = Kodierungs-Rate = konstante Rate
 - Dann ist Fill-Rate = konstante Rate - Paketverlust
- Kurze Verzögerung für Wiedergabe (2-5 Sek.s) um Netzwerk Jitter zu kompensieren

■ TCP

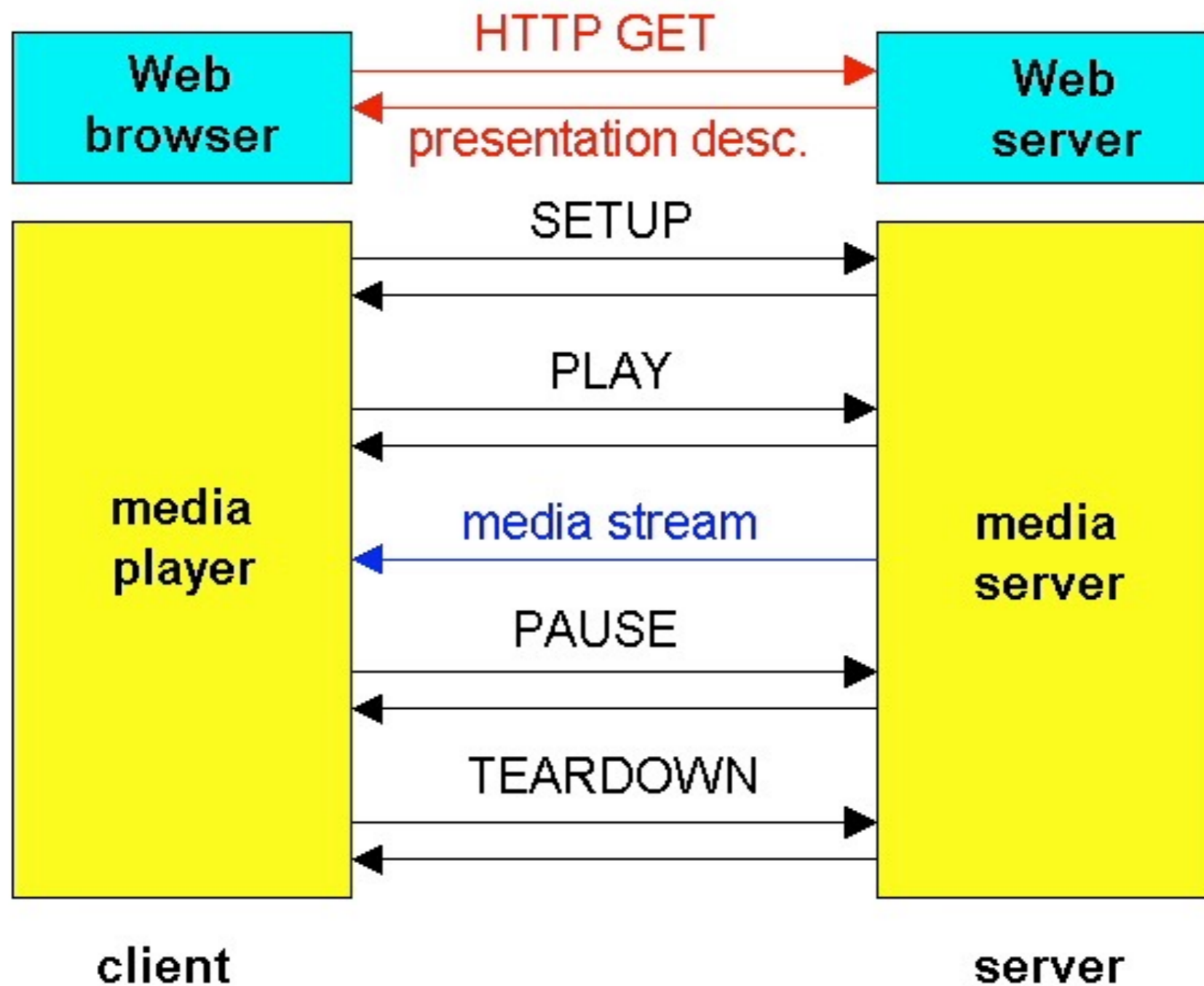
- sendet auf maximal möglicher Datenrate bei TCP
- Fill-Rate verändert sich wegen der TCP-Stauvermeidungsstrategie
- größere Wiedergabe-Verzögerung:
 - gleichmäßig verändernde TCP-Auslieferungsrate
- HTTP/TCP ist für Firewalls durchlässiger



