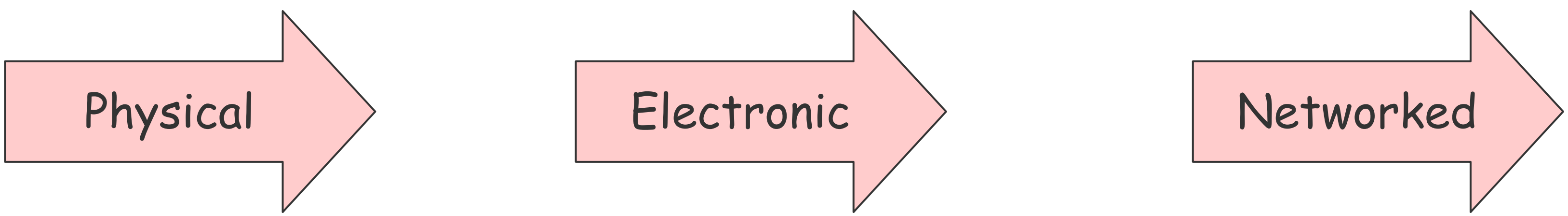
MM Networks - Requirements & Issues

- Distributed Multimedia Applications
- Network Performance Parameters for Multimedia
- Characteristics of Multimedia Traffic Sources
- Factors regarding to Network Performance
- Multimedia Traffic Requirements for Networks
- Quality of Service

Distributed Multimedia Applications

- MM networks facilitate the distribution of MM information among different geographical locations.
- Distributed MM Applications are the current trend:
  - Work in office
  - Board games
  - Library research
  - Store shopping
  - Work on home PC
  - Electronic games
  - CD-ROM research
  - CD-ROM shopping
  - Telecommute/desktop collaboration
  - Multiplayer interactive games
  - Online services research
  - Internet shopping



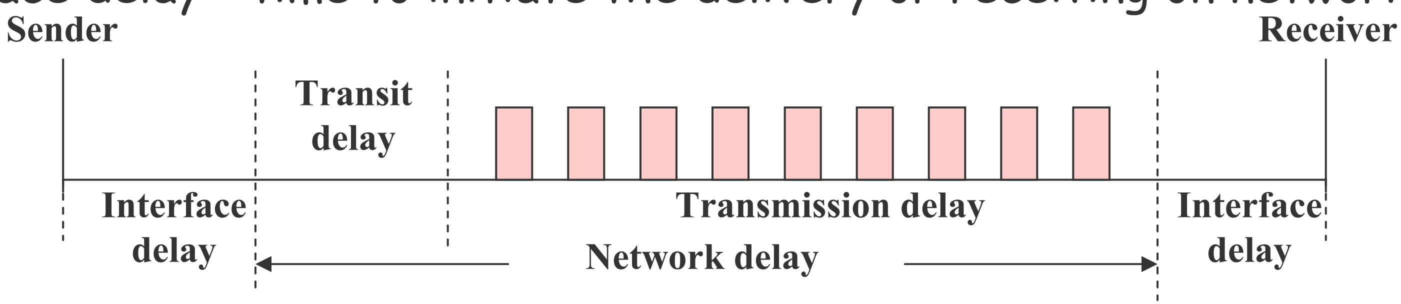
"Content" *"Digitized"* *"Network"*

Multimedia Communications

- Seven layers of the OSI model:
  - Upper layers: Application (7), Presentation (6), Session (5), Transport (4, say, TCP).
  - Lower layers: Network (3, say, Internet Protocol or IP), Data Link (2), Physical (1).
  - Operation modes:
    - Unicast (Peer-to-Peer): this includes individual client-to-server applications, such as home-shopping, online banking, MM e-mail, etc.
    - Multicast (Multi-Peer): like distance learning, MIM (*Multiparty Interactive Multimedia*) from CSCW (*Computer Supported Collaborative Work*), Virtual Café [uncontrolled access].
    - Broadcast (1-to-all): one sender with many receivers, static; only the sender (source) is allowed for information delivery.
- Network Performance Parameters for MM traffics:
  - **Throughput**:
    - Effective bit rate (effective bandwidth)- Physical-link bit rate minus the various transmission overheads. 155.52 Mbps [SONET: ~ 3%, ATM: ~ 9.5%.] to 136 Mbps.
    - Effective frame rate- Highly compressed media streams may result in lower effective frame rate in the presence of erroneous transmissions.
  - **Error Rate**:
    - BER (bit error rate): the ratio of the average number of corrupted bits to the total number of transmitted bits. (10^{-9} to 10^{-12} in most today's networks.)
    - PER (packet error rate): like so, but replacing packets for bits.
    - FER (frame error rate): like so, but replacing frames for bits.
      - ✓ Errors in Audio/Video frames are tolerable, but not in electronic funds.

Other Parameters

- **Delay, End-to-End delay:**
  - Transit delay - the propagation time to send a bit.
    - With light speed, mainly dependent on the distance.
  - Transmission delay - time to transmit a block of data.
    - Dependent on the network bit rate, intermediate nodes' processing delays (routing, buffering).
  - Network delay = Transit delay + Transmission delay.
  - Interface delay - time to initiate the delivery or receiving on networks.



- **Round-trip delay:**
  - The total time required for a sender to send a block of data and receive an ACK, useful to give a better picture of network performance.
- **Delay variation (jitter):** (Non-uniform end-to-end delays among packets.)
  - Video/Audio streams deliver the data packets sequentially. However, most today's packet switch networks cannot ensure their serial reception.
  - The techniques enabling the avoidance of jitters are essential to the distributed MM systems.

Characteristics of MM Traffics

- Throughput variation with time
  - CBR (constant bit rate):
    - CD-ROM (sector size in White Book) applications, ISDN networks.
  - VBR (variable bit rate):
    - A bursty traffic yields varying data rate from time(frame) to time (frame) according to the level of relative activity (scene content + compression).
    - This is often characterized by the ratio of the peak traffic rate over the mean traffic rate over a given period of time.
- Time dependence
  - For video conferencing, the end-to-end delay must be at most 150 ms for participants to be unaware of its effects.
- Bi-directional symmetry
  - Forward and backward channels:
    - Video-on-Demand on cable networks, teleconference.

Network Performance Factors

- The delivery of media streams are determined by the throughput performance of underlying network layers.
- Important factors
  - **Node** or **link failures**: These failures will cause congestion in their immediate vicinity, leading to packet delays & loss, error, losing connectivity.
  - **Congestion**: It results from heavy traffic or bottlenecks on the ill-managed networks. The throughput of the network decreases with increasing load because
    - Networks start to drop packets when node buffers overflow.
    - Management procedures take effect to reduce traffic.
    - Heavily loaded nodes become the bottlenecks.
  - **Bottlenecks**: The inevitable data paths crossing the slow links have the overall bandwidth capacity of the slowest one.
  - **Buffer capacity**: The amount of buffer memory at the end-systems and their network interfaces are limited. Bursty traffics exceeding such limit will result in packet drops.
  - **Flow control**: In the presence of buffer overflow, flow control protocols are often invoked at the end systems to limit the transmission rate.
    - More sophisticated traffic-policing strategies at the higher levels can be employed in junction with the simple flow control protocols.

Reasons for Network Errors

- **Individual bit errors**: Due to noise in the lines or packet switches.
  - Rare in the modern fiber optics networks. Most error detecting codes in switches can detect the presence of a bit error and request the re-transmission.
- **Packet loss**: Mainly due to insufficient buffer space at the receiving ends for the network congestion.
  - Packets lost in transit, or dropped by an intermediate node.
  - The receiving end-system can usually detect and inform the sender.
  - Error detection and correction
    - FEC (Forward Error Correction): Such errors are possibly corrected without resorting to re-transmission.
- **Out-of-order packets**: owing to packet delay variation.
  - The receiver needs to re-arrange the packets arriving out of sequence.
  - In case of greatly delayed, some portion or the entire packet sequence may need the retransmission.
  - Buffers are often used to smooth out the delay variation for the sacrifice of additional buffering delay and space.
  - Allocating buffer resources in advance can improve the quality of received video and audio streams, especially for the random delay.
    - Random delay: packet delay variation mainly comes from the different processing delays in the intermediate nodes (routing/buffering) and interface delay.

MM Traffic Requirements for Networks

- **Throughput requirements**
  - **High Transmission Bandwidth**: MM traffics are expensive in network capacity.
  - **High Storage Bandwidth**: The buffer for stream prefetching demands huge storage bandwidth to sustain the the incoming data stream from the network.
  - **Streaming**: The consumption of the network capacity for delivering media streams will last long enough for their playback duration.
- **Reliability (error control) requirements**
  - Due to the limits of human sensory perception, the error control for MM networks can vary according to the characteristics of given media streams.
  - Dropped packets are more noticeable in a text stream, than an audio stream than in a video stream.
  - The requirements for error control and end-to-end latency are contradictory. (Need careful consideration)
- **Delay requirements**
  - MM data often contain multiple streams of data, such as video and audio streams, each with different delay and reliability requirements.
  - MM applications also need to consider the synchronization of streams' playbacks.

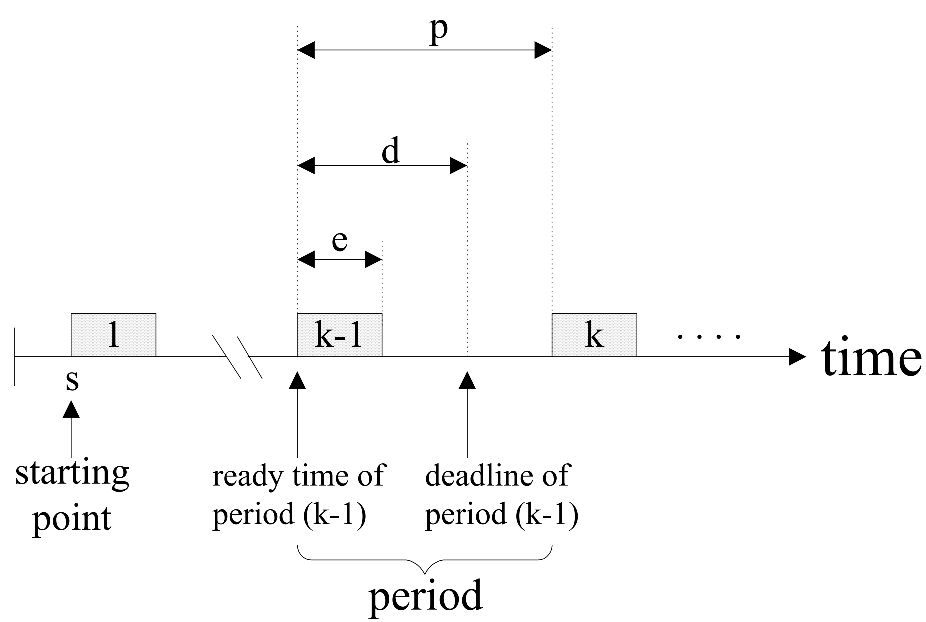
Quality of Service

- QoS indicates how well a network can perform for MM Applications, in terms of QoS parameters.
  - Maximal allowable delay, delay jitter, throughput, error rate, etc.
- New QoS concepts implemented as standard services:
  - Resource Reservation and Scheduling.
    - When an application "knows" in advance the resource requirements, say, a given bandwidth, it can reserve explicitly with the networks.
    - Networks can deny the requests or schedule their allocations.
  - Resource Negotiations.
    - To conserve the resources, the network can negotiate with the requester of other lower QoS parameters.
  - Admission Control.
    - The network can choose not to let some application on to the networks.
  - Guaranteed QoS.
    - It can attract a particular user to its service through advertisement.
    - An accurate estimate from previous experiences can be deducted.

Goals of Resources Scheduling

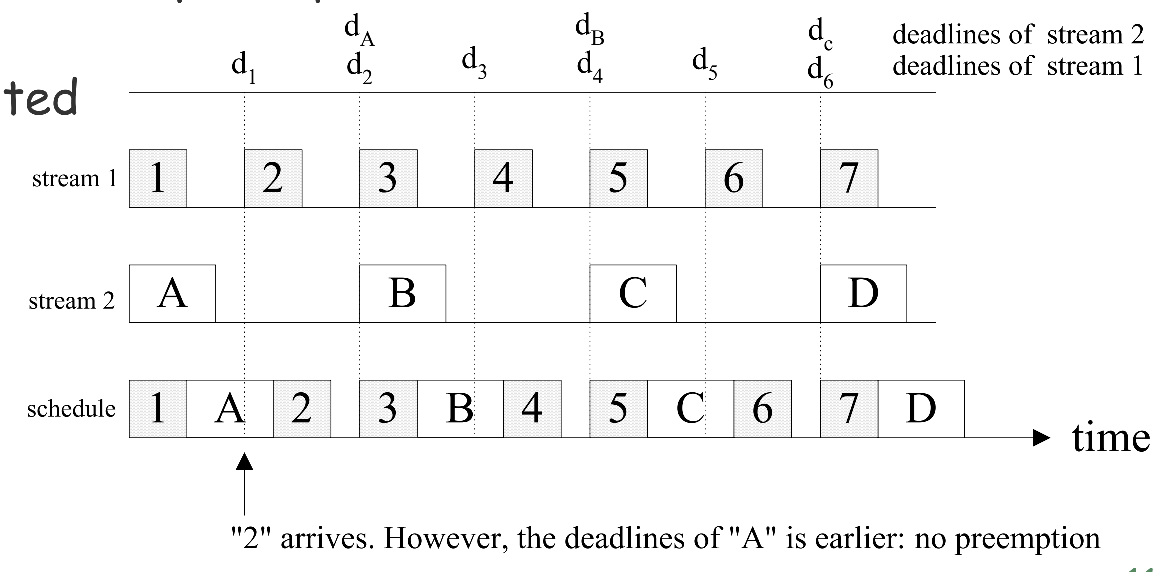
- Traditional Scheduling
  - Optimal Throughput
  - Optimal Resource Utilization
  - Fair Queuing
- Real-Time Scheduling
  - As many tasks meeting their time requirements as possible.

- Characterization of Periodic Tasks
  - The time constraints of a periodic task (traffic flow) are characterized by a 4 -tuple (s, e, d, p)
  - For continuous media tasks, it is assumed that the deadline of the period (*k-1*) is the ready time of period *k*, i.e., $d = p$.



Scheduling Approaches

- Static:
  - A task is scheduled by a static algorithm at the beginning of the task.
  - Subsequently, each task is processed with the priority calculated at the beginning. ⇒ *No further scheduling is required.*
- Dynamic:
  - A dynamic algorithm schedules every instance of each incoming task according to its specific demands.
  - Periodic tasks must be scheduled in each period again. ⇒ *considerable overhead.*
- **Earliest Deadline First (EDF) Algorithm**:
  - At any new arrival, the running task is preempted and the new task is scheduled if its deadline is earlier.
  - The processing of the interrupted task is **continued** according to the EDF algorithm.
  - EDF is an optimal, dynamic algorithm, since if a set of tasks can be scheduled by any dynamic priority assignment, it also can be done by EDF.

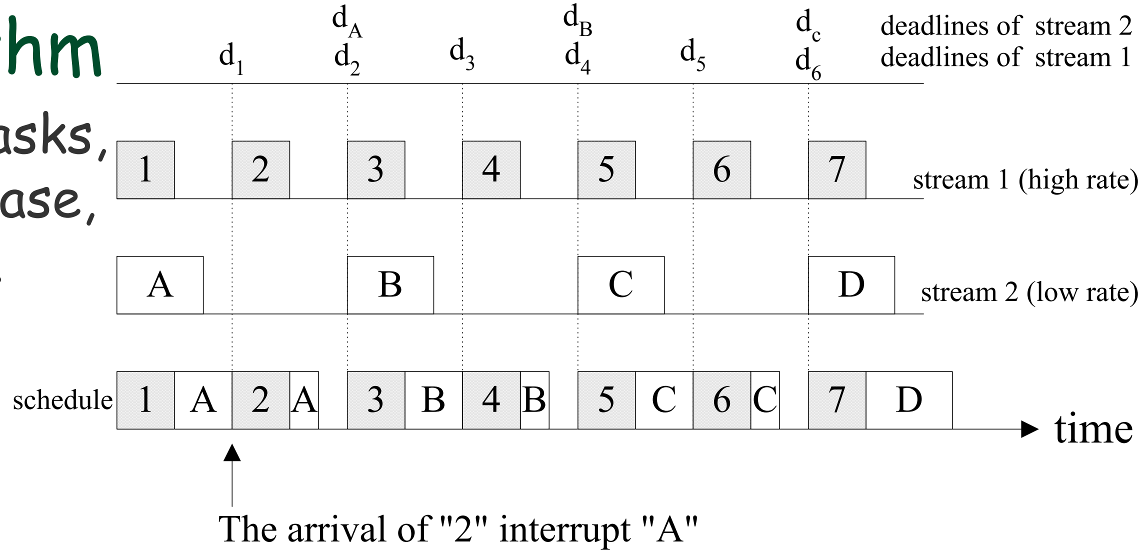


Extensions of EDF

- <u>Time-Driven Scheduler (TDS):</u>
  - Tasks are scheduled according to their deadlines.
  - If an overload situation occurs, the scheduler **aborts** tasks which cannot meet their deadline anymore.
- <u>Priority-Driven EDF:</u>
  - Every task is divided into a **mandatory** and **optional** part.
    - Tasks are scheduled *w.r.t.* the deadline of the ***mandatory*** *parts*.
    - The ***optional*** *parts* are processed if the resource capacity is not fully utilized.

  <u>Notes</u>: This scheme can be used to implement <u>**media scaling**</u>.
    - Each pixel of a monochrome picture is encoded with 16 bits.
    - The processing of the <u>eight most significant</u> bits is mandatory.
    - The processing of the <u>eight least significant</u> bits is optional.

Rate Monotonic Algorithm



- Static priority are assigned to tasks, once at the connection set-up phase, according to their request rates.
- Tasks with higher request rates (e.g., **shorter periods**) will have higher priorities.

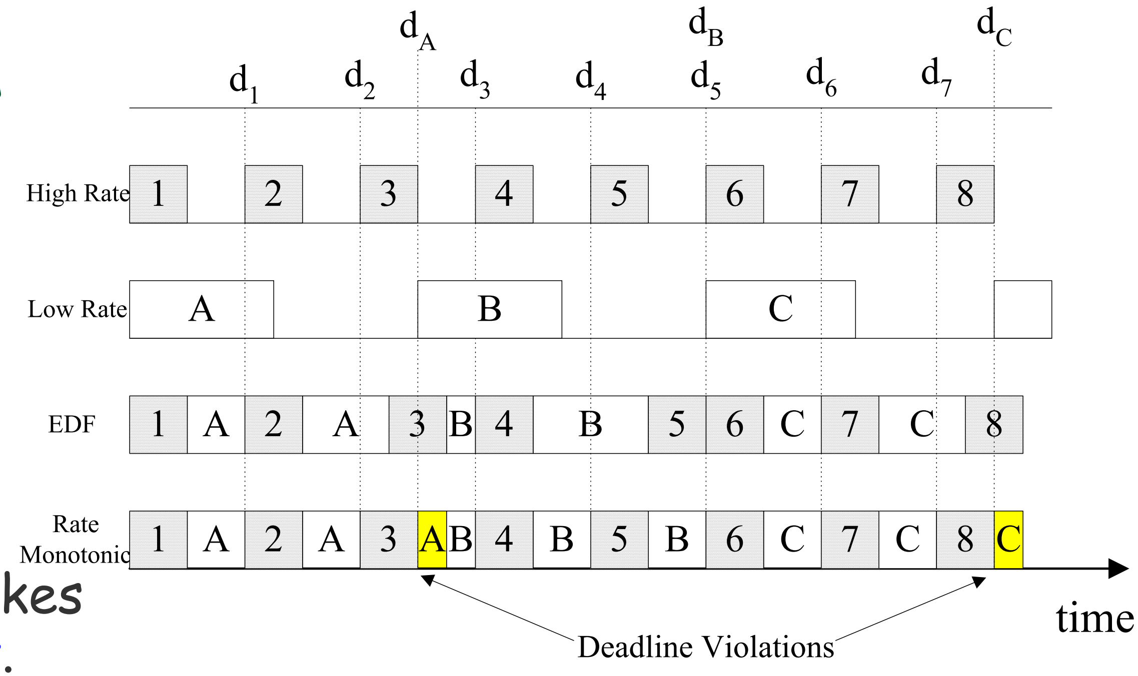
Note: Rate monotonic is an optimal, **static** priority-driven algorithm for preemptive, periodic jobs.

It is optimal since there is no other static algorithm that is able to schedule a task set which cannot be done by the rate monotonic algorithm.

- ✓ Requirements:
  1. The requests for all tasks with deadlines are **periodic**.
  2. The processing of a single task must be **finished** before the next task of the same data stream become ready for execution.
  3. All tasks are **independent**.
  4. Run-time for each request of a task is **constant**.
  5. Any non-periodic task in the system has **no** required deadline.

Utilization Comparisons

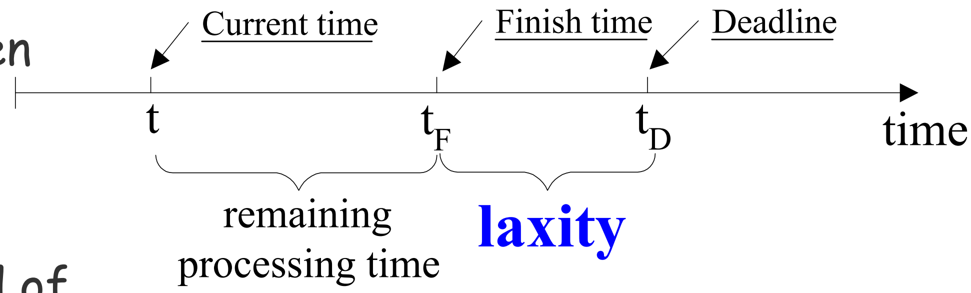
- With EDF, utilization of **100%** can be achieved because all tasks are scheduled dynamically.
- Rate monotonic algorithm is particularly suitable for continuous media because it makes optimal use of their **periodicity**.



