Frame-Aggregated Concurrent Matching Switch

Bill Lin (University of California, San Diego)

Isaac Keslassy (Technion, Israel)

Background

- The Concurrent Matching Switch (CMS) architecture was first presented at INFOCOM 2006
- Based on any fixed configuration switch fabric and fully distributed and independent schedulers
  - 100% throughput
  - Packet ordering
  - O(1) amortized time complexity
  - Good delay results in simulations
- Proofs for 100% throughput, packet ordering, and O(1) complexity provided in INFOCOM 2006 paper, but no delay guarantee was provided

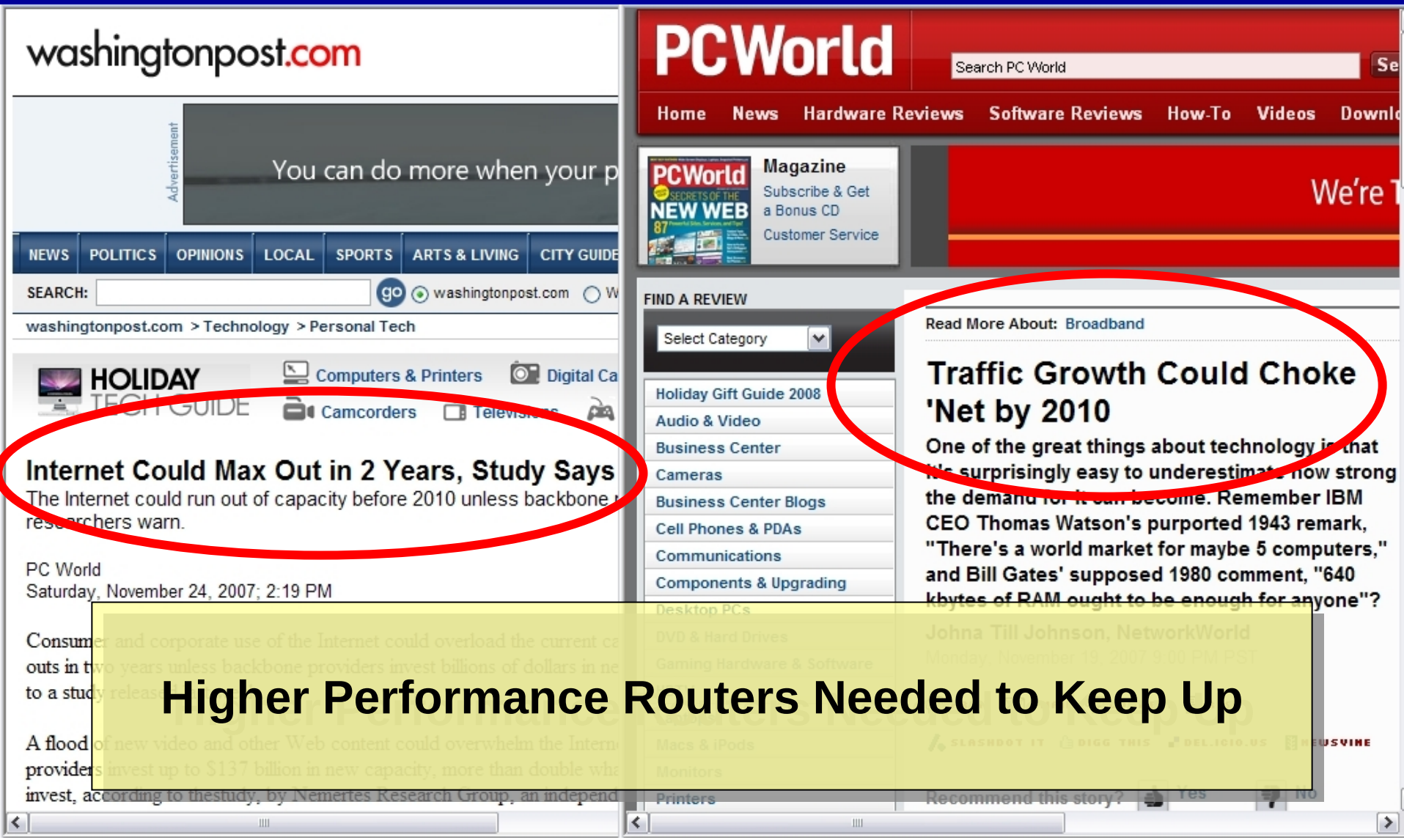
This Talk

- Focus of this talk is to provide a delay bound
- Show O(N log N) delay is provably achievable while retaining O(1) complexity, 100% throughput, and packet ordering
- Show No Scheduling is required to achieve O(N log N) delay by modifying original CMS architecture
- Improves over best previously-known $O(N^2)$ delay bound given same switch properties

This Talk

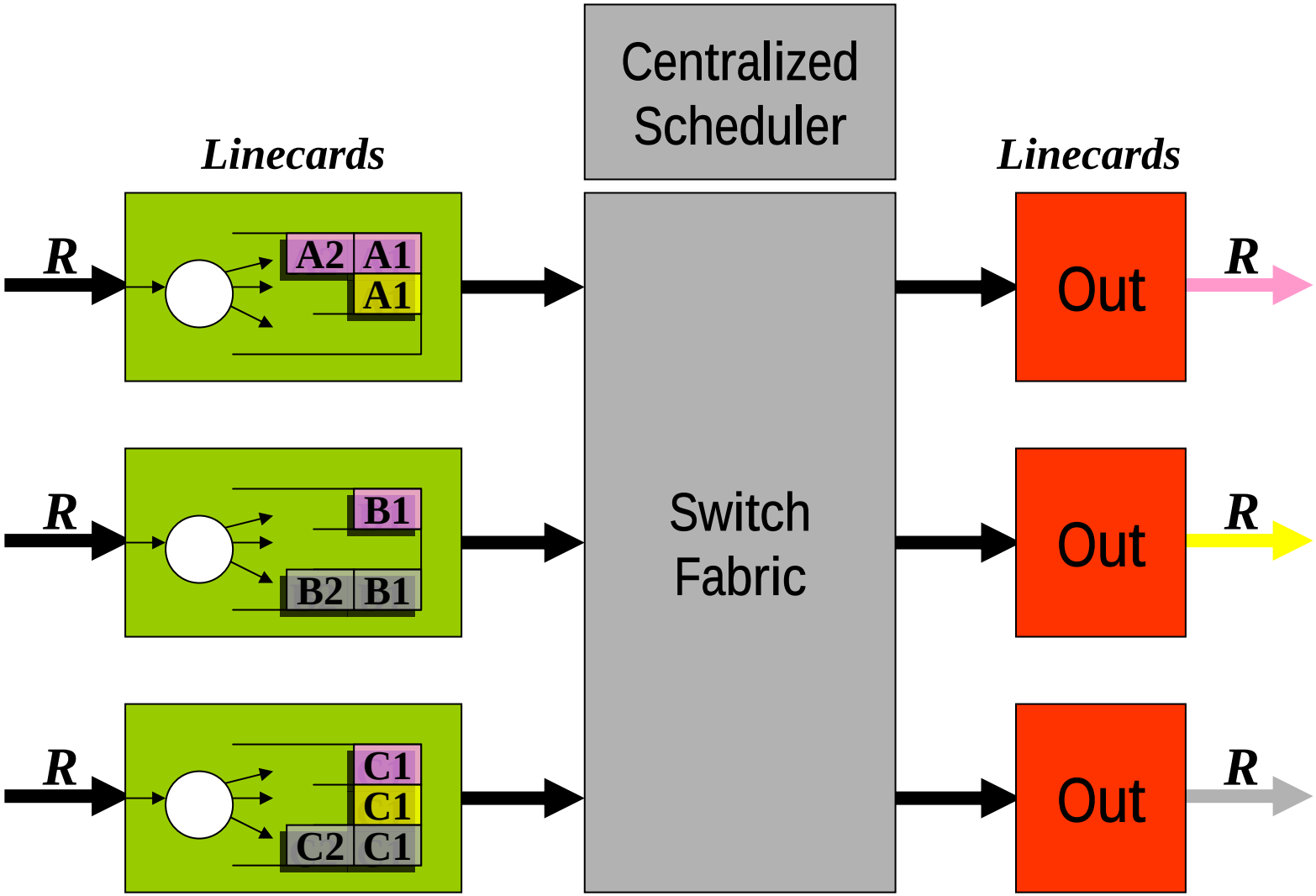
- Concurrent Matching Switch
- General Delay Bound
- O(N log N) delay with Fair-Frame Scheduling
- O(N log N) delay and O(1) complexity with Frame Aggregation instead of Scheduling

