The Reliable Multimedia Streaming Server:

the good, the bad and the ugly

John C.S. Lui
Department of Computer Science & Engineering
The Chinese University of Hong Kong

Outline

- **Motivation**
- **Target Features**
- **Server Module**
  - **Disk Sub-Systems**
    - Data Layout
    - Data Retrieval Process
    - Admission Control
    - Fault Tolerance
  - **Network Module**
- **Client Module**
  - Decoder
  - Buffer Management
- **Current Extensions, Conclusion and Lessons Learn**

Motivation

- **Personal challenge: from paper to real system**
- **Challenges of Continuous Multimedia Delivery:**
  - **large data size (e.g, digital library)**
  - **heterogeneous bandwidth requirements**
  - **continuity requirement**
  - **admission control and QoS support**
  - **multimedia object synchronization**
  - **reliability and fault tolerance**
  - **from LAN to Internet**
- **Platform for future research**

Features

- QoS support (e.g, frame rate, packet loss)
- Efficient Admission Control
- Able to support heterogeneous bandwidth requirements.
- Efficient I/O Scheduling
- Full VCR functionalities
- Reliability and capability to rebuild failed disk
- Platform independent
- Free: under GPL. Installation and user manuals are available.

Current Available Services

- **Movie-on-demand:** Available titles:
  "Terminator I", "The Joy Luck Club", "True Lies",
  "Taxi driver", "A Few Good Men", "Sense & Sensibility".
- **Musical Jukebox:** Jazz, Pop, Symphony, Classical
- **Education-on-demand:** Available lectures:
  - System Performance Evaluation (semester worth)
  - Introduction to Operating System
  - Distributed Systems and Computer Networks
  - Research papers presentation

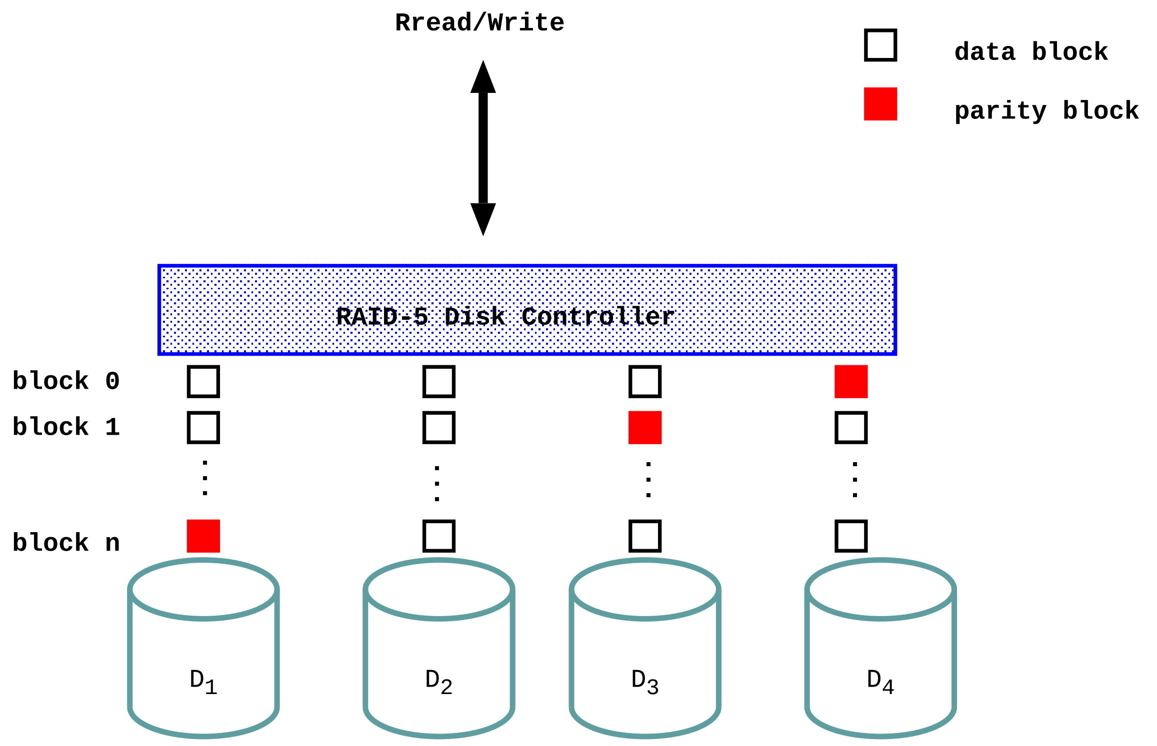
- **Seminars:** Available seminars:
  - DLS by Prof S. Lam on *"Security Services for Internet Flows and Multicasts"*
  - Colloquium by Prof Hui Zhang on *"QoS Control and Resource Management in the Internet"*
  - Colloquium by Prof. D. Yau on *"Real-time and Differentiated Web Services Using Orthogonal Kernel Resource Allocations"*
  - etc.
- **Collaboration with RTHK:**
  - Top 10 Pop Music Award
  - Outstanding Chinese Series
  - etc.

Technical Issues

- **Server**
  - File System and Storage Management
  - I/O scheduling
  - Admission Control
  - Buffer Management: Network transmission and recovery
  - Fault Tolerance
- **Client**
  - Video Decoder
  - Buffer Management: Network transmission, data recovery, data filtering services, object synchronization.

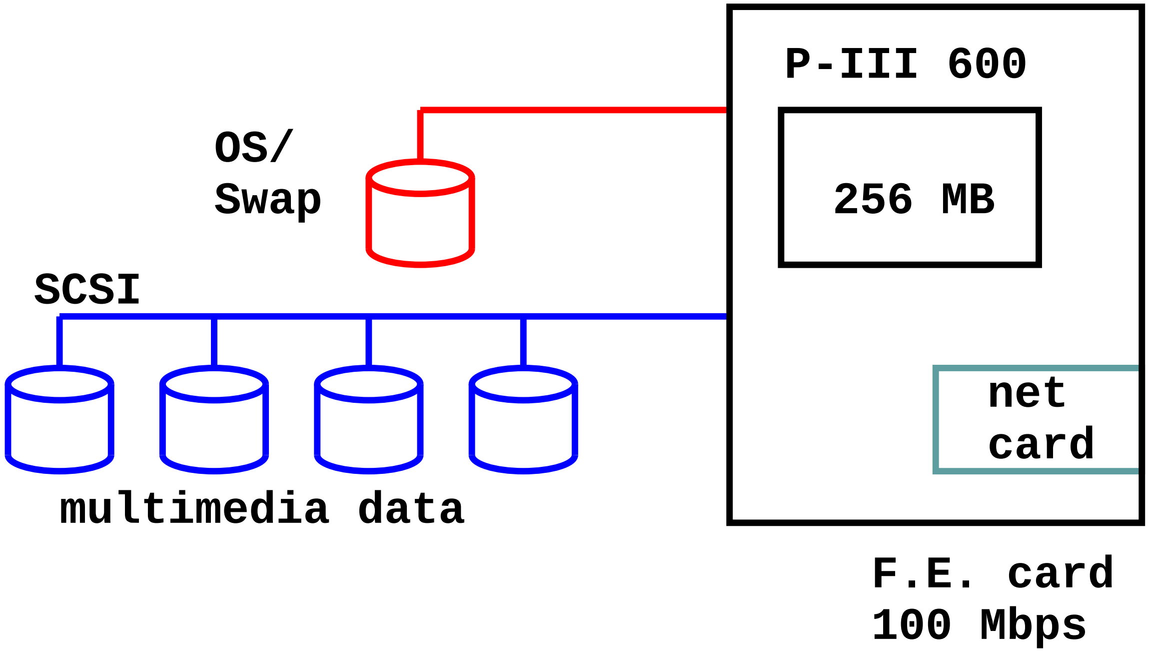
File System

- **Large amount of data, needs many disks.**
- **Also need disk arms to maximize number of concurrent clients.**
- **Since there are many multimedia objects, we need to stripe objects among multiple disks.**
- **Need to consider reliability.**
- **Naive (and WRONG) way is to use RAID-5.**



- Reasons:
  - Can't control and directly access physical data block.
  - Major reduction of bandwidth.

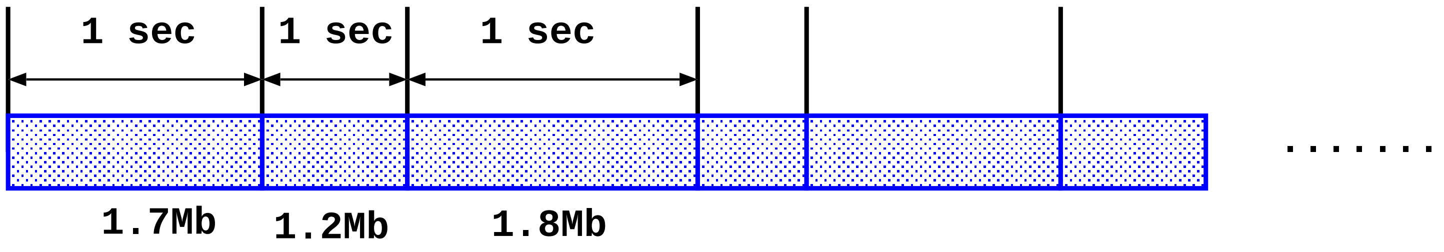
- **Approach:** *raw device* **(need to provide our file system support)**