- Problem: MPEG has no constant processing time per message.
- Solution: schedule these tasks according to their maximum data rate.
- Implication: The utilization is **lower**.
  The idle time can be used to process **non-time-critical** tasks.

Least Laxity First (LLF) Algorithm

- The task with the shortest remaining laxity is scheduled first.
  - The *laxity* of a task is the time between the actual time *t* and the deadline minus the remaining processing time.

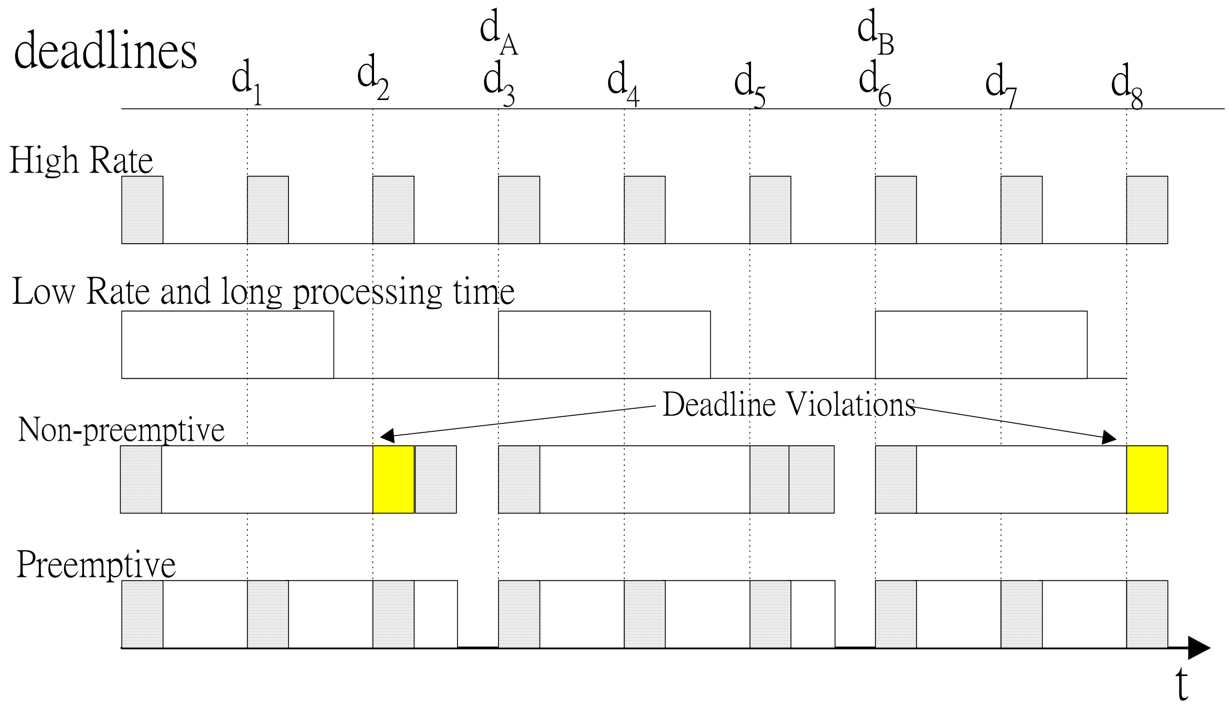


  - At **each** scheduling point (when a packet becomes **available** or at the **end** of a time slice), the laxity of each task must be newly determined.
- Disadvantages:
  1. The laxity of demanding tasks **dynamically** changes over time. Thus, tasks may preempt each other over several times without dispatching a new task.
     $\Rightarrow$ *causing **numerous** context switches.*
  2. Computing the laxity of each task at each scheduling point is **expensive**.

- **Shortest Job First (SJF)**
  - The task with the shortest remaining time is chosen for execution.
  - Advantage: It guarantees that as many tasks as possible meet their deadlines under an overload situation if all of them have the same deadline.

Preemptive vs. Non-preemptive Scheduling

- Non-preemptive: The task is processed w/o interrupt until it is finished or requires further resources.
- Preemptive: The task is interrupted immediately by a request of any higher-priority task.



- The scheduling of non-preemptive tasks is generally less favorable because the number of schedulable task sets is smaller compared to preemptive tasks.

Dynamic Frame Rate Control for Video Streams

by Pejhan, Chiang, Zhang, ACM MM '99

- Motivation:
  - **Video streaming**: each frame of video is immediately decoded and displayed upon reception.
    - A large initial delay and huge storage space is inevitable if the long video is played out only after the completion of the foremost download.
  - **Rate mismatch**: the encoded frame rate is different from the allowable delivery rate or the desirable playback rate.
  - Implementing **VCR-style controls** for users is rather challenging by video streaming.
  - Possible solutions:
    - The server **re-encodes** the video using different **quantization** value to match the client demand - degradation of video **spatial** quality.
    - Sacrificing the **temporal** quality instead, like frame skipping.

Dynamic Frame Rate Control

- Employing **motion estimation** approaches, MPEG, H.263, and the like cannot easily change the encoded bit rate on the fly.

- **Regenerating** motion vectors for a **different** bit rate in real-time is quite computation intensive.
  - One solution: pre-storing the video sequence at **various** frame rates.

    (for the expense of considerable disk storage space)

  - The proposed idea: to store only **motion vectors** for the lower frame rates.
    - M-files recording motion vectors are rather small compared to the complete sequences.
    - Consuming **less** space to achieve the **same** savings of computation for motion vector regeneration.

  - Using H.263 standard as an example.

Overview of H. 263

(ITU-T Recommendation for low bitrate communication)

- Four layers: [syntax]
  - **Picture**: stream = (picture header, picture data i.e. **GOBs**)$^{+}$(EOS code)$^{?}$
    - Picture header = (PSC_{22}, TR_8, $PTYPE_{13}$, $PQUANT_5$, CPM_1, … , PEI_1=0)
      - ✓ PSC (Picture Start Code).
      - ✓ $bit_{6\text{-}8}$ of PTYPE = $001^{\text{if sub-QCIF}}$, $010^{\text{if QCIF}}$, $011^{\text{if CIF}}$, $100^{\text{if 4CIF}}$, etc.
      - ✓ bit_9 of PTYPE = $0^{\text{if INTRA}}$, $1^{\text{if INTER}}$.
      - ✓ bit_{13} of PTYPE = $0^{\text{if normal I- or P-picture}}$, $1^{\text{if PB-frame}}$.
  - **Group of blocks**: GOBs = (GOB header$^{\text{if not first}}$, (MB)$^{+\text{, if not skipped}}$)$^{+}$
  - **Macroblock**:
    - MB = (COD_1 $^{\text{if } bit_9 \text{ of PTYPE = 1}}$, $MCPC_{1\text{-}9}$, … , block 1$^{\text{if coded}}$, … , block 12$^{\text{if coded}}$)
  - **Block**: block = (INTRADC$^{\text{if intra}}$, (run-level VLC)$^{+}$, end_of_block)
    - run-level VLC = entropy-coded of quantized 8x8 DCT coefficients.

- NOTE:
  - If COD=1, MB is skipped; otherwise, some blocks are coded.
  - Some blocks can be coded in INTRA mode even if in predictive frames.

Details of proposed scheme

- Dynamic frame rate control approach:
  - Storing only **the motion vectors** for the lower frame rates in separate motion files additionally.
  - The video server can dynamically switch frame rates in the middle of transmission to adapt each user's specification.
    - A full version, 30 fps, bit-stream is first decoded frame by frame.
    - For each frame needed to be encoded, the motion vectors from the motion file of the desired frame rate is retrieved for delivery.
    - A great deal of computation can be saved through the storage of pre-computation results.
- Motion file format (m-file):
  1. Each frame starts with a 17-bit PSC.
  2. 1/2 bit COD & MB: 0 for COD=1, 10 for INTRA, 11 for INTER mode.
  3. Loop to next MB (2) or frame (1).
- **Full search algorithms** and **large search areas** can be used to achieve maximum compression for off-line motion estimation.

Fast Forward Control

- Possible approaches:
  - **2x speed** (**double consumption**): encoded at 30 fps, decoded at 60 fps.
    - **Bandwidth consumption** is proportional to the fast forward **speed**.
    - A powerful processor is necessary.
  - **Frame skipping**: to just transmit and display the I-frames.
    - Due to the large size, I-frames are few (1 or 2 per 30 frames).
  - **2-Phase Service Model**:
    - Initialization phase: non-adjacent portions of the video are first loaded for FF operations.
    - Normal phase: the rest of video portions are downloaded.
    - Drawbacks: **long** download time, significant local **storage** requirement.
  - **Sampling independently decodable segments** of the video:
    - 3x FF is achieved by sending every 3rd segment. (but, non-uniform FF)
  - The **proposed** dynamic frame rate control scheme.
    - Re-encode a 30 fps at 15 fps, but decode/display at 30 fps.

Error Control Techniques

by Injong Rhee, N-Carolina, SIGCOMM '98

- Full title: Error Control Techniques for Interactive Low-bit Rate Video Transmission over the Internet
- Motion prediction loop:
  - MPEG, H.263, etc., eliminate the temporal redundancy of the video streams using motion-prediction loop. (motion compensation-based codecs)
  - Fact: Errors in a reference frame caused by earlier packet loss will effectively propagate to all the subsequent referencing frames.
  - Correction of these errors will have a profound effect on the overall playback quality of video streams.
- Retransmission-based error control:
  - Advantages:
    - The bandwidth consumption for packets retransmission is more effective than the forward-error correction techniques.
    - The proposed technique re-arranges the **temporal dependency** of frames so that **a displayed frame** is referenced for the decoding of its succeeding dependent frames much later than its display time.
  - Disadvantages:

    Mask out

    - Retransmission takes time, usually experiencing in several round-trip delays.

Overview of the technique

- Characteristics:
  - It is combined with the layered video coding techniques to maintain consistently good video quality even under heavy packet loss (severe congestion).
  - It has been tested with the extensive Internet experiments.
  - This study shows that the layered video coding techniques **alone** cannot result in good performance in the presence of heavy packet loss.
  - The **hybrid** of the proposed retransmission scheme and the layered video coding technique **(QAL, Quality Assurance Layering)** leads to a considerable performance improvement.

- Traditional error control:
  - For video streams, adding more intra frames. (counter effect on temporal redundancy)
  - CU-SeeMe, nv, vic, and the like employ the conditional replenishment.
    - It filters out the blocks that have not changed much from the previous frame and intra-code the remaining blocks. (**increase temporal independency**)
  - Forward Error Corrections: (parity, add redundancy to sustain packet loss)
  - Re-transmission of missing packets: (ARQ, CUDP, Patching, etc.)

Retransmission-based Error Control

- Re-transmission of missing packets: (ARQ, CUDP, Patching, etc.)
  - Most researchers focus on the extended control or playout times to allow retransmitted packets to arrive in time for **display**.
    - The playback time of a frame is delayed by at least three one-way trip times. (two for packet transmissions, and one for a retransmission request)
- **PTDD (Periodic Temporal Dependency Distance)** is proposed for error control.
  - It does not require any artificial extension of control time and playout delays.
  - In this scheme, the frames are played at the normal playback times with no delay.
  - If the packet arrives late, the frame is displayed with errors.
  - Particularly, this "late" packet is used to remove error propagation.
    - The frame was displayed (with errors). But, the late packet can be used to **restore** its frame so as to **stop** the errors from being amplified.
  - PTDD extends TDD of frames, inter-frame delays or frame intervals, between a frame and temporally dependent frame.
  - Every p-th frame (periodic frame) has an extended TDD, while the other frames have TDD=1 (not protected).
    - The usage of TDD or extended-TDD does not effect on the regular playback time of the frames.
    - The TDD of periodic frames is determined by the estimated delay between the sender and the receiver.

QAL (Quality Assurance Layering)

- Since non-periodic frames with TDD=1 are not protected, the packet loss of these frames will cause the frames to be displayed with errors.
  - Error will propagate until the next periodic frame is received.
    - Periodic fluctuation on playback quality of video streams.
  - **QAL (Quality Assurance Layering)** scheme is used for this purpose.
- The usage of **QAL**:
  - Non-periodic frames are divided into two portions: **essential** and **enhancement** signals.
    - Originally, the essential signals are protected by a simple **forward error correction** (FEC) technique – significantly reduce error propagation.
    - Since the amount of data in essential signals is much smaller than the entire, the overhead introduced by FEC is moderate.
    - The frames only temporally depend on the essential signals of their reference frames.
    - ✓ However, under heavy packet loss, even essential signals can be lost, still causing error propagation.
    - ✓ Another drawback of the QAL is that the temporal redundancy present in the enhancement signals are not exploited at all (Low compression efficiency).

Illustration: H.261

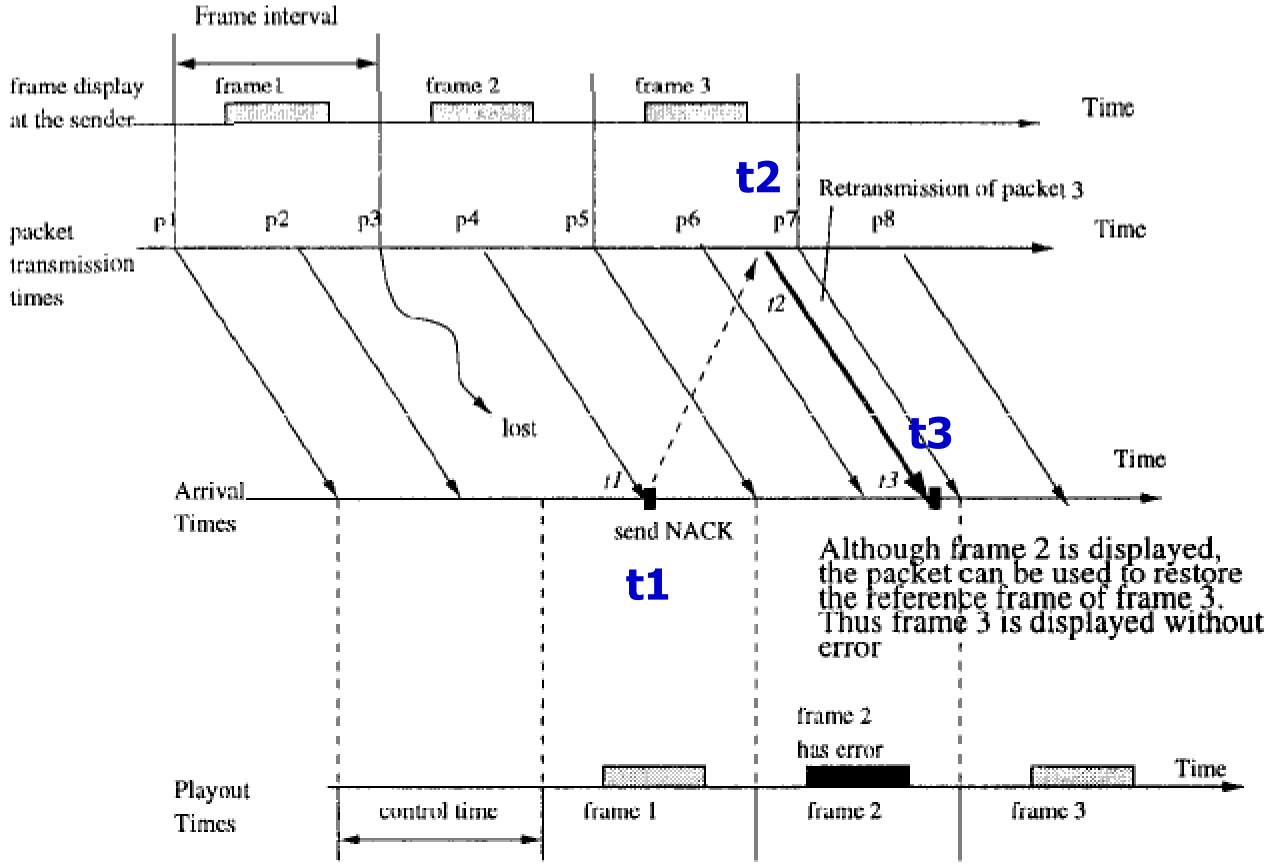
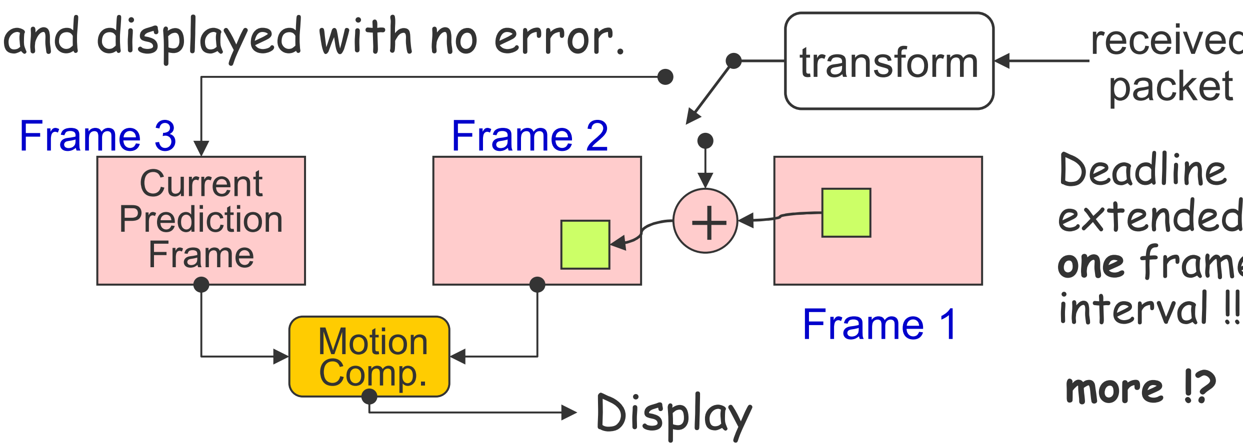


Figure 1: The recovery of frames using retransmission

- Each frame has two packets.
  - Frame 1 contains Packets p1 & p2; Frame 2 has p3 & p4.
  - When p4 arrives at t1, p3 is found lost, rendering a NACK sent to the sender.
  - The sender gets the NACK at t2, and resends p3 which arrives at t3 before Frame 3 is displayed.
  - Despite being late, p3 is used to restore Frame 2, the R-frame of Frame 3.
  - Frame 3 can be decoded and displayed with no error.

- The deadline of a packet is now its arrival time at the receiver after which it is not useful for decoding any frame.

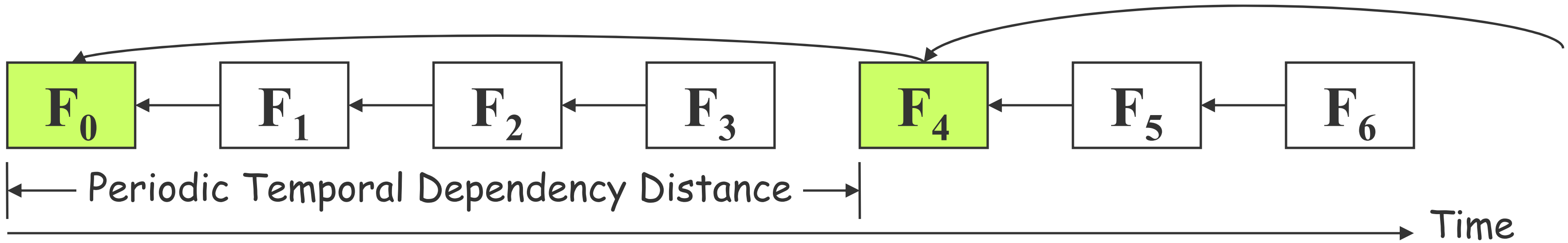


Deadline is extended by **one** frame interval !!

more !?

PTDD, PTDD+QAL

PTDD:



- Only F_0 , F_4 , F_8 and so on are protected with retransmission.