The Problem

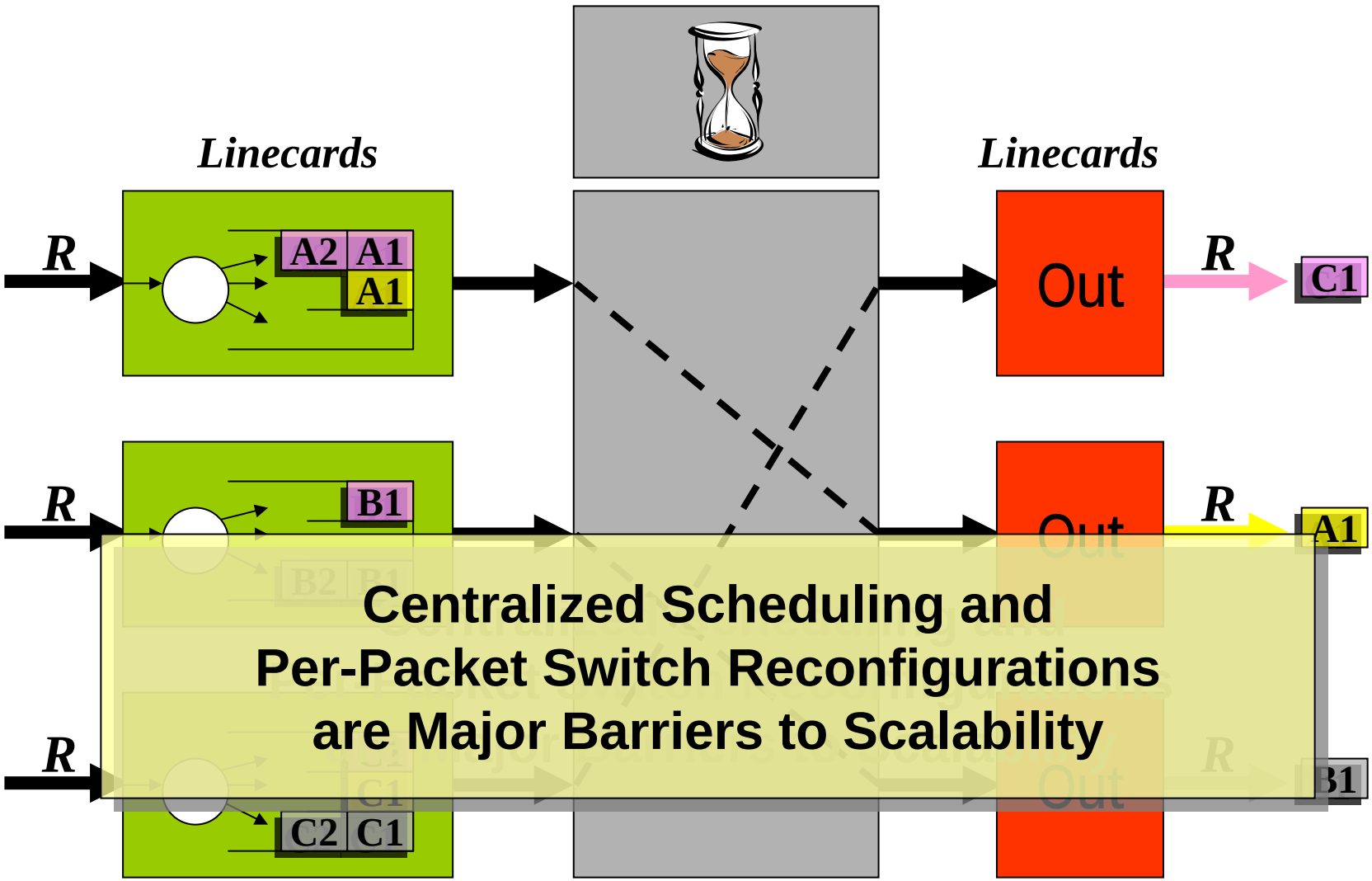


Higher Performance Routers Needed to Keep Up

Classical Switch Architecture



Classical Switch Architecture

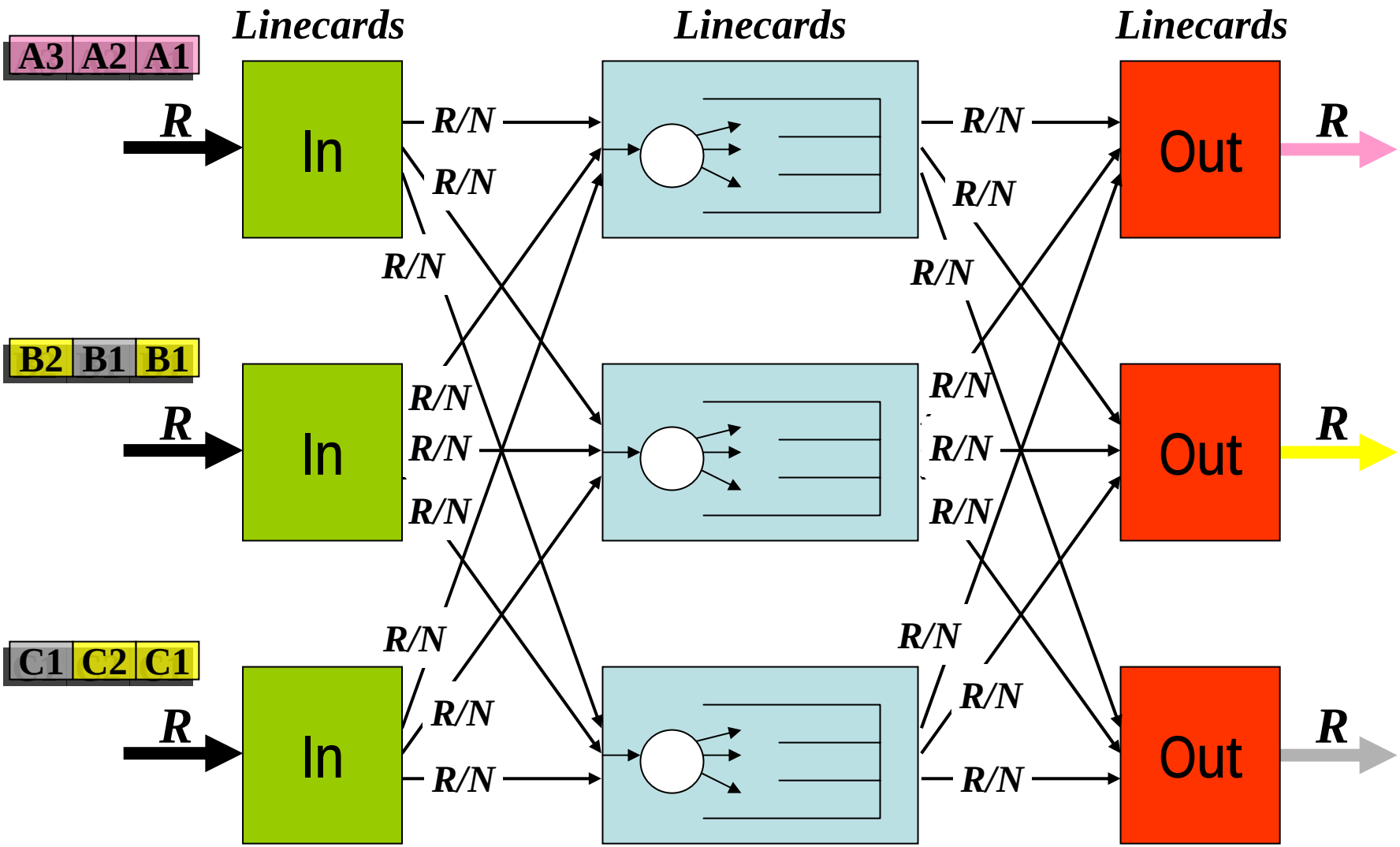


Recent Approaches

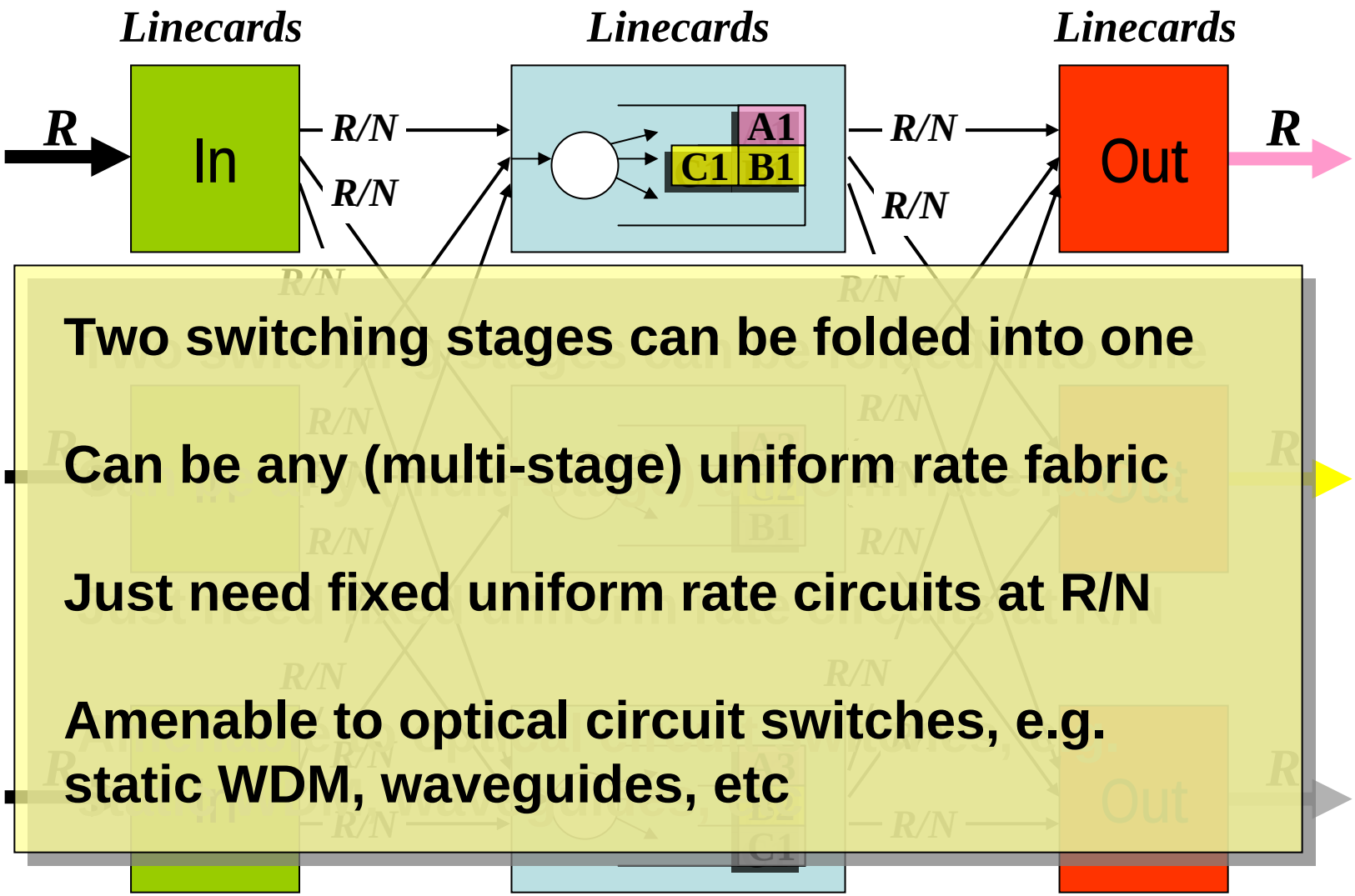
- Scalable architectures
  - Load-Balanced Switch [Chang 2002] [Keslassy 2003]
  - Concurrent Matching Switch [INFOCOM 2006]

- Characteristics
  - Both based on two identical stages of fixed configuration switches and fully decentralized processing
  - No per-packet switch reconfigurations
  - Constant time local processing at each linecard
  - 100% throughput
  - Amenable to scalable implementation using optics