- Wie können unterschiedliche Client-Raten gleichzeitig bedient werden?
 - 28.8 Kbps Modem
 - 100 Mbps Ethernet
- Der Server speichert und überträgt verschiedene Video-Kopien, die sich in der Kodierungsrate unterscheiden

- HTTP
 - zielt nicht auf Multimedia-Inhalte
 - keine Kommandos für Vorspulen, etc.
- RTSP (Real-Time Streaming Protocol): RFC 2326
 - Client-Server-Protokoll in der Anwendungsschicht
 - Benutzerkontrolle:
 - Rückspulen, Vorspulen, Pause, Resume, Neue-Position, etc...
- Was RTSP nicht leistet
 - keine Definition der Audio/Video-Kodierung über das Netzwerk
 - beschreibt nicht den Transport (UDP oder TCP)
 - beschreibt nicht wie der Media-Player Audio und Video puffert

RTSP Operations

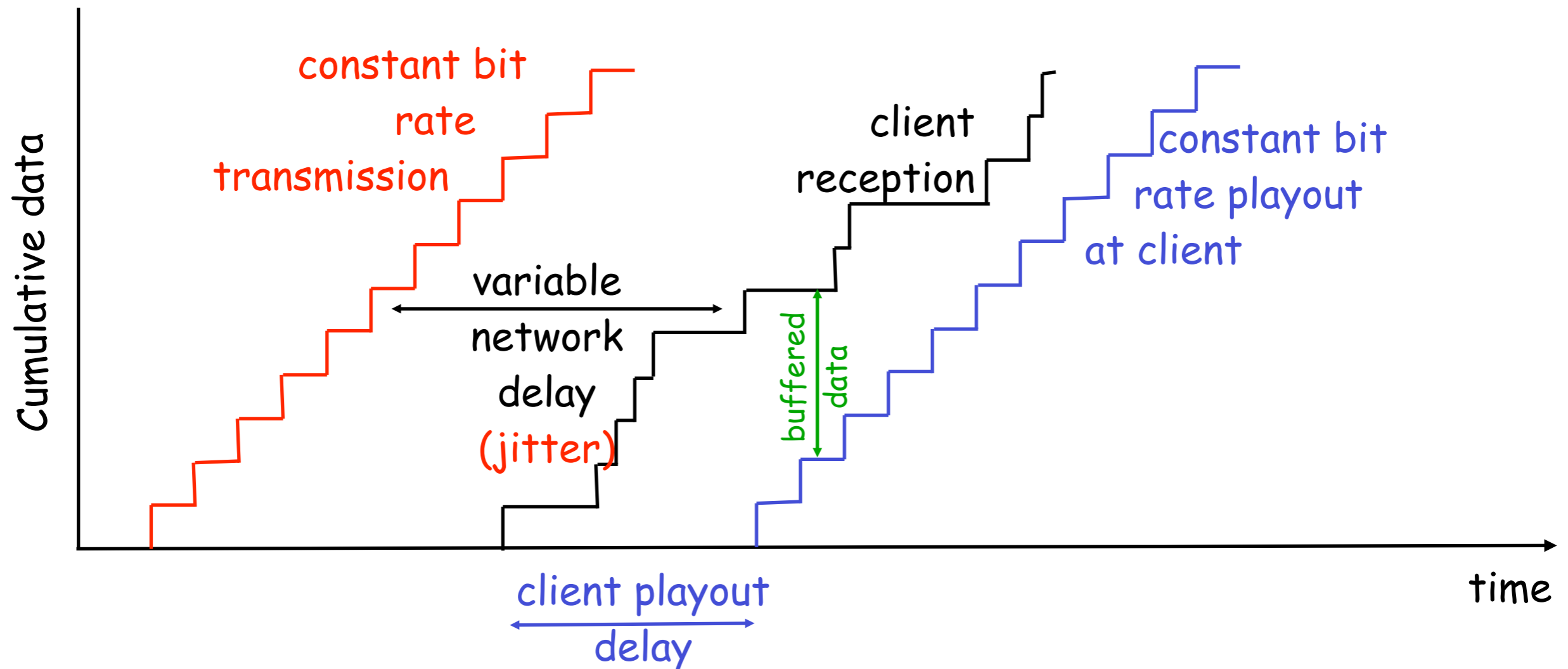


- PC-2-PC phone
 - Skype, iChat, SIP, ...
- PC-2-phone
 - Dialpad, Net2phone, Skype, FaceTime,...
- Video-Konferenz mit Webcams
 - Skype
 - Polycom
 - FaceTime
 - GoogleTalk