PTDD+QAL:

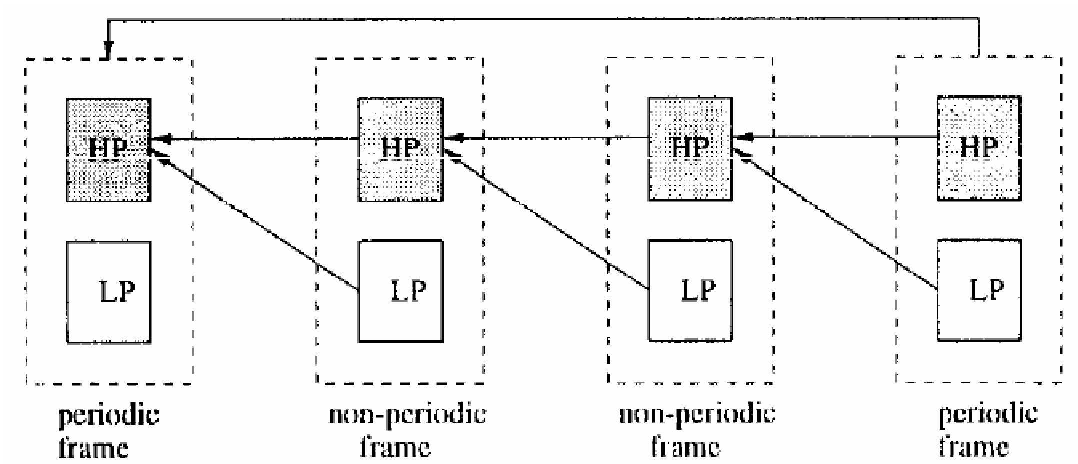
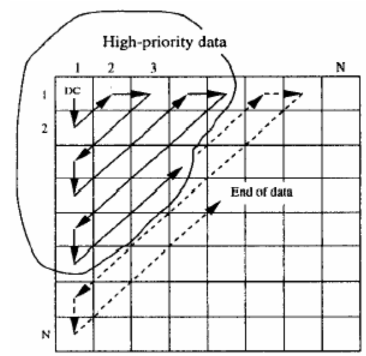
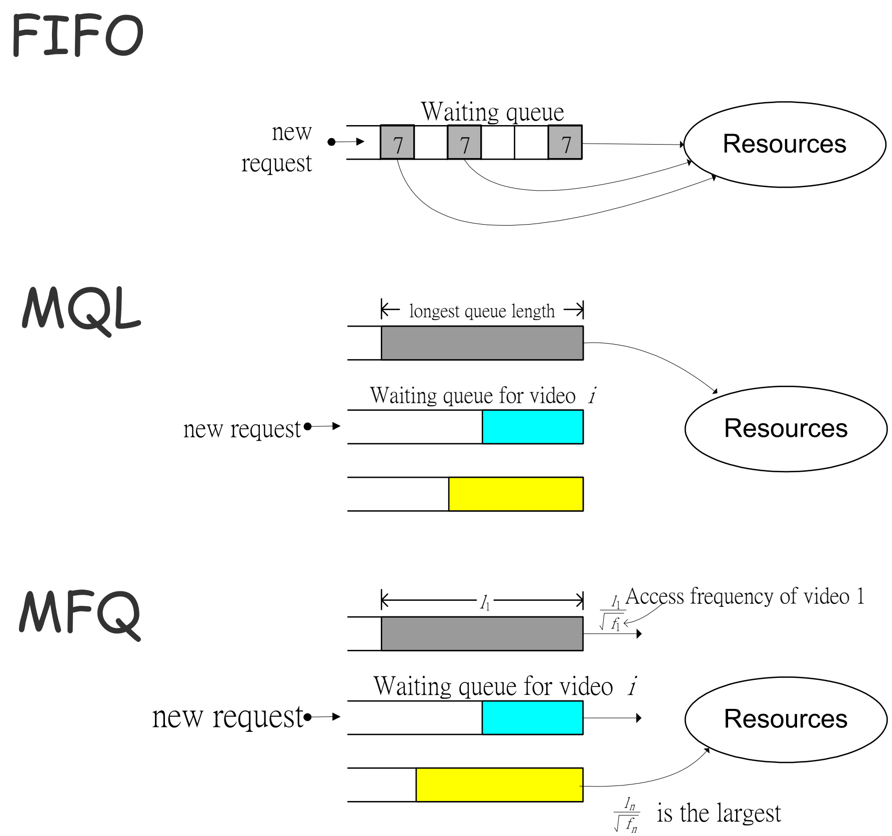


Figure 6: A Temporal Dependency Chain in Layered PTDD (PTDD +QAL)



On-demand Techniques

- Conventional approaches that dedicate one service stream for each request is very expensive (economic concerns)!!
  - The server bandwidth will quickly exhaust as the more clients make requests.
  - The available server capacity will determine the number of concurrent services being served at a time.
- More streams sharing?
- On-demand approaches momentarily queue the requests, and serve the ones for the same video objects in a batch by multicast.
  - The batch approaches can effectively improve the service quality, namely, delay.
  - There are several request-based (on-demand) batching technique.
    - Delay:
    - Throughput:
    - Fairness:

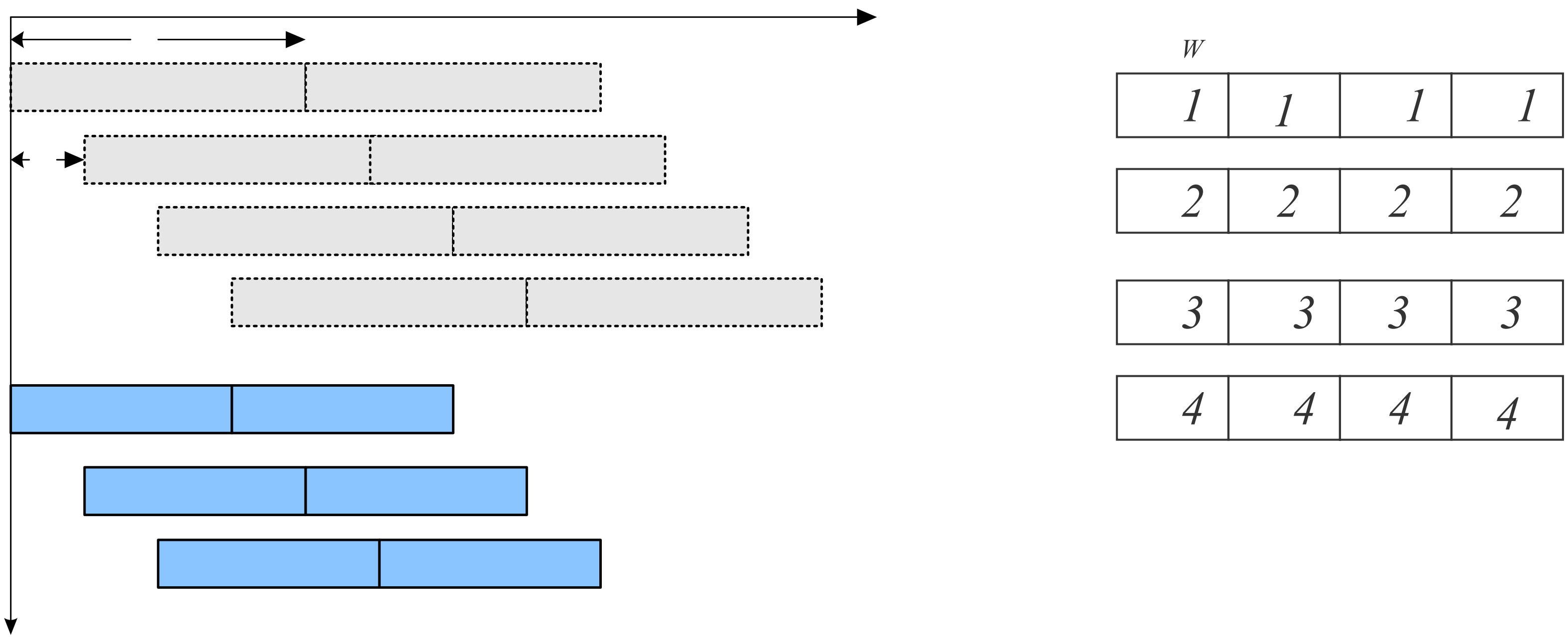


Periodic Broadcast Techniques

- Motivation:
  - User behaviors, 80/20 Rule: Most users (> 80%) request for a few very popular videos (< 20%).
  - Service overhead: Actively "Push" popular videos can serve the majority of users, and reduce most scheduling overhead.
  - Bounded service latency can be guaranteed if the new streams can initiate every certain time intervals.
    - Bounded service delay will favor the system throughput.

Uniform Broadcast (UB)

- A stream is started every B min. for a video.
- The worst latency is the broadcast interval (B). [Dan et al., ACM/MM'94, MS 6/96]



<u>Advantage</u>: Required BW α No. of Videos (not Users). $\Rightarrow$ Data-Centered !!

<u>Disadvantage</u>: Linear Latency Improvement (BW $\uparrow$).

Video Dissemination Techniques

- Subdivide a video into segments:
  - Each segment can therefore be possibly transferred with different periodicity's.

- Typically,
  - To minimize the initial service latency,
    - the fore video segments are delivered more frequently.
  - To save the server bandwidth,
    - the hind video segments are broadcast less often.
  - Beside the current segment being rendered,
    - some of the next segments are also being prefetched;
    - disk buffer serves as the staging area to eliminate the timing-mismatch.

Naïve Dissemination Approaches

- "K" equal segments are yielded:
  - Harmonic Broadcast (HB): [IEEE Trans. on Broadcasting, Sept. '97]
    - A channel of p/i repeatedly broadcasts the i^{th} segment.
      - ✓ "p": playback rate; L: video length; H(K): K^{th} harmonic number.
    - Total BW "B" for a video is p * H(K).
    - Latency, L/K, reduced vastly as B increases.
  - Pagoda Broadcast (PaB): [SPIE '99]

S_1	S_1	S_1	S_1	S_1	S_1
S_2	S_4	**S_2**	S_5	**S_2**	S_4
S_3	S_6	S_8	**S_3**	S_7	S_9

l=2k=4:	**S_z**	S_{2z} S_{2z+1}	**S_{z+1}**	S_{2z+2} S_{2z+3}	**S_{z+2}**	…	**$S_{3z/2-1}$**	S_{3z-2} S_{3z-1}
2k+1=5:	**$S_{3z/2}$**	S_{3z} S_{3z+1}	**S_{4z}** **S_{4z+1}**	S_{3z+2} S_{3z+3}	**S_{4z+2}** **S_{4z+3}**	…S_{2z-1}	**S_{4z-2}** **S_{4z-1}**	S_{5z-2} S_{5z-1}

z=10

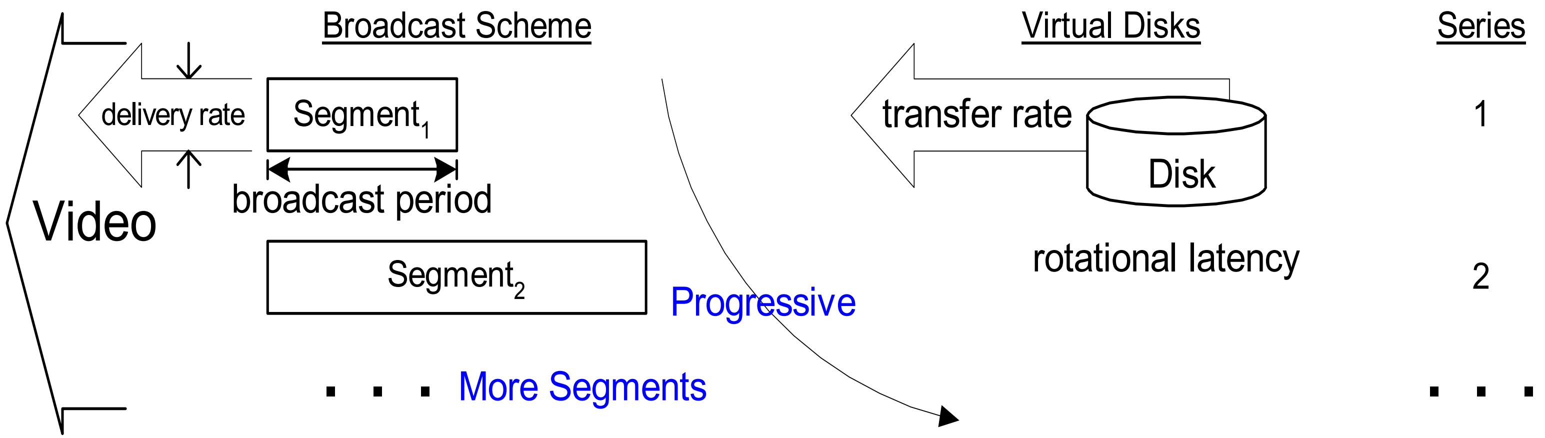
z	z	z	z

z/2	z/2	z	2z

- Disadvantages:
  - Any client needs to tune on all channels to download all segments at the same time.
    - Considerable BW required on the network & storage I/O's.
    - Streams of p, p/2,…,p/157, p/158,…,are hardly achieved.
  - Large disk space requirement: up to 50% of a video.

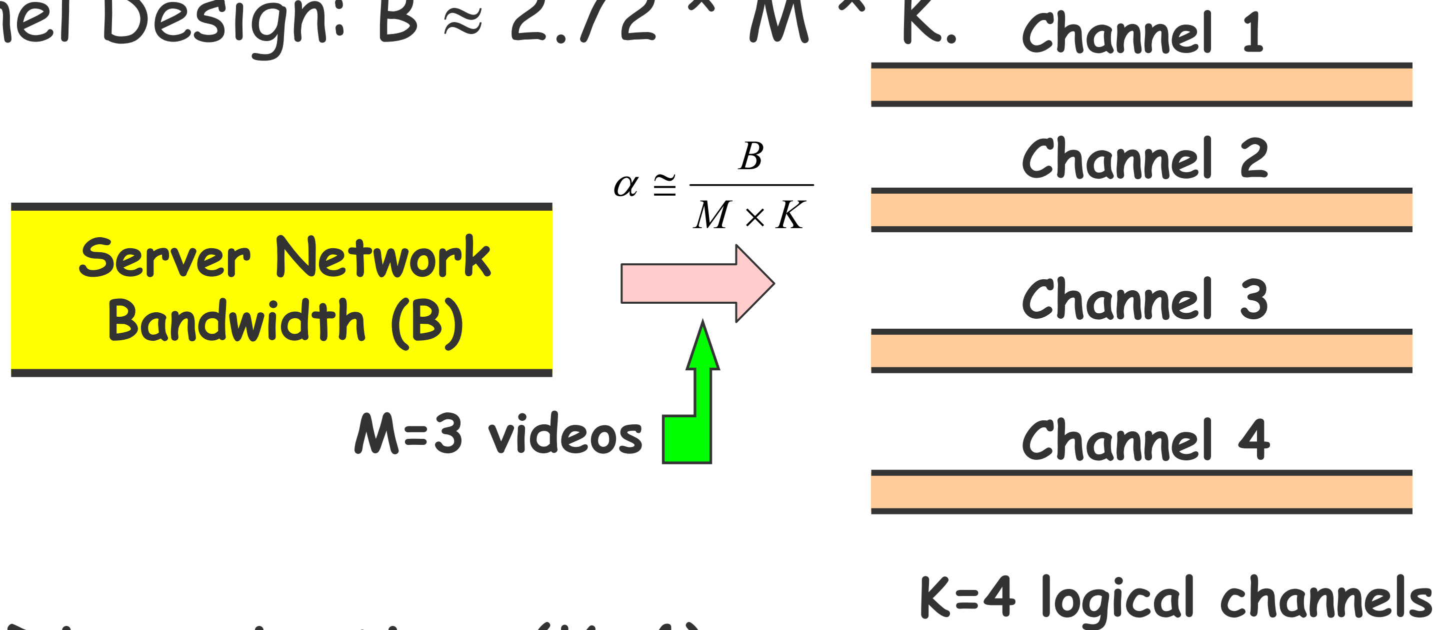
Tactical Dissemination Approaches

- "K" segments of increasing sizes are yielded:
  - Each is repeatedly broadcast on its channel at the same speed.
- Broadcast Series:
  - Normalized by the 1st segment, segment sizes form a series, which is used to depict the segmentation method.

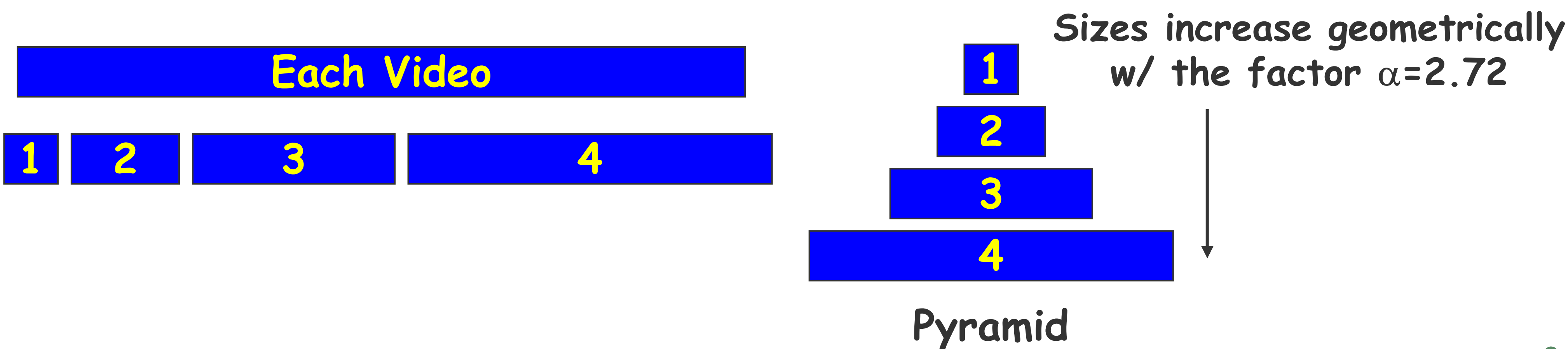


Pyramid Broadcast (PB) [Multimedia Systems, Aug. '96]

- Channel Design: B ≈ 2.72 * M * K.

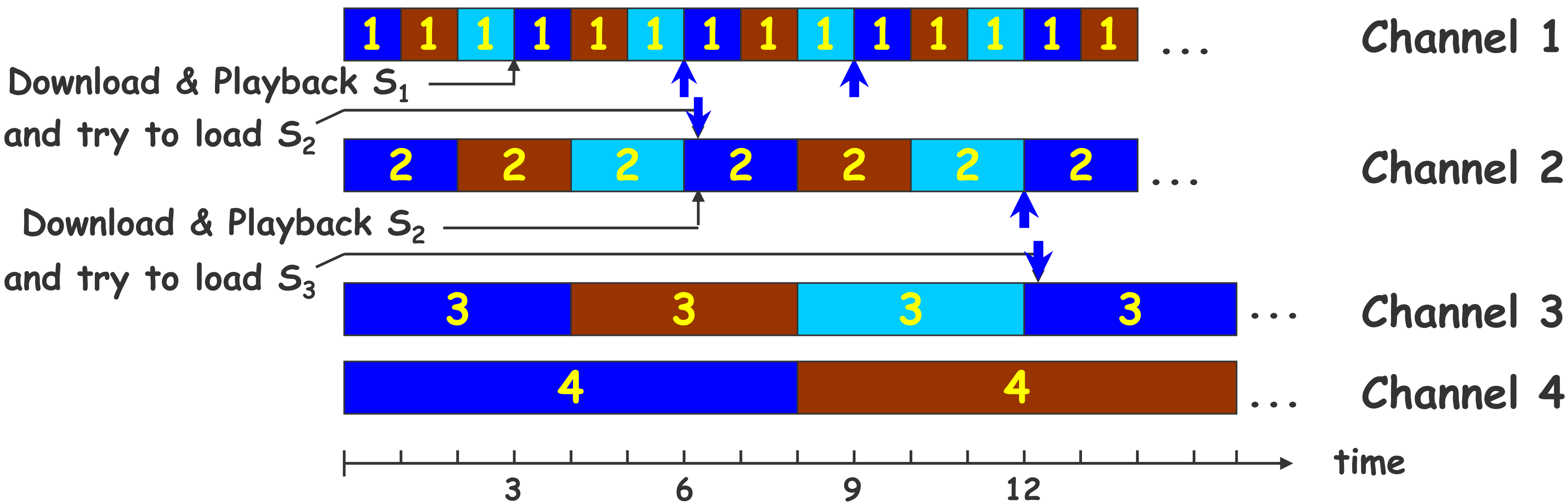


- Data Dissemination: (K=4)



Pyramid Broadcast (PB)

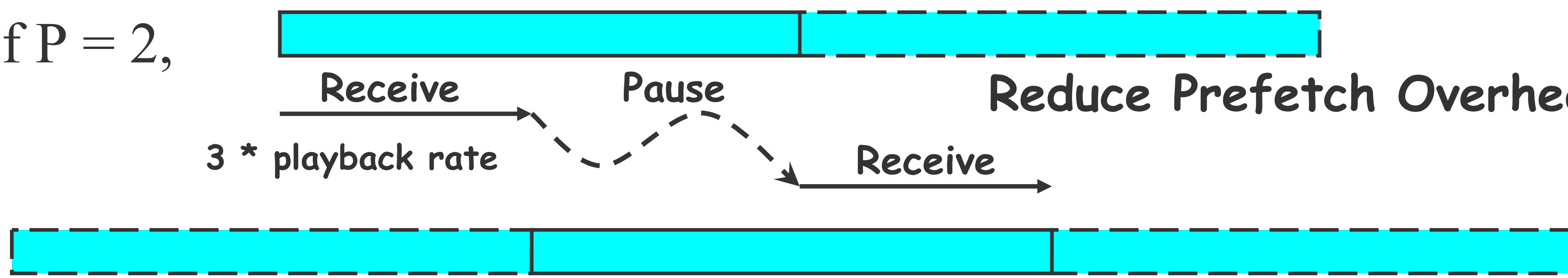
- Delivery Schedule & Playback Strategy:



Permutation-Based Pyramid Broadcast (PPB)

- Each logical channel = M * P subchannels. [IEEE/ICMCS'96]
- Each fragment is multicast on P subchannels.
- Clients can tune into these P subchannels to collect the data for a given fragment.

If P = 2,



Reduce Prefetch Overhead !!

Complex Control to tune into the middle of broadcast !!

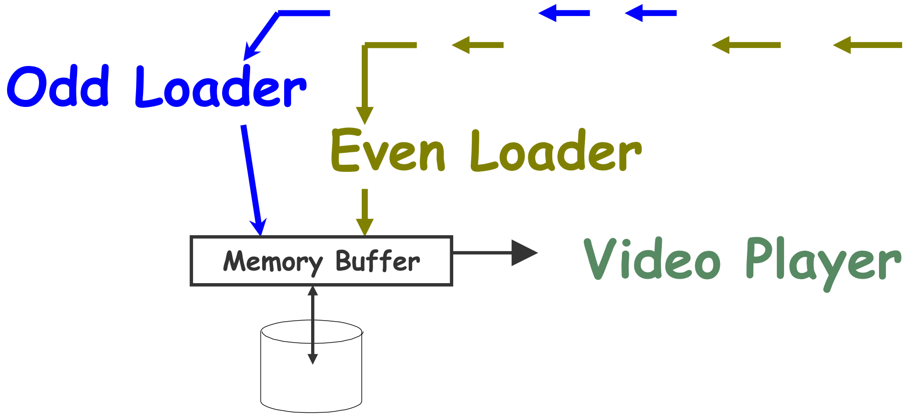
Skyscraper Broadcast (SB) [SIGCOMM '97]

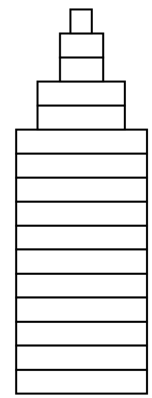
- K segments per video, each at the playback rate.
- Relative length progression:
  - An exemplary series used:

Size of larger segments are constrained to W (width of the "skyscraper").

