Overflow Management with Multipart Packets

Dror Rawitz
Tel-Aviv University

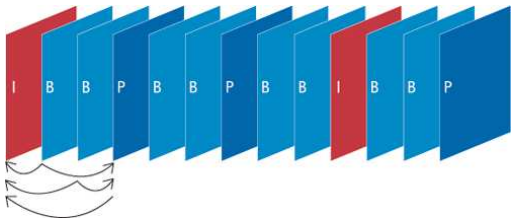
Joint work with Yishay Mansour and Boaz Patt-Shamir
To appear in INFOCOM 2011

Motivation

Video Transmission Over Networks:

- **Large Data Items**:
  - Sequence consisting of large frames
    (I-frames: SD hundreds of Kb; HD several Mb)

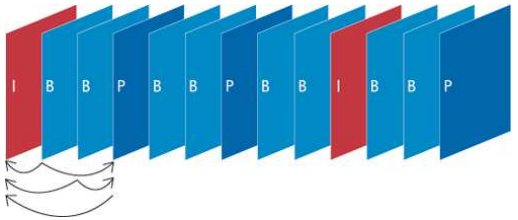
- **Small Transfer Units**:
  - IP packet size ≤ 64Kb
  - Practically ≤ 1.5Kb (Ethernet)



Motivation

Video Transmission Over Networks:

- **Large Data Items**:
  - Sequence consisting of large frames
    (I-frames: SD hundreds of Kb; HD several Mb)



- **Small Transfer Units**:
  - IP packet size ≤ 64Kb
  - Practically ≤ 1.5Kb (Ethernet)

- **Problem**:
  - Packets arrive in bursts at an outgoing link of a router
  - Which packets are dropped when buffer is full?
  - If enough packets are dropped, the whole frame may be lost!

Online Scheduling of Superpackets

Input:

- Superpackets consisting of k packets
- Packets arrive in bursts of size $\leq \sigma$
- Server capacity is c packets/time unit
- Server has a buffer of size b
- Packets that are not transmitted or saved in buffer are lost
- Superpacket is useful only if $k^* \stackrel{\triangle}{=} (1-\beta)k$ of its packets survive (FEC)

Online Scheduling of Superpackets

Input:

- Superpackets consisting of k packets
- Packets arrive in bursts of size $\leq \sigma$
- Server capacity is c packets/time unit
- Server has a buffer of size b
- Packets that are not transmitted or saved in buffer are lost
- Superpacket is useful only if $k^* \stackrel{\triangle}{=} (1-\beta)k$ of its packets survive (FEC)

Question: which packet to serve?!

Online Scheduling of Superpackets

Input:

- Superpackets consisting of k packets
- Packets arrive in bursts of size $\leq \sigma$
- Server capacity is c packets/time unit
- Server has a buffer of size b
- Packets that are not transmitted or saved in buffer are lost
- Superpacket is useful only if $k^* \triangleq (1-\beta)k$ of its packets survive (FEC)

Question: which packet to serve?!

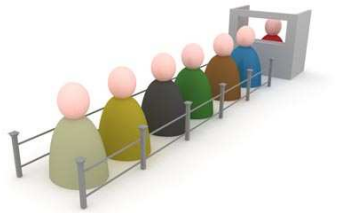
Special Cases:

- Capacity ($c = 1$ or $c > 1$)
- Buffer ($b = 0$ or $b > 0$)
- Redundancy ($\beta = 0$ or $\beta > 0$)

Related Work: Deterministic Algorithms

Special Case: FIFO buffer, unit capacity ($c = 1$), no redundancy ($\beta = 0$)

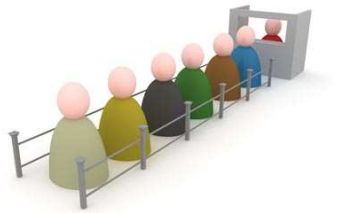
- FIFO Buffer Management [Kesselman, Patt-Shamir, Scalosub 09]
  - No competitive online algorithm, even for $k = 2$
  - Assumption: order respecting sequences
    - $\Omega(k)$ lower bound
    - $O(k^2)$-competitive algorithm