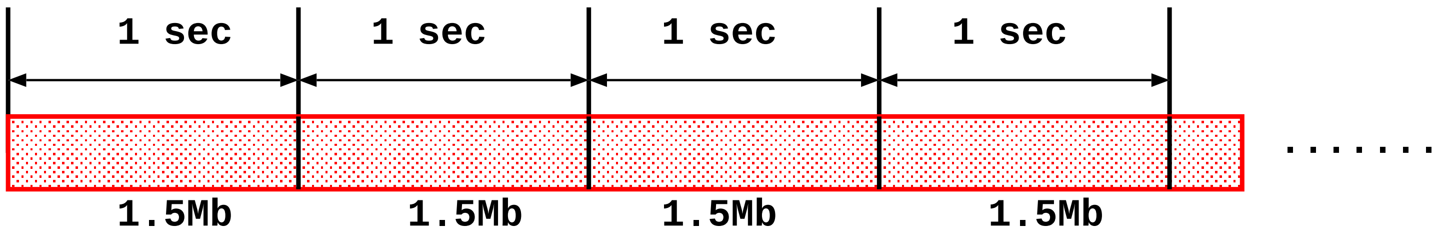


- **Currently, we can support up to 55 concurrent MPEG1 clients.**

Data Striping

- Striping choices
  - Time striping
  - Space striping

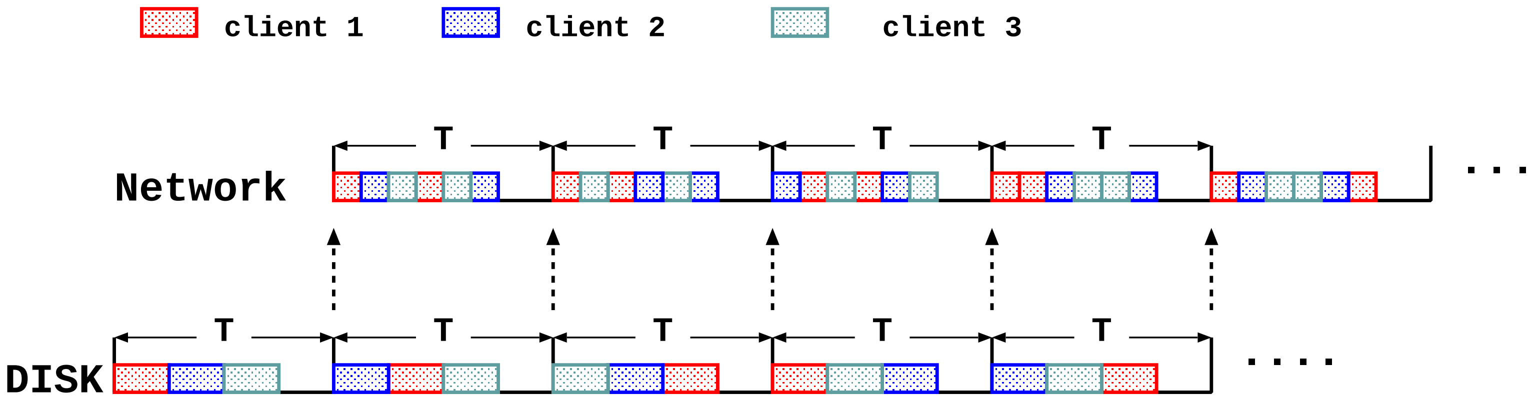




- **Time striping requires minimum client's buffer, but many difficult issues in:**
  - **VBR transmission**
  - **Admission Control**
  - **VCR support**
- **Space striping requires move "buffer" management and adaptation. (Stick to Internet Principle).**

Disk Scheduling: Cycle-based Scheduling

- each cycle is of length T, let N be the number of data blocks we can retrieve from a disk.
- the transmission of data retrieved in the i^{th} cycle does not start until the beginning of the $(i+1)^{th}$ cycle.



- T dictates bandwidth efficiency and startup latency

- **Remember, we use space striping**
- **Need to** *guarantee* **that every** T**, the storage system can retrieve the requested data.**
- **Two forms of QoS guarantee:**
  - **deterministic guarantee**
  - **stochastic guarantee**
- **Concentrate on a single disk first, extend to multiple disk later**

Deterministic Guarantee

- N can be determined by considering the maximum number of data blocks that we can retrieve within a design value of T (e.g, 1 sec).

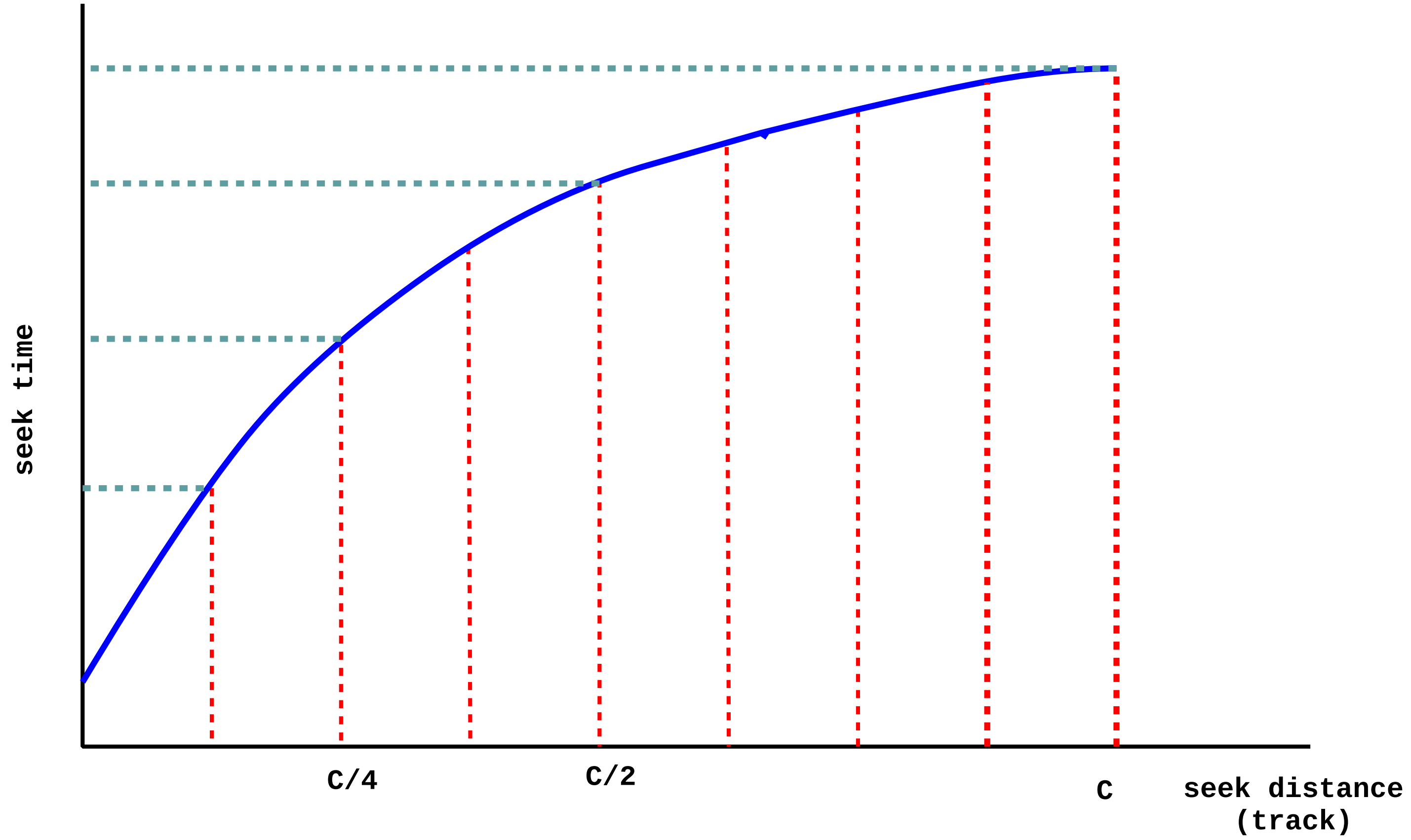
$$T = \tau_{seek}^{max}(N) + N * \left(\tau_{rot}^{max} + \tau_{tfr}^{max}\right) \quad (1)$$

where:

τ_{rot}^{max}	:	*worst case* rotational latency
τ_{tfr}^{max}	:	*worst case* transfer time (i.e., CBR vs. VBR)
$\tau_{seek}^{max}(N)$	:	*worst case* seek time for servicing N requests which are **evenly spaced**

How to obtain these parameters?

- rotational latency
- transfer time



Claim for even-spacing to get worst seek time

Let $T_s(n) = T(d_1) + T(d_2) + \cdots + T(d_n)$ where $T_s(n)$ is the total seek time for servicing n requests in a sweep and $T(x)$ is a seek time given that the seek distance is x. We want:

$$\begin{aligned} \max T_s(n) &= T(d_1) + T(d_2) + \cdots + T(d_n) \\ s.t \quad & \sum_{i=1}^{n} d_i = D \end{aligned} \tag{2}$$

where D is the maximum seek distance.

Because $T()$ is a *concave* function, we can use the *Lagrange Multiplier Method* to show that the worst sum seek time is when

$$d_i = \frac{D}{n} \qquad \forall i$$

Stochastic Guarantees

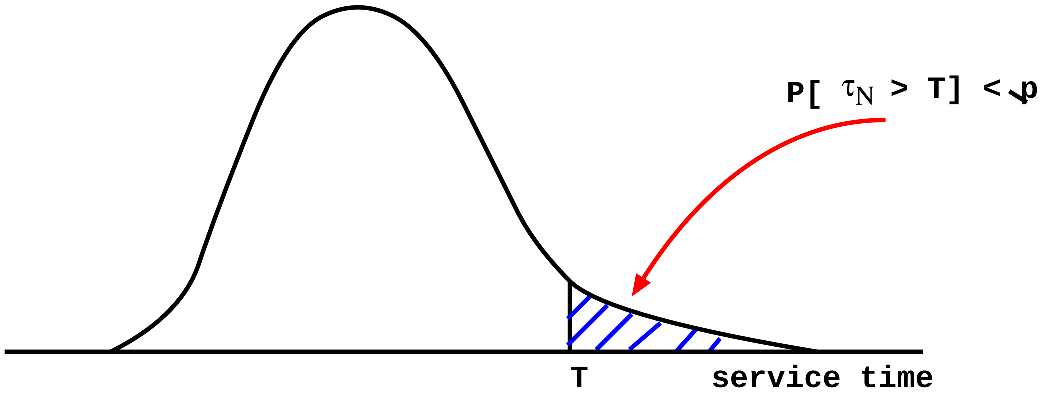
- Define τ_N and related random variables.

$$\begin{aligned}\tau_N &= \tau_{seek}(N) + N \times \tau_{rot} + N \times \tau_{tfr} \\ &\leq \tau_{seek}^{max}(N) + N \times \tau_{rot} + N \times \tau_{tfr}\end{aligned}$$

We want

$$\text{Prob}[\tau_N > T] \leq p \tag{3}$$

where p is some predefined system parameter (e.g., $p = 0.01$).