Basic Load-Balanced Switch

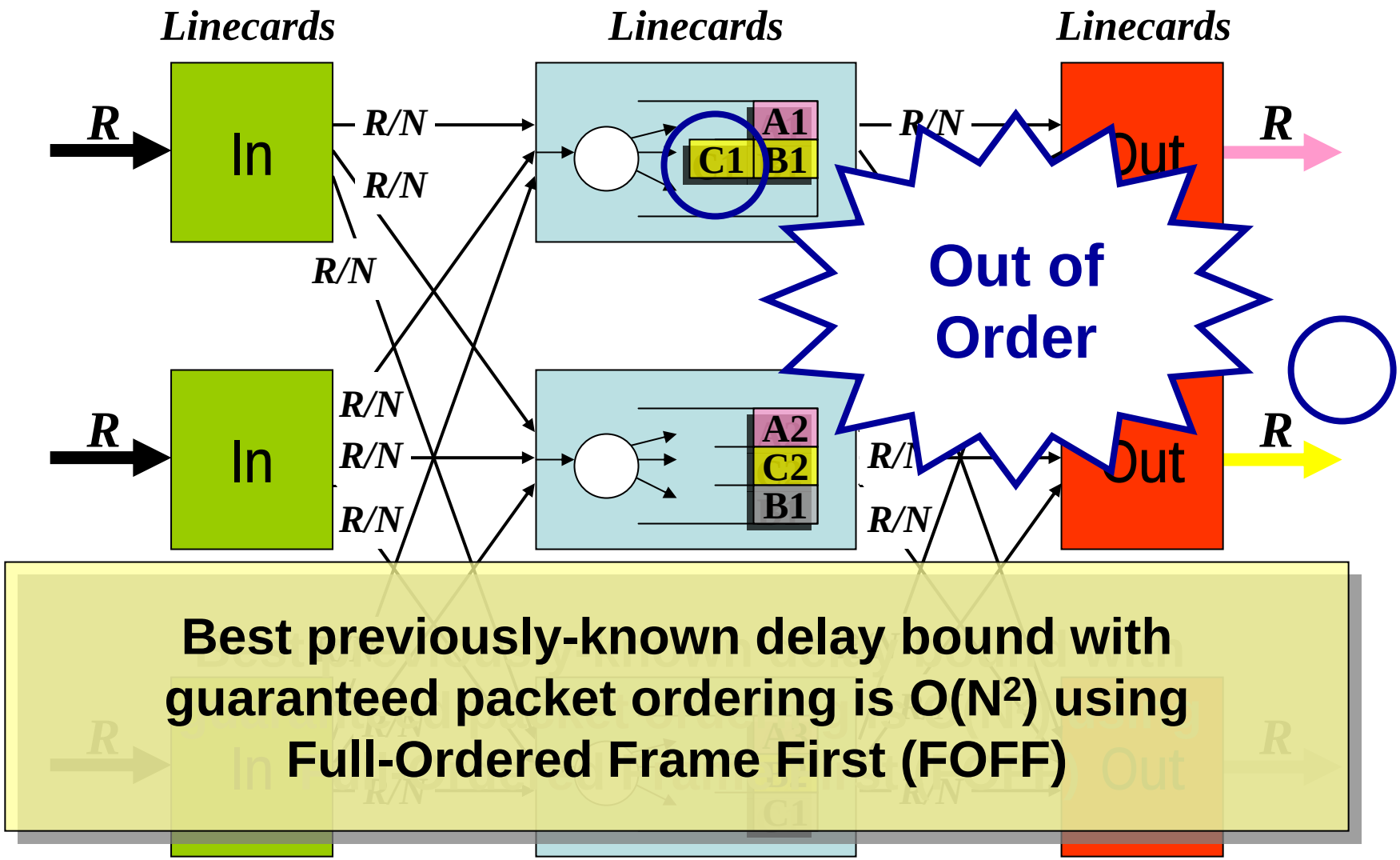


Basic Load-Balanced Switch



- Two switching stages can be folded into one
- Can be any (multi-stage) uniform rate fabric
- Just need fixed uniform rate circuits at R/N
- Amenable to optical circuit switches, e.g. static WDM, waveguides, etc

Basic Load-Balanced Switch



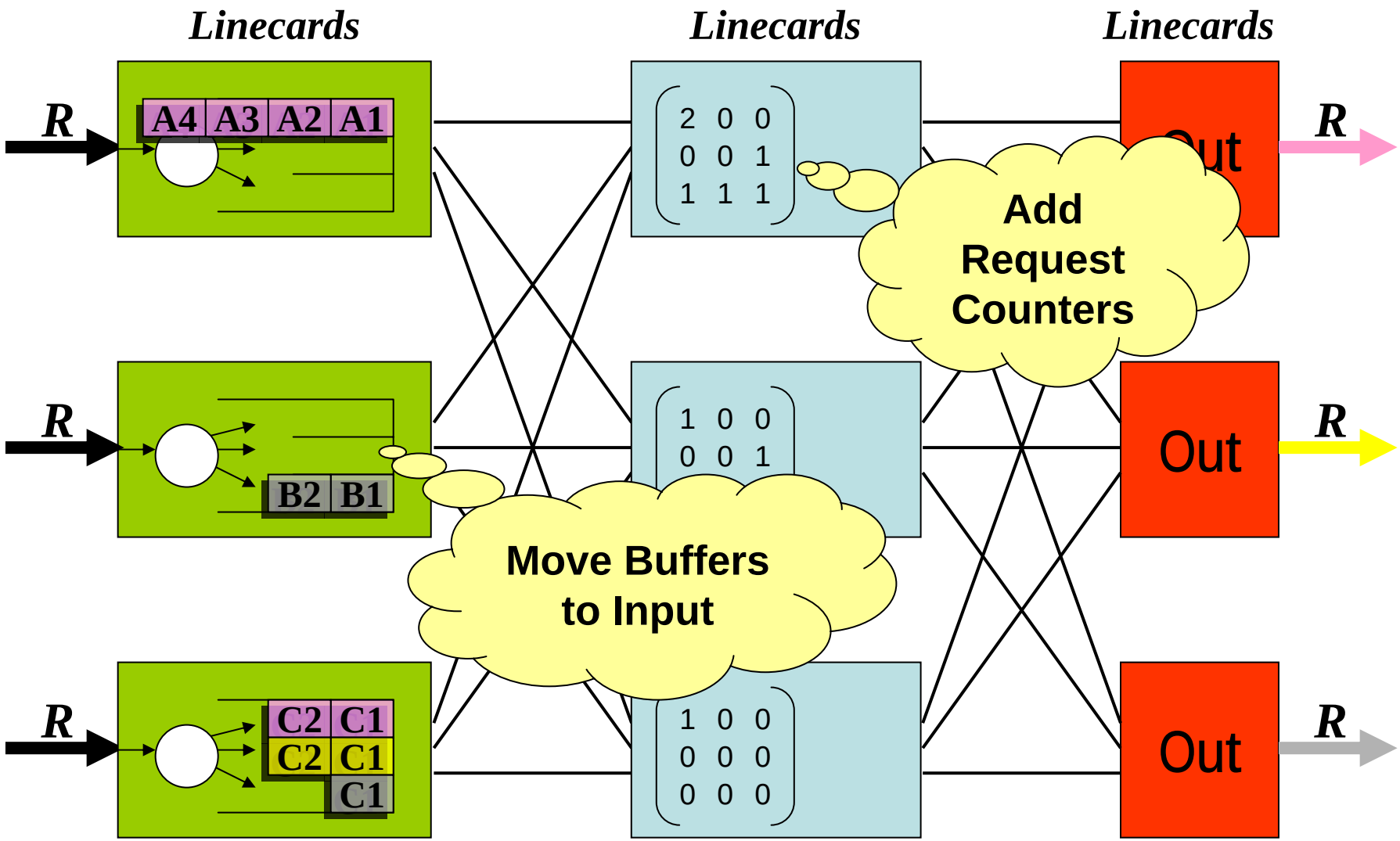
Best previously-known delay bound with guaranteed packet ordering is $O(N^2)$ using Full-Ordered Frame First (FOFF)

Concurrent Matching Switch

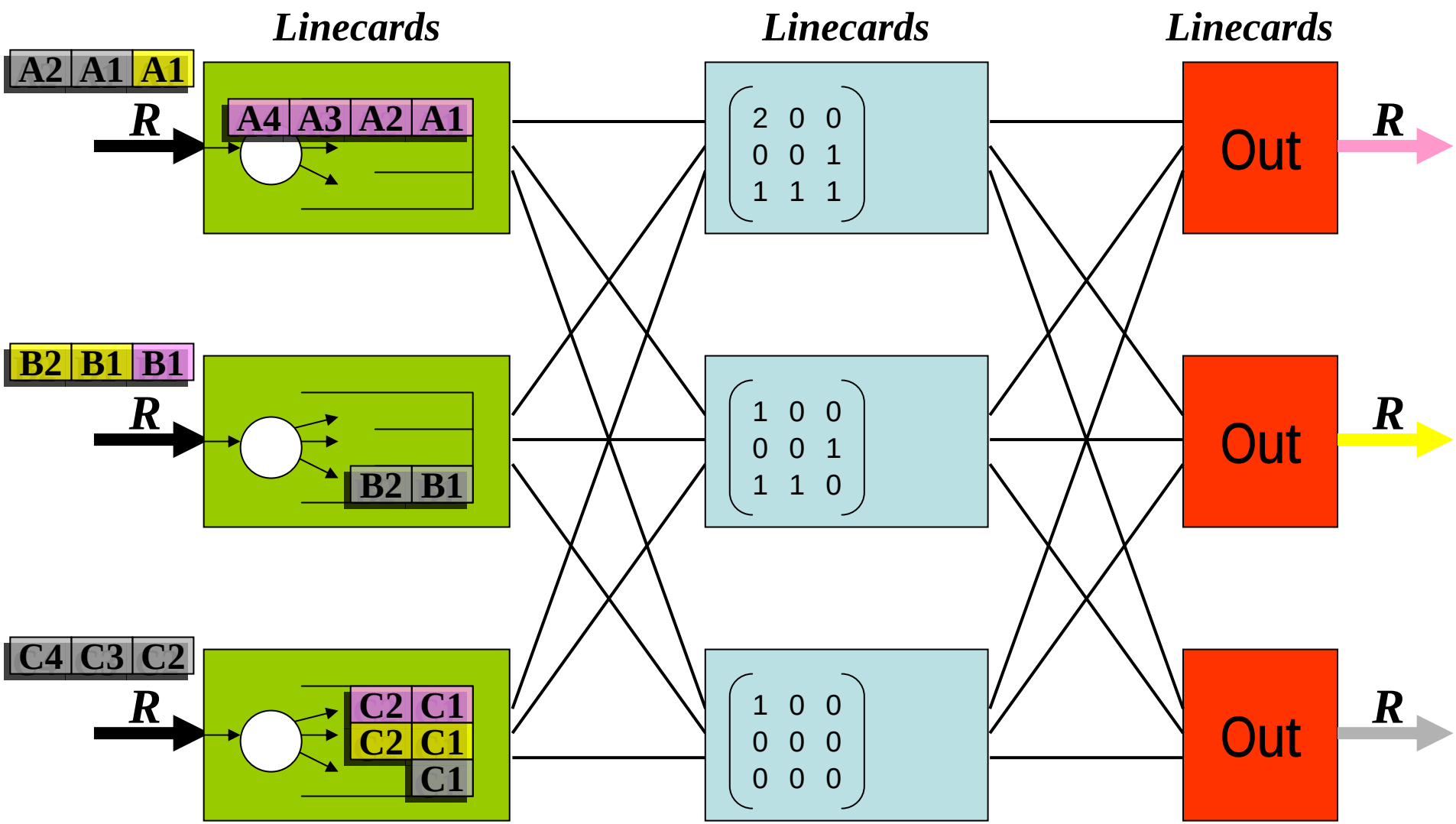
- Retains load-balanced switch structure and scalability of fixed optical switches
- Load-balance **“requests”** instead of packets to **N** parallel “schedulers”
- Each scheduler **independently** solves its own matching
- Scheduling complexity amortized by factor of **N**
- **Packets delivered in order** based on matching results