Even group | Odd group

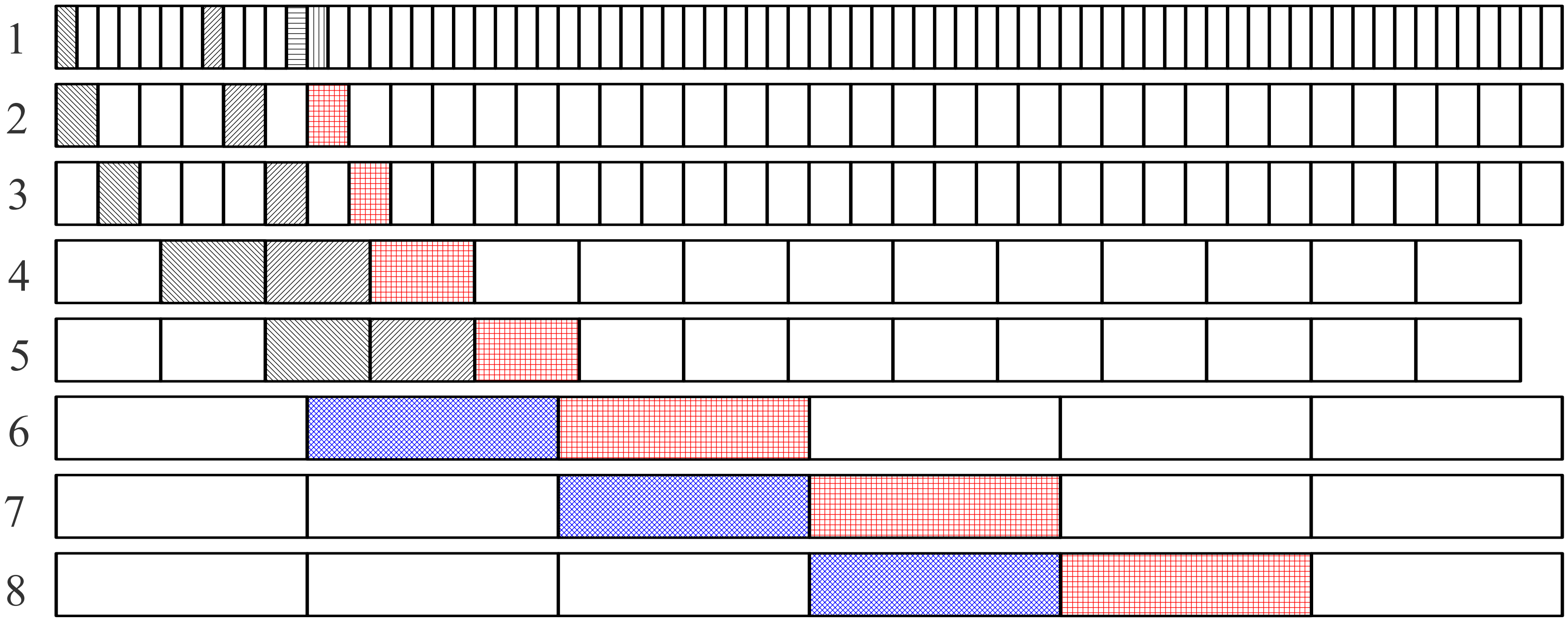
[1, 2, 2, 5, 5, 12, 12, 25, 25, … ,W, W, … , W]



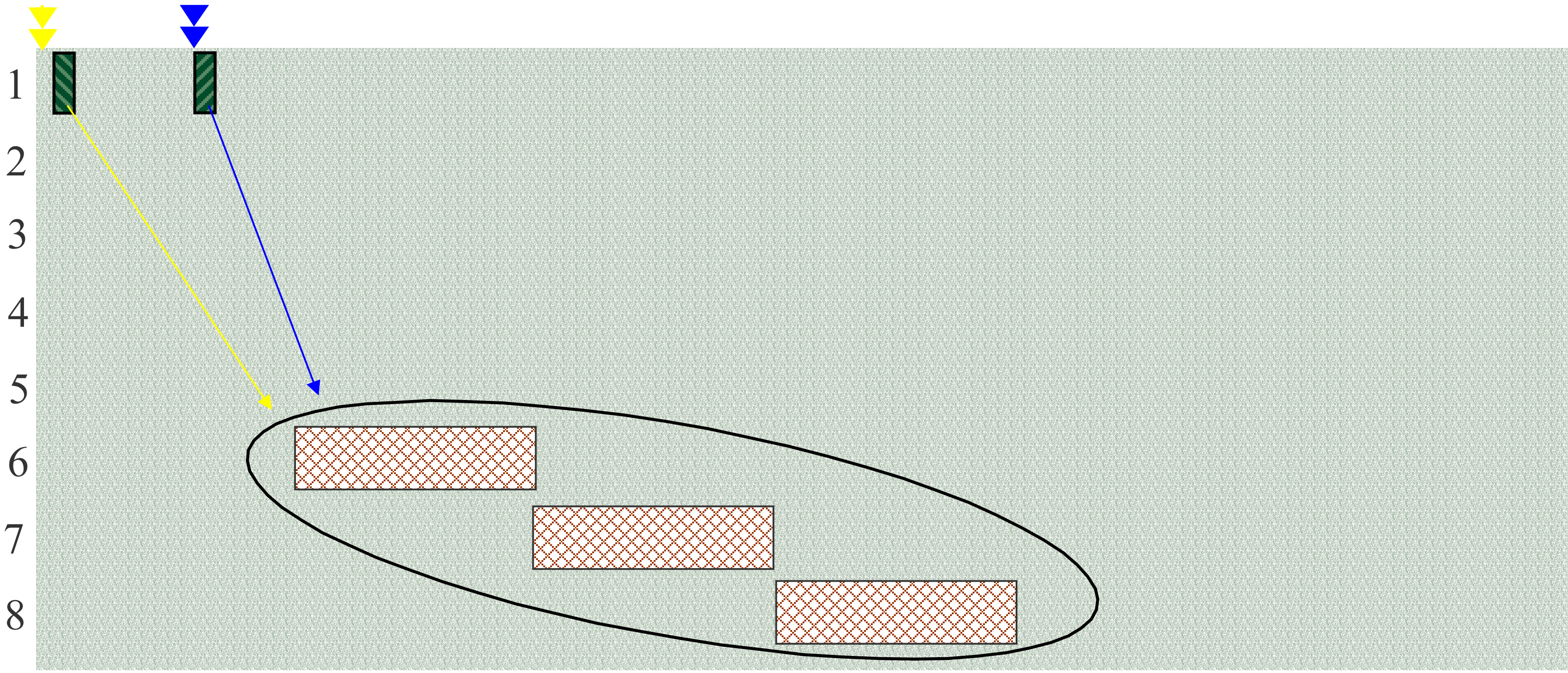


$$Service \quad Latency \approx \frac{\text{video length}}{\#\,\text{segment} \times w}$$

Receiving Data

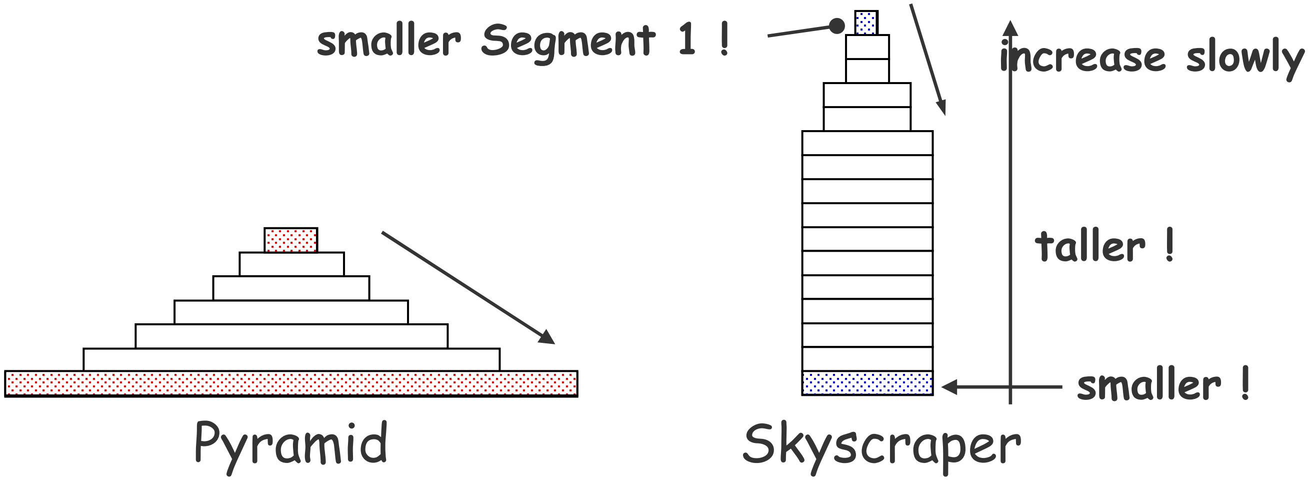


Receiving Data



- Batch Merging (Slot Aggregation) to save bandwidth.

Comparison



- SB yields more segments for a video using much less expensive channels. => better latency !
- SB makes last segments (W) smaller. => less buffer space !
- SB employs only two loaders to receive at most two segments simultaneously, each only at the playback rate.
  => less disk/network bandwidth overhead !

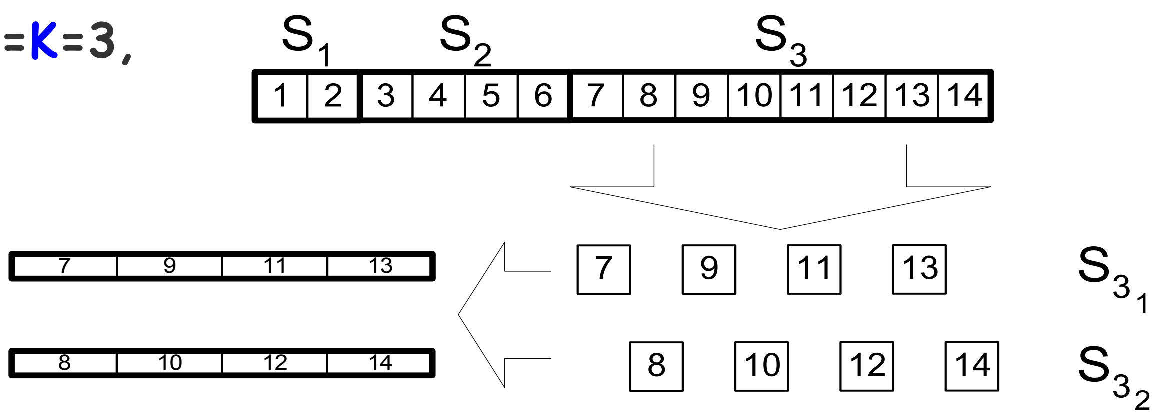
Other Variants

- Dynamic Skyscraper Broadcast (DSB): [MIS '98]
  - [1, 2, 2, 6, 6, 12, 12, 36, 36, …] is used.
  - 3, out of K, segments need to be received simultaneously.
- Client Centric Approach (CCA): [IC3N '98]
  - $[1, 2, \ldots, 2^{g-1}, 2^{g-1}, \ldots, 2^{2g-1}, \ldots]$ is used.
  - g, out of K, segments need to be received simultaneously.
- Mayan Temple Broadcast (MTB): [ACM MM '99]
  - g is set to be K, the extreme case of CCA.
- Pagoda Broadcast (PaB): [SPIE MCN '99]
  - All the K channels needs simultaneous reception.

Striping Broadcast (StB)

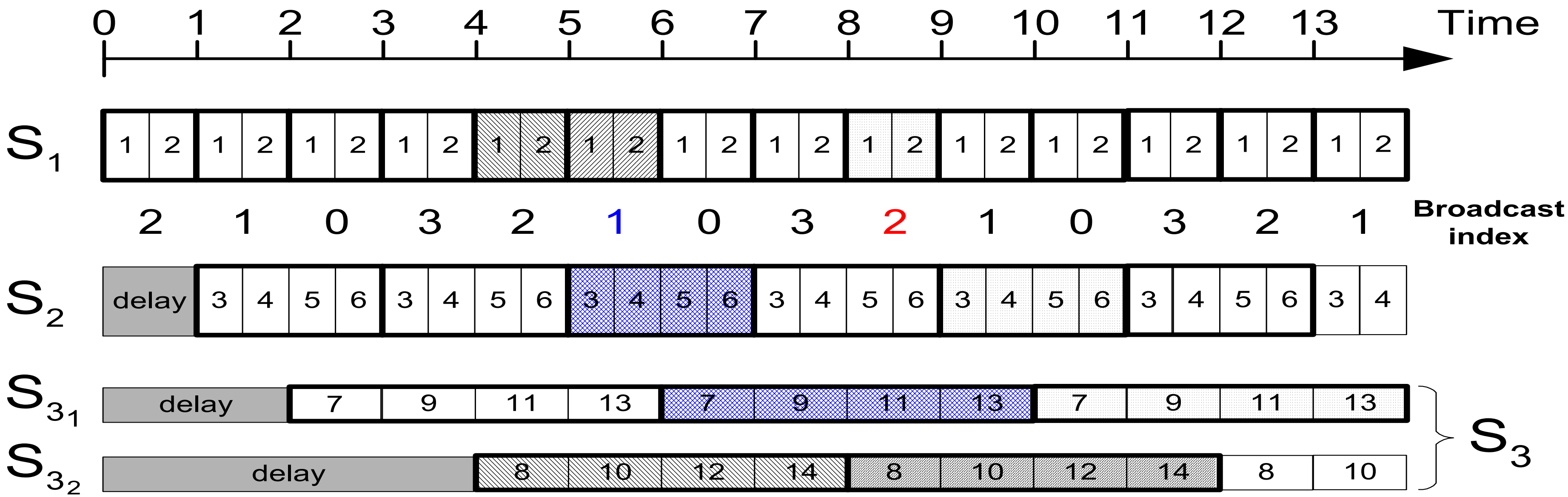
- Consider clients that can receive 3 segments simultaneously.
- Geometric series ‹1, 2, 4, … , 2^{n-1}, … , 2^{n-1}›: K horizontal segments, $n=1+\log_2 W$.
- Each of the n ~ K-th segments are further vertically partitioned into 2 subsegments:

For instance, n=K=3,



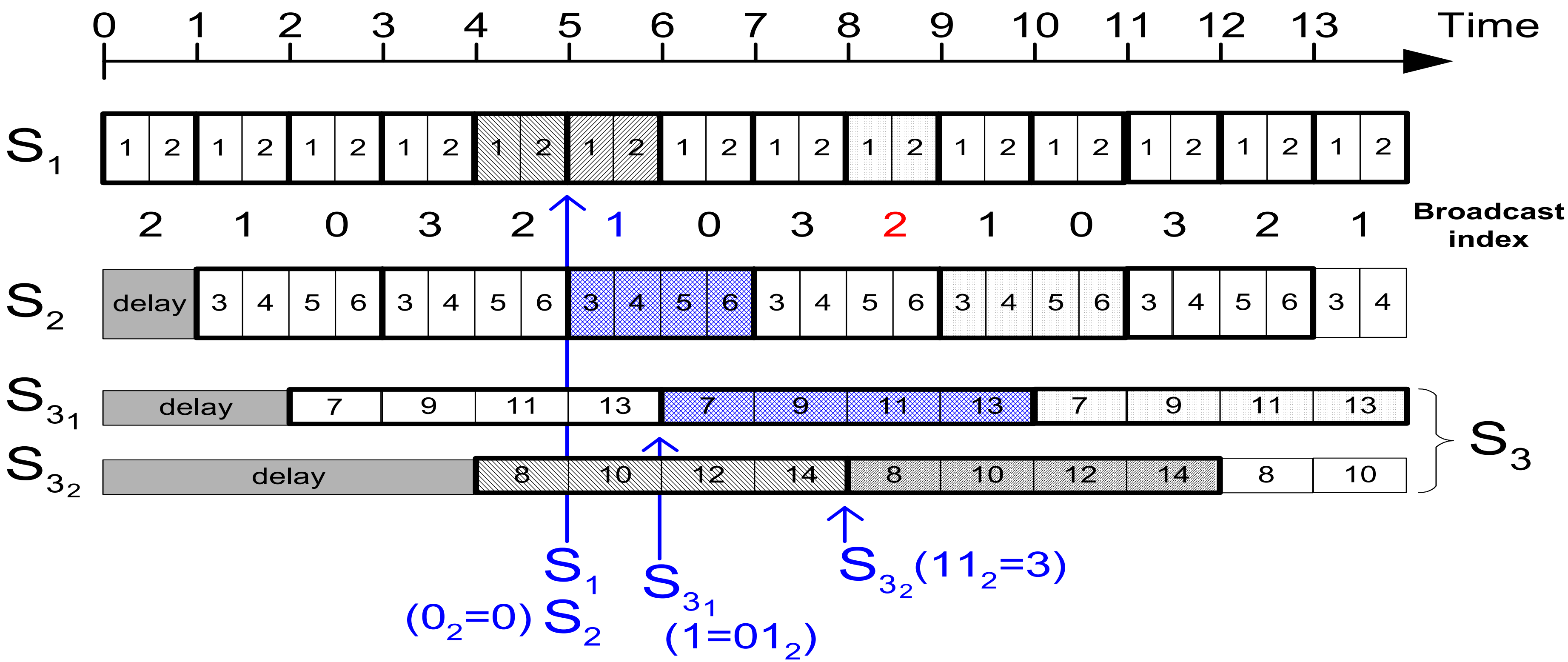
- Each subsegment is multicast at half playback rate.

Receiving Data



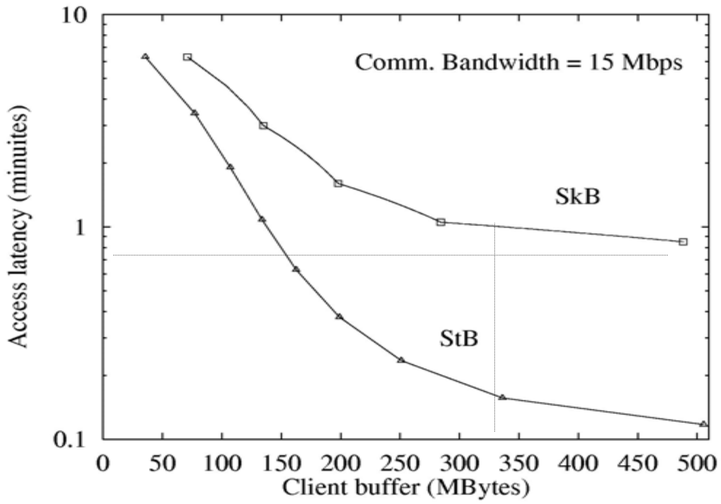
- Apply "delay" to offset the phase of broadcast schedule; Striping the virtual disks.
- Sharing Multicast Streams.

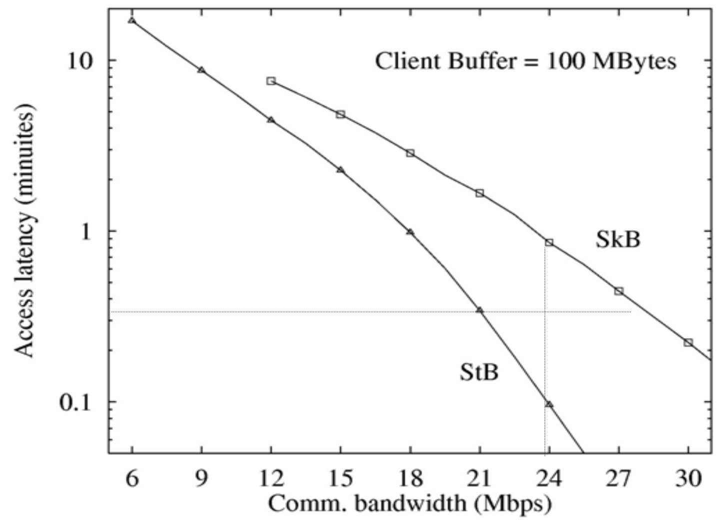
Receiving Data



❖ Deterministic Tuning Schedule:

Comparison

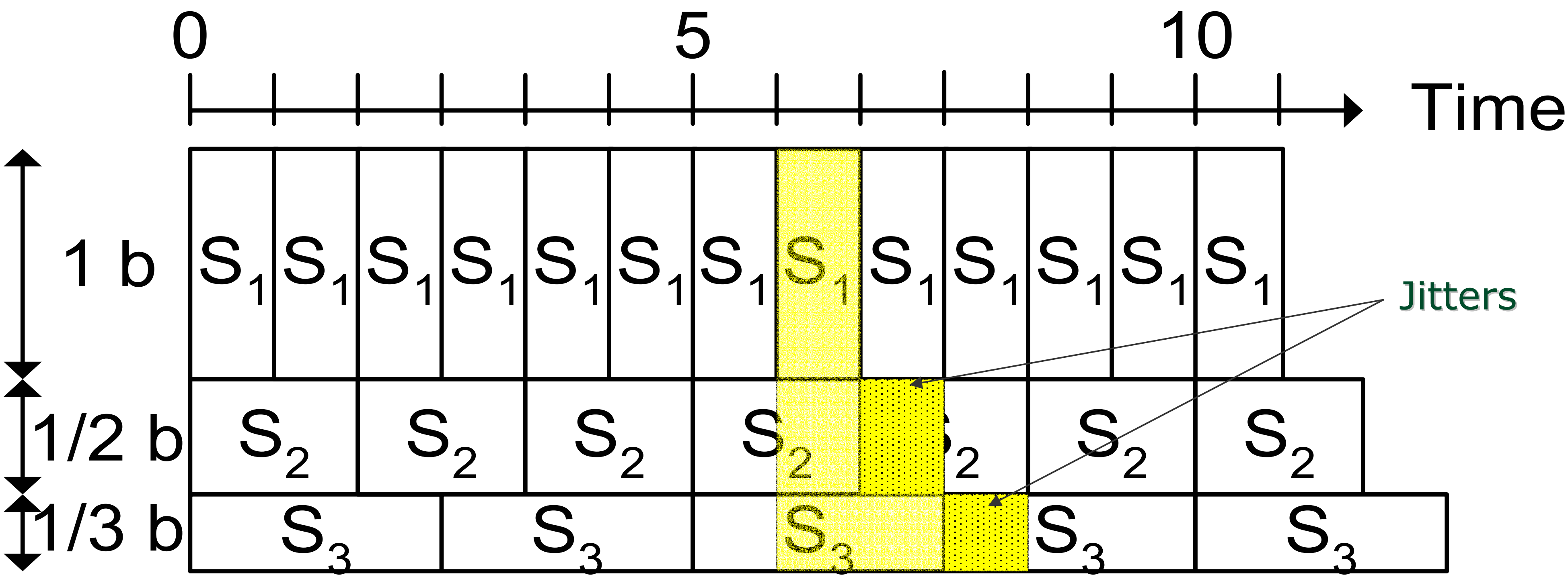




- StB further improves latency and server bandwidth, while using tiny disk buffer at each client end.

