Advanced Computer Networks

Content-Centric Networking (III)

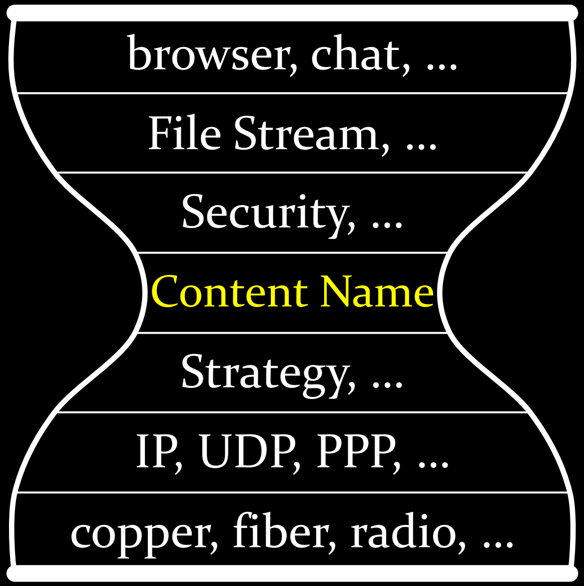
Instructor: *Prof. Dr. Xiaoming Fu*
Presenter: *Jiachen Chen*
Computer Networks Group, Institute of Computer Science
Georg-August-Universität Göttingen

Review

- **Why COPSS?**
  - Temporal separation between providers (publishers) and consumers (subscribers)
  - NDN cannot achieve this via pure query/response model
  - The add-on systems to mitigate the mismatch also introduces overhead

REVIEW

- Why COPSS?
- **How does COPSS achieve Content-Centric Pub/Sub? (protocol level)**



Content Name:

/ugoe.edu/jchen/acn14-ICN.pdf/_v1/_s1

Content Descriptors:

/networking/ICN
/ugoe.edu/acn/2014
/ugoe.edu/jchen

Review

- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)
- **What are the 2 new packet types in COPSS?**

Content Name
Selector (order preference, publisher filter, scope, ..)
Nonce

Interest (Request)

Content Descriptor
Selector (order preference, publisher filter, scope, ..)
Nonce

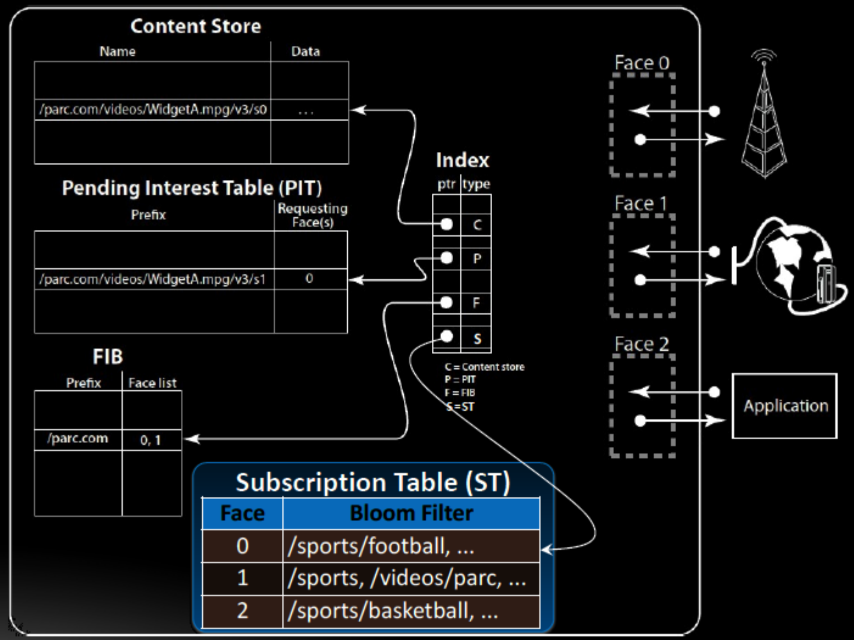
Subscription

Content Name
Content Descriptors
Signature (digest algorithm, witness, …)
Signed Info (publisher ID, key locator, stale time, …)
Data