Goal to provide low average delay with Packet Ordering while retaining 100% throughput and scalability

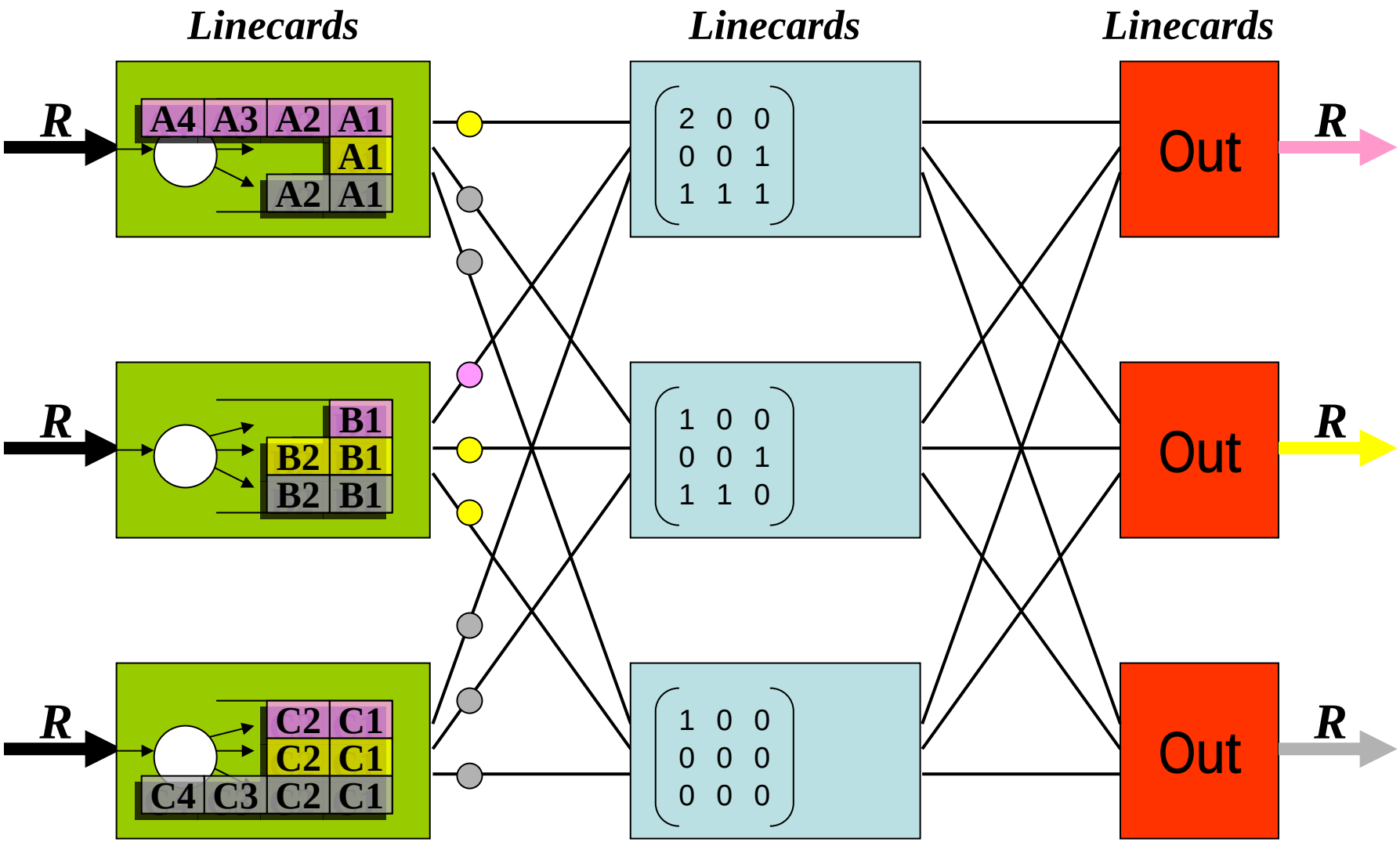
Concurrent Matching Switch



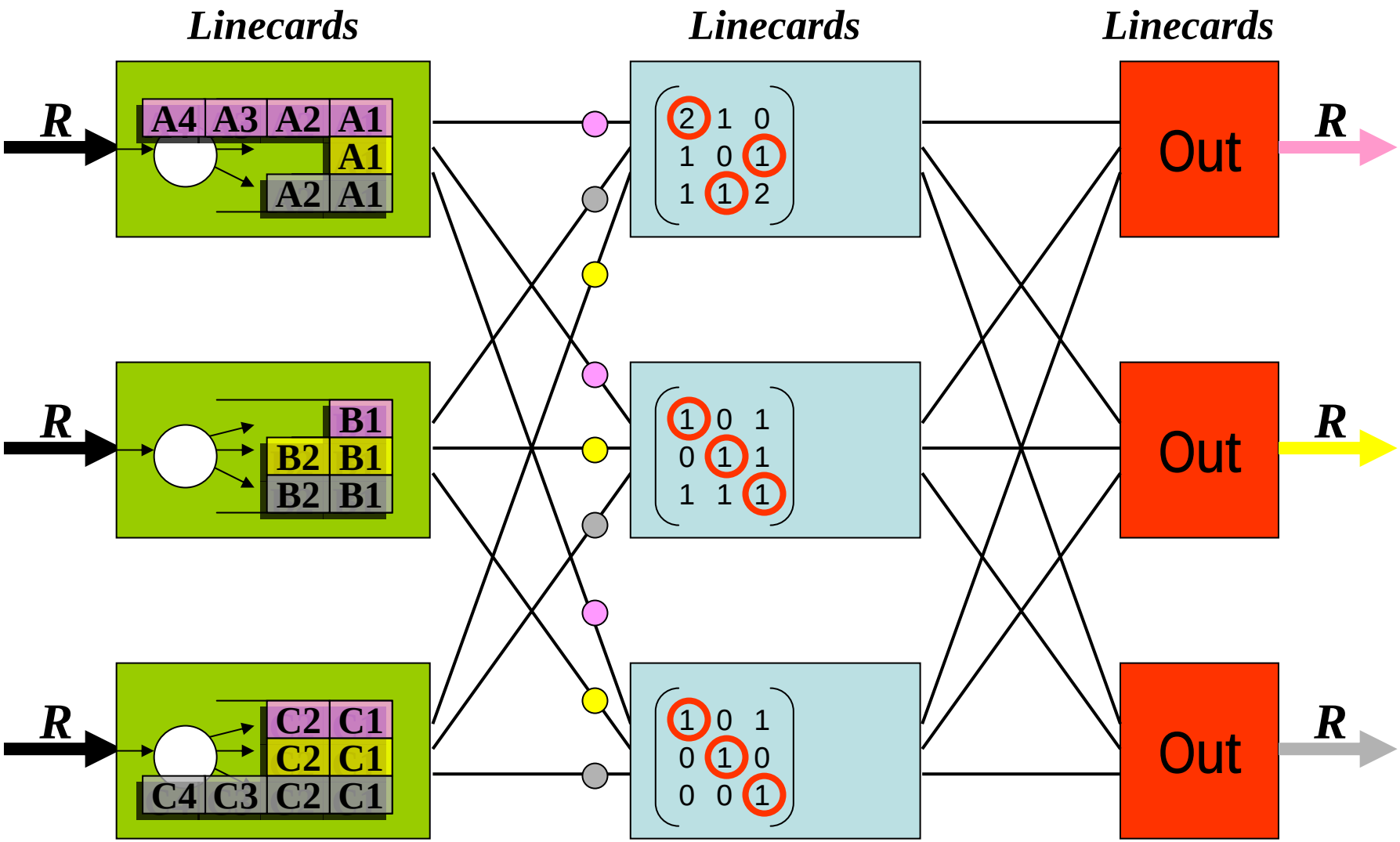
Arrival Phase



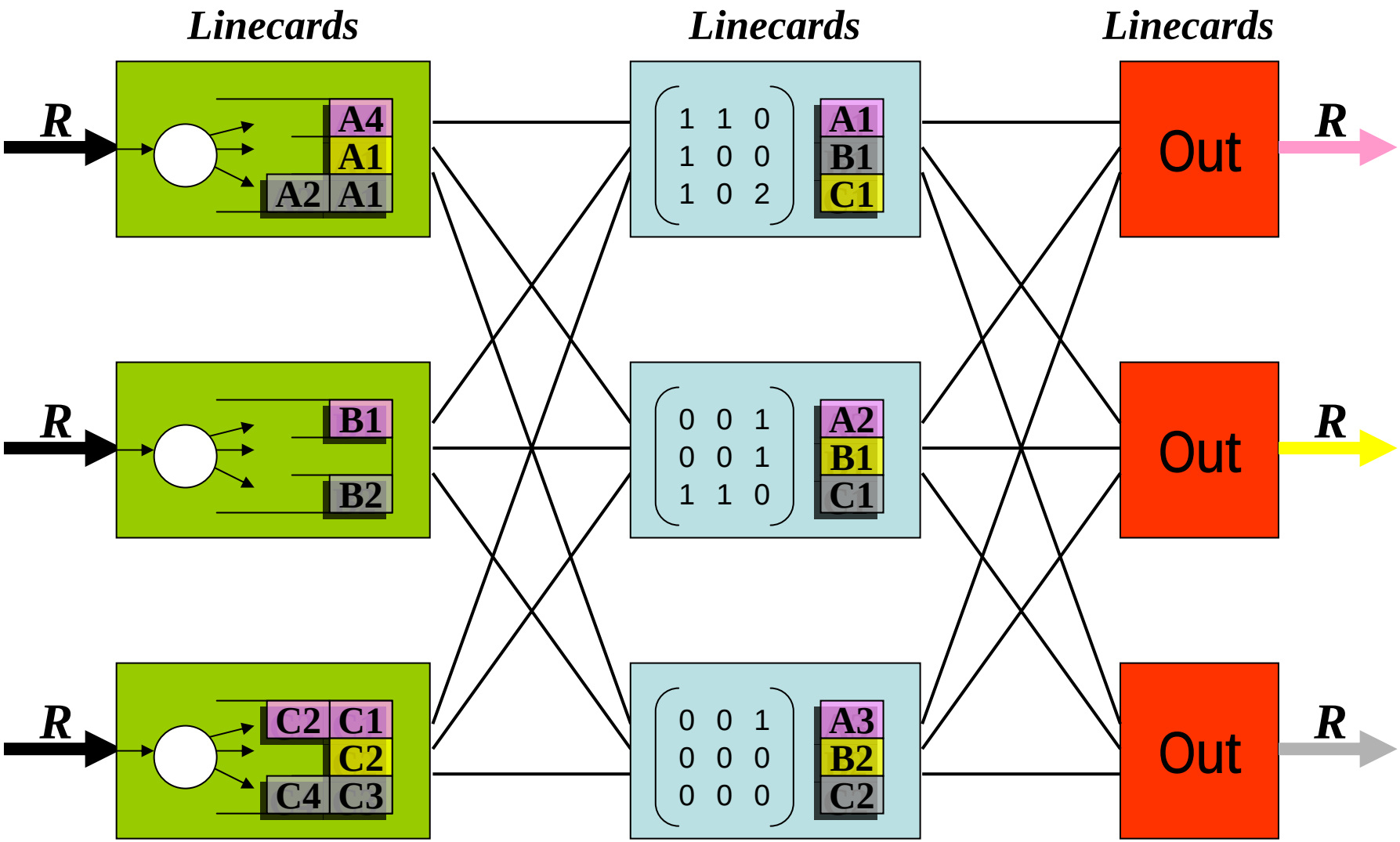
Arrival Phase



Matching Phase



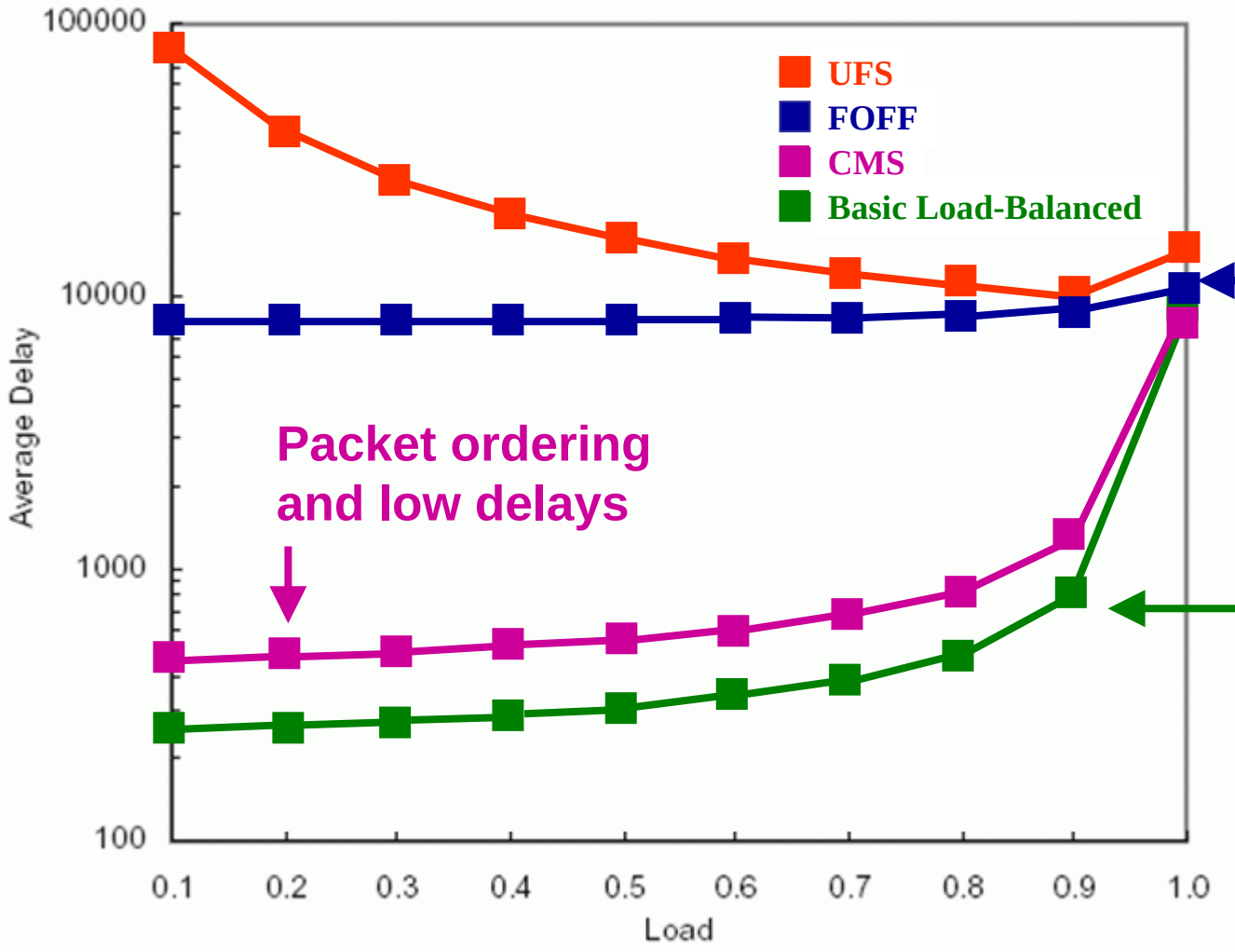
Departure Phase



Practicality

- All linecards operate in parallel in fully distributed manner
- Arrival, matching, departure phases pipelined
- Any stable scheduling algorithm can be used
- e.g., amortizing well-studied randomized algorithms [Tassiulas 1998] [Giaccone 2003] over N time slots, CMS can achieve
  - O(1) time complexity
  - 100% throughput
  - Packet ordering
  - Good delay results in simulations

Performance of CMS



FOFF guarantees packet ordering at $O(N^2)$ delay

No packet ordering guarantees

N = 128, uniform traffic

This Talk

- Concurrent Matching Switch
- **General Delay Bound**
- O(N log N) delay with Fair-Frame Scheduling
- O(N log N) delay and O(1) complexity with Frame Aggregation

Delay Bound

- **Theorem:** Given Bernoulli i.i.d. arrival, let **S** be strongly stable scheduling algorithm with average delay $\mathbf{W_S}$ in single switch. Then CMS using **S** is also strongly stable, with average delay $\mathbf{O(N\ W_S)}$

- **Intuition:**
  - Each scheduler works at an internal reference clock that is N times slower, but receives only $1/N^{th}$ of the requests
  - Therefore, if $\mathbf{O(W_S)}$ is average waiting time for request to be serviced by S, then average waiting time for CMS using S is N times longer, $\mathbf{O(N\ W_S)}$

Delay Bound