Related Work: Deterministic Algorithms

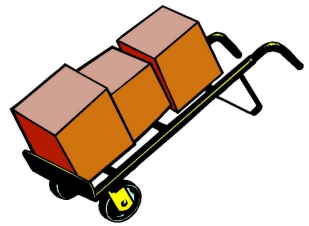
Special Case: FIFO buffer, unit capacity ($c = 1$), no redundancy ($\beta = 0$)

- FIFO Buffer Management [Kesselman, Patt-Shamir, Scalosub 09]
  - No competitive online algorithm, even for $k = 2$
  - Assumption: order respecting sequences
    - $\Omega(k)$ lower bound
    - $O(k^2)$-competitive algorithm



- Aggregated Streaming Data [Scalosub, Marbach, and Liebeherr 10]
  - Assumption: traffic consists of M aggregated streams
    - $\Omega(kM/b)$ deterministic lower bound
    - $O((kMb + M)^k + 1)$-competitive algorithm
    - Simulations: algorithm outperforms versions of tail-drop

Related Work: Online Set Packing



Special Case: no buffer ($b = 0$), no redundancy ($\beta = 0$)

- Online Set Packing

  [Emek, Halldórsson, Mansour, Patt-Shamir, Radhakrishnan, R 10]

  - σ^{k-1} deterministic lower bound
  - $\tilde{\Omega}(k\sqrt{\sigma})$ randomized lower bound (unit capacity, centralized)
  - $O(k\sqrt{\frac{\sigma}{c}})$-competitive randomized algorithm (capacitated, distributed) (can be refined depending on uniformity of parameters)

Algorithm Priority

Idea: Consistent randomization

Algorithm:

- For each superpacket S pick a random priority $r(S) \sim U[0,1]$
- Upon arrival of a burst:
  Service c superpackets with highest priority in burst

Algorithm Priority

Idea: Consistent randomization

Algorithm:

- For each superpacket S pick a random priority $r(S) \sim U[0,1]$
- Upon arrival of a burst:
  Service c superpackets with highest priority in burst

Advantages:

- Very simple
- Easily distributed
- Extends to weighted superpackets

Our Results

- Online set packing with redundancy
  - $O(\sqrt{kk^*\sigma/c})$ upper bound on the competitive ratio,
    where $k^* \triangleq (1-\beta)k$
    (can be refined depending on uniformity of parameters)

Our Results

- Online set packing with redundancy
  - $O(\sqrt{kk^*\sigma/c})$ upper bound on the competitive ratio, where $k^* \triangleq (1-\beta)k$ (can be refined depending on uniformity of parameters)

- Large capacity/buffer with uniform burst size
  - $b = 0$, $c = O(\frac{1}{\varepsilon^2} \log \frac{k^*}{\varepsilon})$ $\Rightarrow$ goodput $\geq (1-\varepsilon)^2(1-\beta)$
  - Dual buffer, $b = O(\frac{6k}{\varepsilon^2} \log \frac{k}{\varepsilon})$ $\Rightarrow$ goodput $\geq (1-\varepsilon)^3(1-\beta)$

Our Results

- Online set packing with redundancy
  - $O(\sqrt{kk^*\sigma/c})$ upper bound on the competitive ratio, where $k^* \triangleq (1-\beta)k$ (can be refined depending on uniformity of parameters)

- Large capacity/buffer with uniform burst size
  - $b = 0$, $c = O(\frac{1}{\varepsilon^2} \log \frac{k^*}{\varepsilon})$ $\Rightarrow$ goodput $\geq (1-\varepsilon)^2(1-\beta)$
  - Dual buffer, $b = O(\frac{6k}{\varepsilon^2} \log \frac{k}{\varepsilon})$ $\Rightarrow$ goodput $\geq (1-\varepsilon)^3(1-\beta)$