Data (Response)
or
Publish

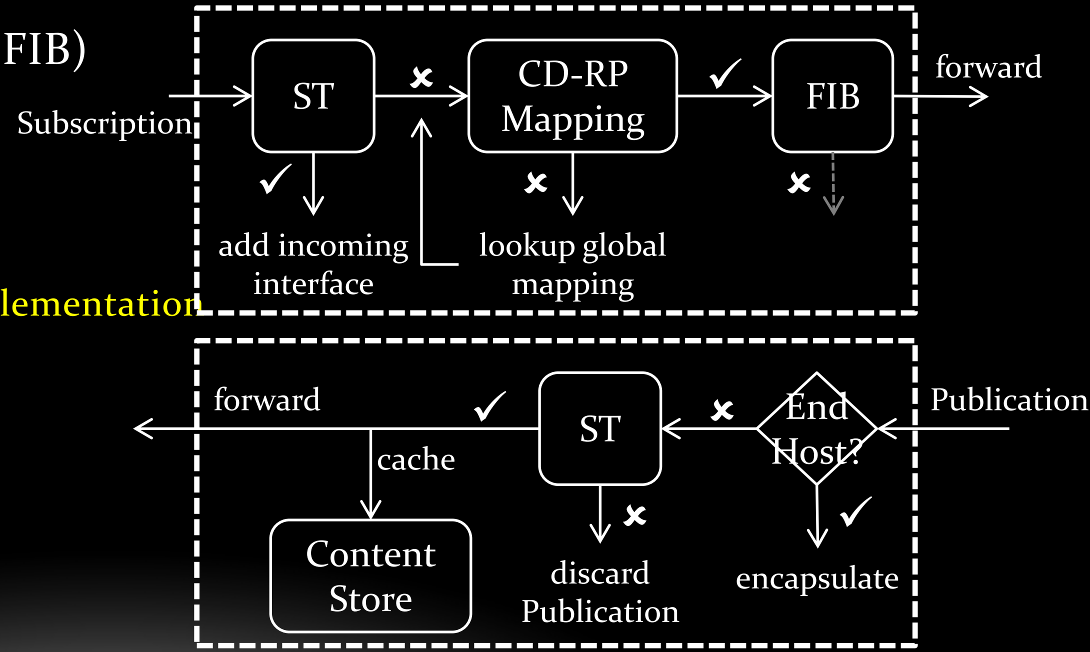
Review

- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)
- What are the 2 new packet types in COPSS?
- **What are the data structures in a COPSS forwarding engine? And functions?**
  - Forwarding Information Base (FIB)
  - Pending Interest Table (PIT)
  - Subscription Table (ST)
  - Content Store



Review

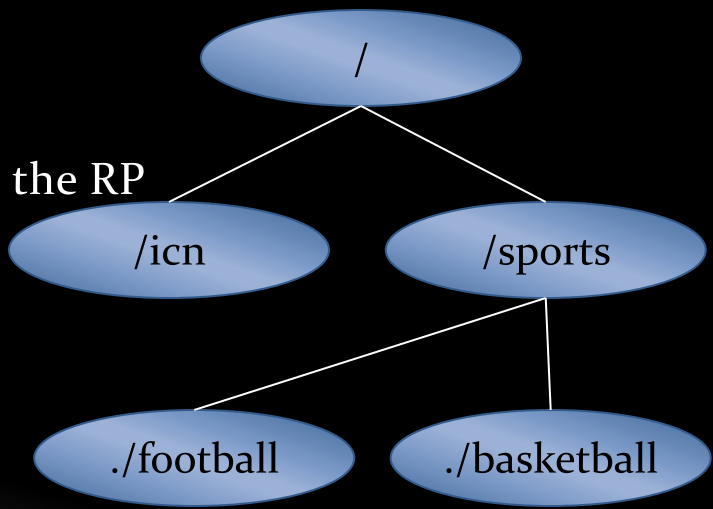
- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)
- What are the 2 new packet types in COPSS?
- **What are the data structures in a COPSS forwarding engine? And functions?**
  - Forwarding Information Base (FIB)
  - Pending Interest Table (PIT)
  - Subscription Table (ST)
  - Content Store
  - * You can separate RP module in implementation



Review

- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)
- What are the 2 new packet types in COPSS?
- What are the data structures in a COPSS forwarding engine? And functions?
- **What is Rendezvous-Point (RP) based communication? How to avoid information concentration?**
  - An RP serves … ch)
  - All the public … gh the RP
  - Automatic RP
  - Note: RP is ju

Prefix	RP Name
/	/RP1
/icn	/RP2
/sports	/RP3
/sports/football	/RP4
/sports/basketball	/RP5



Review

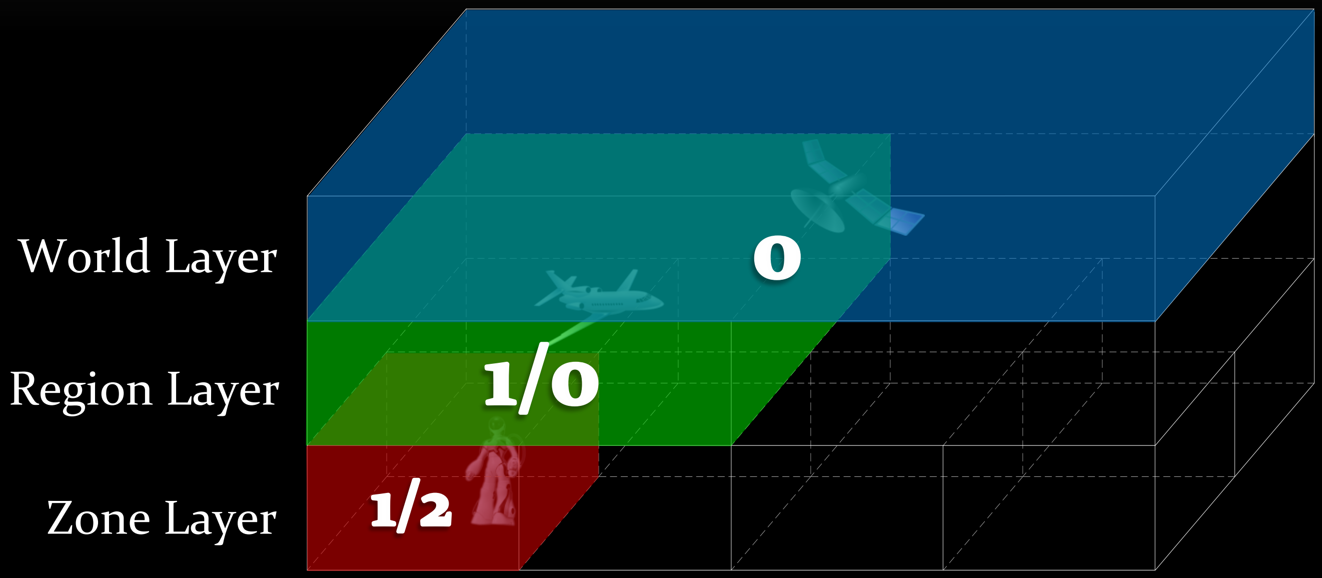
- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)
- What are the 2 new packet types in COPSS?
- What are the data structures in a COPSS forwarding engine? And functions?
- What is Rendezvous-Point (RP) based communication? How to avoid information concentration?
- **How to control CD-RP Map size and ST size for scalability?**
  - CD-RP lookup like DNS
  - Bloom-Filter ST