- Any stable scheduling algorithm can be used with CMS
- Although we previously showed good delay simulations using a randomized algorithm called SERENA [Giaccone 2003] that is amortizable to O(1) complexity, no delay bounds (W_S) are known for this class of algorithms
- Therefore, delay bounds for CMS using these algorithms are also unknown

O(N log N) Delay

- In this talk, we want to show CMS can be provably bounded by O(N log N) delay for Bernoulli i.i.d. arrival, improving over the previous $O(N^2)$ bound provided by FOFF

- This can be achieved using a known logarithmic delay scheduling algorithm called Fair-Frame Scheduling [Neely 2004], i.e. $W_S = O(\log N)$, therefore O(N log N) for CMS

Fair-Frame Scheduling

- Suppose we accumulate incoming requests for frame of

$$T = \left\lceil \frac{\log(2N/\delta)}{\log(1/\gamma)} \right\rceil$$

  consecutive time slots, where γ is a constant with respect to the load ρ

- Then the row and column sums of the arrival matrix **L** is bounded by **T** with high probability

Fair-Frame Scheduling

- For example, suppose **T = 3** and

$$\mathbf{L} = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix}$$

then it can be decomposed into **T = 3** permutations

$$\begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

- Logarithmic delay follows from **T** being O(log N)
- Small probability of “overflows” serviced in future frames when max row/column sum less than **T**

CMS with Fair Frame Scheduling

- O(N log N) delay
- 100% throughput and packet ordering
- O(log log N) amortized time complexity by solving matrix decomposition with edge-coloring

Question: Can O(N log N) delay be guaranteed with O(1) complexity?

