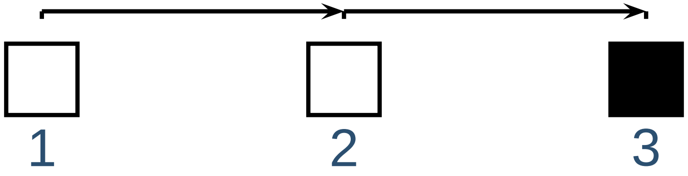
Rate vs. Buffer Size -
Greedy Information Gathering on the Line

Adi Rosén
LRI
CNRS - U. Paris 11

Gabriel Scalosub
Computer Science Department
Technion

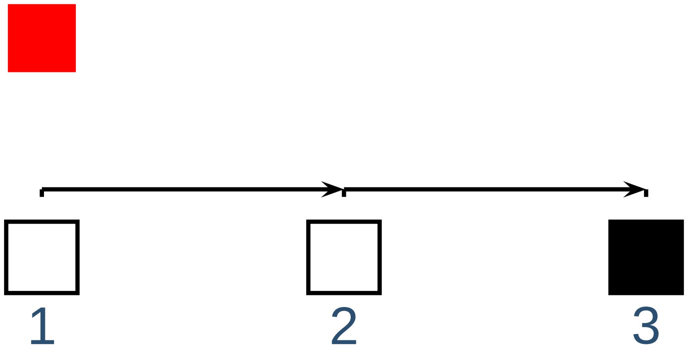
Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.



Preliminary Example

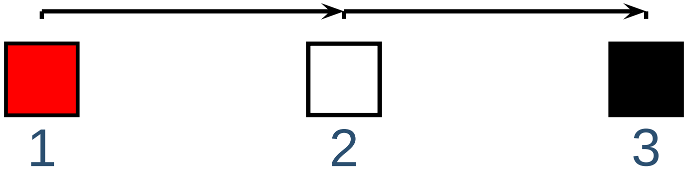
- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.



Preliminary Example

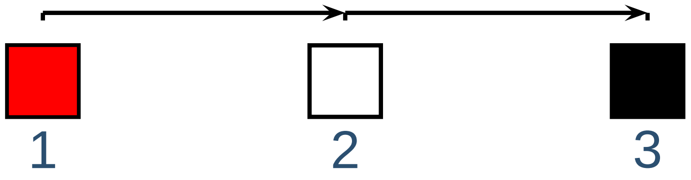
- 3 nodes, buffer size $B = 1$.
- Adversary:
    - Injects 1 packet/time-unit.
    - Chooses source.
    - Packet destination: node 3.
- Goal: maximize the *throughput*.
    - The number of packets successfully delivered.

Packet is stored in the buffer



Preliminary Example

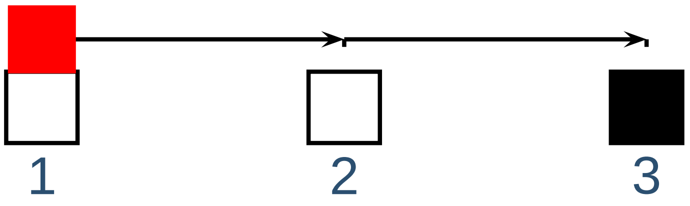
- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.


First case: the packet is forwarded

Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

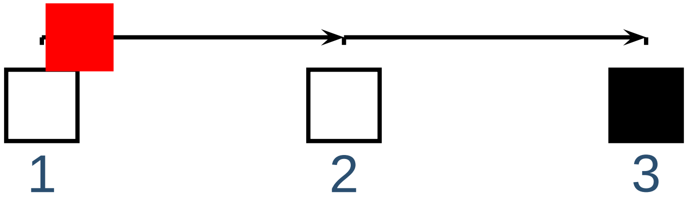
First case: the packet is forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

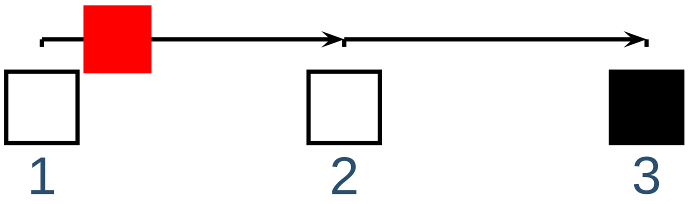
First case: the packet is forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

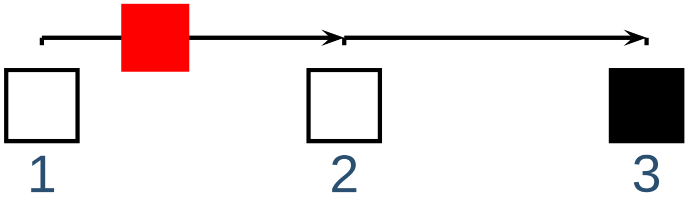
First case: the packet is forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