Review

- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)
- What are the 2 new packet types in COPSS?
- What are the data structures in a COPSS forwarding engine? And functions?
- What is Rendezvous-Point (RP) based communication? How to avoid information concentration?
- How to control CD-RP Map size and ST size for scalability?
- **Why gaming is related to COPSS?**
  - Online gaming needs a communication infrastructure
  - Gaming is content-centric
  - Gaming is pub/sub

Review

- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)



Satellite:
- Location: 0
- Pub: /0
- **Sub: /**

Plane:
- Location: 1/0
- Pub: /1/0
- **Sub: /1, /0**

Soldier:
- Location: 1/2
- Pub: /1/2
- **Sub: /1/2, /1/0, /0**

Review

- Why COPSS?
- How does COPSS achieve Content-Centric Pub/Sub? (protocol level)
- What are the 2 new packet types in COPSS?
- What are the data structures in a COPSS forwarding engine? And functions?
- What is Rendezvous-Point (RP) based communication? How to avoid information concentration?
- How to control CD-RP Map size and ST size for scalability?
- Why gaming is related to COPSS?
- Hierarchical map partitioning → Hierarchical CD structure?
- **Two-step communication?**
  - Subscriber interest
  - Policy control

CONGESTION CONTROL

- Survey: Was TCP designed for congestion control?
  - Yes and no…?
- History:
  - TCP – RFC 675 (1974)[1]
  - Congestion Control – RFC 896 (1984)[2]
  - TCP Congestion Control – RFC 2001 (1997)[3], RFC 2581 (1999)[4], RFC 5681 (2009)[5]
- Question: Why congestion control??
  - Congestion Collapse!

> In heavily loaded pure datagram networks with end to end retransmission, as switching nodes become congested, the round trip time through the net increases and the count of datagrams in transit within the net also increases. This is normal behavior under load. As long as there is only one copy of each datagram in transit, congestion is under control. Once retransmission of datagrams not yet delivered begins, there is potential for serious trouble. [2]

Why TCP Works? (w/o C.C.)

- Basic Send & ACK mode:

Why TCP Works? (w/o C.C.)

- Now, let's have a window = 6

Why TCP Works? (w/o C.C.)

- Now, let's have a window = 22

Why TCP Works? (w/o C.C.)

- Now, let's have a window = 22
- Why 22?
  - Available Bandwidth = 20 Mbps
  - RTT (Delay) = 12ms + *
  - $B \times D = \frac{20 \times 1000000b}{1000ms} \times 12ms = 240000b$
  - Packet Size: 12000b
  - Therefore: > 20 packets

Why TCP Does Not Work? (w/o C.C.)

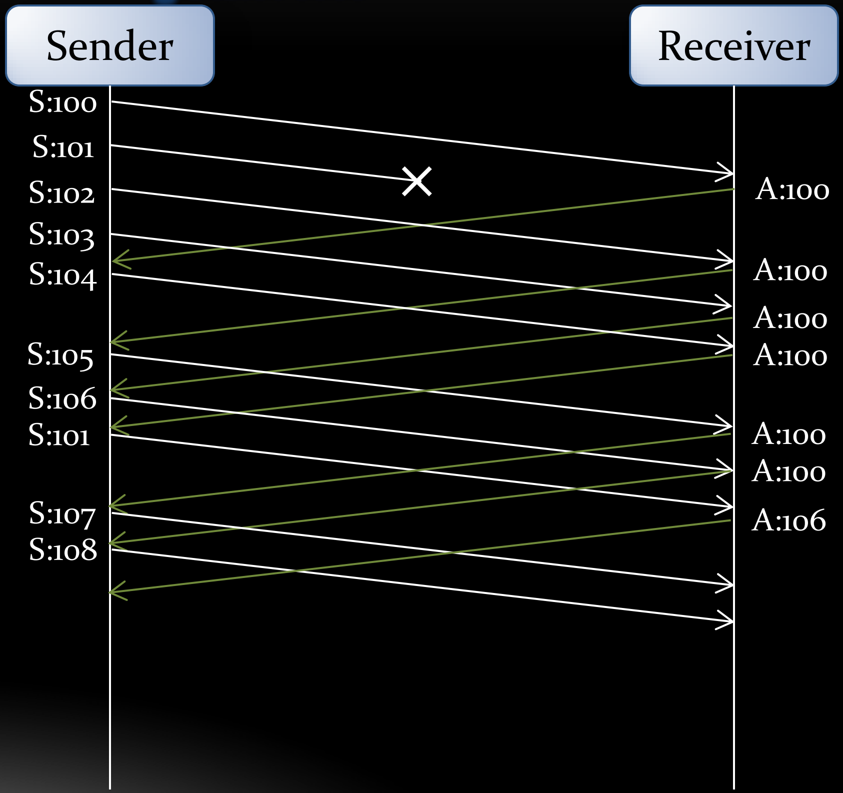
- What about… window size = 40??