Answer: Yes, and with No Scheduling

This Talk

- Concurrent Matching Switch
  - 100% throughput, packet ordering, O(1) complexity, good delays, but no delay bound previously provided
- General Delay Bound
  - O(N W_S) delay, N times the delay of scheduling algorithm used
- CMS with Fair-Frame Scheduling
  - O(N log N) delay, O(log log N) complexity

➡ **Frame Aggregated CMS**

- **O(N log N) delay, O(1) complexity, and No Scheduling**

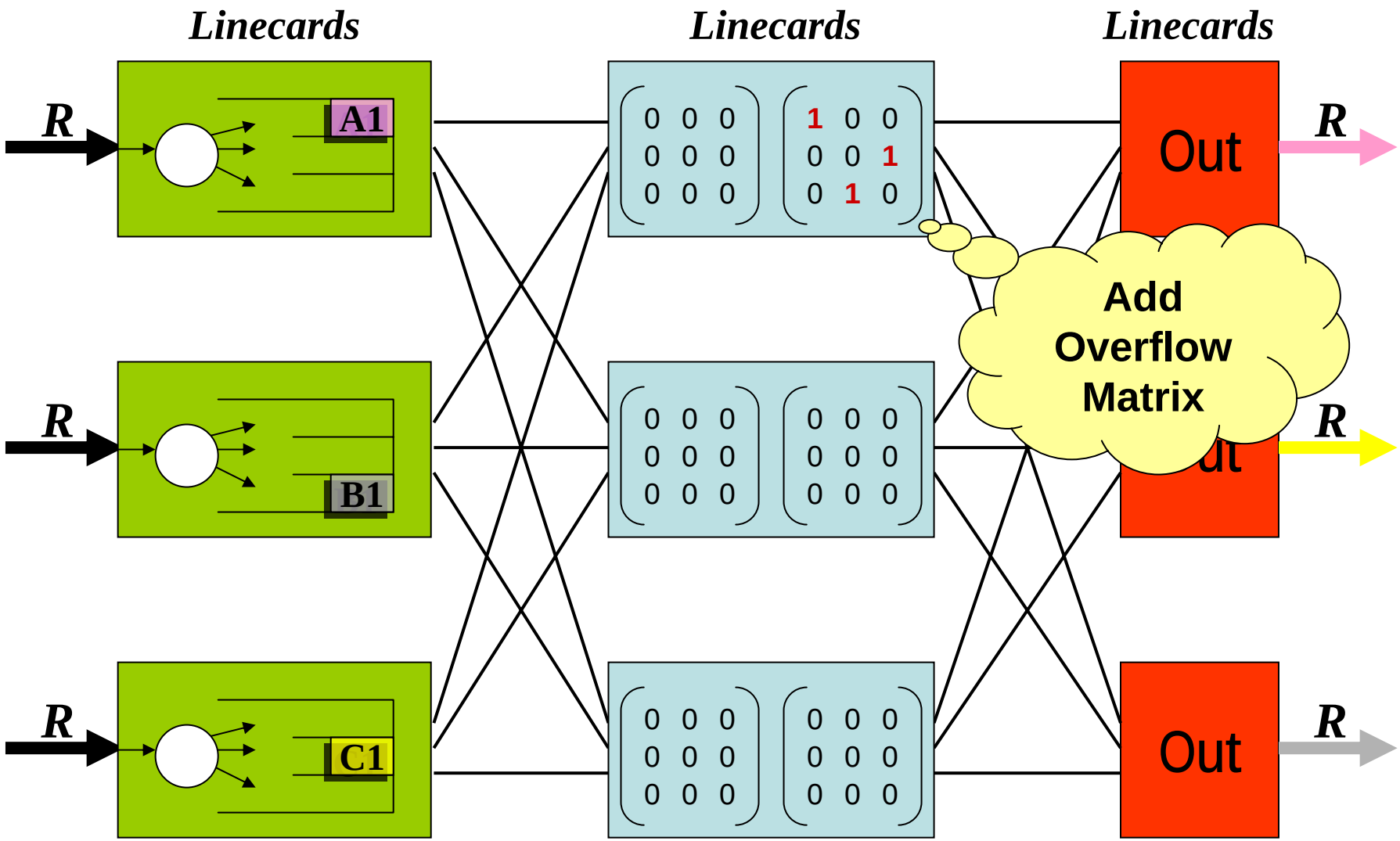
Frame-Aggregated CMS

- Operates just like CMS, but each intermediate linecard accumulates requests for a **superframe** of **N T** time slots before sending back grants in batch

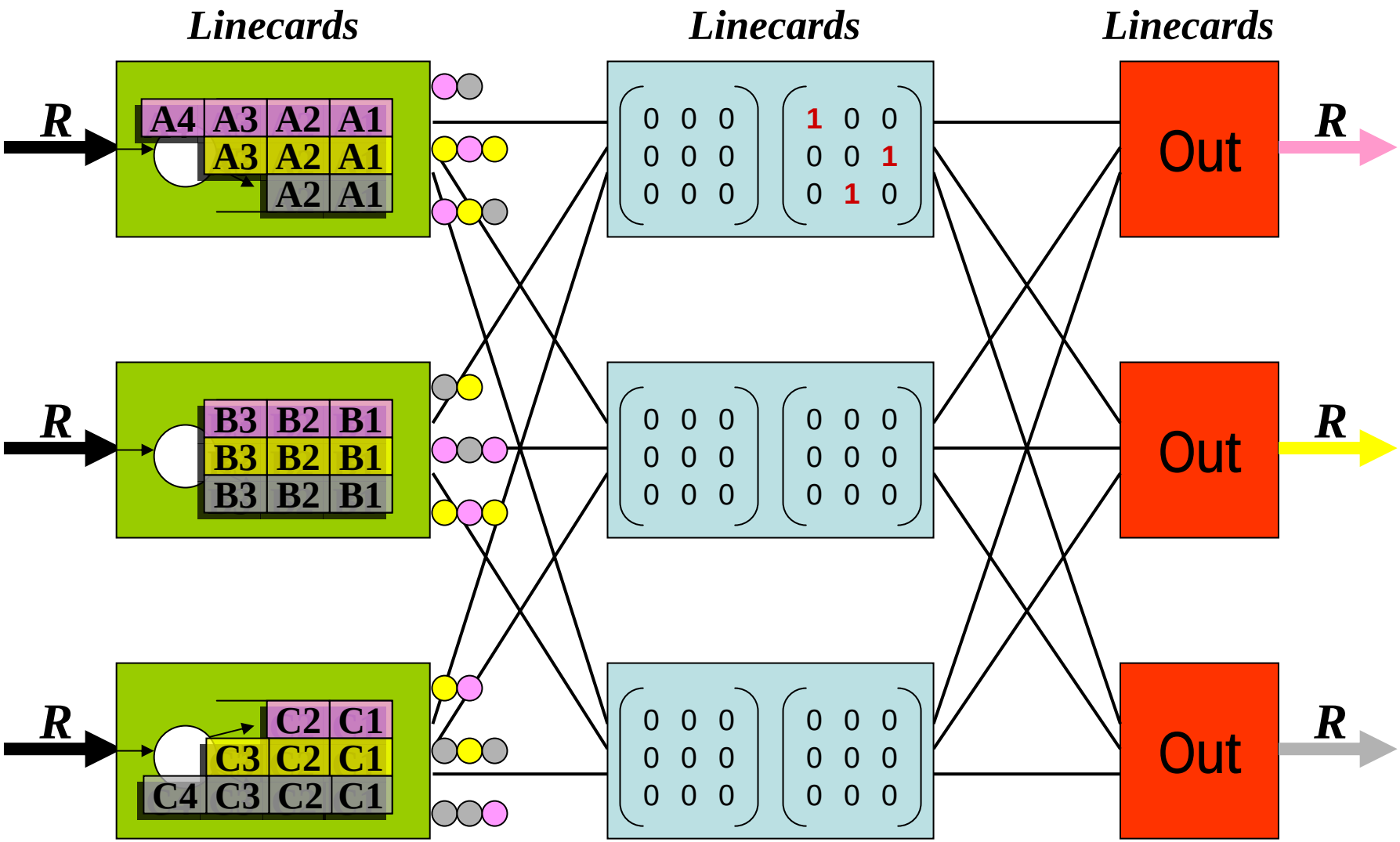
- **T** determined using same logarithmic formula as fair-frame scheduling

- **Main Idea**: When arrival request matrix **L** to an intermediate linecard has row/column sums bounded by **T**, **No Need to Decompose L** before returning grants (No Scheduling). "Overflow" requests defer to future superframes.