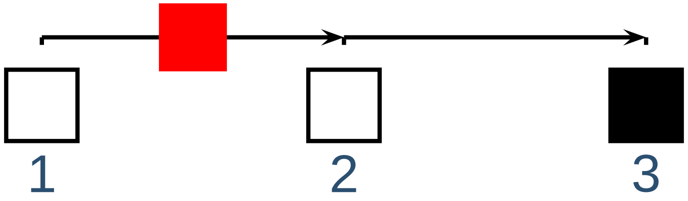
First case: the packet is forwarded



Preliminary Example

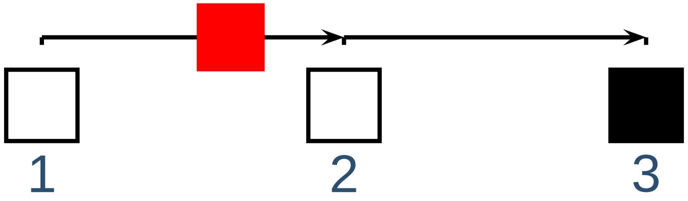
- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

First case: the packet is forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

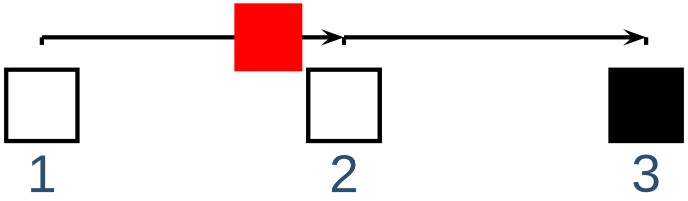


First case: the packet is forwarded

Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

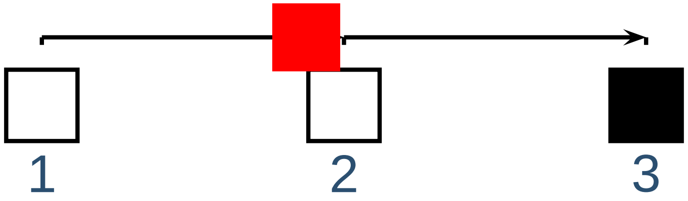
First case: the packet is forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

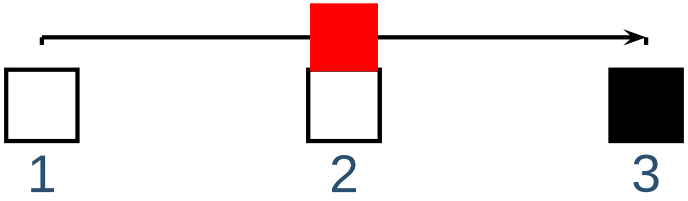
First case: the packet is forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

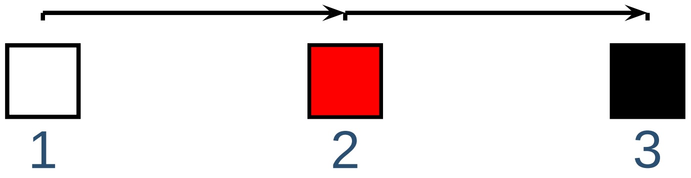
First case: the packet is forwarded



Preliminary Example

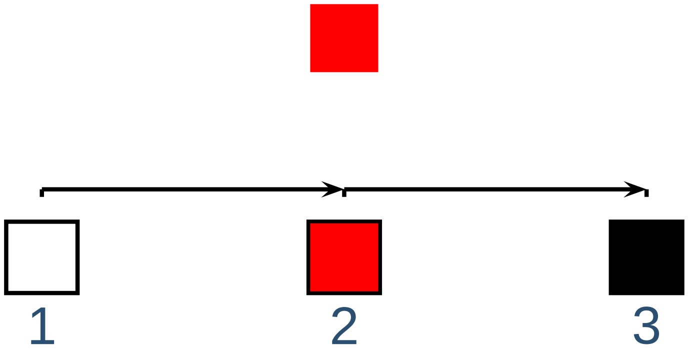
- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

First case: the packet is forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