Why TCP Does Not Work? (w/o C.C.)

- It will be more severe under load…

TCP C.C. General Idea

- Window Control: AIMD
  - Additive Increase: Increase window by 1 on successful delivery of 1 window
  - Multiplicative Decrease: Decrease window by half on packet loss
- Key: Fast Recovery/Retransmit
  - ACK: ACK the last position correct in order
  - On triple duplicate ACK, start fast recovery
    - On the 3rd duplicate ACK
      - ssthresh = cwnd / 2
      - cwnd = ssthresh + 3
      - Retransmit the missing packet
    - On the following duplicate ACKs
      - cwnd++
      - Send new packet if cwnd allows
    - On the 1st non-duplicate ACK
      - cwnd = ssthresh
      - Go to congestion avoidance



TCP C.C. General Idea

- Fast Retransmit Demo

TCP C.C. General Idea

- Fast Retransmit Demo (competition)

Classification

Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
Window	Independent	✓	✓
Rate	Assisted	✗	✗
N/A	Dependent		

CLASSIFICATION

- TCP:

Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
Window	Independent	✓	✓
Rate	Assisted	✗	✗
N/A	Dependent		

SOLUTION 1: ICP [6]

- Classification:

Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
Window	Independent	✓	✓
Rate	Assisted	✗	✗
N/A	Dependent		

- Thoughts:
  - Apply TCP directly in ICN

Solution 1: ICP [6]

- AIMD:
  - W is increased by α/W on Data reception ($\alpha = 1$)
  - W is decreased to $\beta \times W$ on timer expiration ($\beta < 1$)
- Timer:
  - $\tau = RTT_{min} + (RTT_{max} - RTT_{min})^{\delta}$
  - Maximum and minimum RTT averaged over a history of samples (20 Data)
  - $\delta = 0.5$
- Issue:
  - RTT can vary drastically due to the caching

SOLUTION 2: CONTUG [7]

- Classification:

Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
Window	Independent	✓	✓
Rate	Assisted	×	×
N/A	Dependent		

- Thoughts:
  - Solve the issue: different sources might have different delay

SOLUTION 2: CONTUG [7]

- Every source marks the packet with its ID
- Per-source Conceptual $CWND$: $CCWND = \sum_i CCWND_i$
- Per-source RTT Measurement (EWMA):
  - $\overline{RTT_i} = \alpha \times \overline{RTT_i} + (1 - \alpha) \times Sample_i$
- Weak congestion indication (like TCP Vegas): increased RTT estimation

$$ExpectedRate_i = \frac{CCWND_i \times SegSize}{BASERTT_i} \quad ActualRate_i = \frac{CCWND_i \times SegSize}{RTT_i}$$

  - $diff < 1$: slow start; $diff < \alpha$: congestion avoidance; $diff > \beta$: linear decrease
- Strong congestion indication: timeout

$$Timeout_i = C \times \overline{RTT_i} \qquad Timeout = C \times Max_{i \in \{0,n\}} RTT_i$$

  - Reduce window by half on timeout

SOLUTION 3: MULTI-PATH ICP [8]

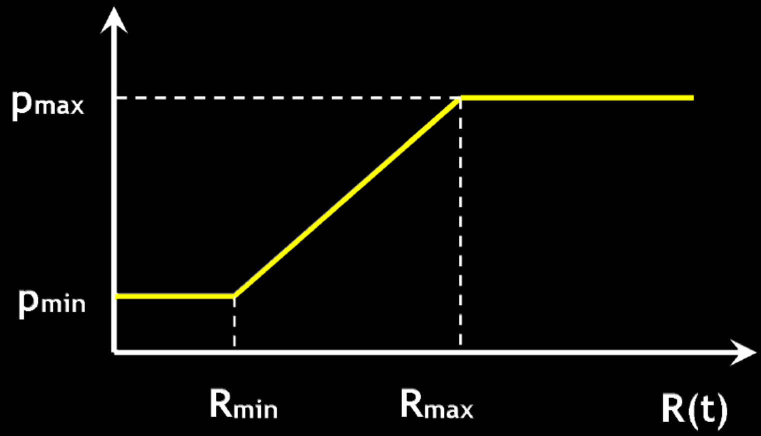
- Classification:

Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
Window	Independent	✓	✓
Rate	Assisted	×	×
N/A	Dependent		

- Thoughts:
  - Solve the issue: different sources might have different delay
  - With some privacy

Solution 3: Multi-Path ICP [8]

- Route labeling:
  - Source and intermediate routers mark the packet with its ID / face
- Route delay monitoring:
  - Min, max RTT from a specific route
- Per-route probabilistic window decrease (Active Queue Management):
  - $p_r(t) = p_{min} + (p_{max} - p_{min}) \frac{R_r(t) - R_{r,min}(t)}{R_{r,max}(t) - R_{r,min}(t)}$



- Issues:
  - Receiver has to know all the paths?
  - What about the new paths?
- Extend reading: [9]

SOLUTION 4: HOBHIS [10]

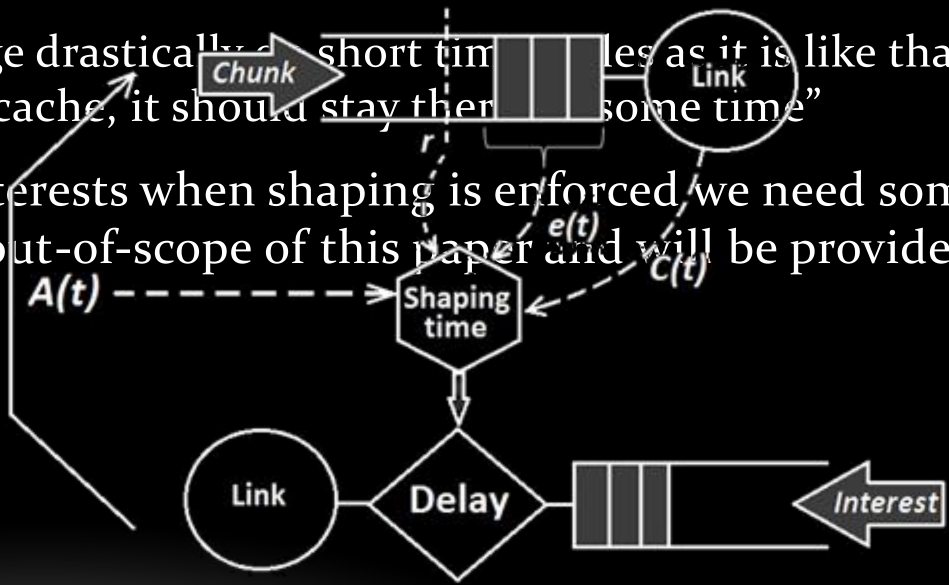
- Classification

Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
Window	Independent	✓	✓
Rate	Assisted	×	×
N/A	Dependent		

- Thoughts:
  - Flow fairness controlled directly at the routers

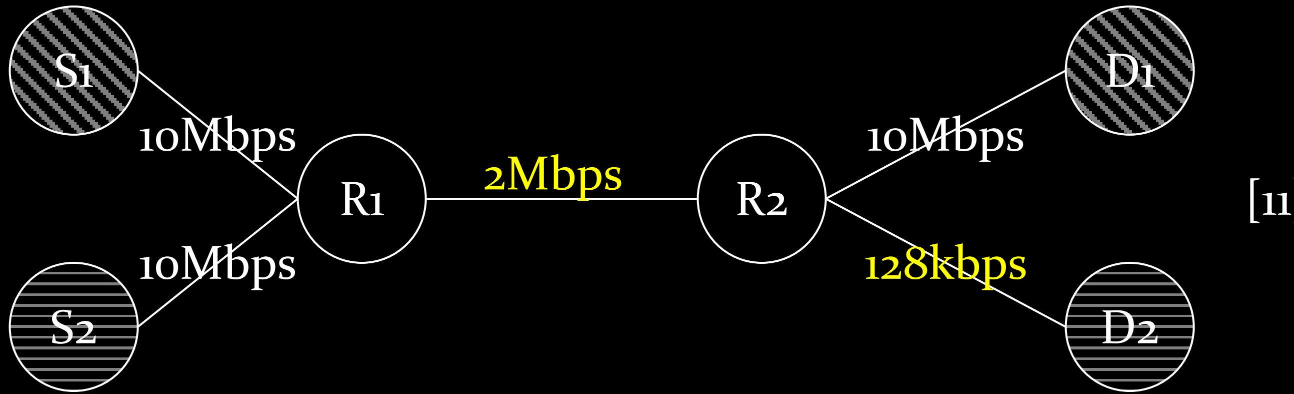
SOLUTION 4: HOBHIS [10]

- "The congestion control scheme based on hop-by-op interest shaping was preferred to an end-to-end mechanism such as TCP"
- Basic idea: limit the Interest forwarding to control the return rate after a "Response Delay"
- Assumption:
  - Response Delay "should not change drastically on short time scales as it is like that when the data is stored in a given cache, it should stay there for some time"
- "In order to avoid the losses from Interests when shaping is enforced, we need some back-pressure mechanism which is out-of-scope of this paper and will be provided in future work"



Solution 4: HoBHIS [10]

- Issues:
  - Flow state maintenance
  - Efficiency on multiple bottlenecks



- Extend readings:
  - Flow aware traffic control [12]: similar idea, control the flow fairness at the routers
  - Improved HoBHIS [13]: HoBIS + receiver reaction (AIMD)
  - HR-ICP [14]: ICP + bit-bucket (per-flow credit) on the routers
  - CCTCP [15]: Put everything into the packet (route, RTT, etc.)