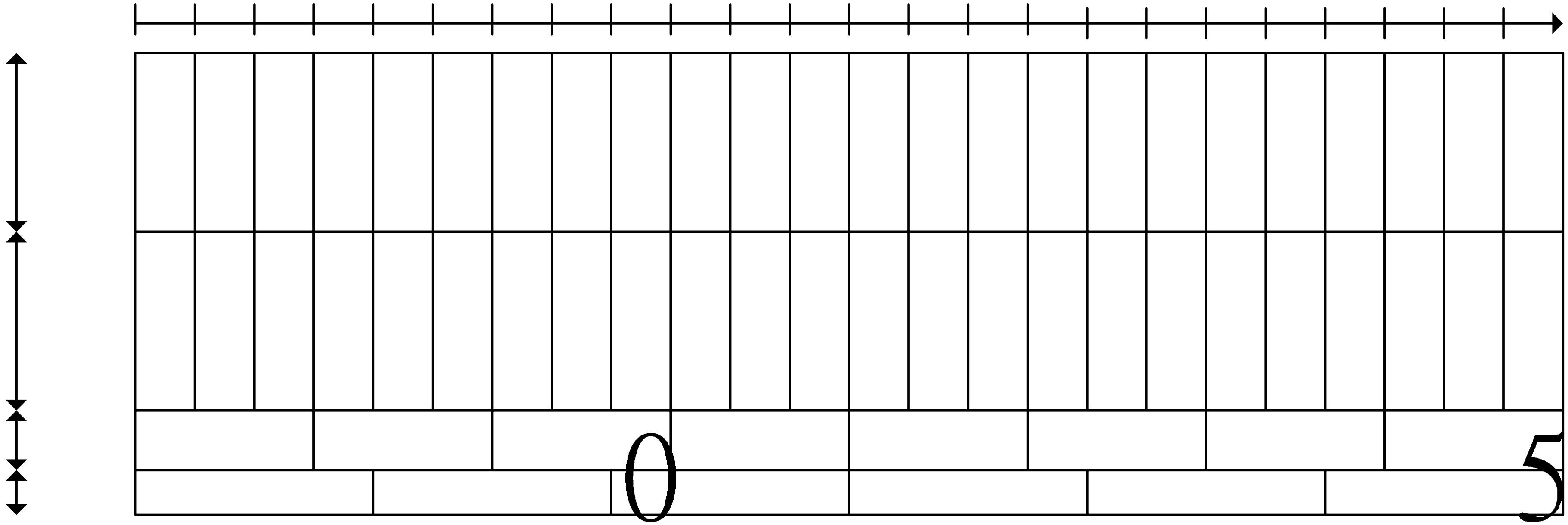
- The cost is to tune to only one additional channel.

Jitters in Harmonic Broadcast



Harmonic Broadcast

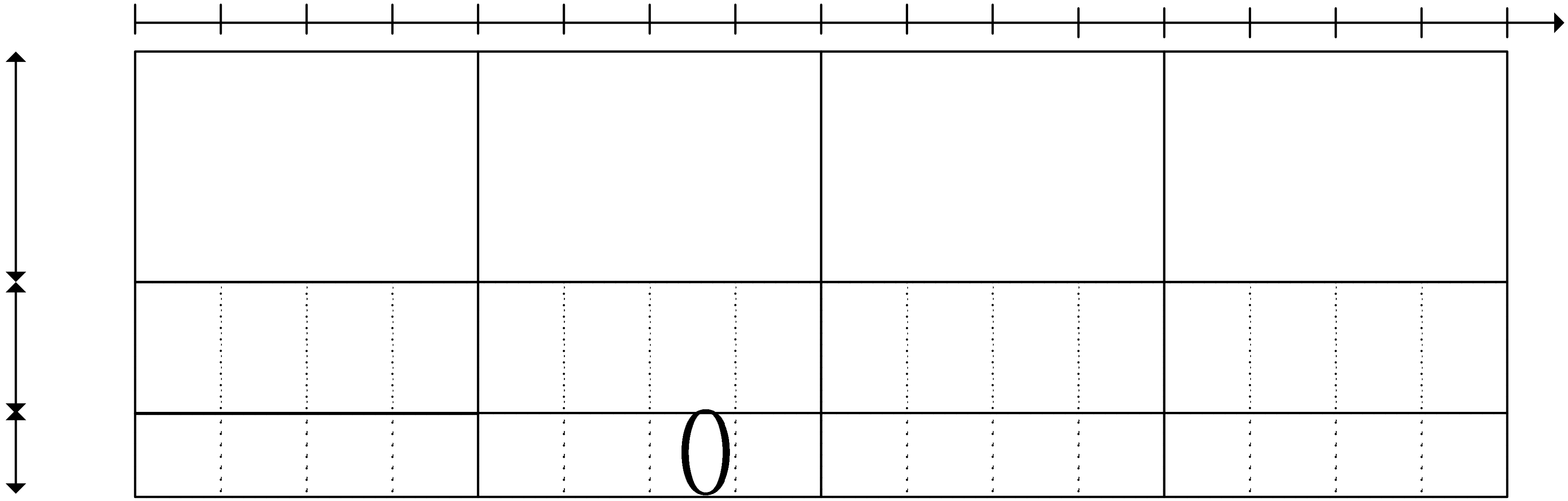
Cautious Harmonic Broadcast(CHB)



$$B_{CHB}(n) = 2b + \sum_{i=3}^{n-1} \frac{b}{i} = \frac{b}{2} + bH_{n-1}.$$

$$B_{HB}(N) = \sum_{i=1}^{n} \frac{b}{i} = bH_{n}.$$

Quasi-Harmonic Broadcast(QHB)



$$S_i \Rightarrow S_{i,1} \cdots S_{i,i\times m-1} : \quad S_2 \Rightarrow S_{2,1} \cdots S_{2,7}; \quad S_3 \Rightarrow S_{3,1} \cdots S_{3,11}$$

- Each slot is broken into m equal subslots.
  - The last subslot of each slot is used to broadcast the first i-1 subsegments.
  - The k<u>th</u> subslot of slot j is used to broadcast subsegment i· k+j-1 mod i· m. (others)

$C_i = \frac{b}{i}$ in HB $\Rightarrow \frac{mb}{im-1}$ in QHB.

$$B_{QHB}(n,m) = b + \sum_{i=2}^{n} \frac{mb}{im-1} = bH_n + \sum_{i=2}^{n} \frac{b}{i(im-1)}.$$

1 b

S_1

Poly-Harmonic Broadcast (PHB) or Enhanced Harmonic (EHB)

HB: $\underbrace{\boxed{}\boxed{}\boxed{}\boxed{}}_{\frac{b}{1}} \quad \underbrace{\boxed{}\boxed{}\boxed{}\boxed{}}_{\frac{b}{2}} \quad \cdots$

PHB: $\boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{}\ \boxed{} \quad \cdots$

For the same initial latency

EHB: $\frac{b}{4} \quad \frac{b}{5} \quad \frac{b}{6} \quad \frac{b}{7} \quad \frac{b}{8} \quad \frac{b}{9} \quad \frac{b}{10} \quad \frac{b}{11}$ $\quad$ m = 4

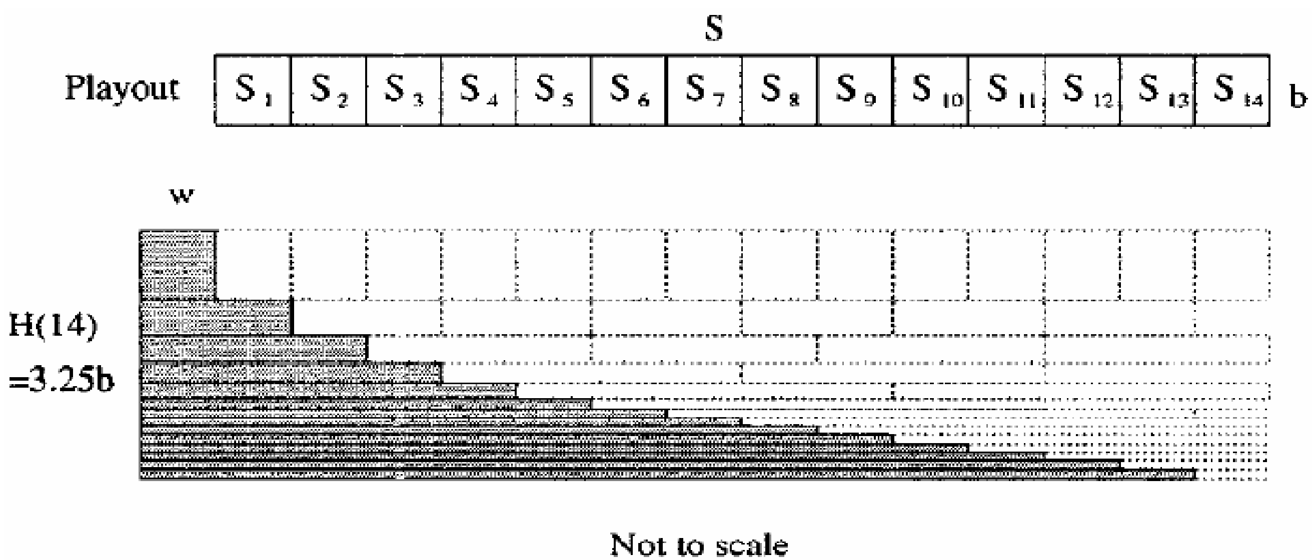
$$C_i = \tfrac{b}{i} \text{ in HB} \Rightarrow C_{i,1} = \tfrac{b}{im}, \cdots, C_{i,m} = \tfrac{b}{(i+1)m-1} \text{ in PHB or EHB.}$$

$$B_{PHB}(n,m) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{b}{im+j-1} = b(H_{nm-1} - H_{m-1}).$$

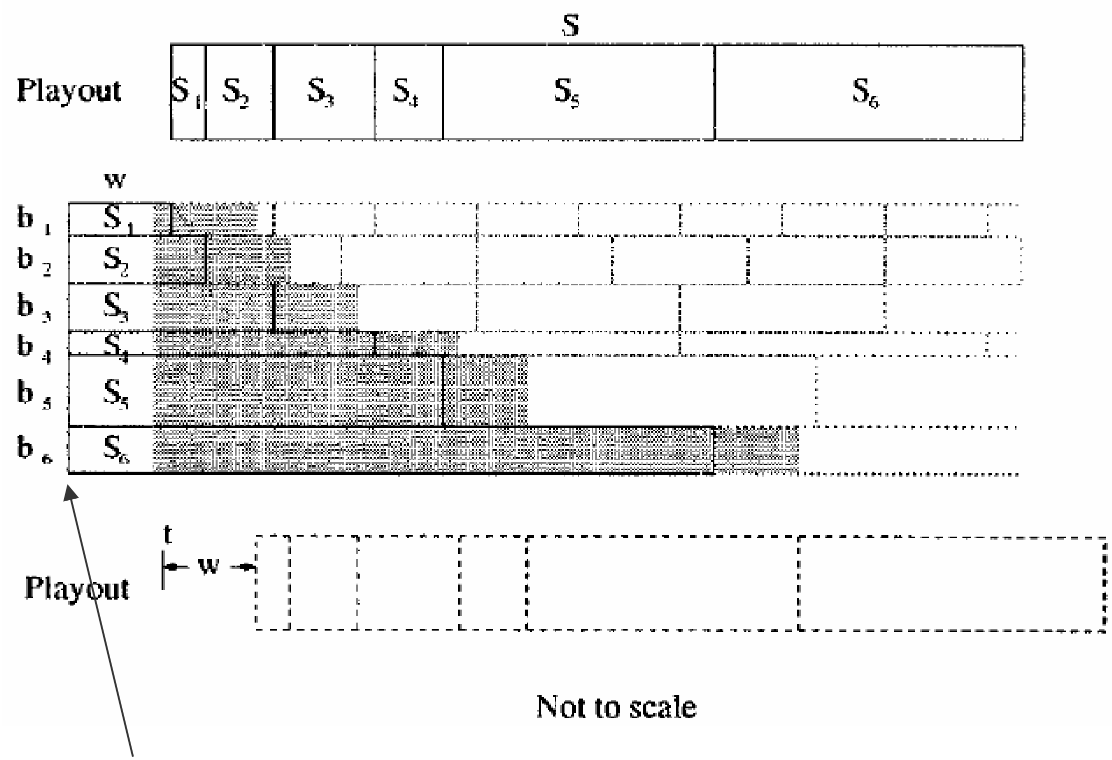
Upon arrival, a client starts to receive all segments even during the middle of the broadcasts.

A Comprehensive Study Ailan Hu, INFOCOM 01

- Generalized analytical approach
  - Temporal-bandwidth map
  - 3 design rules ⇒ optimally-structured scheme
    - No repeat of the segment prior to its playback.
    - Only needed data for just-in-time playback. (no blank)
    - At least one period starting within the *good downloading time*.
    - Poly-harmonic protocol meets these rules.



Optimally-Structured Schemes



- All segments are started from the same time.
- Given w = 1 & b = 1.
- All segments on channels are received simultaneously.

$$\text{minimize} \sum_{i=1}^{n} b_i$$

subject to

$$b_i(1 + \sum_{j=1}^{i-1} S_j) = S_i \quad i = 1, 2, \ldots, n$$

$$\sum_{i=1}^{n} S_i = S$$

$$S_i > 0, b_i > 0$$

Note that

$$(1 + b_i) = \frac{1 + \sum_{j=1}^{i} S_j}{1 + \sum_{j=1}^{i-1} S_j} \quad i = 1, 2, \ldots, n$$

$$\prod_{i=1}^{n}(1 + b_i) = 1 + \sum_{j=1}^{n} S_j = S + 1$$

$$b_i = b^* = \sqrt[n]{S+1} - 1$$

$$\begin{aligned} S_i &= b^*(1 + b^*)^{i-1} \\ &= (\sqrt[n]{S+1} - 1)(\sqrt[n]{S+1})^{i-1} \end{aligned}$$

Greedy Equal Bandwidth Broadcast (GEBB)

Given w = 1 & b = 1: $C_i = b_i b = b_i \rightarrow S_i$

$$\text{Mininize } \sum_{i=1}^{n} b_i \qquad \text{subject to} \quad b_i(1+\sum_{j=1}^{i-1} S_j) = S_i, i = 1..n.$$

$$\sum_{i=1}^{n} S_i = S$$

$$\Rightarrow (1+b_i) = \frac{1+\sum_{j=1}^{i} S_j}{1+\sum_{j=1}^{i-1} S_j} \quad \Rightarrow \prod_{i=1}^{n}(1+b_i) = 1+\sum_{j=1}^{n} S_j = 1+S = \text{constant!}$$

$$\therefore \quad b_i = b_{i+1} = \cdots = b^*, (1+b_i)^n = 1+S, b^* = \sqrt[n]{S+1}-1.$$

$$B_{GEBB}(n) = \sum_{i=1}^{n} b_i = nb^* = n(\sqrt[n]{S+1}-1).$$

$$S_i = b_i(1+\sum_{j=1}^{i-1} S_j) = b^*\prod_{j=1}^{i-1}(1+b_j) = (\sqrt[n]{S+1}-1)(\sqrt[n]{S+1})^{i-1}, i = 1..n.$$

120	1										
n	1	2	3	4	5	6	7	8	9	10	11
b*	120	10	3.946	2.317	1.609	1.224	0.984	0.821	0.704	0.615	0.546
n b*	120	20	11.838	9.266	8.047	7.344	6.888	6.569	6.334	6.154	6.011

Closer Look

$$b^* = \sqrt[n]{\frac{S}{W}+1}-1$$

- Given S = 120 min. & W = 1 min.

n	1	2	3	4	5	6	7	8	9	10	11
b*	120	10	3.946	2.317	1.609	1.224	0.984	0.821	0.704	0.615	0.546
n b*	120	20	11.838	9.266	8.047	7.344	6.888	6.569	6.334	6.154	6.011

- Given S = 120 min. & W = 0.5 min.

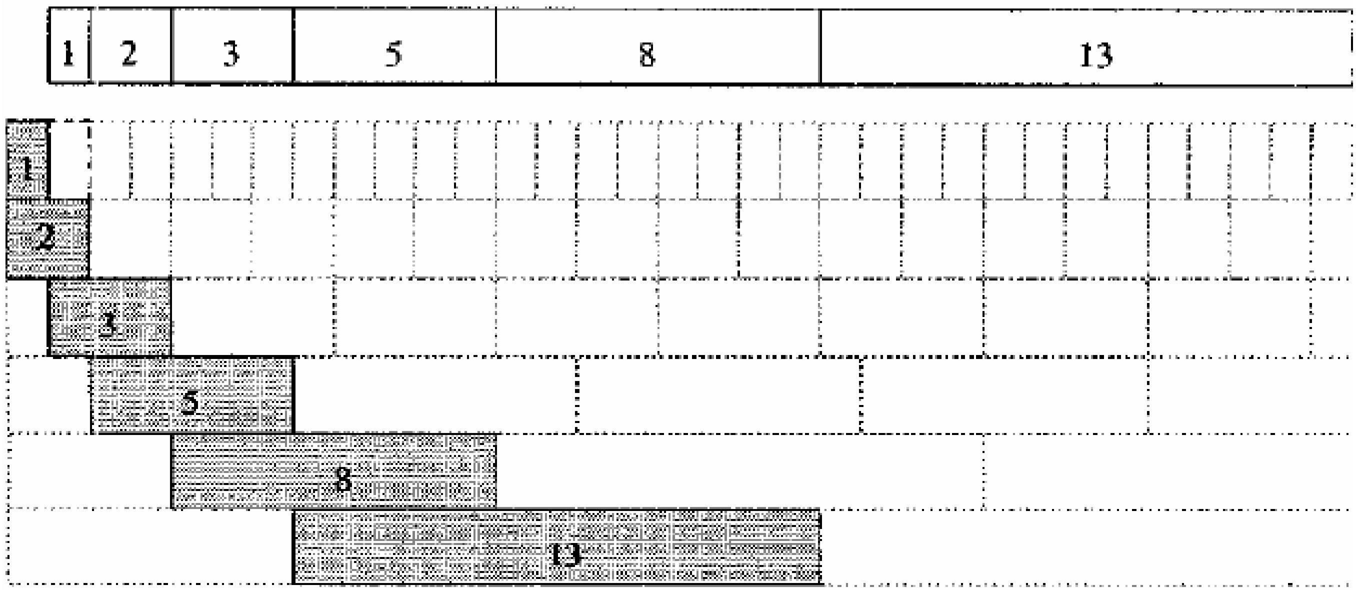
n	1	2	3	4	5	6	7	8	9	10	11
b*	240	14.524	5.223	2.940	1.995	1.495	1.189	0.985	0.839	0.731	0.646
n b*	240	29.048	15.669	11.760	9.975	8.968	8.324	7.880	7.554	7.306	7.111

- Given S = 120 min. & W = 10 min.

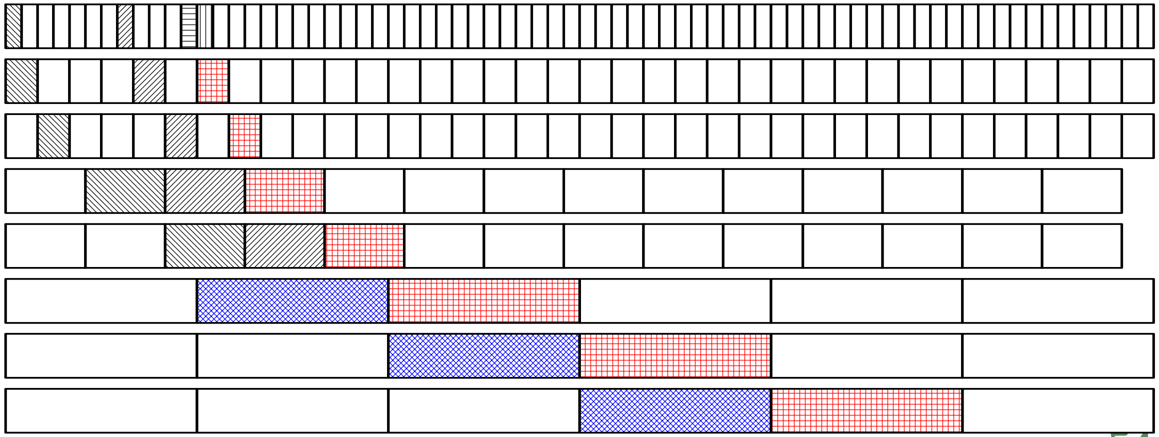
n	1	2	3	4	5	6	7	8	9	10	11
b*	12	2.6056	1.351	0.899	0.670	0.533	0.443	0.378	0.330	0.292	0.263
n b*	12	5.2111	4.054	3.595	3.351	3.200	3.098	3.024	2.968	2.924	2.889

Client I/O BW & Storage

- Fibonacci series is employed.