- Ignoring large bursts
  - Large bursts satisfy $\sigma(t) > \alpha \cdot c$
  - Can be ignored using redundancy
  - Analysis of Algorithm Priority when ignoring large bursts

Online Set Packing with Redundancy

Special Case: no buffer ($b = 0$), unit capacity ($c = 1$)

- Consider a superpacket S
  - $S' \subseteq S$ such that $|S'| = k^* = (1 - \beta)k$
  - $N(S') = \{T : T \text{ is in conflict with } S'\}$

Online Set Packing with Redundancy

Special Case: no buffer ($b = 0$), unit capacity ($c = 1$)

- Consider a superpacket S
  - $S' \subseteq S$ such that $|S'| = k^* = (1 - \beta)k$
  - $N(S') = \{T : T \text{ is in conflict with } S'\}$
  - Observation: $|N(S')| < k^*\sigma$

Online Set Packing with Redundancy

Special Case: no buffer ($b = 0$), unit capacity ($c = 1$)

- Consider a superpacket S
  - $S' \subseteq S$ such that $|S'| = k^* = (1 - \beta)k$
  - $N(S') = \{T : T \text{ is in conflict with } S'\}$
  - Observation: $|N(S')| < k^*\sigma$

  $\Rightarrow$ $\Pr[S \in \text{ALG}] \geq \frac{1}{k^*\sigma}$

Online Set Packing with Redundancy

Special Case: no buffer ($b = 0$), unit capacity ($c = 1$)

- Consider a superpacket S
  - $S' \subseteq S$ such that $|S'| = k^* = (1 - \beta)k$
  - $N(S') = \{T : T \text{ is in conflict with } S'\}$
  - Observation: $|N(S')| < k^*\sigma$

  $\Rightarrow$ $\Pr[S \in \text{ALG}] \geq \frac{1}{k^*\sigma}$

$$\Rightarrow \quad \mathbb{E}[|\text{ALG}|] \;=\; \sum_S \Pr[S \in \text{ALG}] \;\geq\; \frac{n}{k^*\sigma} \;\geq\; \frac{|\text{OPT}|}{k^*\sigma}$$

Online Set Packing with Redundancy

Special Case: no buffer ($b = 0$), unit capacity ($c = 1$)

- Consider a superpacket S
  - $S' \subseteq S$ such that $|S'| = k^* = (1 - \beta)k$
  - $N(S') = \{T : T \text{ is in conflict with } S'\}$
  - Observation: $|N(S')| < k^*\sigma$

  $\Rightarrow$ $\Pr[S \in \text{ALG}] \geq \frac{1}{k^*\sigma}$

$$\Rightarrow \quad \mathbb{E}[|\text{ALG}|] \;=\; \sum_S \Pr[S \in \text{ALG}] \;\geq\; \frac{n}{k^*\sigma} \;\geq\; \frac{|\text{OPT}|}{k^*\sigma}$$

- Algorithm Priority is $k^*\sigma$-competitive

Online Set Packing with Redundancy

Special Case: no buffer ($b = 0$), unit capacity ($c = 1$)

- Consider a superpacket S
  - $S' \subseteq S$ such that $|S'| = k^* = (1-\beta)k$
  - $N(S') = \{T : T \text{ is in conflict with } S'\}$
  - Observation: $|N(S')| < k^*\sigma$

  $\Rightarrow$ $\Pr[S \in \text{ALG}] \geq \frac{1}{k^*\sigma}$

$$\Rightarrow \quad \mathbb{E}[|\text{ALG}|] \;=\; \sum_S \Pr[S \in \text{ALG}] \;\geq\; \frac{n}{k^*\sigma} \;\geq\; \frac{|\text{OPT}|}{k^*\sigma}$$