- Internet-Telefonie als Beispiel
- Audio: Reden und Schweigen wechseln ab
 - 64 kbps während des Redens
 - Pakete entstehen nur während dessen
 - 20 ms Pakete mit 8 Kbytes/sec: 160 bytes data
- Application Header kommt hinzu
- Daten und Header zusammen in UDP-Paket
- Anwendung sendet UDP Paket alle 20ms während Teilnehmer spricht

- IP-Pakete werden wegen Netzwerkstau verloren
 - Router Puffer läuft über
- Verzögerungs-Verlust (delay loss)
 - IP-Paket kommt zu spät zum Abspielen beim Empfänger an
 - Delays entstehen durch Verarbeitung, Warteschlangen im Netzwerk, Sender und Empfänger-Delays
 - Maximal tolerierter Delay 400 ms
- Verlust-Toleranz
 - hängt von der Stimmkodierung ab
 - Paketverlustraten von 1% bis 10% sind tolerierbar

Internet Phone: Delay Jitter



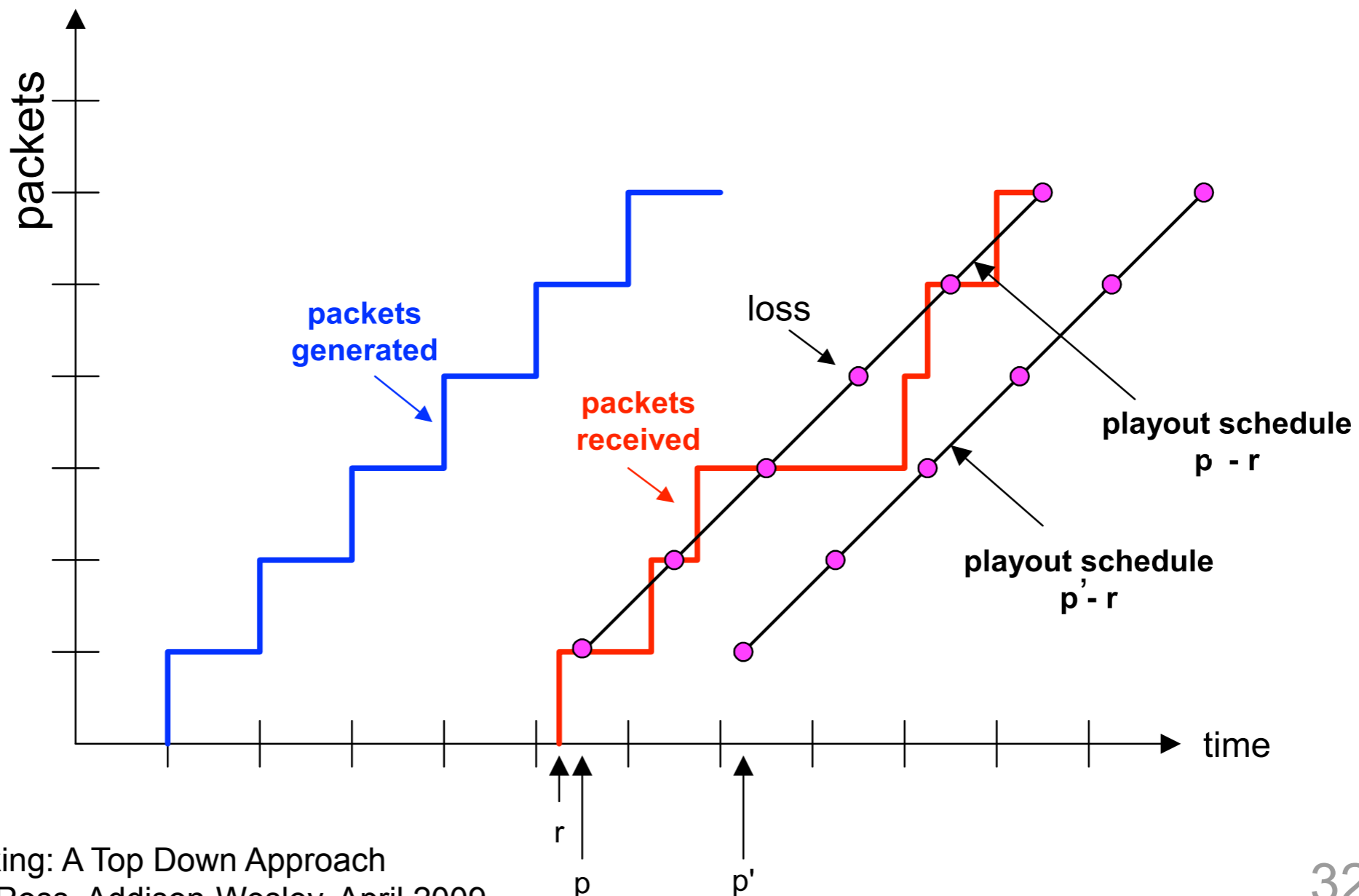
- Gesamtverzögerungen von zwei aufeinander folgenden Pakete kann um 20 ms variieren (transmission time difference)