- We find this bound, simply apply *Chernoff's bound.*

$$F_N^*(s) = e^{(-s\ \tau_{seek}^{max}(N))} \left[F_{rot}^*(s) F_{tfr}^*(s)\right]^N$$

- From disk spec, we can derive $F_{rot}^*(s)$ and $\tau_{seek}^{max}(N)$. Because of space striping, we can derive $F_{tfr}^*(s)$.
- Apply Chernoff's bound, we have:

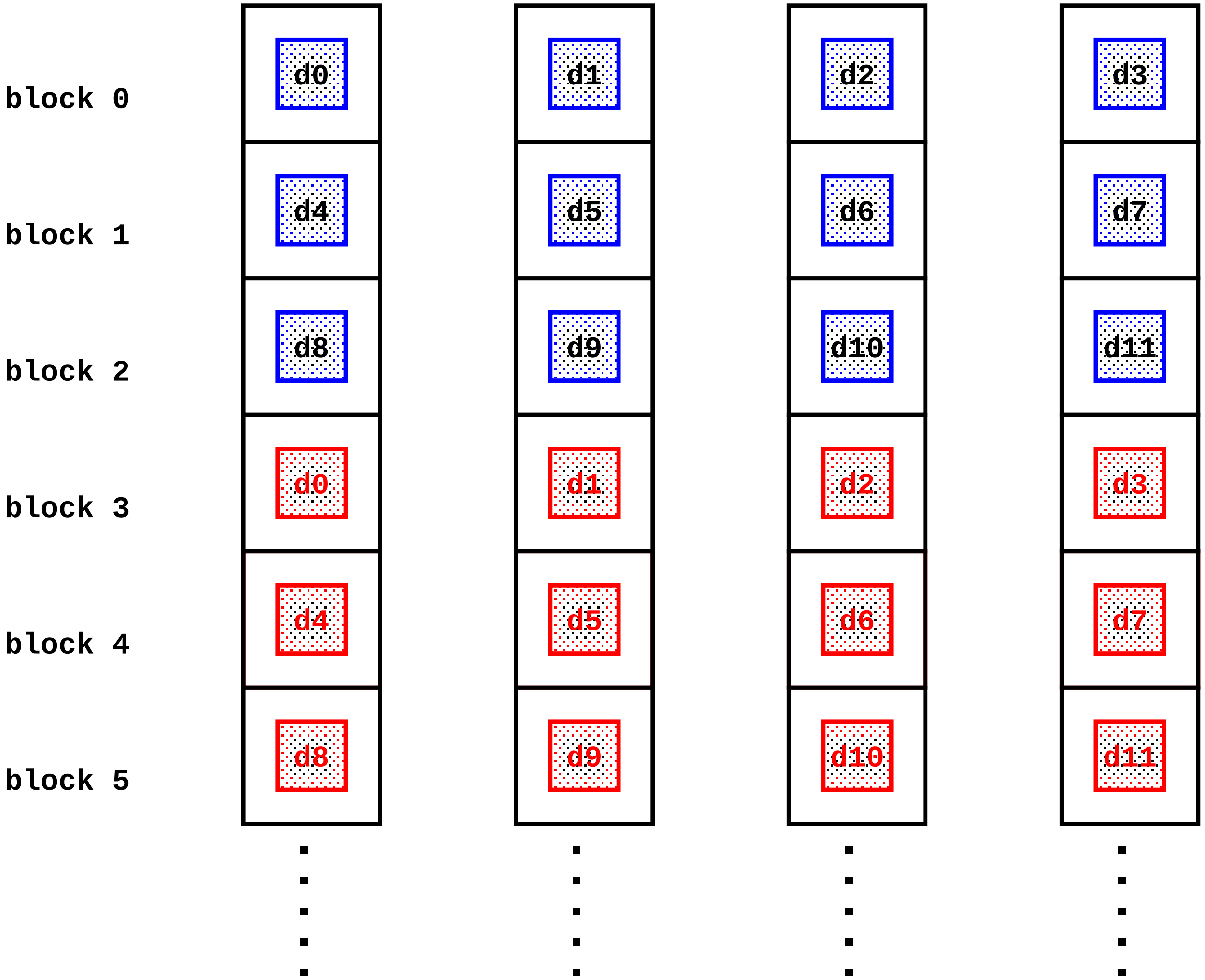
$$\text{Prob}[\tau_N > T] = p \leq \inf_{\theta \geq 0} \left\{ \frac{F_N^*(-\theta)}{e^{\theta T}} \right\}$$

- Efficient numerical solution, that is, given p, T and disk spec, we can find N.
- We now know how to provide QoS for a single disk !!!

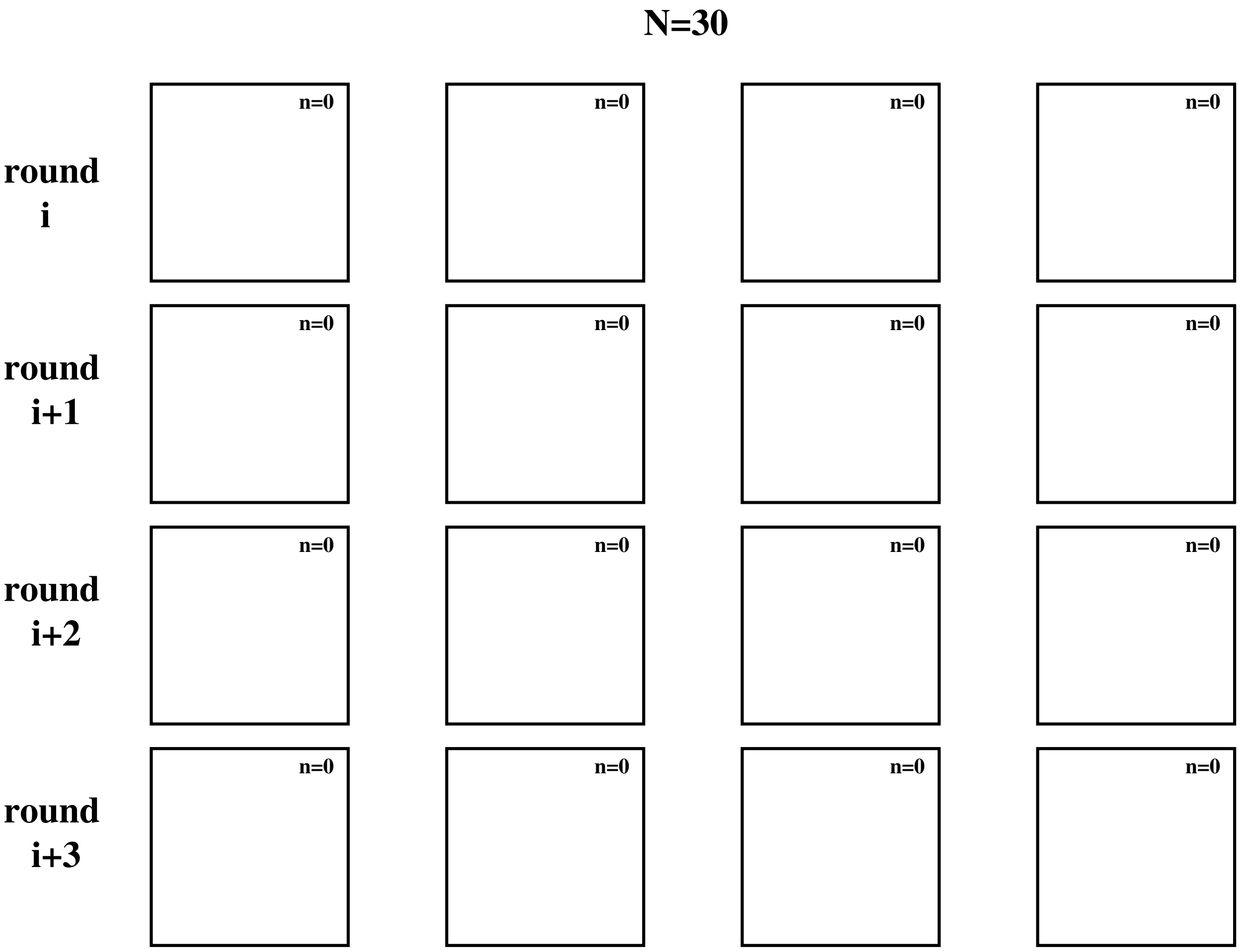
Admission Control

- **Let N_d be the number of disks we have in the server.**
- **Given a movie M, we use space striping. For simplicity of illustration, assume we use round-robin assignment.**
- **We use an admission map to perform the admission control. The admission control map is a simple $N_d \times N_d$ array, with each storing the number of admitted streams.** *The admission map* **is memory resident.**
- **Admission control is a** *simple* **memory look up.**