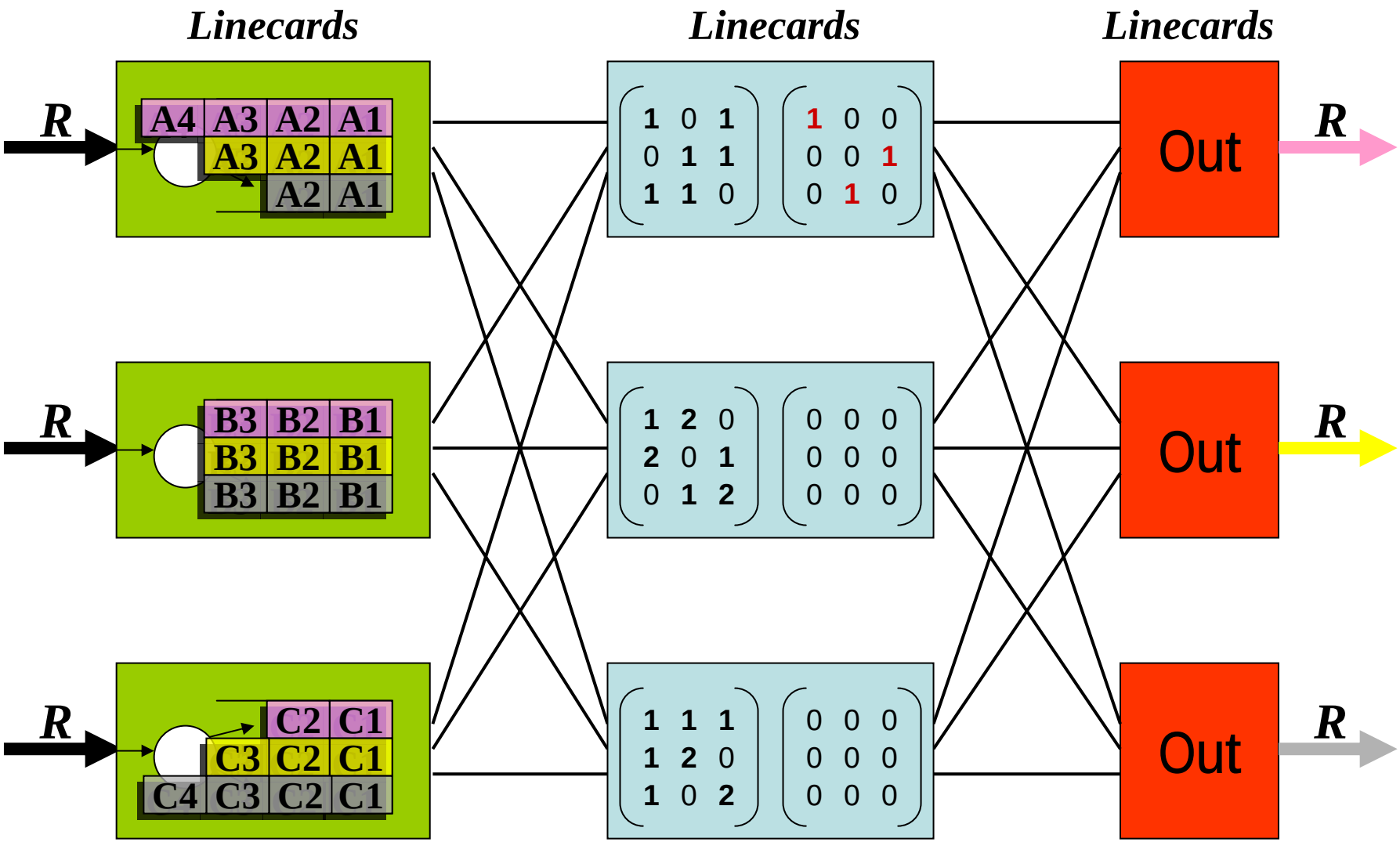
Frame-Aggregated CMS



Frame-Aggregated CMS



Frame-Aggregated CMS

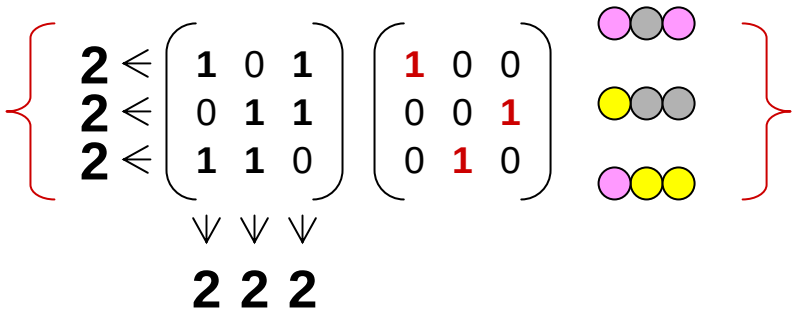


Frame-Aggregated CMS

T = 3

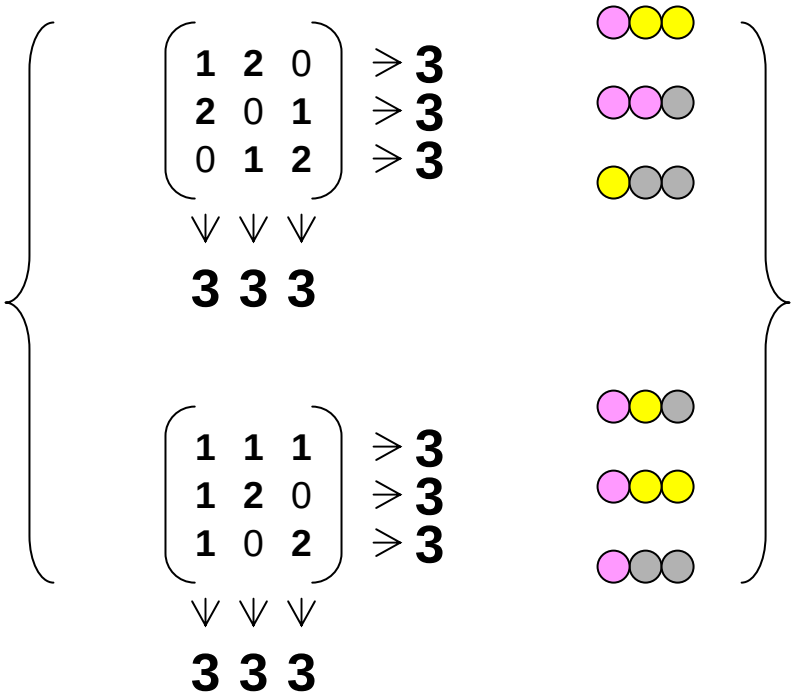
Requests **Overflow**

Max Row/Col Sum: 2 < T



Fill with "Overflows"