SOLUTION 6: DEADLINE-BASED [16]

- Classification

Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
Window	Independent	✓	✓
Rate	Assisted	×	×
N/A	Dependent		

- Thoughts:
  - Can we use computer scheduling mechanisms to schedule the network?

Solution 6: Deadline-based [16]

- Idea:
  - Prioritize the packets with short available delay
  - Drop packets that cannot meet the requirement

- Issues:
  - Global clock / Router delay measurement
  - Scheduling issues?

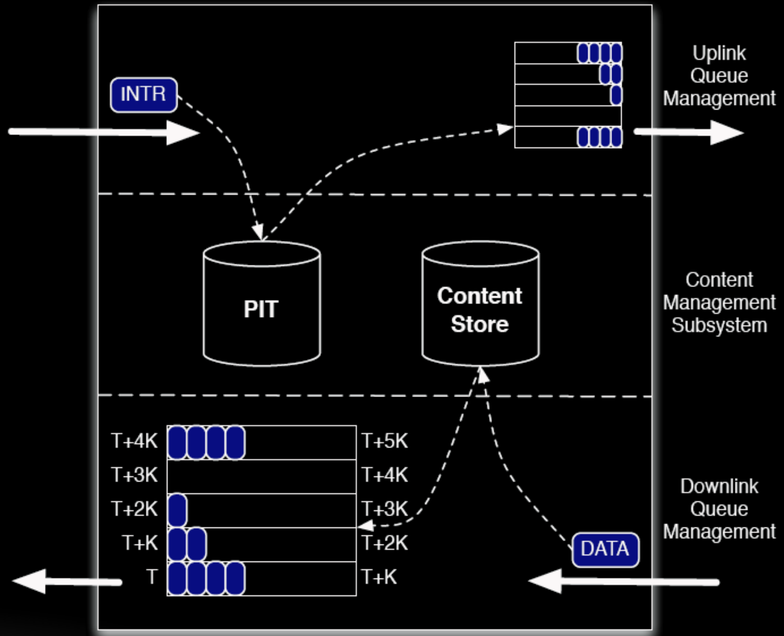
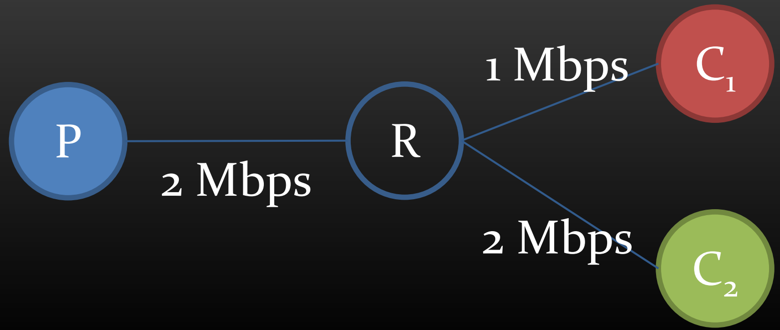
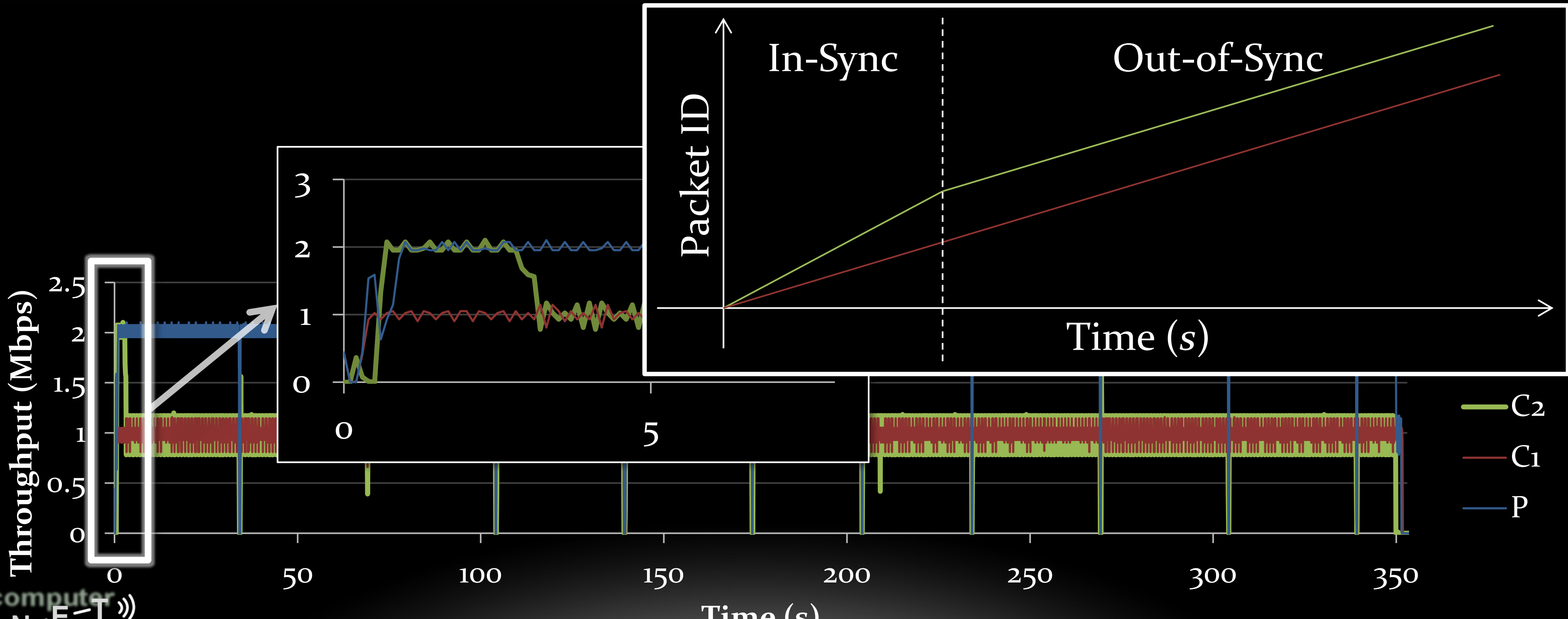