- Algorithm Priority is $k^*\sigma$-competitive

- Our result: Algorithm Priority is $O(\sqrt{kk^*\sigma/c})$-competitive

Large Capacity/Buffer

- Large Capacity
  - Let $r(S)$ be the priority of S
  - If σ is uniform and c is "large enough"
    - roughly $r(S) \cdot \sigma$ superpackets from burst have smaller priorities
    - $r(S) \geq 1 - \frac{c}{\sigma} + \delta \Rightarrow S$ survives all its bursts w.h.p.
    - $r(S) \leq 1 - \frac{c}{\sigma} - \delta \Rightarrow S$ is dropped in all its bursts w.h.p.
  - Priority's goodput is close to $(1 - \beta)$

Large Capacity/Buffer

Large Capacity

- Let $r(S)$ be the priority of S
- If σ is uniform and c is "large enough"
  - roughly $r(S) \cdot \sigma$ superpackets from burst have smaller priorities
  - $r(S) \geq 1 - \frac{c}{\sigma} + \delta \Rightarrow S$ survives all its bursts w.h.p.
  - $r(S) \leq 1 - \frac{c}{\sigma} - \delta \Rightarrow S$ is dropped in all its bursts w.h.p.
- Priority's goodput is close to $(1 - \beta)$

Large Buffer

- We use Priority with a Dual Buffer to "simulate" large capacity
- If σ is uniform and b is "large enough" then goodput is close to $(1 - \beta)$