Internet Phone: Feste Abspielverzögerung

- Empfänger versucht jedes Paket (chunk) genau q ms nach dessen Erzeugung abzuspielen
 - Chunk mit Zeitstempel t wird zum Zeitpunkt $t+q$ abgespielt
 - Chunks, die nach $t+q$ ankommen, werden übersprungen
- Tradeoff für die Wahl von q :
 - Großes q : seltener Paketverlust
 - Kleines q : bessere Interaktion

Internet Phone: Feste Abspielverzögerung (Fixed Playout Delay)

- Sender erzeugt Paket alle 20 ms wenn ein Tonsignal anliegt
- erstes Paket kommt zum Zeitpunkt r
- erster Abspiel-Zeitpunkt: p : 4. Paket wird nicht abgespielt
- zweiter Abspiel-Zeitpunkt: p' : Alle Pakete werden abgespielt



Internet Phone: Adaptive Abspiel-Verzögerung (1)

- Ziel: minimiere Abspielverzögerung und halte Verlustrate niedrig
- Ansatz: Angepasste Abspiel Verzögerung
 - schätze Netzwerk-Verzögerung und passe die Abspielverzögerung nach jeder Sprechphase an
 - Stille wird verkürzt oder verlängert
 - Pakete werden immer noch alle 20 ms während Sprechzeit gesendet
- t_i = Zeitstempel des i -ten Pakets
- r_i = Zeitpunkt, wenn Paket beim Empfänger ankommt
- p_i = Zeitpunkt, wenn Paket abgespielt wird
- $r_i - t_i$ = Netzwerkverzögerung (network delay)
- d_i = Schätzung der Netzwerkverzögerung nach dem Empfangen des i -ten Pakets
 - Dynamischer Schätzwert beim Empfänger:
 - $d_i = (1-u) d_{i-1} + u (r_i - t_i)$
 - für Konstante, z.B. $u = 0,01$

Internet Phone: Adaptive Abspiel-Verzögerung (2)

- v_i = Abschätzung für die mittlere Abweichung
 - $v_i = (1-u) v_{i-1} + u |r_i - t_i - d_i|$
- Für das erste Paket ist die Abspielzeit
 - $p_i = t_i + d_i + K v_i$
 - für eine positive Konstante K
- Die Abspielzeit der anderen Pakete ergibt sich daraus