- Skyscraper Broadcast:
  - [1, 2, 2, 5, 5, 12, 12, 25, 25, 52, 52, …].



Reliability

- The tolerance of packet loss during delivery.
  - Network congestion $\Rightarrow$ packet loss.
- Error propagation
  - Due to the compression, loss of key-frames' packets may contaminate some other frames.
  - Error concealment.
- Metrics
  - PSNR: rate-distortion based measures.
  - $\hat{s}$ metric [SPIE 93]: a hybrid measure of spatial/temporal information.
  - Subjective Metrics: ITU-R 500-5, etc.
  - MPQM: moving pictures quality metric from EPFL in Lausanne.

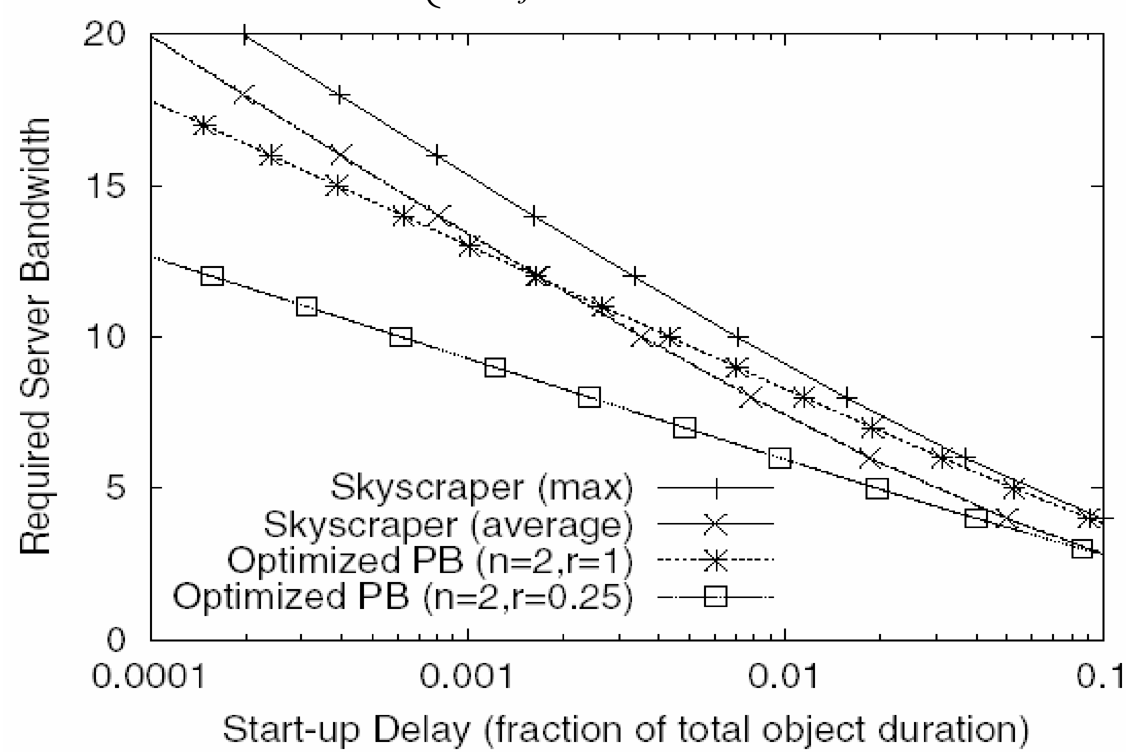
Optimized PB Protocols Mahanti, et. al. SIGCOMM 01

- Broadcast series:
  - Baseline: n = r s.
    - Fibonacci series: r=1, s=2.

$$l_k = \begin{cases} l_1 + \sum_{j=1}^{k-1} l_j & 1 < k \le s \\ \sum_{j=k-s}^{k-1} l_j & s < k \end{cases}$$

  - Using r ≤ 1:

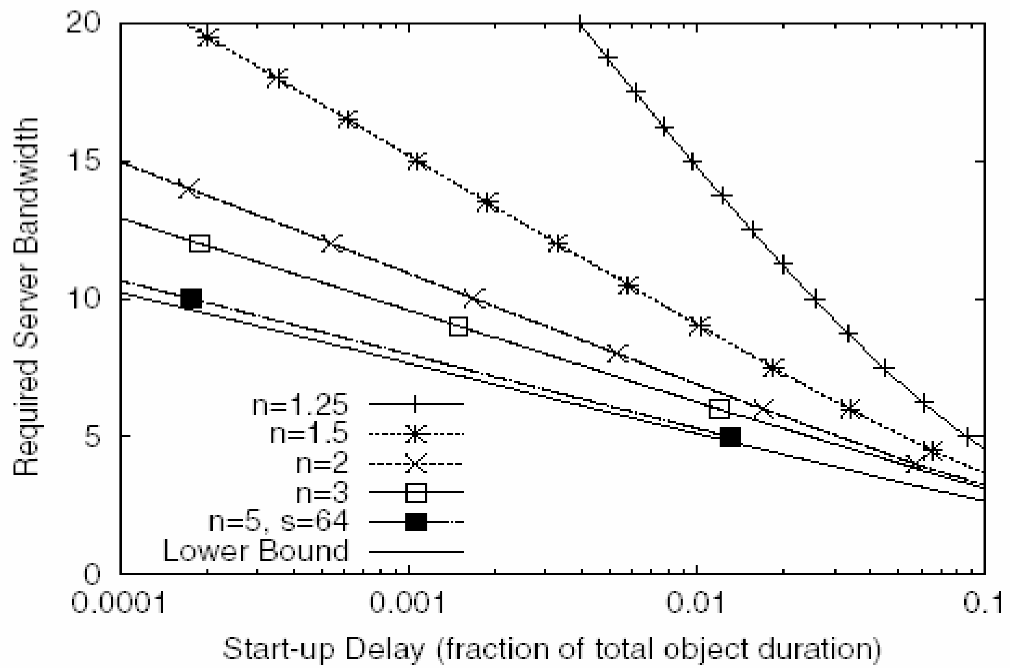
$$\tfrac{1}{r} l_k = \begin{cases} \tfrac{1}{r} l_1 + \sum_{j=1}^{k-1} l_j & 1 < k \le s \\ \sum_{j=k-s}^{k-1} l_j & s < k \end{cases}$$



Reliability

- Basic:

$$\frac{1}{1-p}\frac{1}{r}l_k = \begin{cases} \frac{1}{1-p}\frac{1}{r}l_1 + \sum_{j=1}^{k-1} l_j & 1 < k \le s \\ \sum_{j=k-s}^{k-1} l_j & s < k \end{cases}$$



(b) 10% Packet Loss

- General: $p_1, \dots, p_k$.

$$\frac{1}{1-p_k}\frac{1}{r}l_k = \begin{cases} \frac{1}{1-p_1}\frac{1}{r}l_1 + \sum_{j=1}^{k-1} l_j & 1 < k \le s \\ \sum_{j=k-s}^{k-1} l_j & s < k \end{cases}$$

Interactivity

- VCR-styled operations:
  - On-demand approaches.
  - Periodic Broadcast-based techniques.
- IPB (Interactive Pagoda Broadcast):
  - Clients retain all the segments received into the local storage.
  - The local storage and the **server patch** will fulfill the VCR operations.

Pagoda Broadcasting

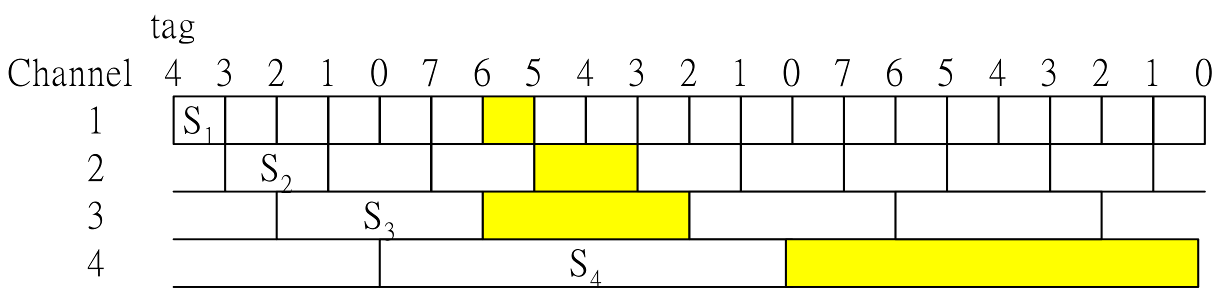
User arrival

Channel 1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1
Channel 2	S_2	S_3	S_2	S_3	S_2	S_3	S_2	S_3	S_2	S_3	S_2	S_3
Channel 3	S_4	S_6	S_8	S_5	S_7	S_9	S_4	S_6	S_8	S_5	S_7	S_9

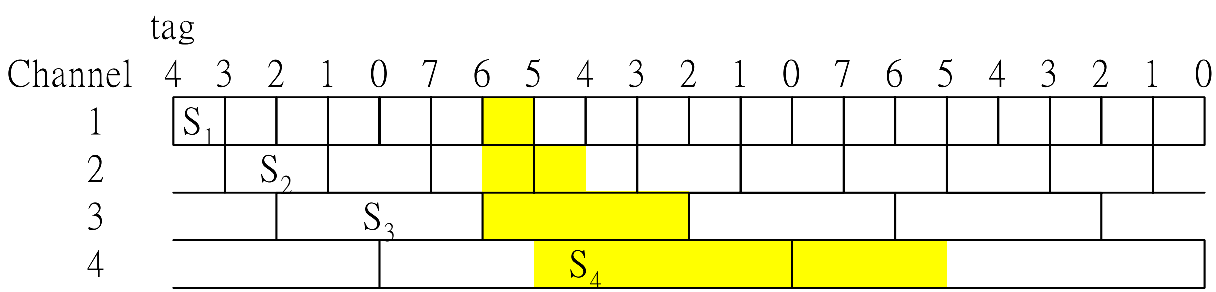
Time

Cost-effective Interactive Broadcast

- Based on the Striping Broadcast.



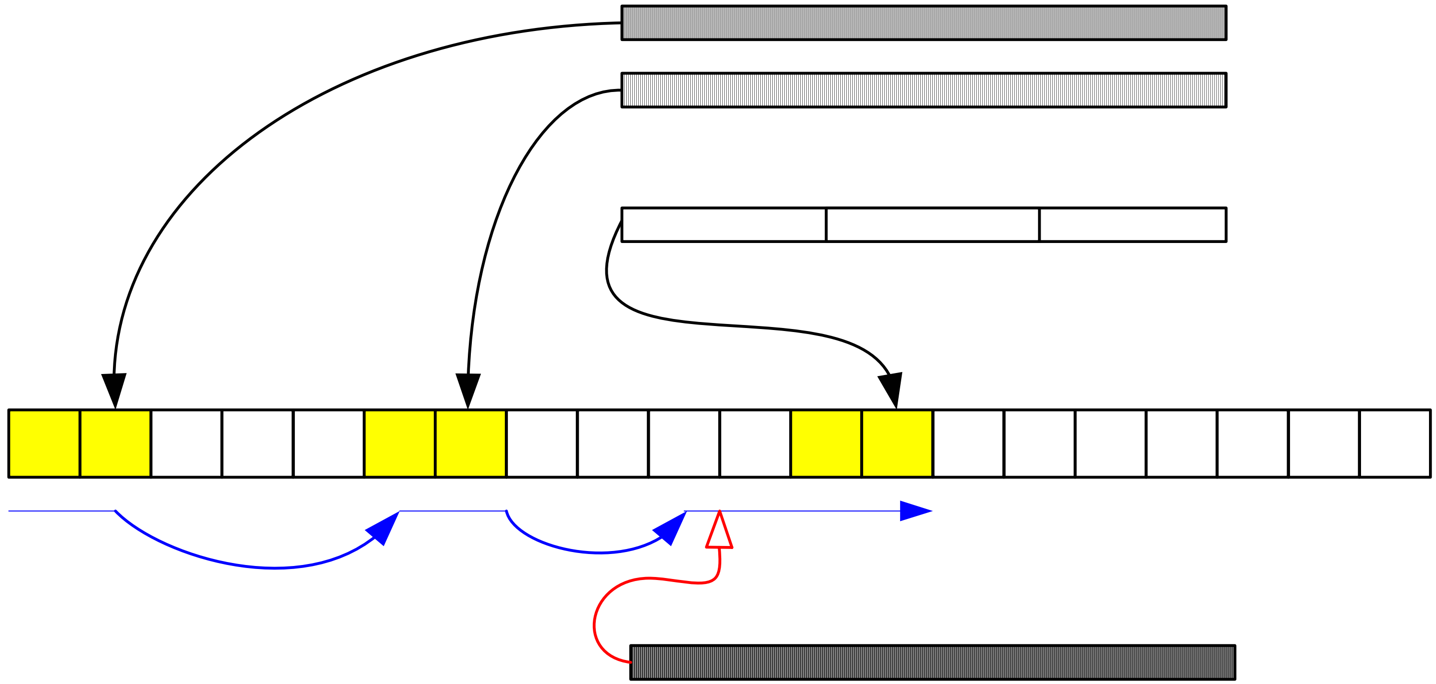
Origin Receive Pattern



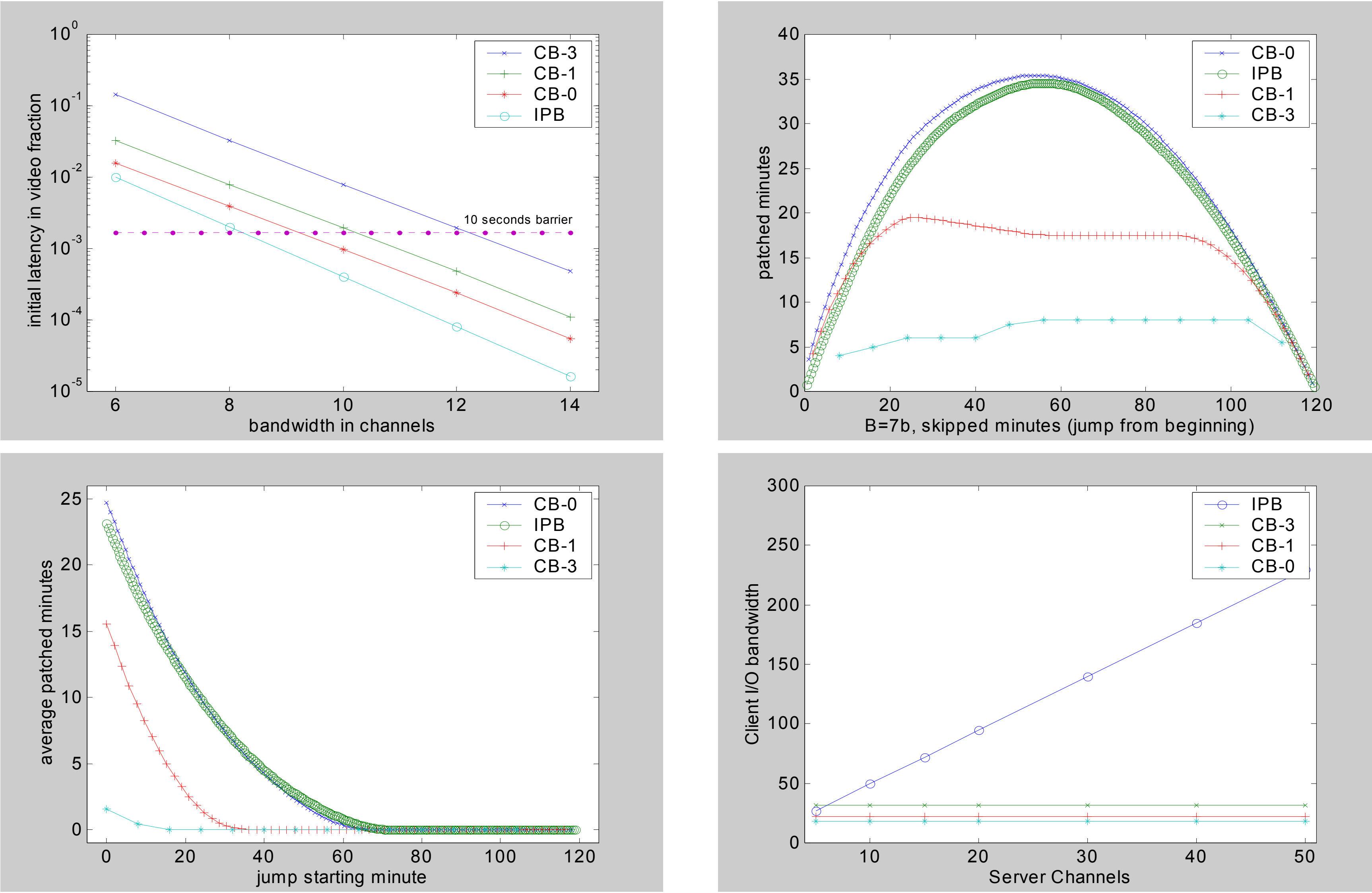
New Receive Pattern

- Using <u>supplemental channels</u> to expedite the downloading of the segments locally.

Illustration



Comparisons



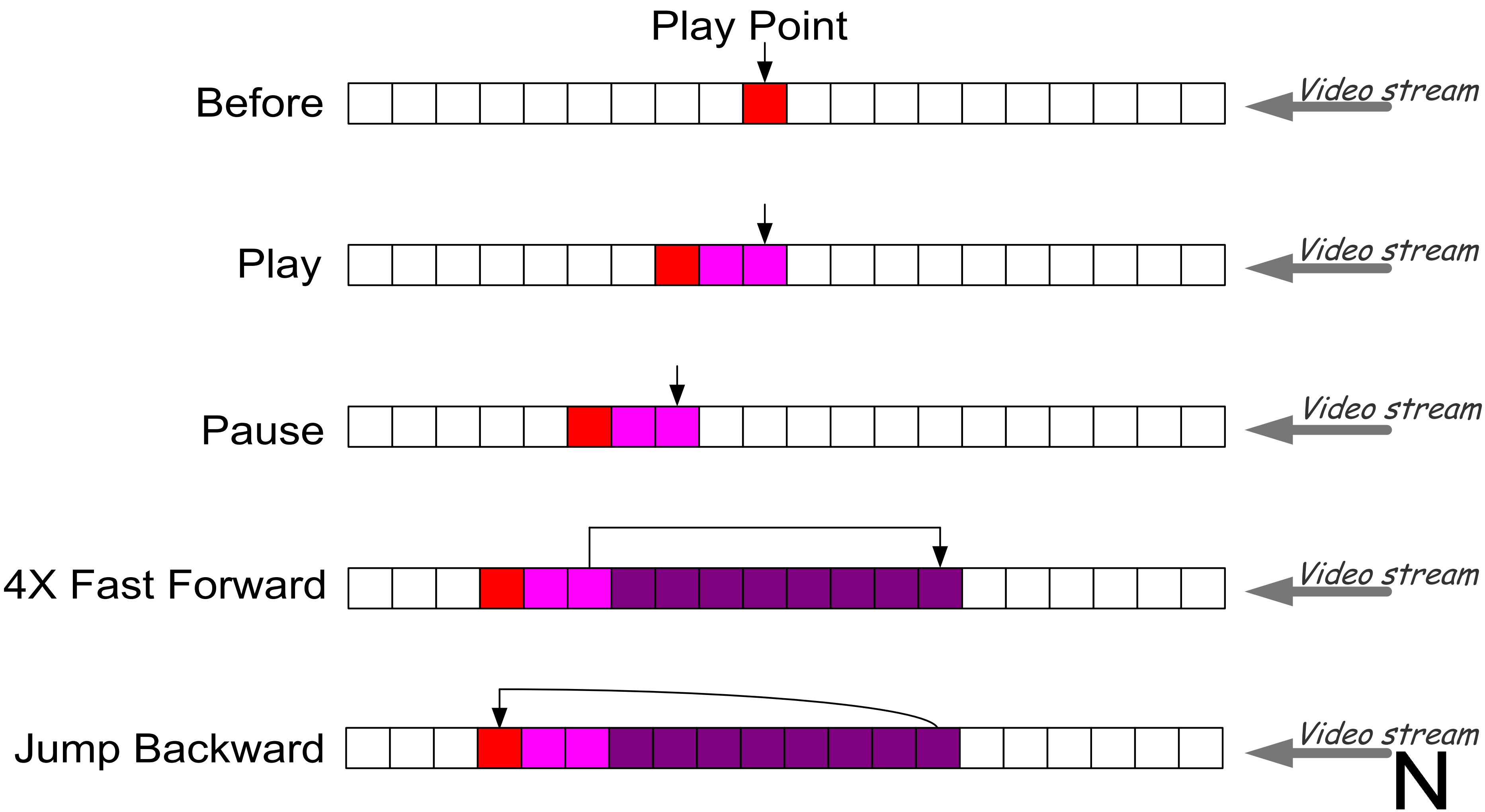
✓ Further enhancements are undergoing.

VCR-like Interactivity

- Challenges:
  - 2x, 4x, or more faster playbacks.
    - Slow motion: $\frac{1}{2}$x, $\frac{1}{4}$x, etc.
    - Pause:
  - Jump Forward/Backward.