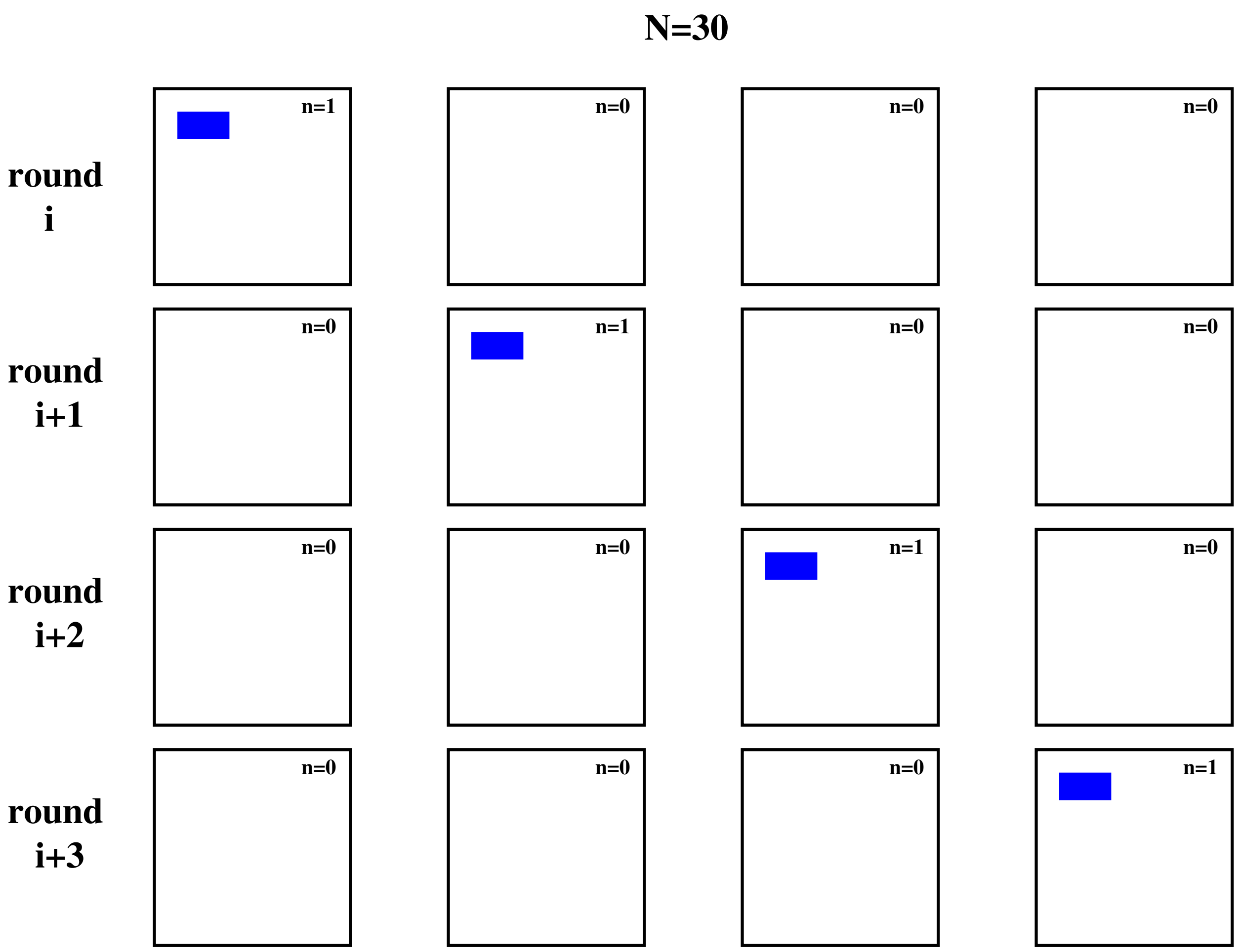
Data Placement where $N_d = 4$



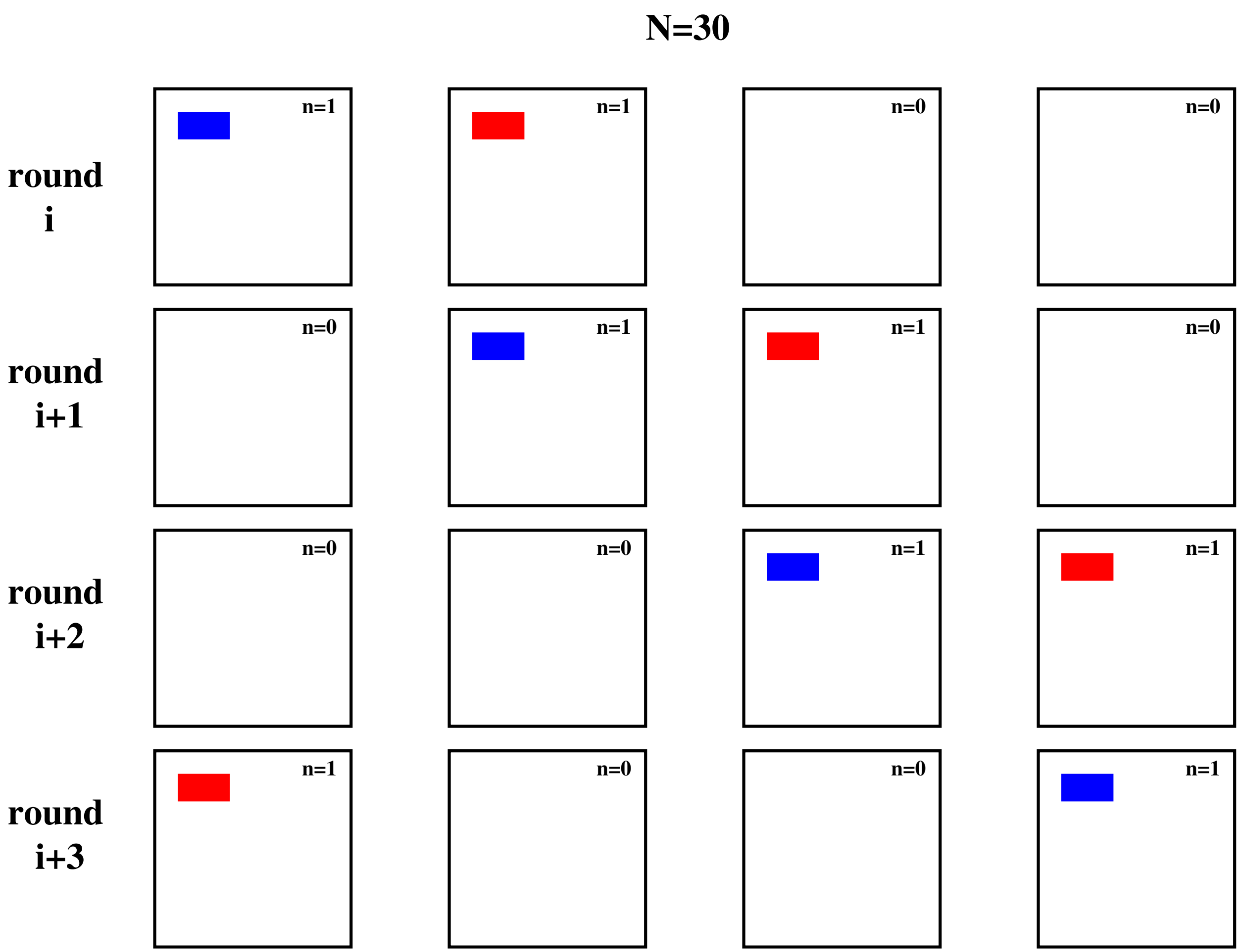
Admission Control Map where $N_d = 4$



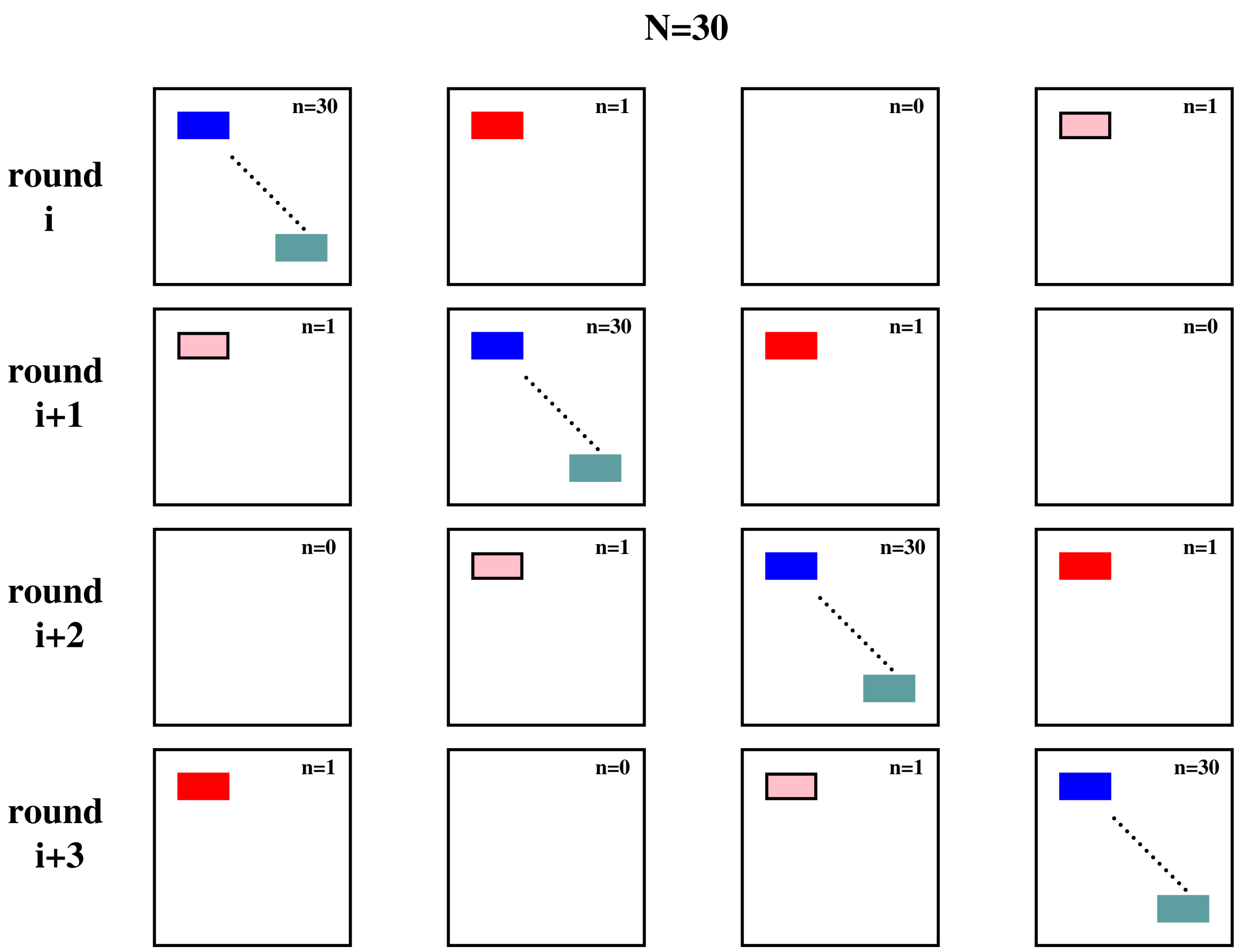
Admission Control Map where $N_d = 4$



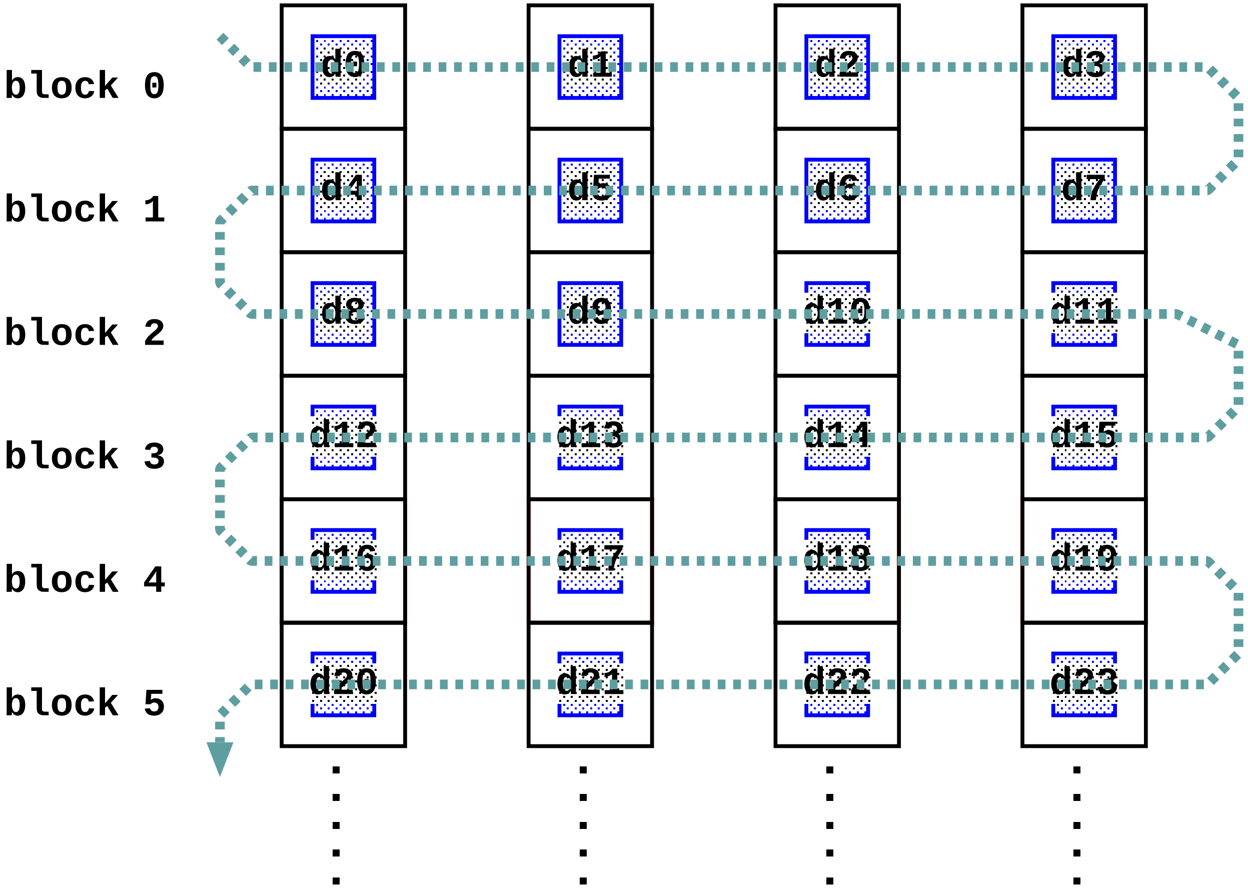