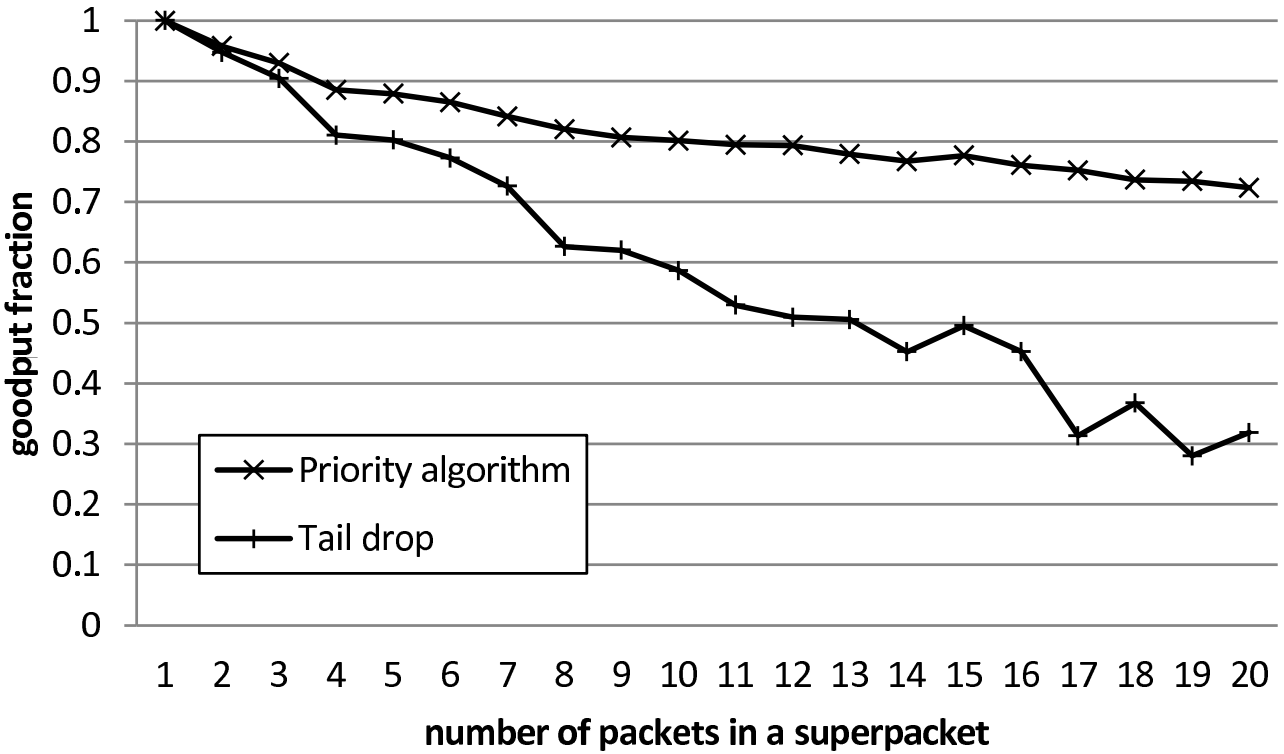
Simulations

- Our tunable parameters:
  - $k =$ number of packets/superpacket
  - $c =$ capacity
  - $b =$ buffer size
  - $\beta =$ redundancy

- Traffic:
  - Aggregate of 10 on/off processes with a tunable parameter $\lambda = \lambda_{\text{on}}/\lambda_{\text{off}}$
  - Packets are associated with superpackets using a random permutation
  - Each data point represents the average of 10 runs

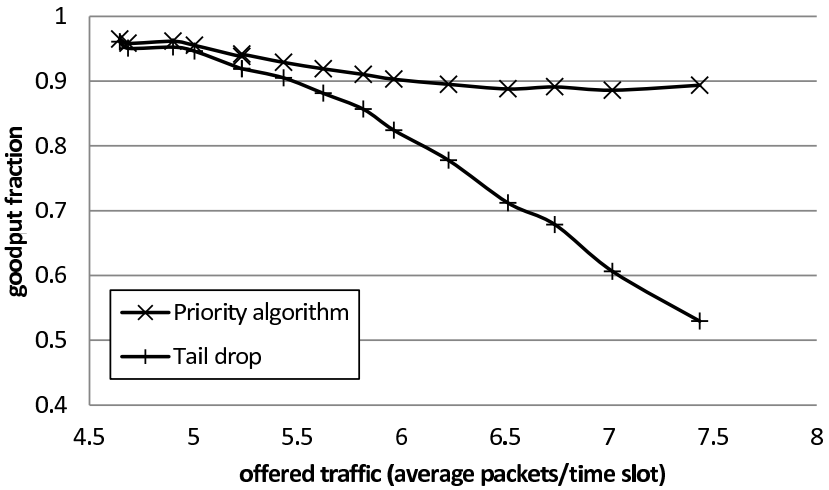
- Performance compared to:
  - Upper bound on maximum possible goodput
  - Tail-drop policy

Size of Superpacket (k)

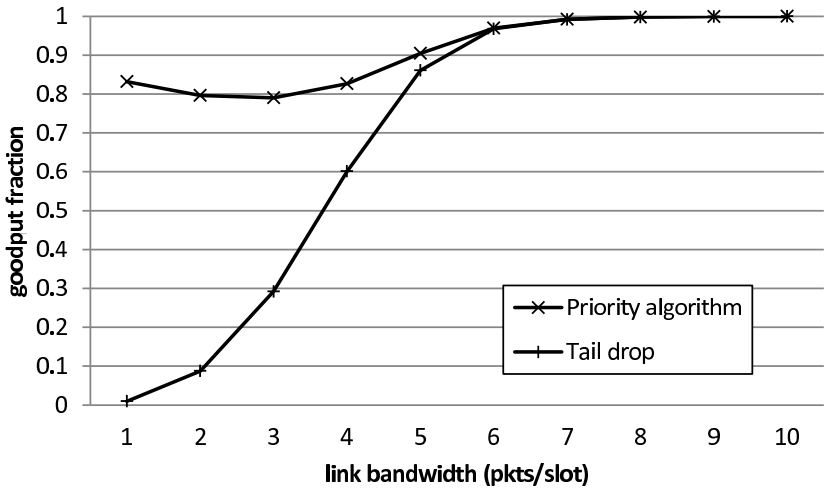


- $c = 5$; $b = 10$; $\overline{\sigma} = 4.67$

Offered Load ($\overline{\sigma}$) and Capacity (c)