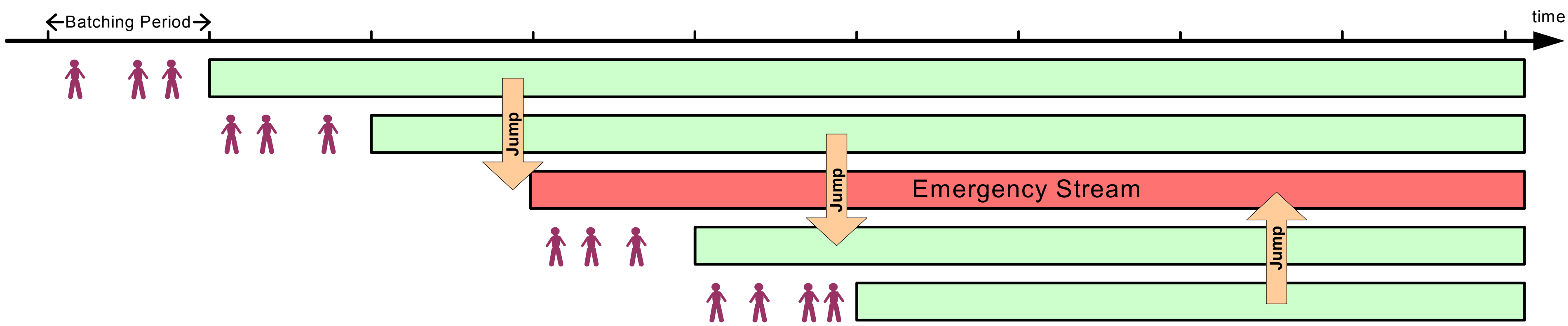
- Existing Solutions:
  - Server Patching: set aside some channels.
    - Blocking!
  - Large client buffer: cache the entire video.
    - Alternative: using VCR windows.

VCR Interaction Using Client Buffer



Interaction Using Batching [Almeroth96]

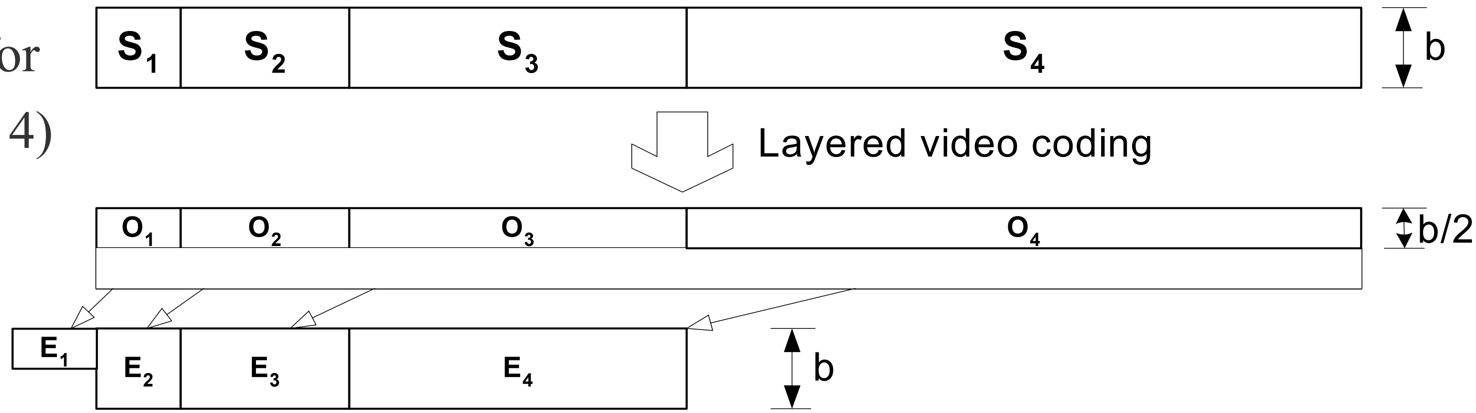
- Requests arriving during a time slot form a multicast group
- Jump operations can be realized by switching to an appropriate multicast group
- Use an emergency stream if a destination multicast group does not exist



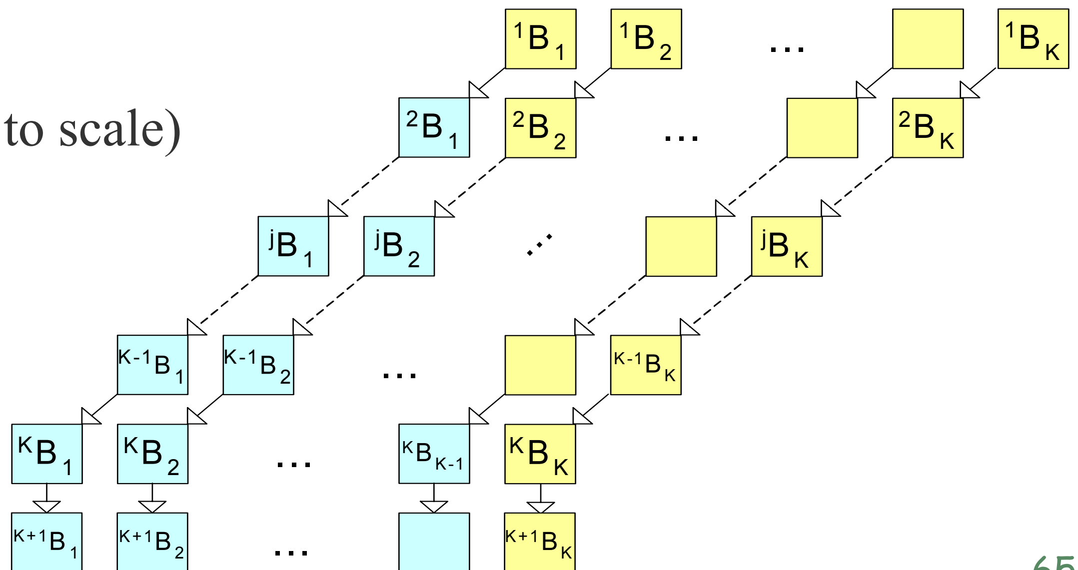
Joint Broadcast (JB)

- Interleaving Frames for r-x playbacks
  - 30 fps for 2x, instead of 60 fps.

(a) Data Segmentation for 1x & 2x playbacks. ($K = 4$)

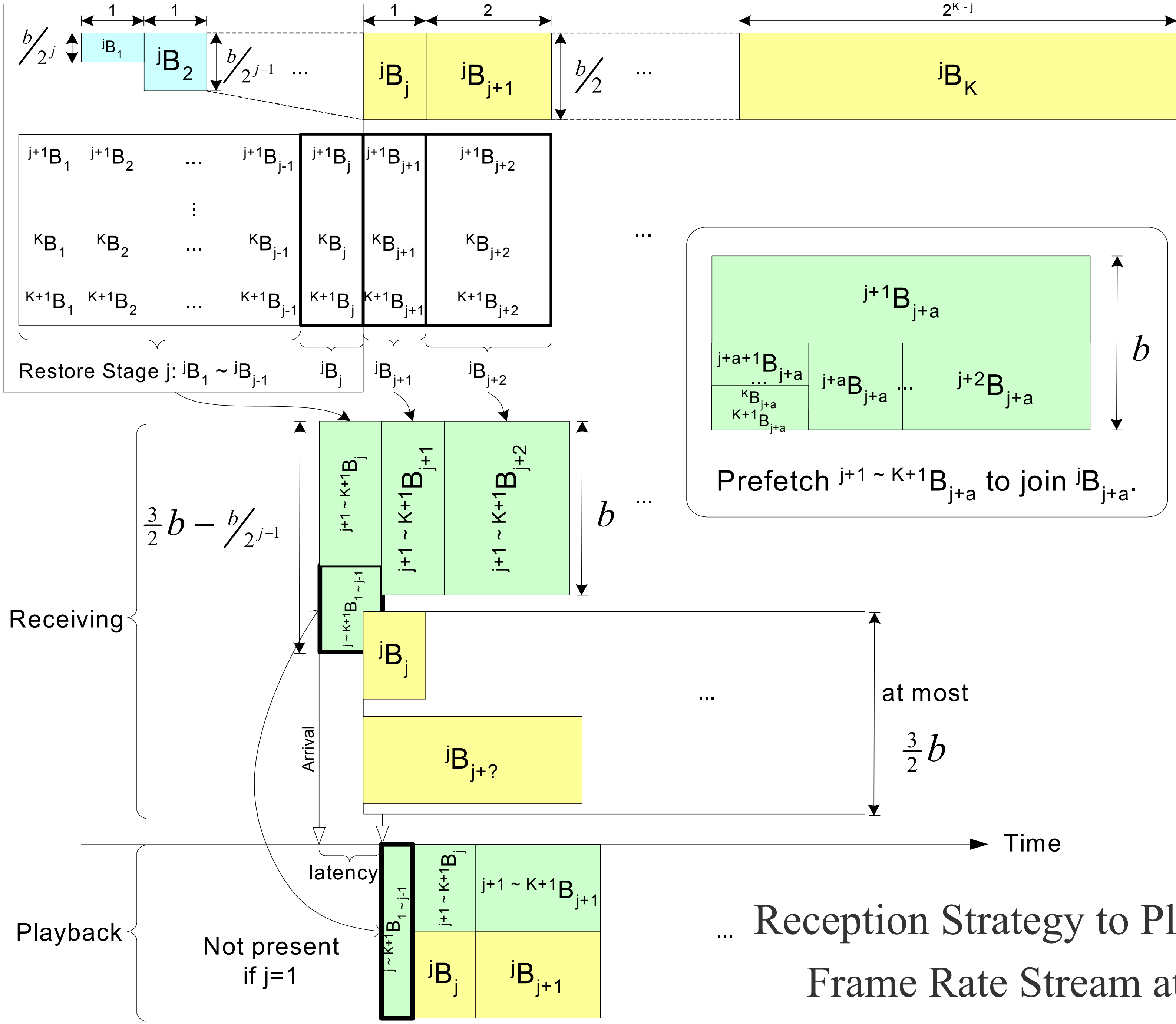


(b) Formulation of Recursive Clusters (not to scale)



⇒ Joint Broadcast !!

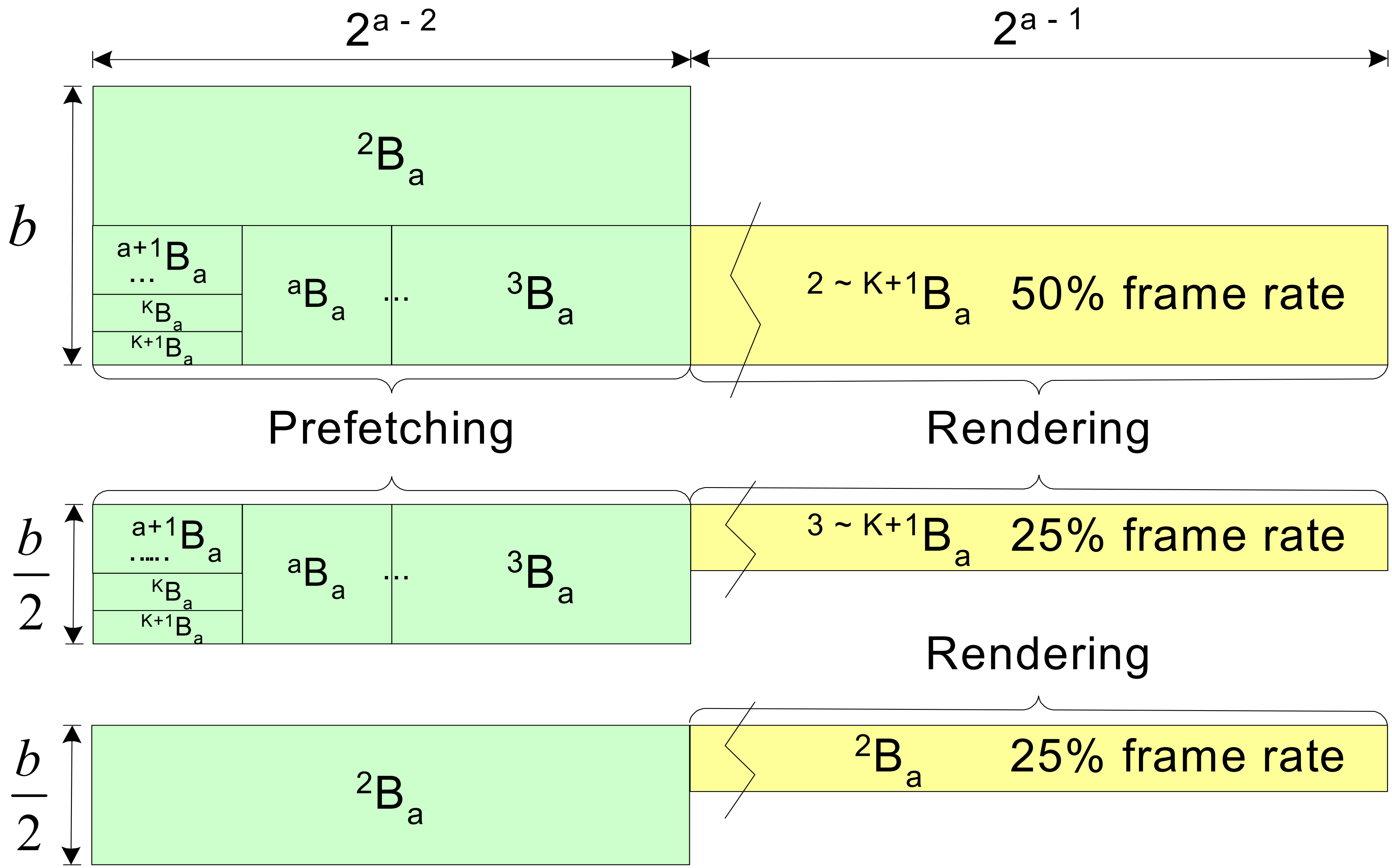
Clusters of Segment Blocks.



Reception Strategy to Playback the Full Frame Rate Stream at 2^{j-1}x speed.

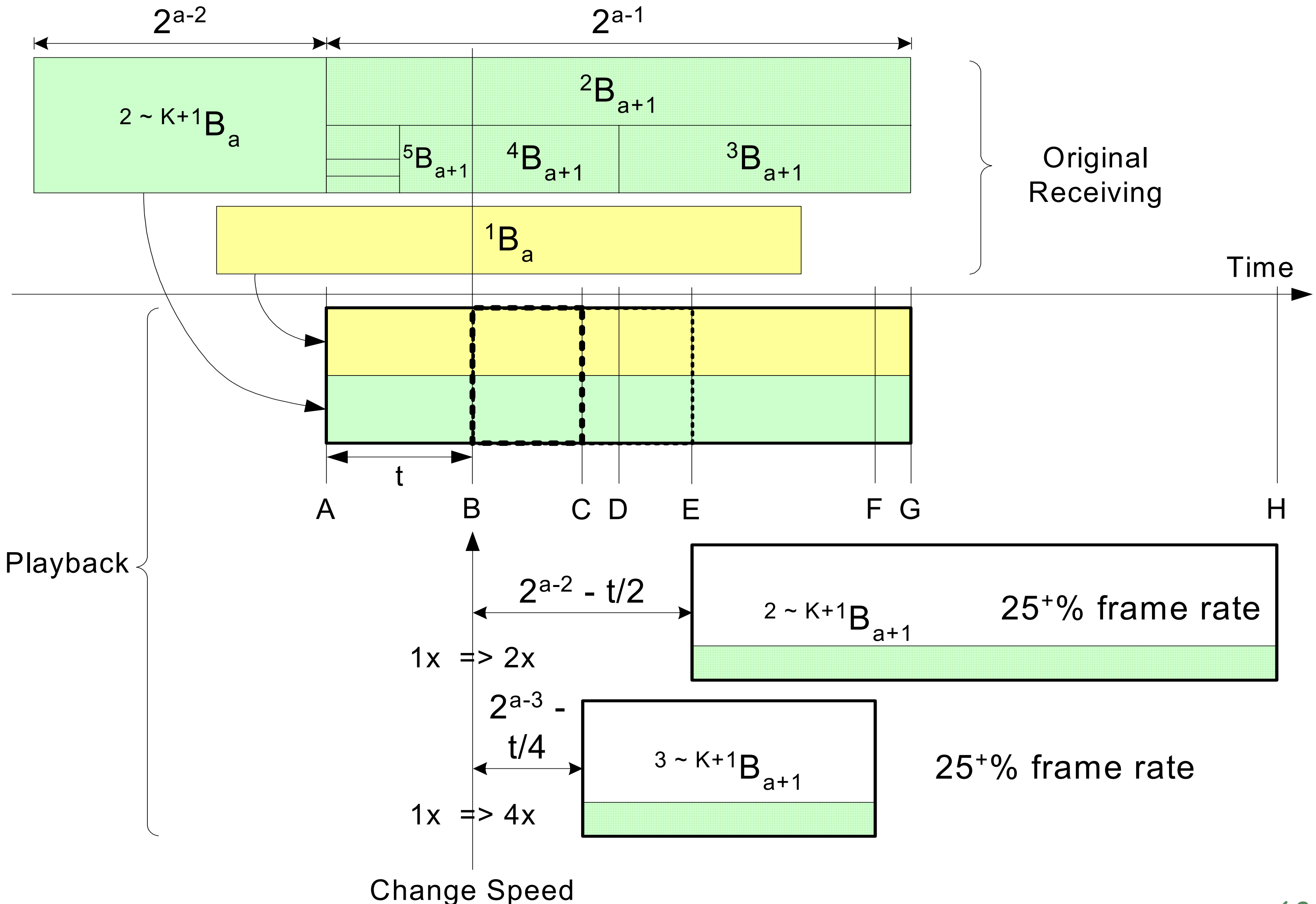
Adaptive Tuning for BW Heterogeneity

Customizable Reception Strategies.



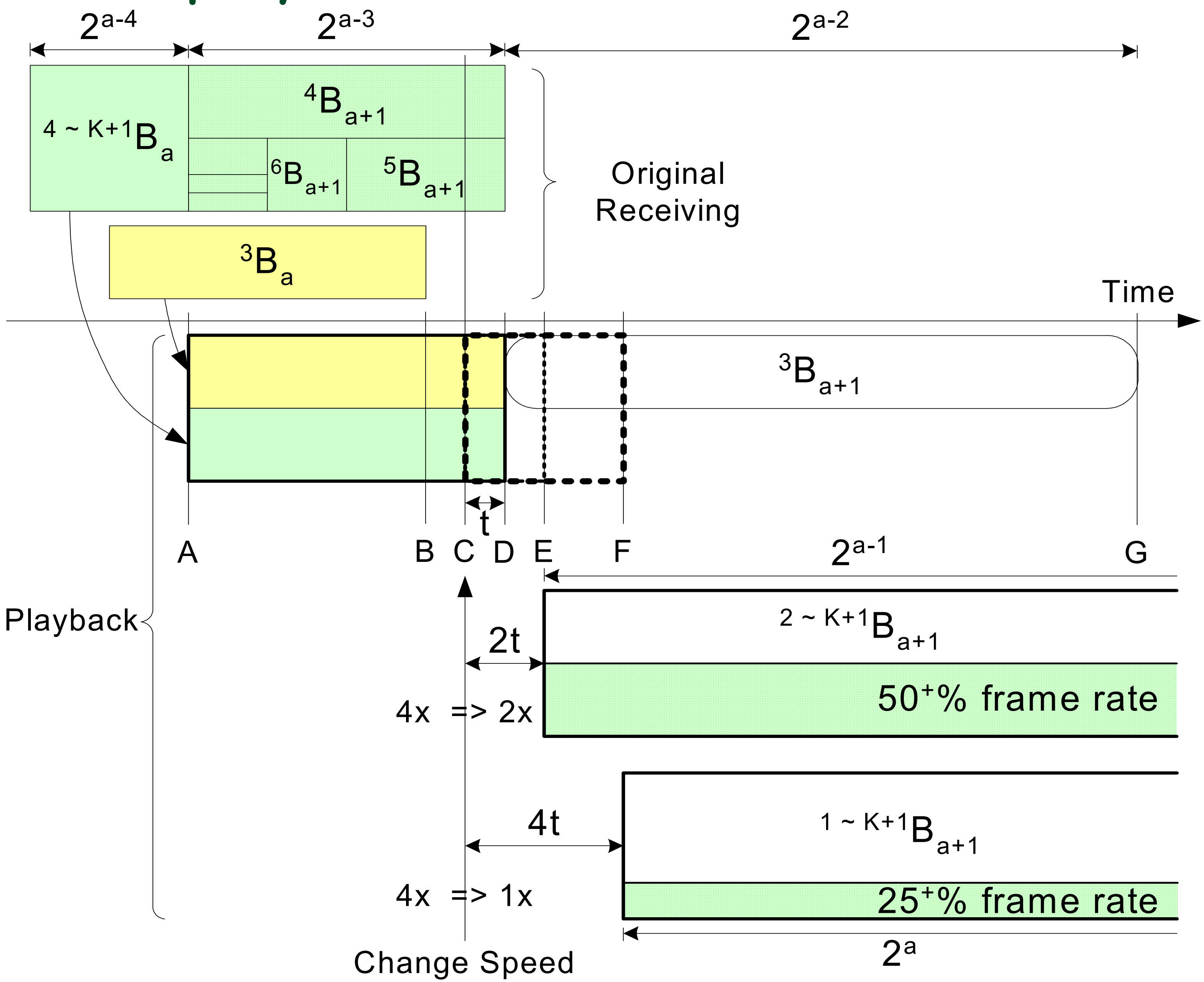
Speed up the playback

1x ⇒ 2x or 4x.