Figure 1: Queuing model for lifetimes.

From another Perspective

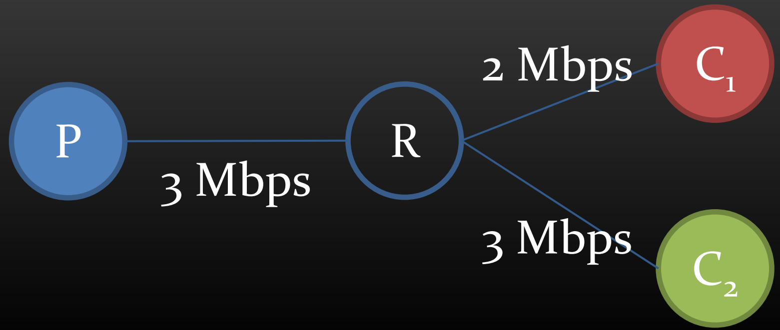
- Efficiency??

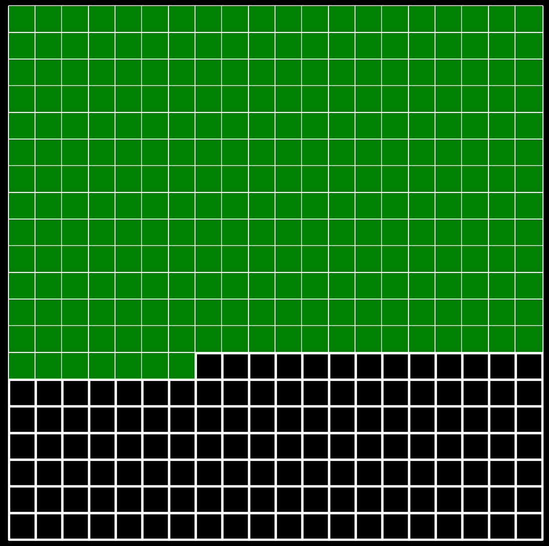


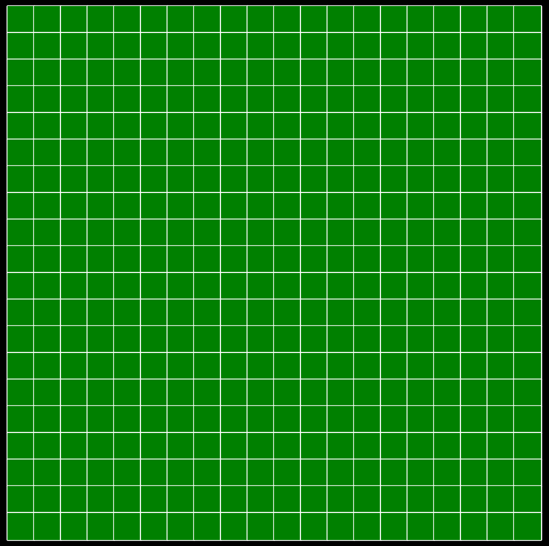


From another Perspective

- Efficiency??