Admission Control Map where $N_d = 4$



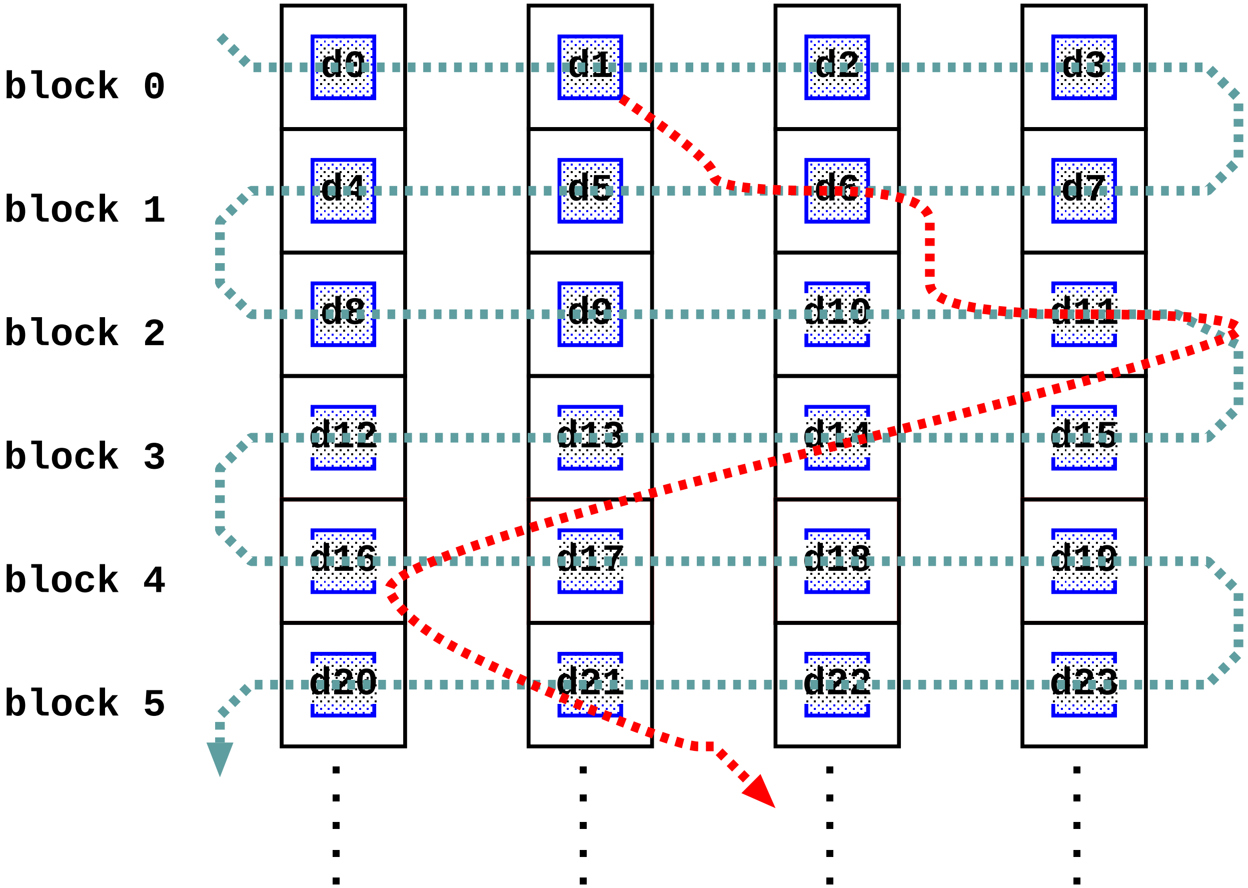
Admission Control Map where $N_d = 4$



VCR Function (where $N_d = 4$) : Fast Forward



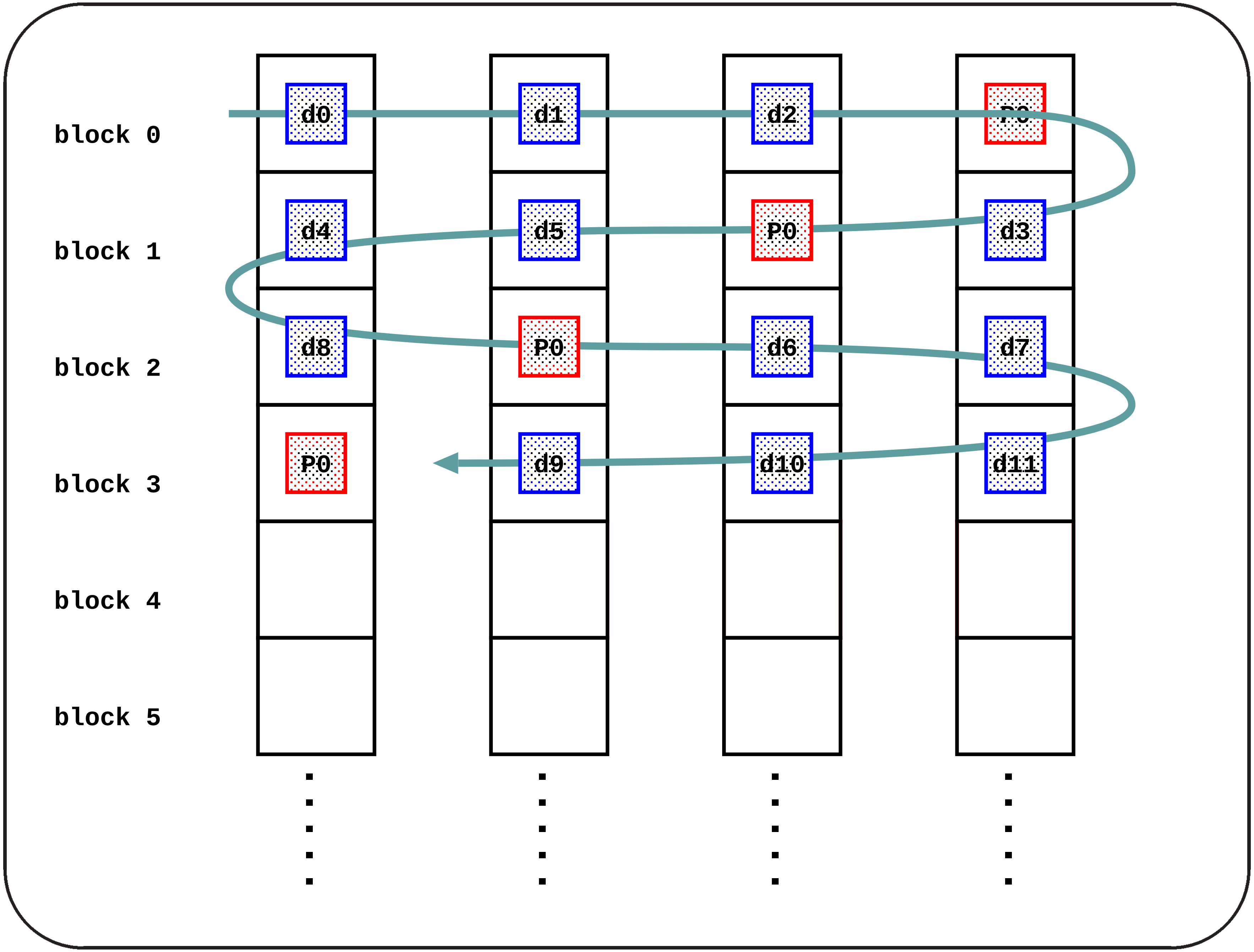
VCR Function (where $N_d = 4$) : Fast Forward