- Wie erkennt man die Sprechzeit?
 - Ohne Paketverlust ergibt sich das aus den Zeitstempel
 - Falls die Zeitdifferenz folgender Pakete größer ist als 20ms, dann beginnt die Sprechzeit einer neuen Redesequenz
 - Mit Verlust muss der Empfänger auf Zeitstempel und Sequenznummern achten
 - Falls Zeitdifferenz größer ist als 20 ms und Sequenznummern kontinuierlich, dann beginnt die Redesequenz

Verarbeitung von Paketverlusten (1): Forward Error Correction

- für jede Gruppe von n Paketen werden durch Xor ein neues Paket erzeugt
- sende diese $n+1$ Pakete (Bandweite wird um $1/n$ erhöht)
- beliebige n Pakete können benutzt werden um ein verlorenes Paket zu rekonstruieren
- Abspielverzögerung muss groß genug sein um alle $n+1$ Pakete zu erhalten
- Tradeoff:
 - größeres n erzeugt kleineren Bandweitenverlust
 - größeres n erhöht Abspielverzögerung
 - größeres n sorgt dafür dass u.U. mehrere Pakete verloren gehen

z.B. $n=4$:



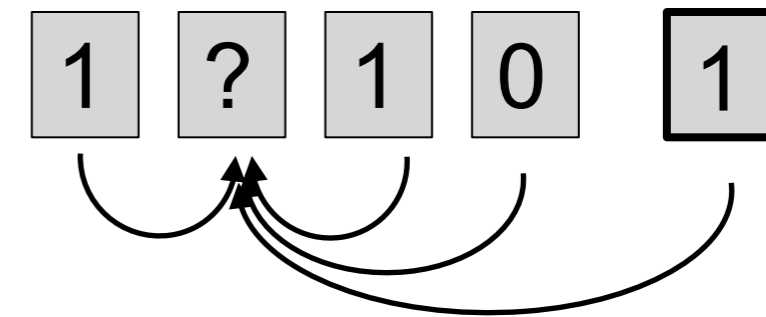
Verarbeitung von Paketverlusten (1): Forward Error Correction

- für jede Gruppe von n Paketen werden durch Xor ein neues Paket erzeugt
- sende diese $n+1$ Pakete (Bandweite wird um $1/n$ erhöht)
- beliebige n Pakete können benutzt werden um ein verlorenes Paket zu rekonstruieren
- Abspielverzögerung muss groß genug sein um alle $n+1$ Pakete zu erhalten
- Tradeoff:
 - größeres n erzeugt kleineren Bandweitenverlust
 - größeres n erhöht Abspielverzögerung
 - größeres n sorgt dafür dass u.U. mehrere Pakete verloren gehen

z.B. $n=4$:

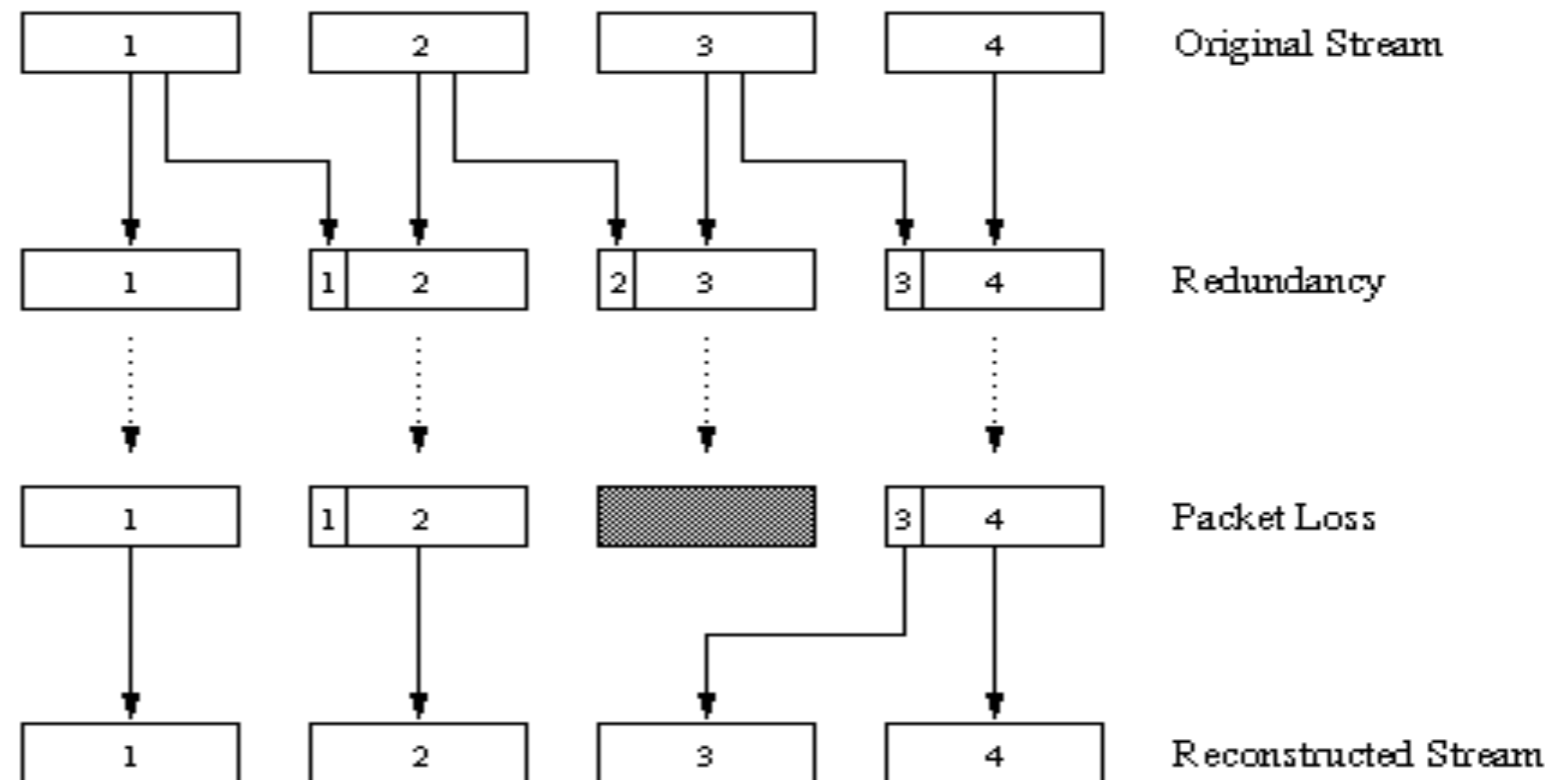


2. Paket verloren?

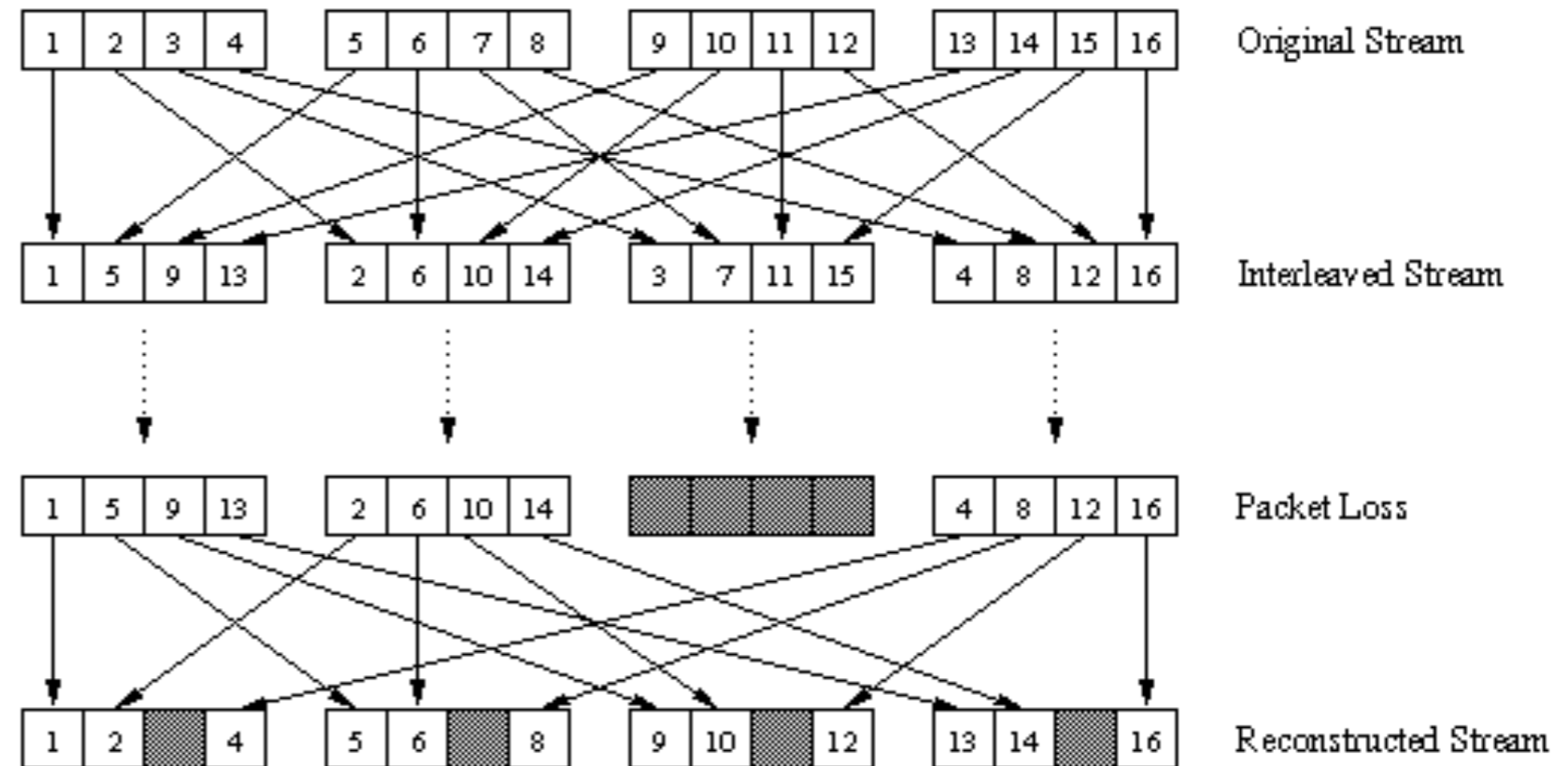


Verarbeitung von Paketverlusten (2): Zweites FEC Schema

- Audio-Strom mit geringerer Qualität wird mit geschickt
- Niedrigere Auflösung liefert redundante Information
 - z.B. nominaler PCM-Strom mit 64 kbps und redundanter Strom mit 13 kbps.
- Sobald es einen nicht zusammenhängenden Verlust gibt, kann der Empfänger den Verlust verbergen



- Pakete werden in kleinere Einheiten unterteilt (z.B. 5ms pro Paket)
- Pakete enthalten kleinere Einheiten von verschiedenen Paketen
- Bei Paketverlust hat man immer noch die meisten Einheiten von jedem Paket
- Keine Erhöhung der Redundanz



- Verwendung von UDP statt TCP um durch Stauvermeidung erzeugte Verzögerungen für zeitnah zu versendende Pakete zu vermeiden
- **Serverseitig** wird die Strom auf die zur Verfügung stehende Bandweite zum Empfänger angepasst
 - Auswahl aus vorkodierten Raten
 - Dynamische Server-Kodierungsrate
- **Empfängerseitig** angepasste Abspielverzögerung um Verzögerung zu kompensieren
- Verlustverarbeitung (Anwendungsschicht wegen UDP)
 - FEC, Interleaving, Fehlerkaschierung
- Content Distribution Networks
 - Inhalt wird in die Nähe des Empfängers gebracht

Systeme II

2. Multimedia

Thomas Janson[°], Kristof Van Laerhoven*, Christian Ortolf[°]

Folien: Christian Schindelbauer[°]

Technische Fakultät

[°]: Rechnernetze und Telematik, *: Eingebettete Systeme

Albert-Ludwigs-Universität Freiburg

Version 29.04.2015