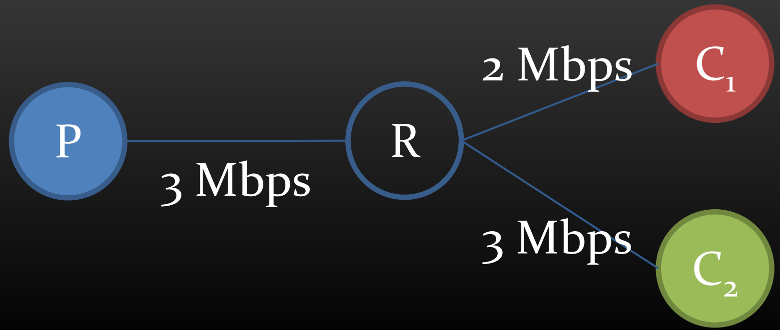


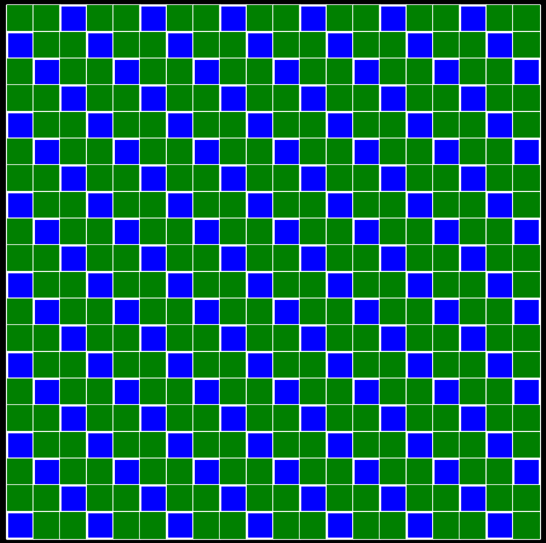


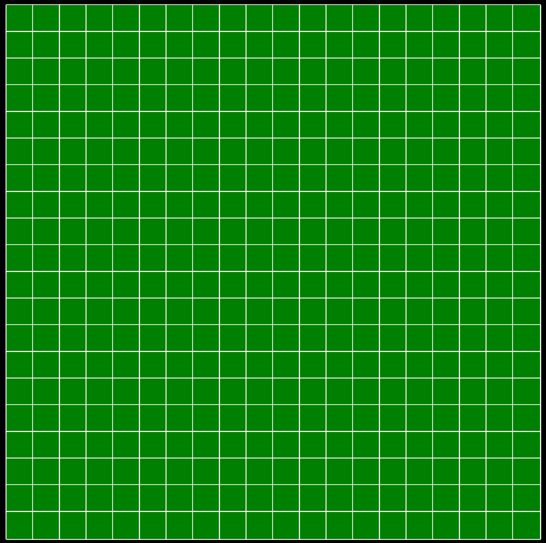


FROM ANOTHER PERSPECTIVE

- Efficiency??







Recap

- Issues with ICN congestion control:
  - No fast recovery due to receiver-driven
  - Multiple sources have different delay
- Solutions:

	Control Metrics	Router Assistance	Flow Awareness	Multipath Awareness
ICP [6]	Window	Independent	✗	✗
ConTug [7]	Window	Independent	✗	✓
MultiPath ICP [8]	Window	Assisted	✗	✓
Optimal Multipath ICP [9]	Window	Assisted	✗	✓
HoBHIS [10]	N/A	Dependent	✓	✗
Flow-aware traffic ctrl. [12]	Window	Dependent	✓	✗
Improved HoBHIS [13]	Window	Assisted	✓	✗
HR-ICP [14]	Window	Assisted	✓	✗
CCTCP [15]	Window	Assisted	✓	✓
Deadline based [16]	Rate	Dependent	✗	✗

- Further issues:
  - Out of sync??

REFERENCES

1. Cerf, Vinton, *et al*. "*RFC 675: Specification Internet Transmission Control Program*" (1974).
2. Nagle, John. "*RFC 896: Congestion control in IP/TCP internetworks.*" (1984).
3. Stevens, W. Richard. "*RFC 2001: TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms*." (1997).
4. Allman, Mark , *et al*. "*RFC 2581: TCP congestion control.*" (1999).
5. Allman, Mark, *et al*. "*RFC 5681: TCP Congestion Control*." (2009).
6. Carofiglio, Giovanna, *et al*. "*ICP: Design and evaluation of an interest control protocol for content-centric networking*." NOMEN, 2012.
7. Arianfar, Somaya, *et al*. "*Contug: A receiver-driven transport protocol for content-centric networks*." Under submission, 2010.
8. Carofiglio, Giovanna, *et al*. "*Multipath congestion control in content-centric networks*." NOMEN, 2013.
9. Carofiglio, Giovanna, *et al*. "*Optimal multipath congestion control and request forwarding in Information-Centric Networks.*" ICNP, 2013.
10. Rozhnova, Natalya, *et al*. "*An effective hop-by-hop Interest shaping mechanism for CCN communications*." NOMEN, 2012.
11. Floyd, Sally, *et al*. "*Promoting the use of end-to-end congestion control in the Internet*." Transactions on Networking (TON), 1999: 458-472.
12. Oueslati, Sara, *el al*. "*Flow-aware traffic control for a content-centric network.*" INFOCOM, 2012.
13. Wang, Yaogong, et al. "*An improved hop-by-hop interest shaper for congestion control in named data networking*." ICN, 2013.
14. Carofiglio, Giovanna, *et al*. "*Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks*." ICN, 2012.
15. Saino, Lorenzo, *et al*. "*Cctcp: A scalable receiver-driven congestion control protocol for content centric networking*." ICC, 2013.
16. Arianfar, Somaya, *et al*. "*Deadline-based resource management for information-centric networks*." ICN, 2013.