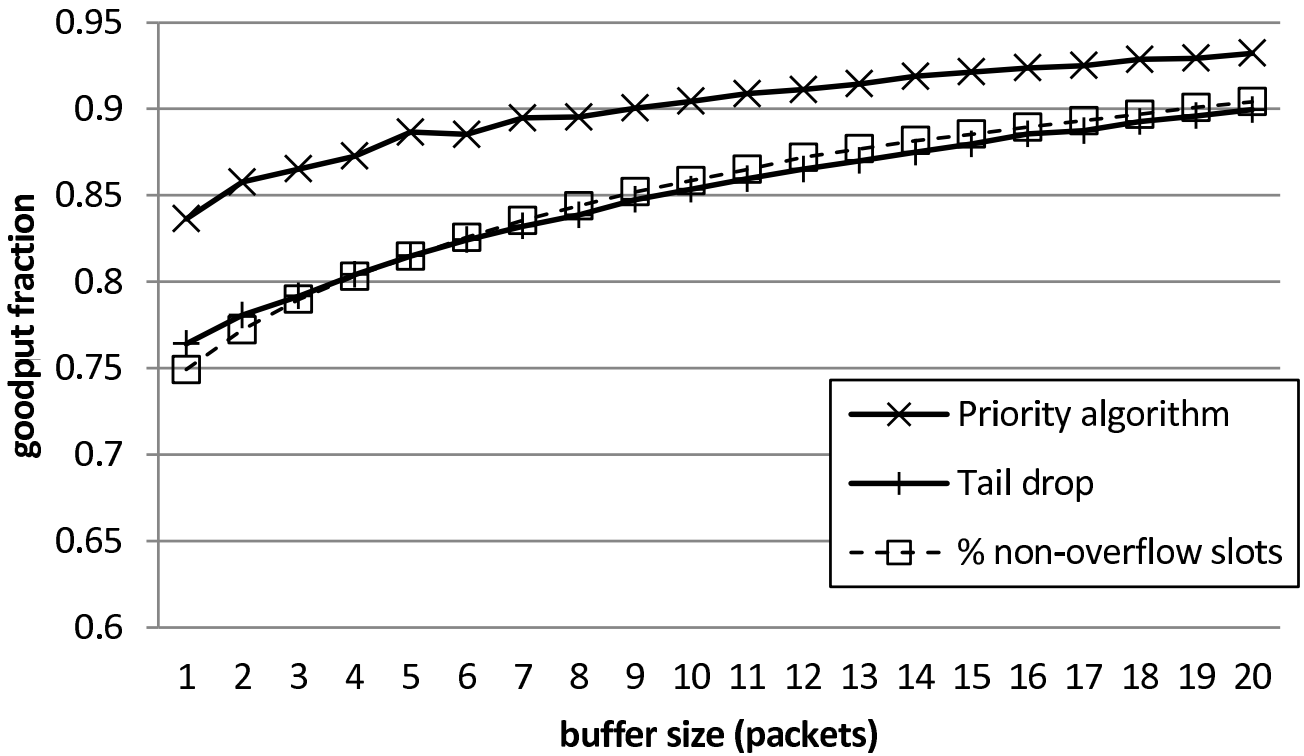


$k = 4;\ c = 6;\ b = 10$



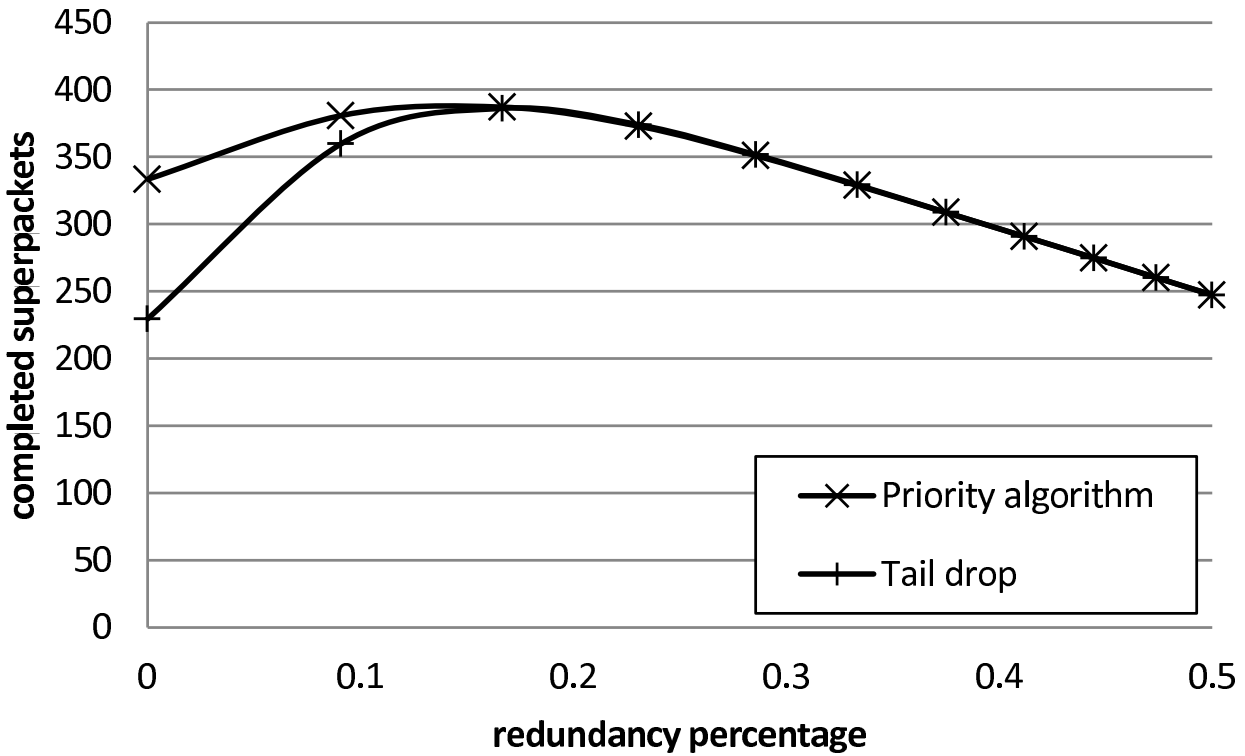
$k = 4;\ b = 10;\ \overline{\sigma} = 4.5$

Buffer Size (b)



- $k = 4$; $c = 5$; $\overline{\sigma} = 4.66$

Redundancy Percentage (β)



- $c = 5$; $b = 10$; $\overline{\sigma} = 4.94$
- $k^* = 10$

Conclusion

- Previous papers considered special cases
  (e.g., order respecting sequence, no buffer)
- This work:
  - Consider capacities and a FIFO buffer
  - Introduce superpacket redundancy
  - Analyze OSP with redundancy
  - Provide both theoretical and experimental evidence that Algorithm Priority works well when β is small

Conclusion

- Previous papers considered special cases (e.g., order respecting sequence, no buffer)
- This work:
  - Consider capacities and a FIFO buffer
  - Introduce superpacket redundancy
  - Analyze OSP with redundancy
  - Provide both theoretical and experimental evidence that Algorithm Priority works well when β is small

Open Questions:

- Analysis of Algorithm Priority with buffer
- Variable packet sizes (with or without a buffer)
- Another algorithm for large redundancy?

Thank you!