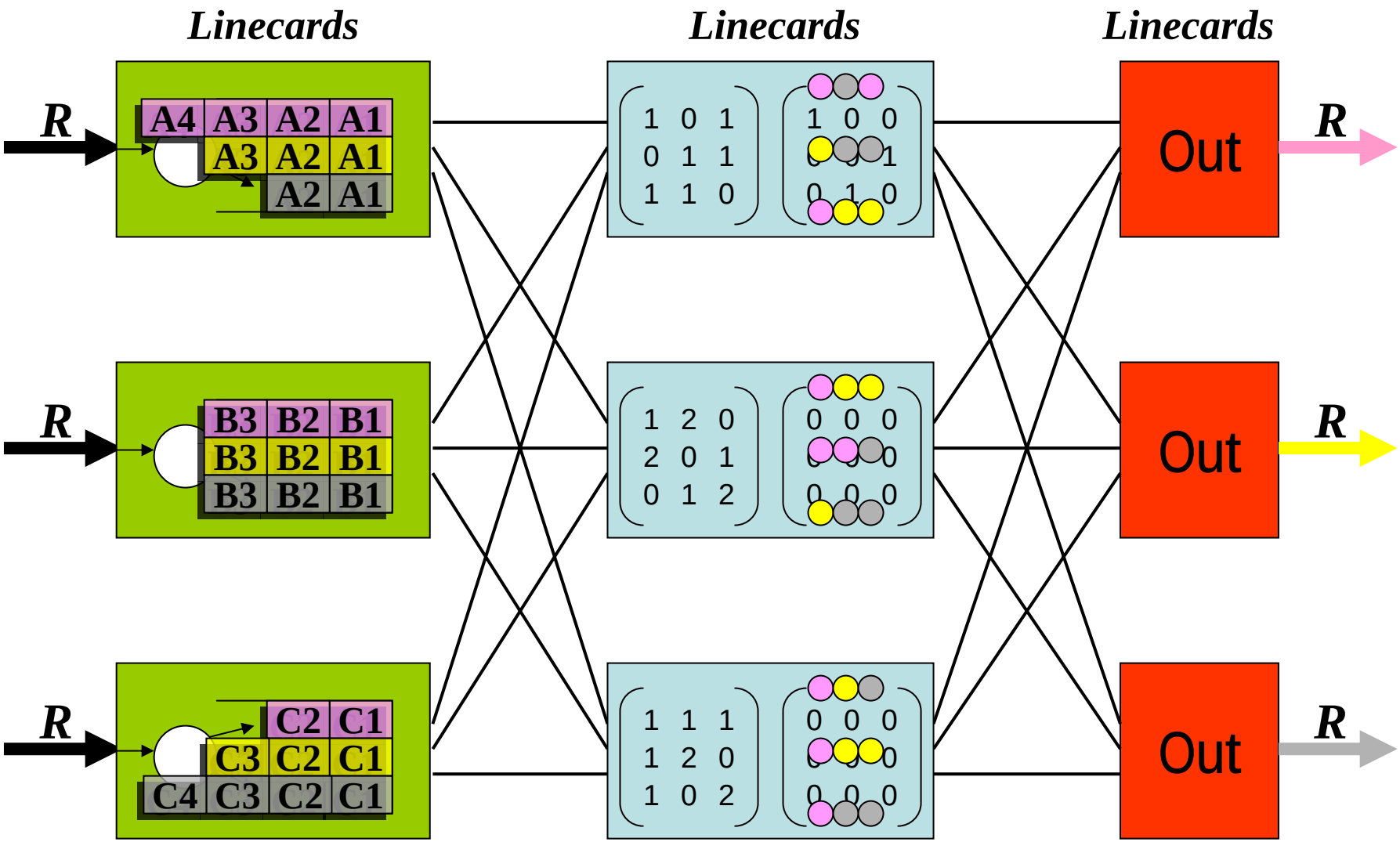
Max Row/Col Sum: 3 = T



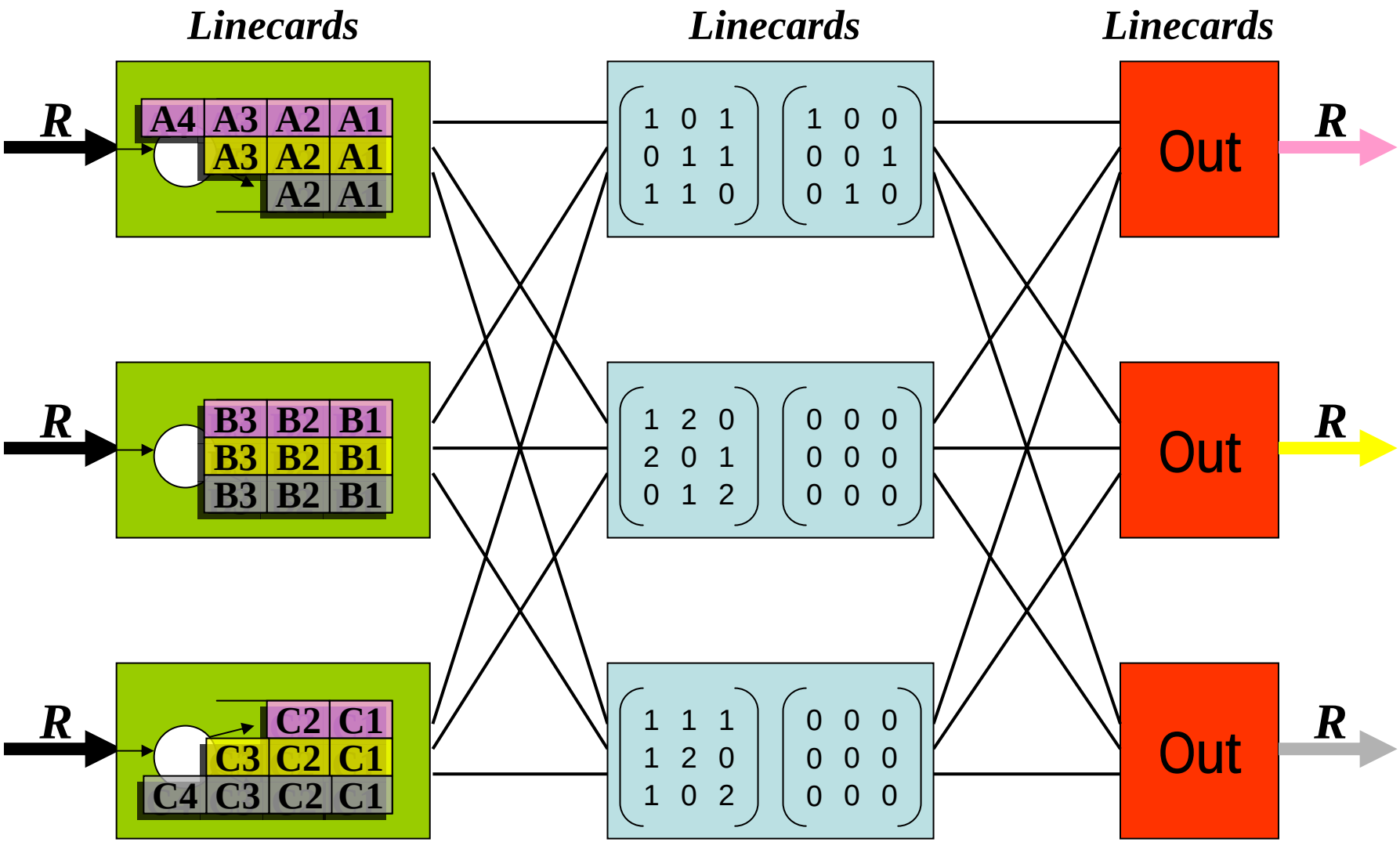
Grants can be sent in Batch

No Scheduling

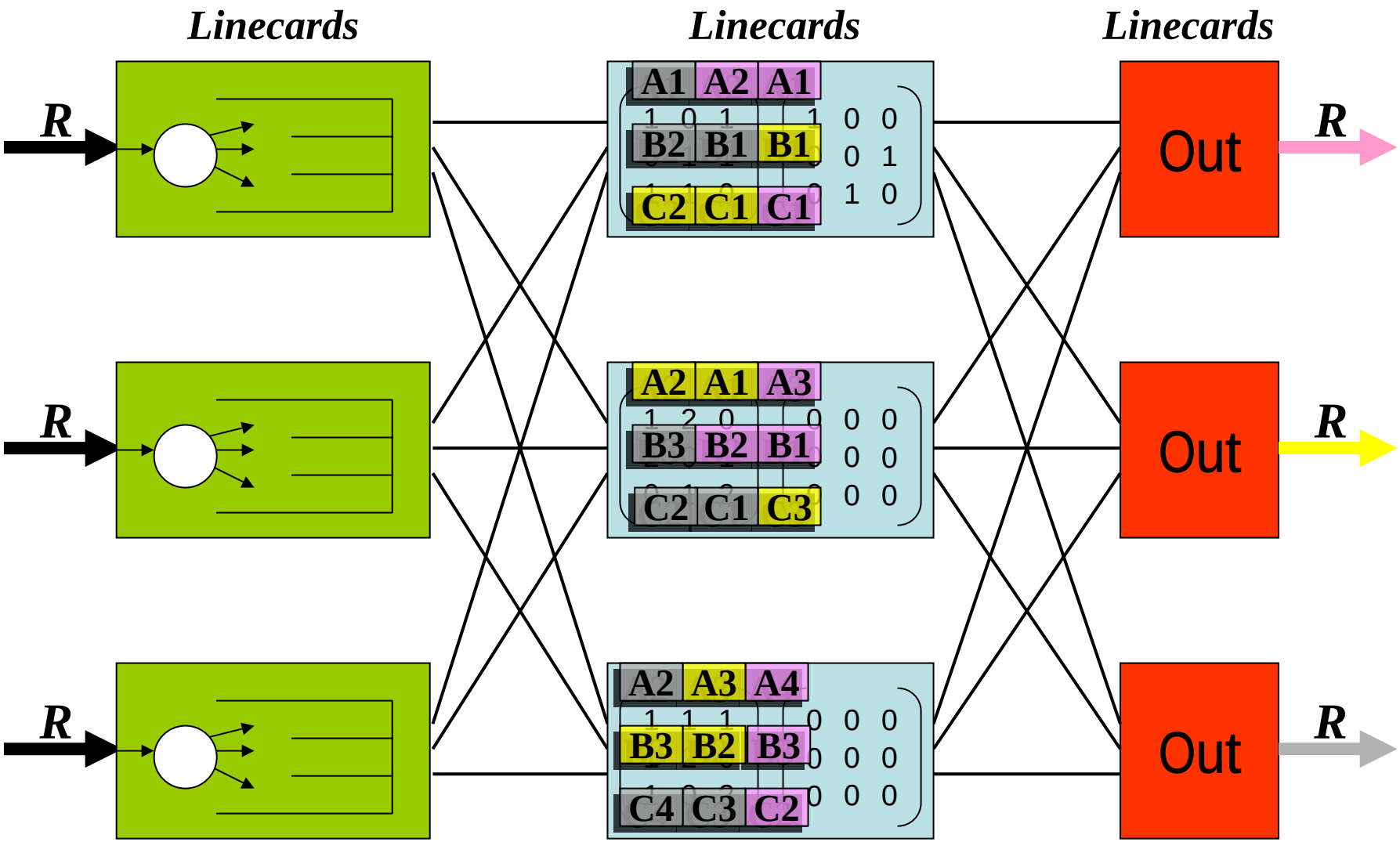
Frame-Aggregated CMS



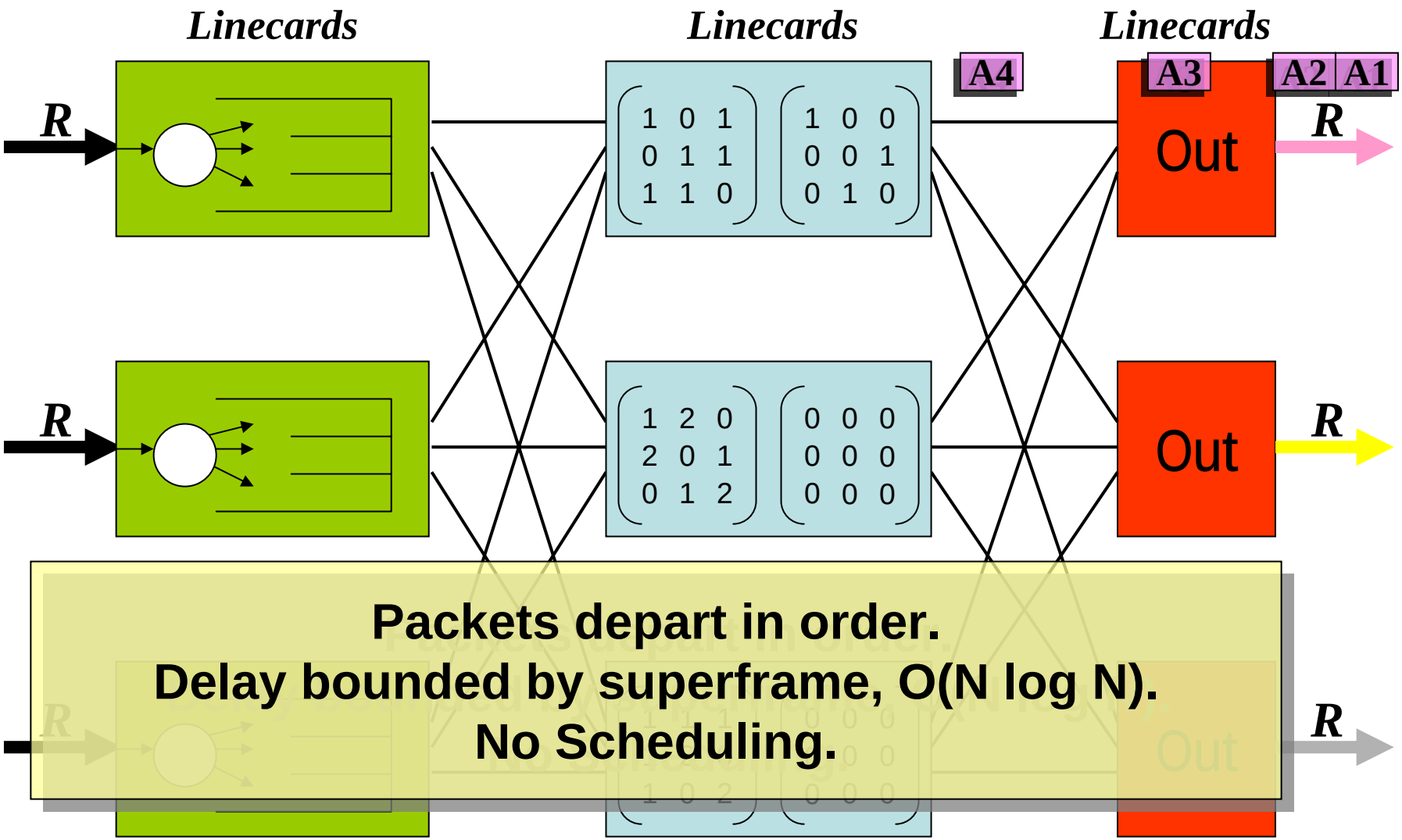
Frame-Aggregated CMS



Frame-Aggregated CMS



Frame-Aggregated CMS



Summary

- We provided a general delay bound for the CMS architecture
- We showed that CMS can be provably bounded by O(N log N) delay for Bernoulli i.i.d. arrival by using a fair-frame scheduler
- We further showed that CMS can be provably bounded by O(N log N) with No Scheduling by means of “Frame Aggregation”, while retaining packet ordering and 100% throughput guarantees
- Our work on CMS and Frame-based CMS provides new way of thinking about scaling routers and connects huge body of existing literature on scheduling to load-balanced routers

Thank You