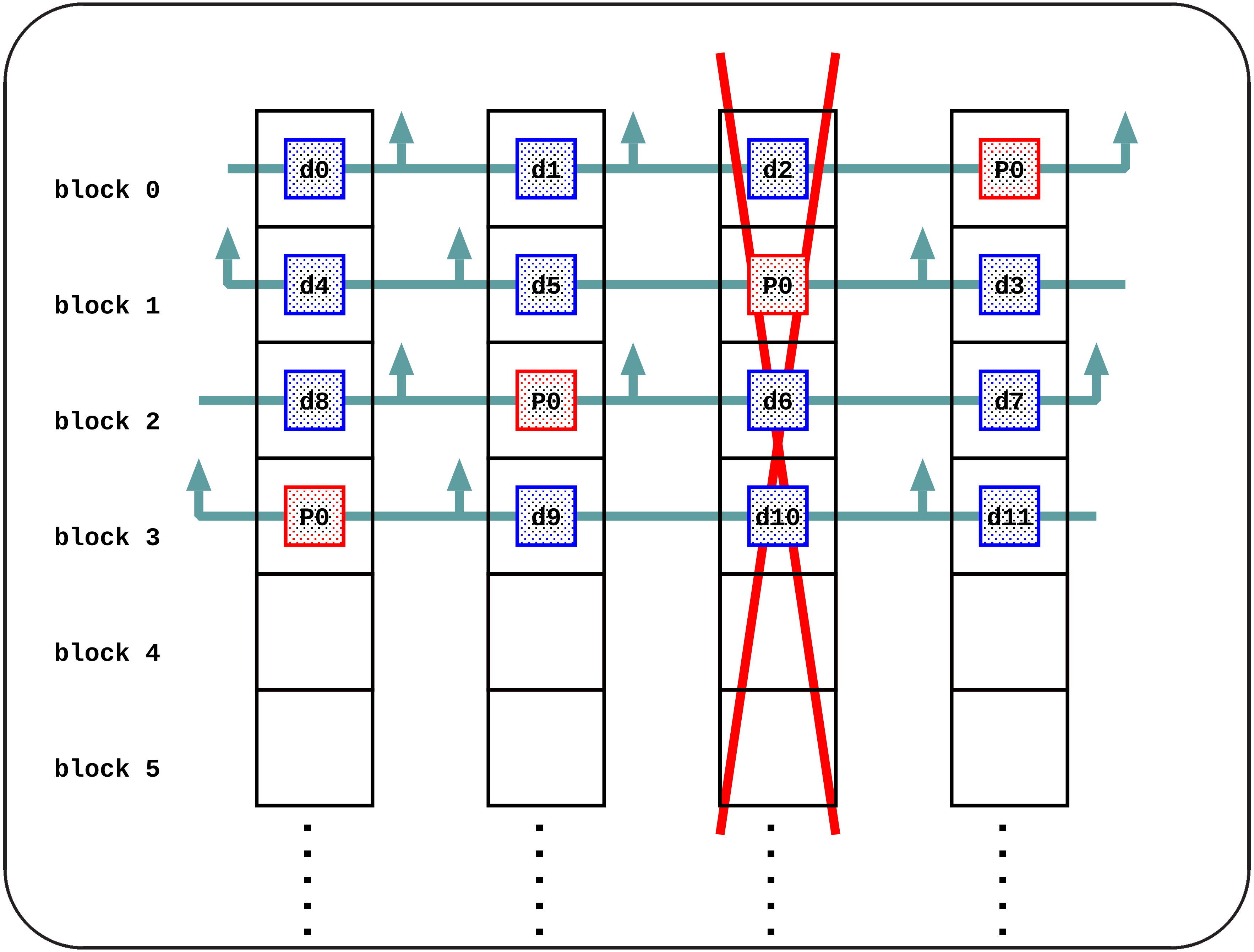


- VCR Speed: integer multiple of 4 times normal display rate

Extension From RAID-5

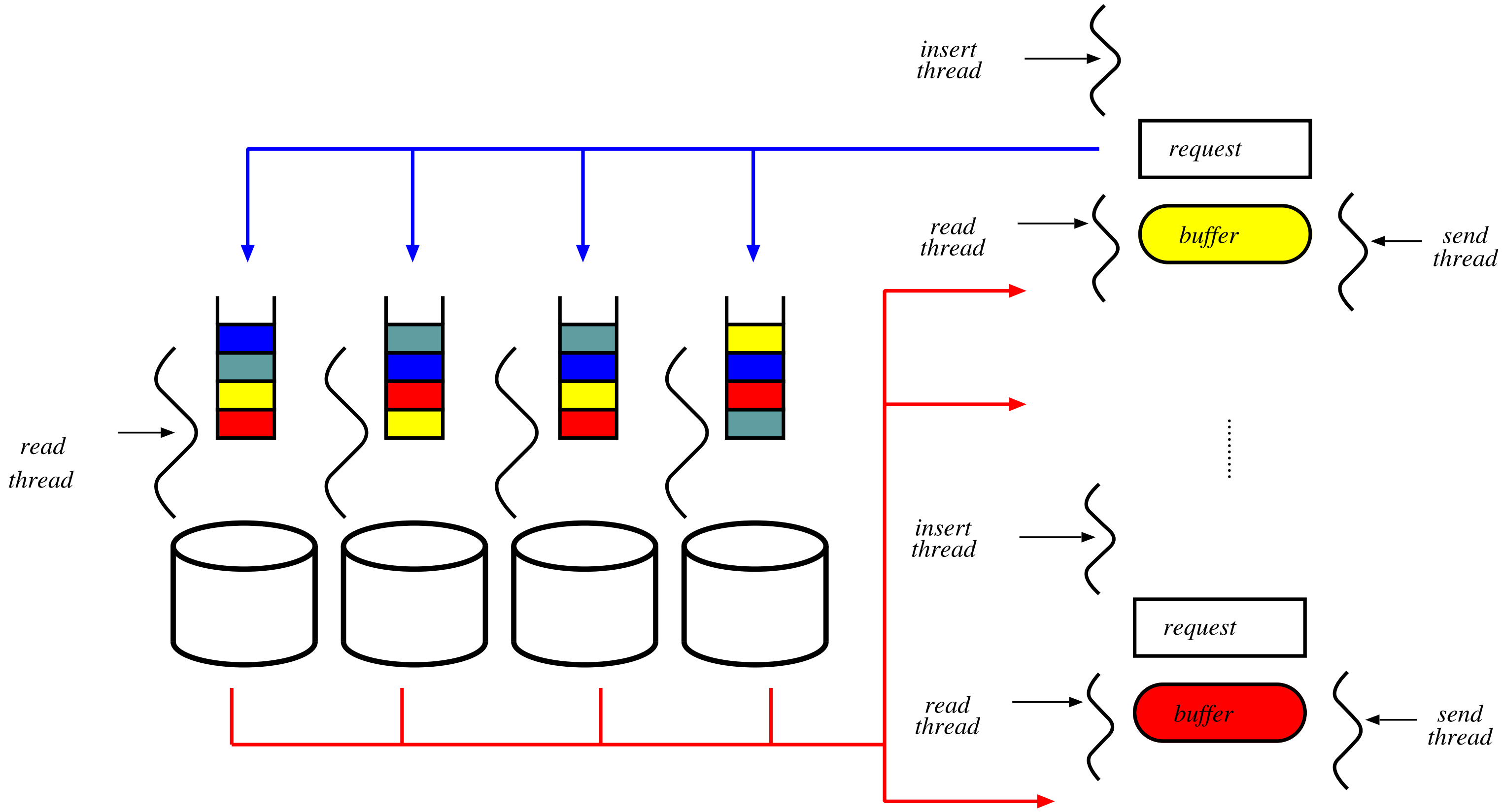
- **Error protection by parity**
- **Maintain Load Feature even in a single disk fault**
- **We provide transparent rebuild utility**


block 0
block 1
block 2
block 3
block 4
block 5
d0
d1
d2
P0
d4
d5
P0
d3
d8
P0
d6
d7
P0
d9
d10
d11


block 0
block 1
block 2
block 3
block 4
block 5
d0
d1
d2
P0
d4
d5
P0
d3
d8
P0
d6
d7
P0
d9
d10
d11

Network Model

- Upon request arrival, refer to memory map for admission control.
- there is a retrieval thread for each disk.
- client's resource include:
  - disk read assignment thread (invoke periodically).
  - buffer for fetching from disk.
  - network sending thread.