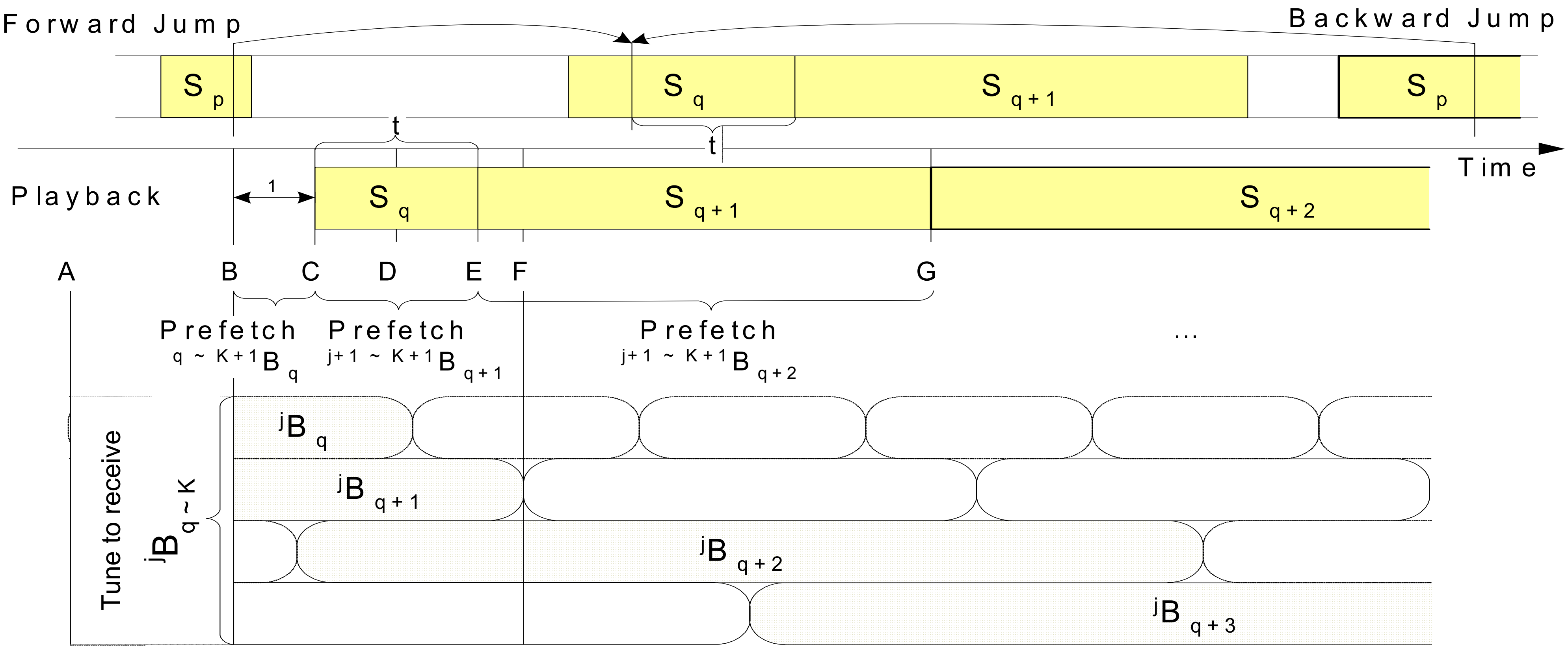
Slow down the playback

$4x \Rightarrow 2x$ or $1x$



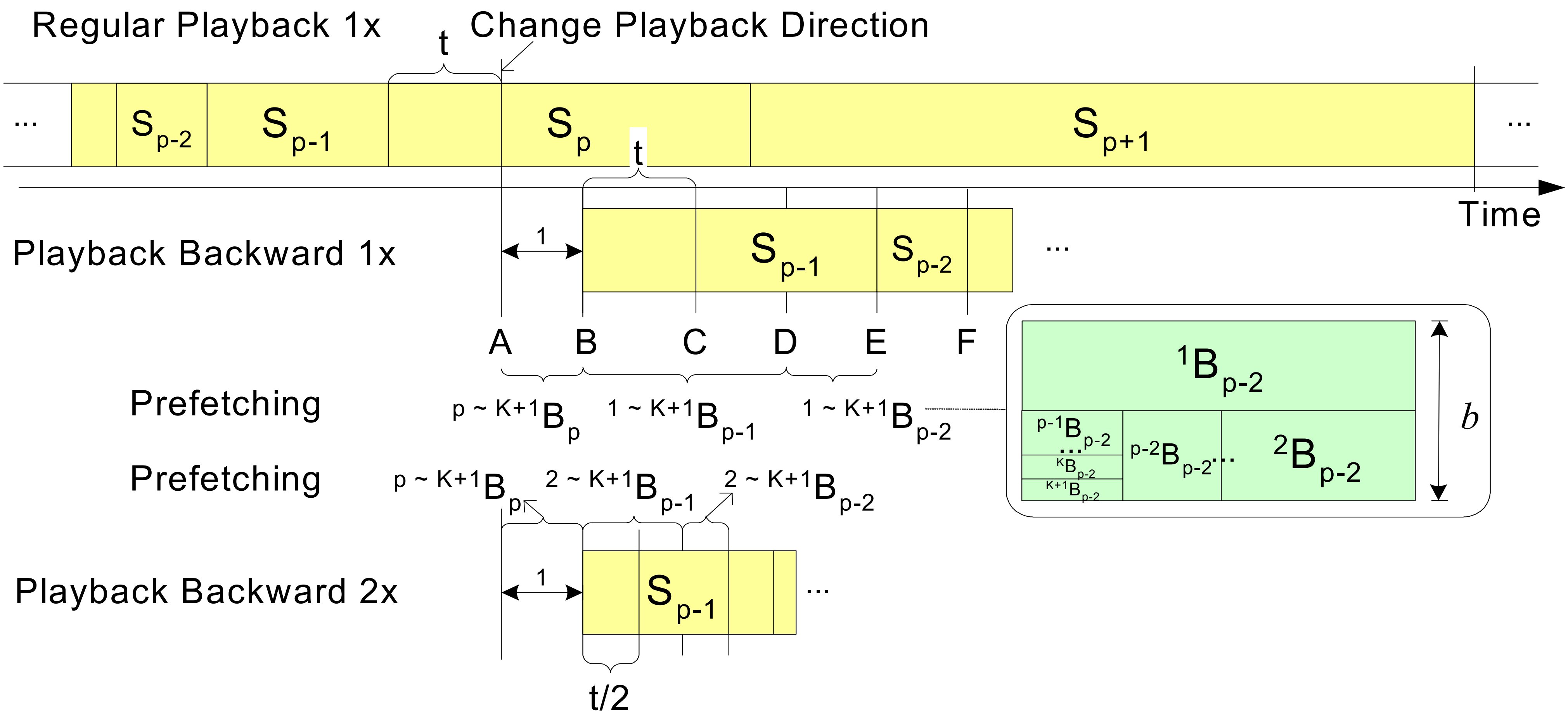
Forward/Backward Jumps

Assume 2^{j-1}x speed



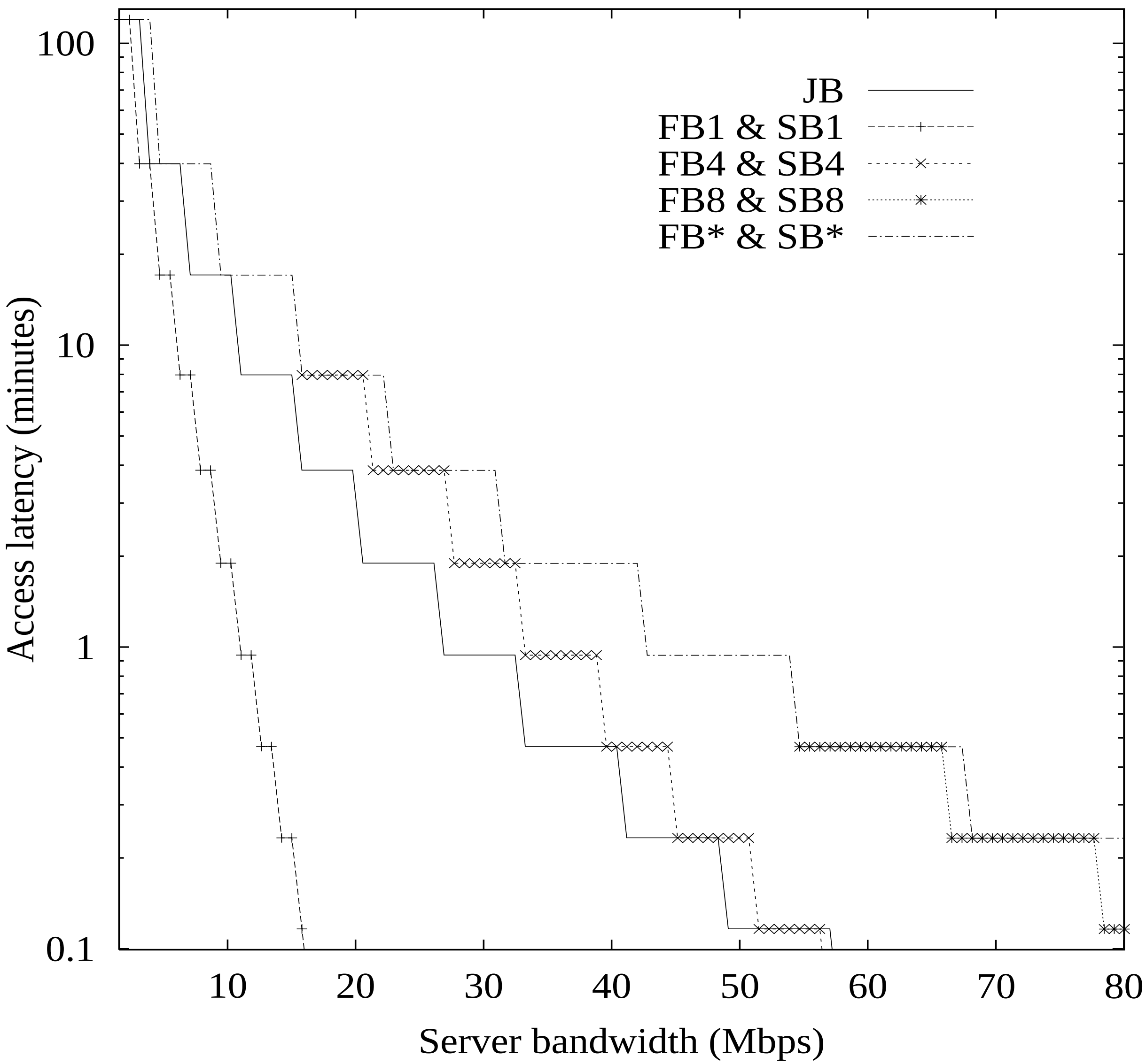
Play Backward Operations

1x ⇒ -1x or -2x

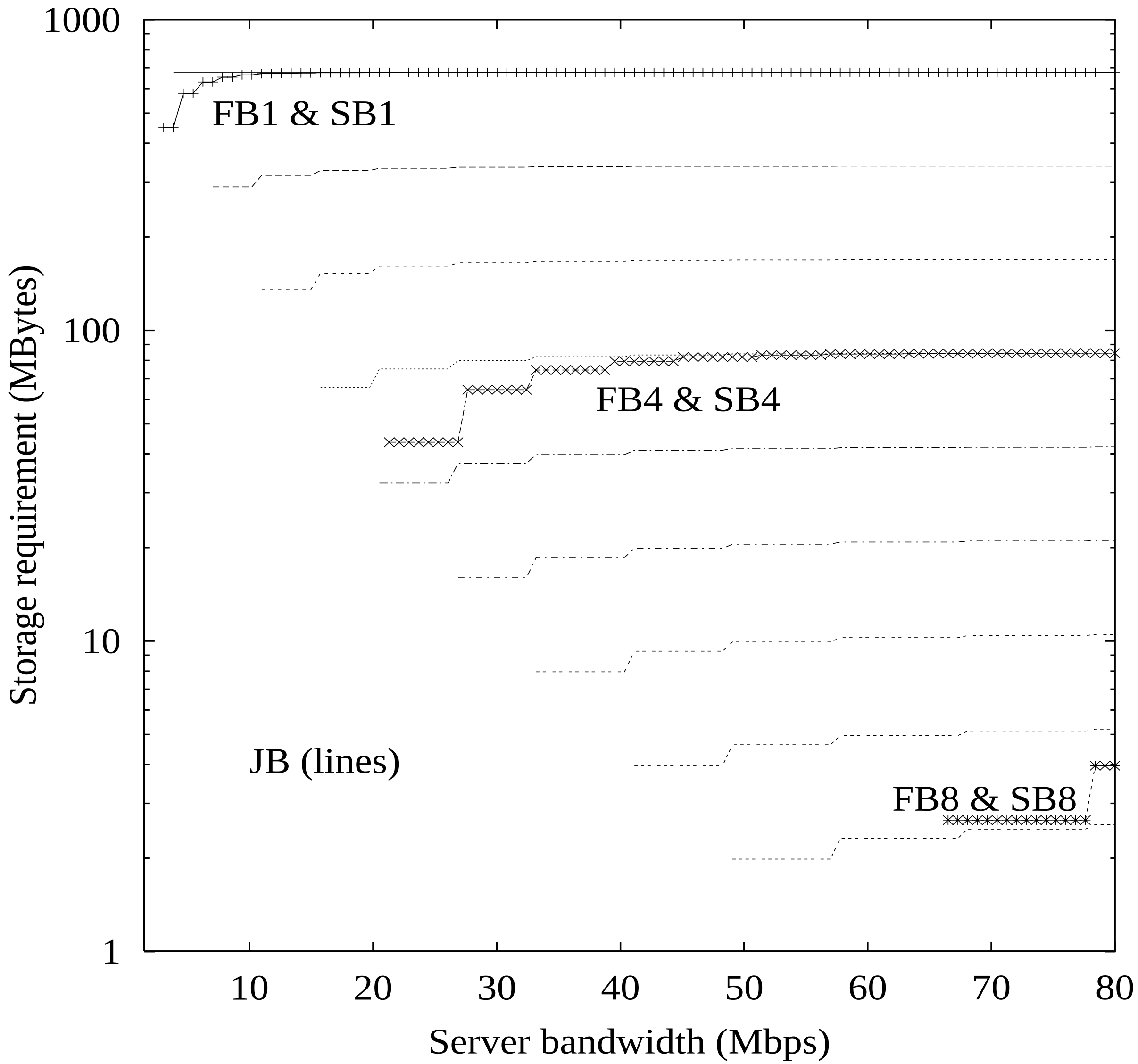


Performance Study

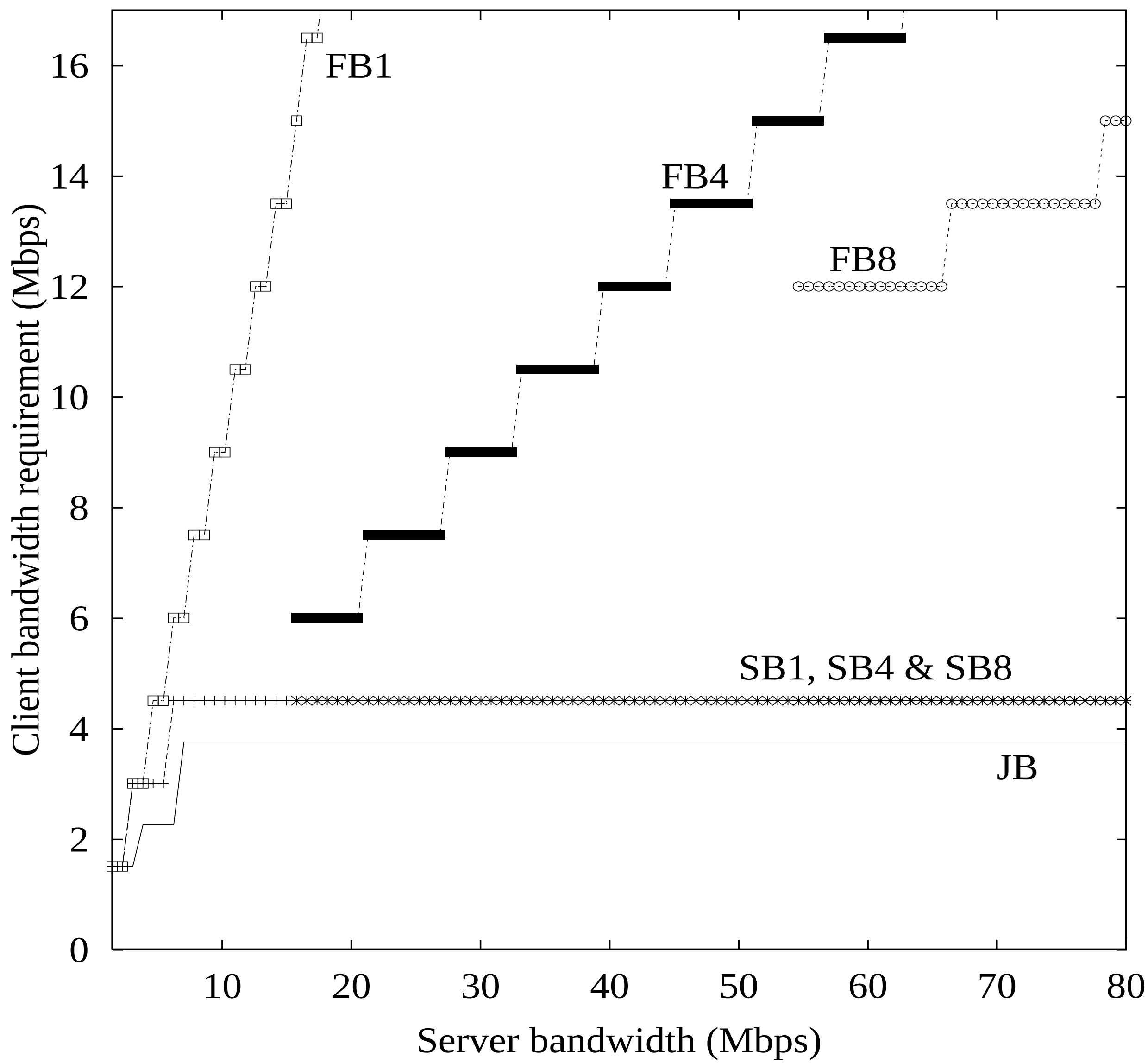
❖ Access Latency:



Storage Requirements



Client BW Requirements

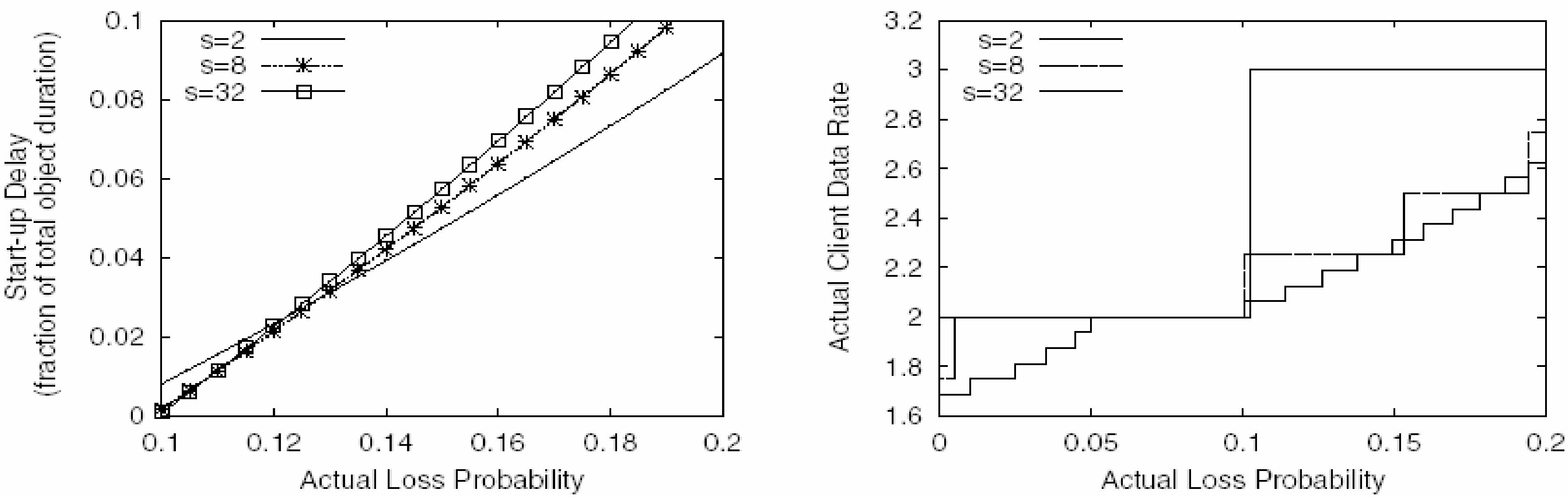


Remarks

- The provision of flexible VCR-styled operations requires faster delivery.
  - Multiple recursive clusters design.
    - Multiple speeds support
    - No need to receive data beyond full fps.
  - Scalability: Non-blocking
    - Server intervention is completely eliminated.
    - VCR Delay = Initial Delay.
    - Consuming less server bandwidth.
  - Client Heterogeneity.
    - At a cost of reduced frame rate.

Heterogeneity

- Reliable PB allows to trade a higher startup delay for a larger packet loss rate.
- It also allows a client with higher BW to listen to more streams so as to sustain higher loss rate.



(a) Start-up Delay vs. Loss Probability

(b) Client Data Rate vs. Loss Probability

Figure 9: RPB Performance for Heterogeneous Clients $(B = 10, n = 2, p = 0.1)$

Heterogeneity

- Fine Granularity Scalability in MPEG-4.
  - Bit-plane coding.
  - Various VLC tables for each bit-plane.
- Encoding methods:
  - Layered video coding approaches.
  - Different code rates for I, P, & B frames.
  - MPEG-2 scalability:
    - SNR
    - Spatial
    - Temporal
    - Frequency

Concluding Remarks

- Goal of Streaming Protocols:
  - Convenient:
    - They should allow a tunable small startup delay.
  - Tolerant:
    - They should resist heterogeneous packet loss rates.
  - Reliable:
    - They may incorporate FEC-like correction capability.
  - Efficient:
    - They should require minimal client receiving bandwidth.
  - Scalable:
    - The broadcast schedule should be generalized for clients with different BWs, arrival times, VCR operations, etc.