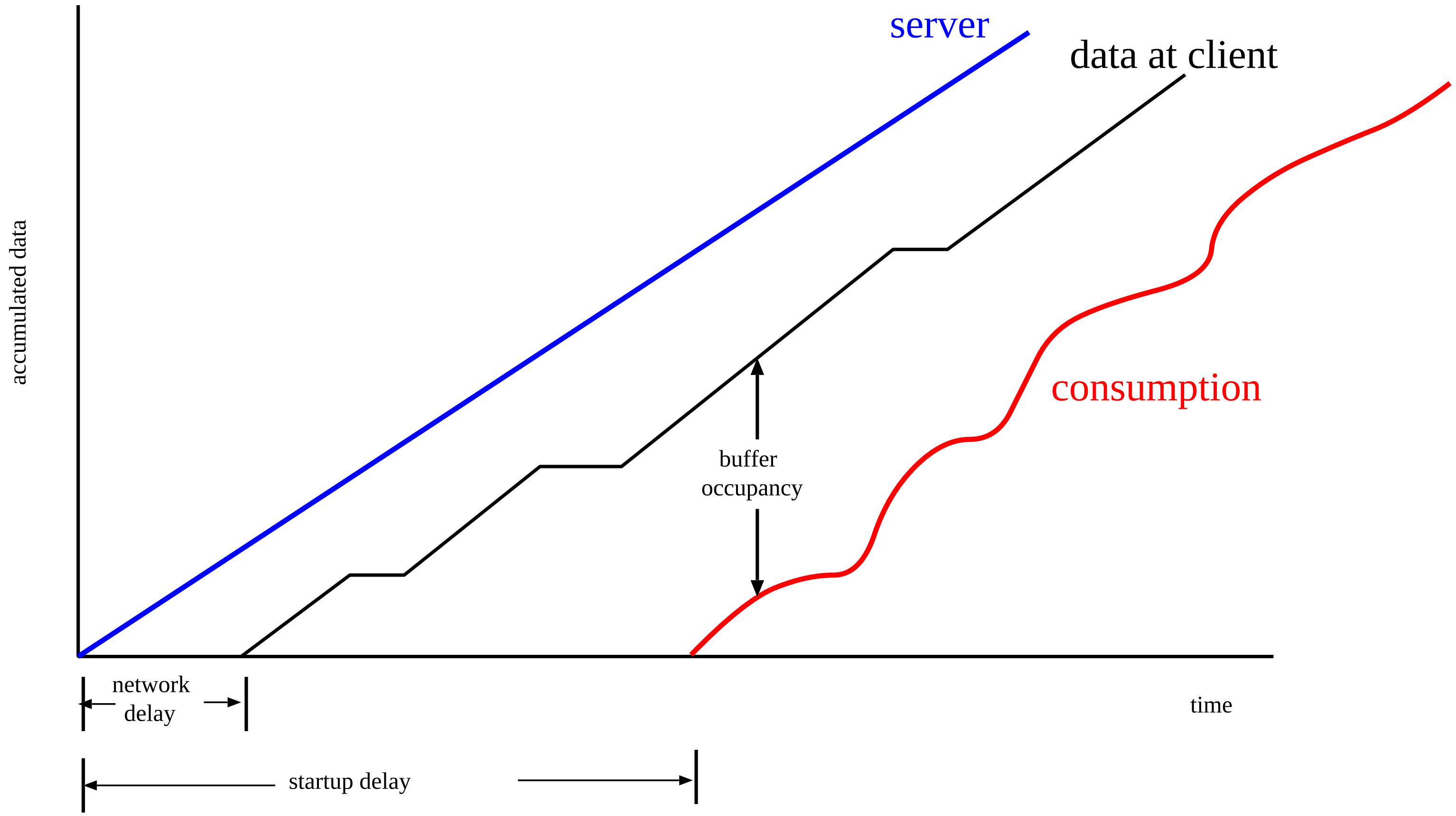


- **Currently, we use UDP for transmission.**

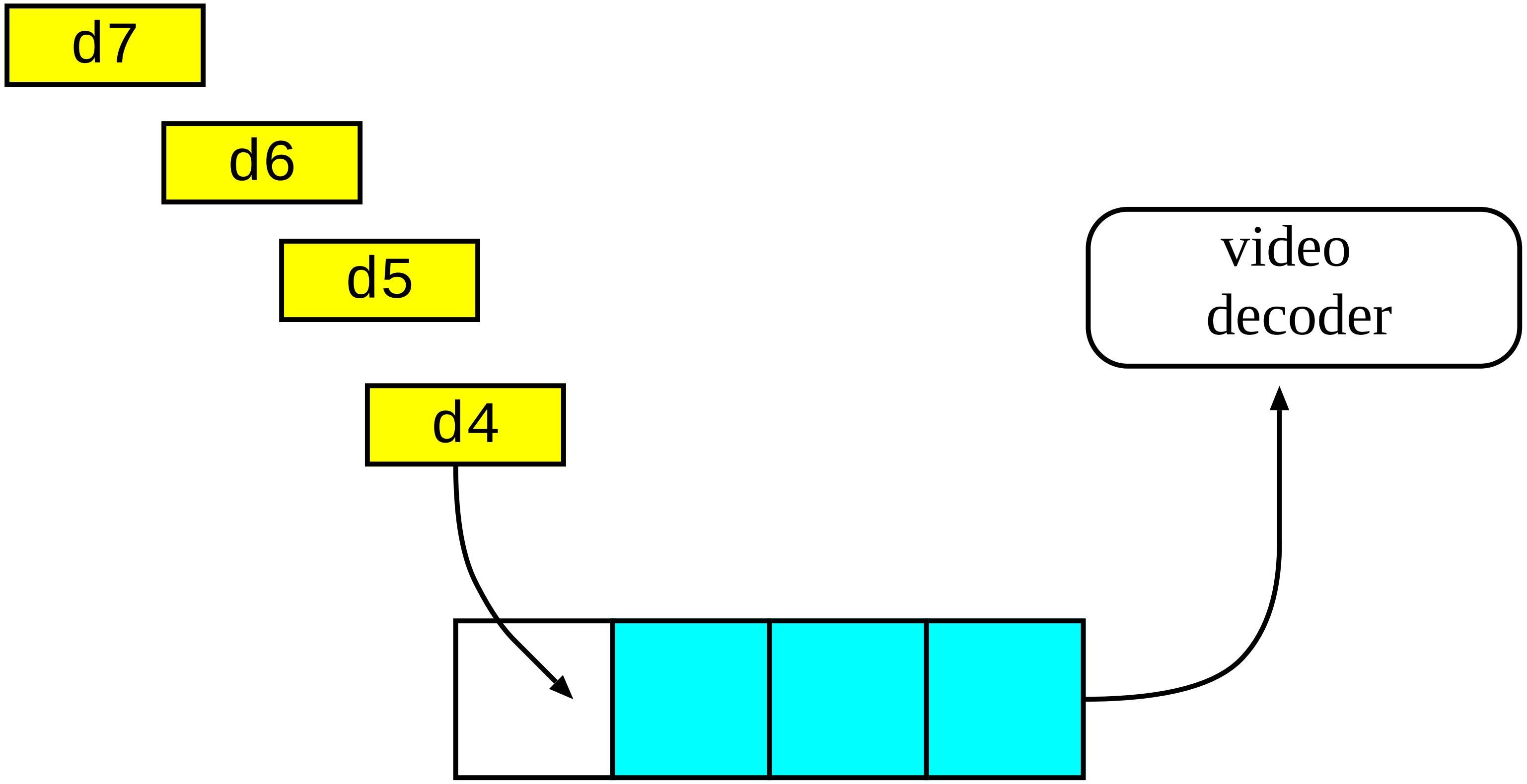
Client Module

- **Client's architecture**
- **Currently, we have MPEG1, MPEG3, QuickTime decoder.**
- **Linux and Solaris: MpegTV.**
- **Window: Java JMF (Java Media Framework)**
- **Buffer Management:**
  - **mask out network bandwidth variation**
  - **packet filtering**
  - **avoid starvation and overflow**
  - **avoid packet loss (best-effort)**

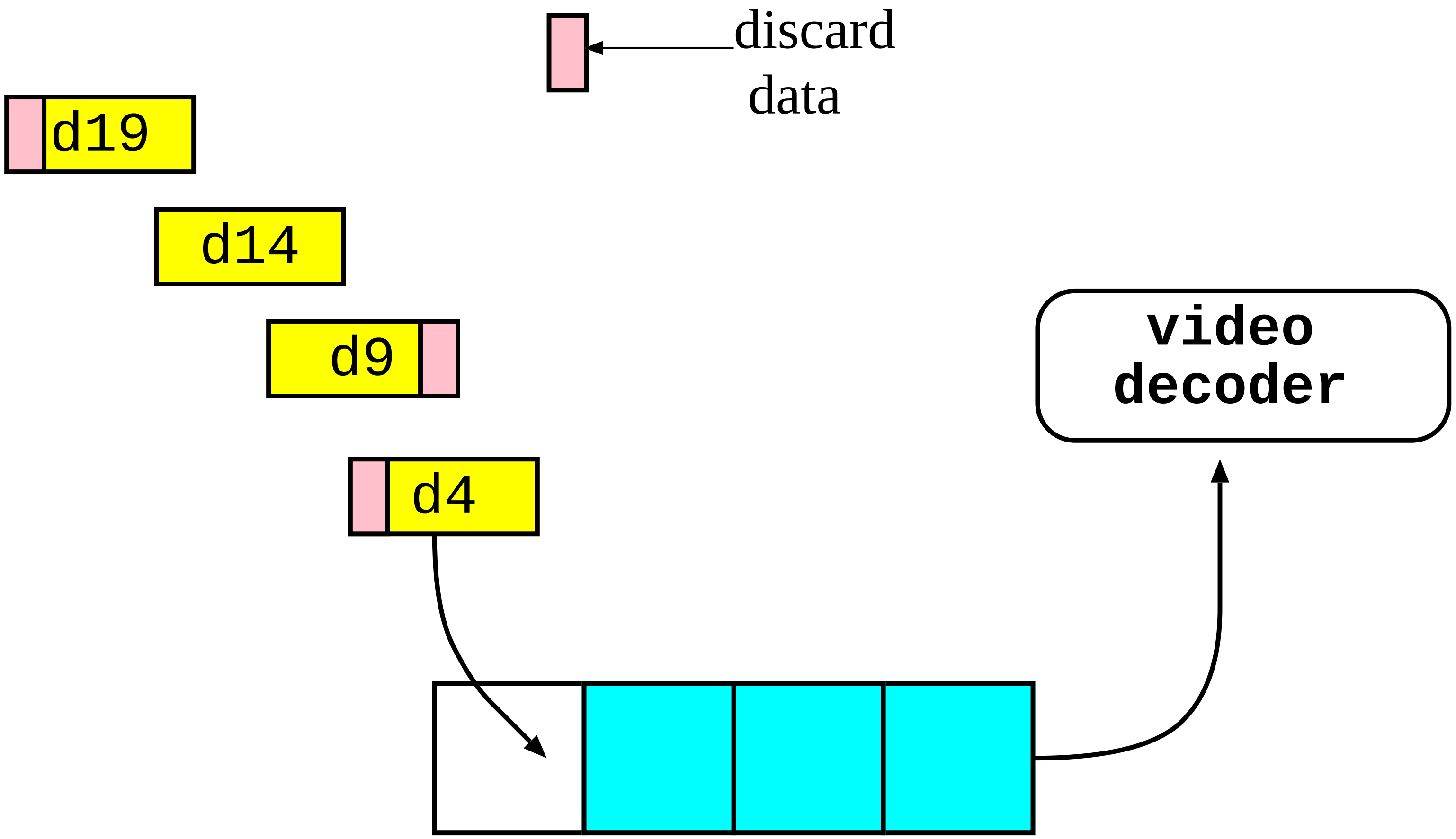
Client:Mask out Network Variation



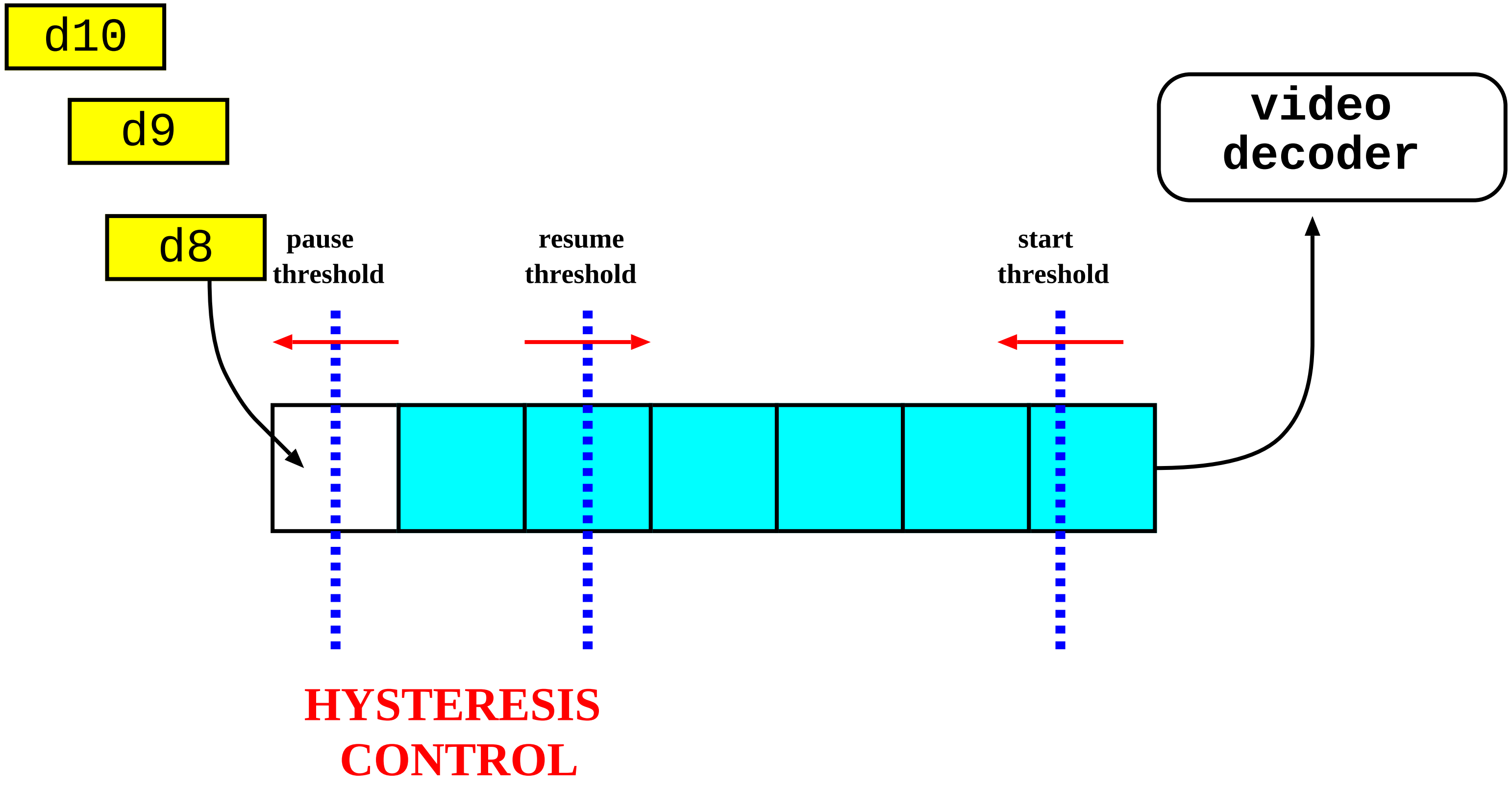
Client: Packet Filtering



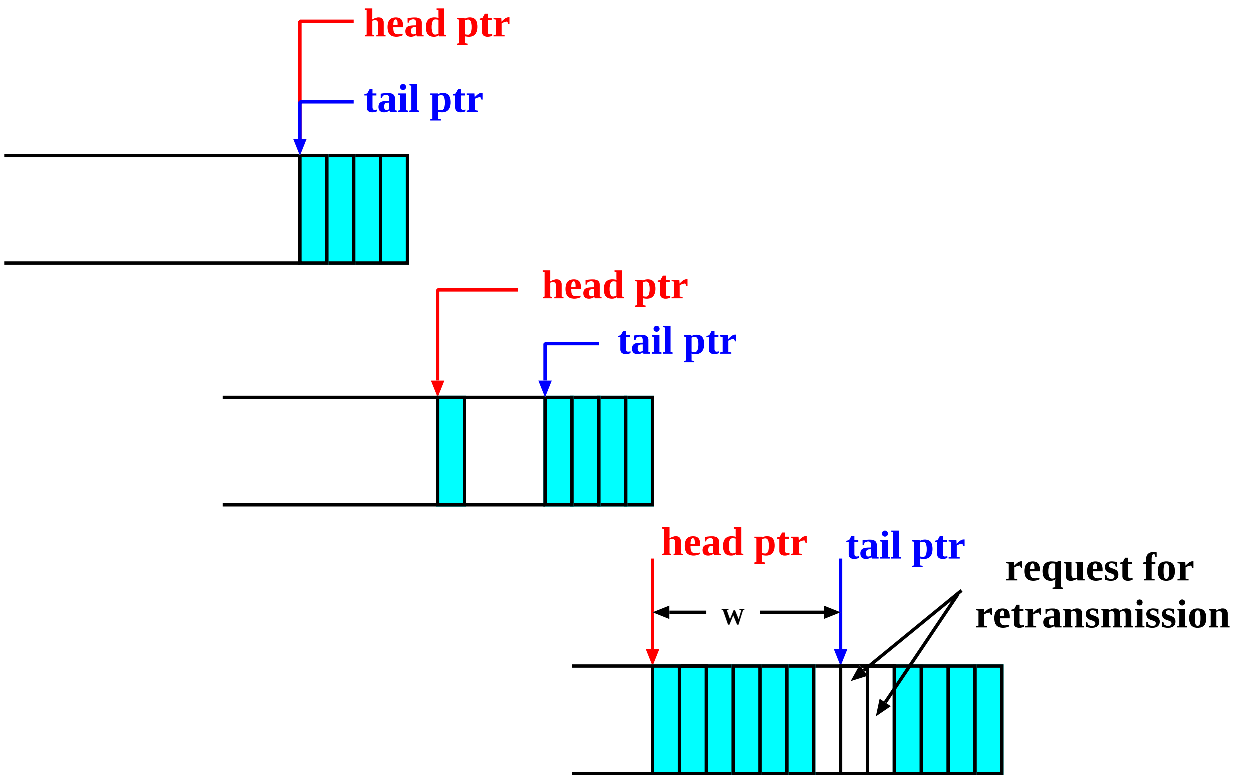
Client: Packet Filtering



Client: Avoid Starvation and Overflow



Client: Avoid Packet Loss



- Value of W
- Can extend to multiple recovery windows

Client: Other features

- MP3 decoder (public domain mpg123 tool)
- synchronization software between
  - video
  - lecture-notes

For this, we use a simple program to control the ghostscript.

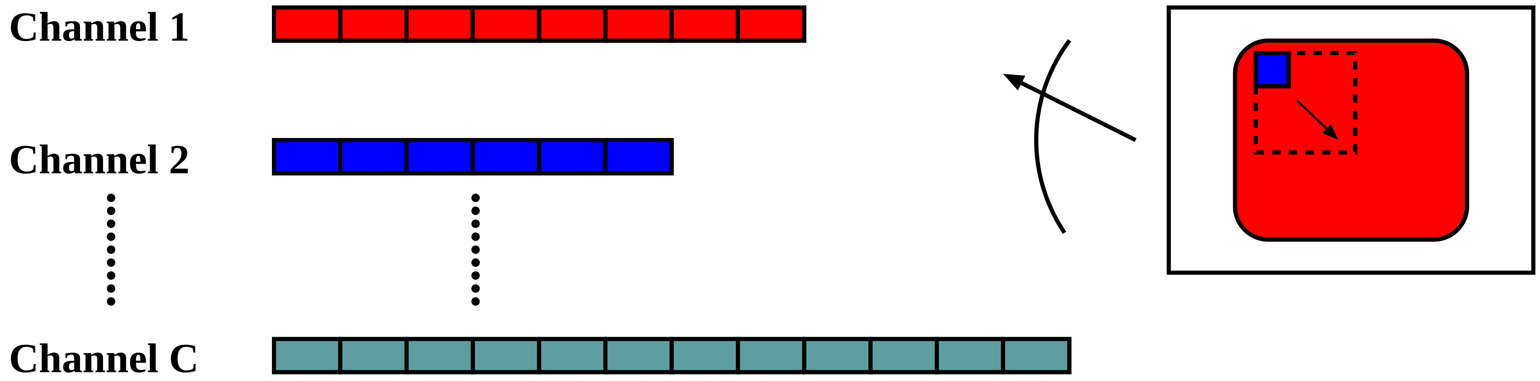
Current Extensions

Basically, the server described above is an *"on-demand"* server. To make the system scalable and efficient, we need to find other methods.

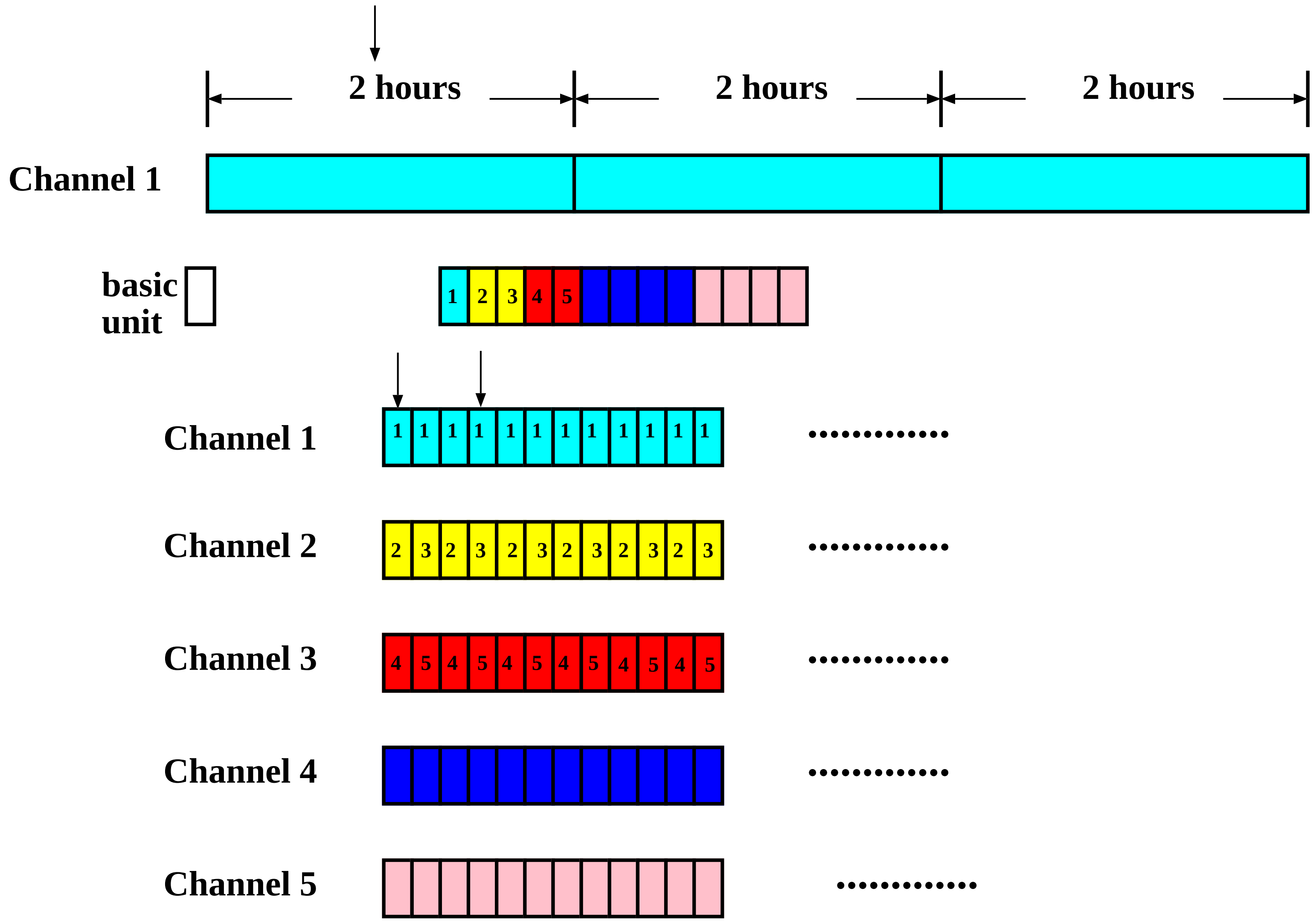
- **Simple Multicast Extension: Need to send "Mpeg header" info.**
- **Picture-in-Picture Feature**
- **Channel Switching Feature**
- **Dynamic Multicast Extension**
- **Real-time Encoding Feature**
- **MBone Support, ..., etc**

Simple Multicast with P-in-P Support

- Window Resizing
- Windows Flipping
- Window Relocation



Dynamic Multicast (Dynamic Skyscraping)



Platform for experiments

- TCP-friendliness (especially for wireless environment)
- Internet Streaming without Resource Reservation
- Bandwidth Skimming upon Congestion.
- Security and Protection
- Cluster Servers
- Collaborative Proxy Server
- Multimedia Document Distribution

Conclusion & Lessons Learn

- **Reliable multimedia streaming server** (GPL, QoS Support, Fault Tolerance, Heterogeneous Media Support, On-demand or Multicast,...,etc)
- **Personal**
  - Good System Work $\Longleftrightarrow$ Good Theoretical Work
  - Contribution to a larger community
- **Students**
  - Realization of good hacking **AND** good fundamental.
  - Creates a sense of excitement.

- **LAST BUT NOT LEAST:**

"We reject kings and presidents, we believe in running codes."

(*quote from IETF*)