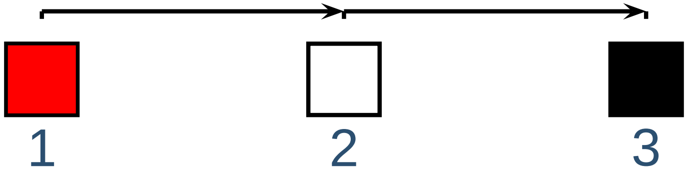


A packet is lost in node 2

Preliminary Example

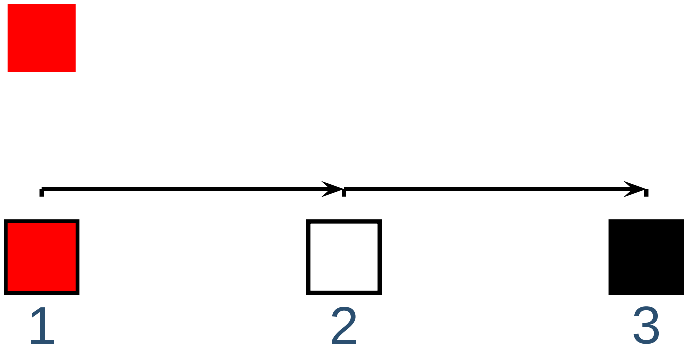
- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

Second case: the packet is not forwarded



Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.



A packet is lost in node 1

Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

$$\Rightarrow \quad \frac{\text{Adversary's throughput}}{\text{Protocol's throughput}} \geq 2.$$

Preliminary Example

- 3 nodes, buffer size $B = 1$.
- Adversary:
  - Injects 1 packet/time-unit.
  - Chooses source.
  - Packet destination: node 3.
- Goal: maximize the *throughput*.
  - The number of packets successfully delivered.

$$\Rightarrow \quad \frac{\text{Adversary's throughput}}{\text{Protocol's throughput}} \geq 2.$$

- What happens if $B = 2$?
  - Greedy is optimal.

Motivation

- Main concern:

  Provide high throughput guarantees in adversarial settings.

- Why adversarial traffic?
  - Globally applicable.
  - Good traffic characterization: hard to find.
- Adversary does not control:
  - Protocol, and
  - Buffer provisioning.
- Online local-control protocols.
- Fundamental networking problems, e.g., in Sensor Networks.

Model

- Digraph $G = (V, E)$.
- Buffer of size B at the tail of every edge.
- Packet: source, destination, and path.
- Discrete time units.
- In every time unit:
    - At most one packet traverses an edge.
    - A packet either arrives to destination, or is stored in a buffer.
- If the buffer is full - packets must be dropped.
- Online local control algorithm.
- Measure: Competitive Ratio
    - A protocol has competitive ratio c if for any input sequence σ,

$$\mathrm{Protocol}(\sigma) \geq \frac{1}{c}\,\mathrm{OPT}(\sigma).$$

Model

Competitive Network Throughput (CNT) Model [Aiello *et al.* ’03]

- Digraph $G = (V, E)$.
- Buffer of size B at the tail of every edge.
- Packet: source, destination, and path.
- Discrete time units.
- In every time unit:
  - At most one packet traverses an edge.
  - A packet either arrives to destination, or is stored in a buffer.
- If the buffer is full - packets must be dropped.
- Online local control algorithm.
- Measure: Competitive Ratio
  - A protocol has competitive ratio c if for any input sequence σ,

$$\mathrm{Protocol}(\sigma) \geq \frac{1}{c}\,\mathrm{OPT}(\sigma).$$

Model (cont.)

Extending the model:

- r-adversary (for $r > 0$):
  For every edge e, at most $\lceil rt \rceil$ packets that use e injected in t time units.

- Goal: Analyze the performance guarantees in terms of:
  - Network size - done in CNT,
  - Buffer size,
  - Adversary's rate.

Model (cont.)

Extending the model:

- r-adversary (for $r > 0$):
  For every edge e, at most $\lceil rt \rceil$ packets that use e injected in t time units.

- Goal: Analyze the performance guarantees in terms of:
  - Network size - done in CNT,
  - Buffer size,
  - Adversary's rate.

$\left.\begin{array}{l} \\ \\ \end{array}\right\}$ NEW

Greedy Information Gathering on the Line

As a first step, we consider:

- Line topology:

- Information gathering:
  - All packets destined to node n.
- The Greedy algorithm:
  - Accept/send if you can.

Previous Work

- Sensor Networks & Wireless Ad-hoc Networks - Information gathering (on the line).

[Florens *et al.* ’04, Kothapalli *et al.* ’03, Kothapalli *et al.* ’05]

- Competitive Network Throughput (CNT) model.
- General Topologies:
    - NTG has bounded competitiveness.
    - FTG has unbounded competitiveness.
- Line topology, $B = 1$:
    - Greedy is $\Theta(n)$-competitive.
- Line topology, $B \geq 2$:
    - NTG is $O(n^{2/3})$-competitive.
    - Greedy is $\Omega(\sqrt{n})$-competitive (even for information gathering).

[Aiello *et al.* ’03]

-
    - Centralized polylog competitive alg’s.

[Angelov *et al.* ’05, Azar and Zachut ’05]

- Information gathering on the line, $B \geq 2$:
    - Greedy is $O(\sqrt{n})$-competitive.

[Angelov *et al.* ’05]

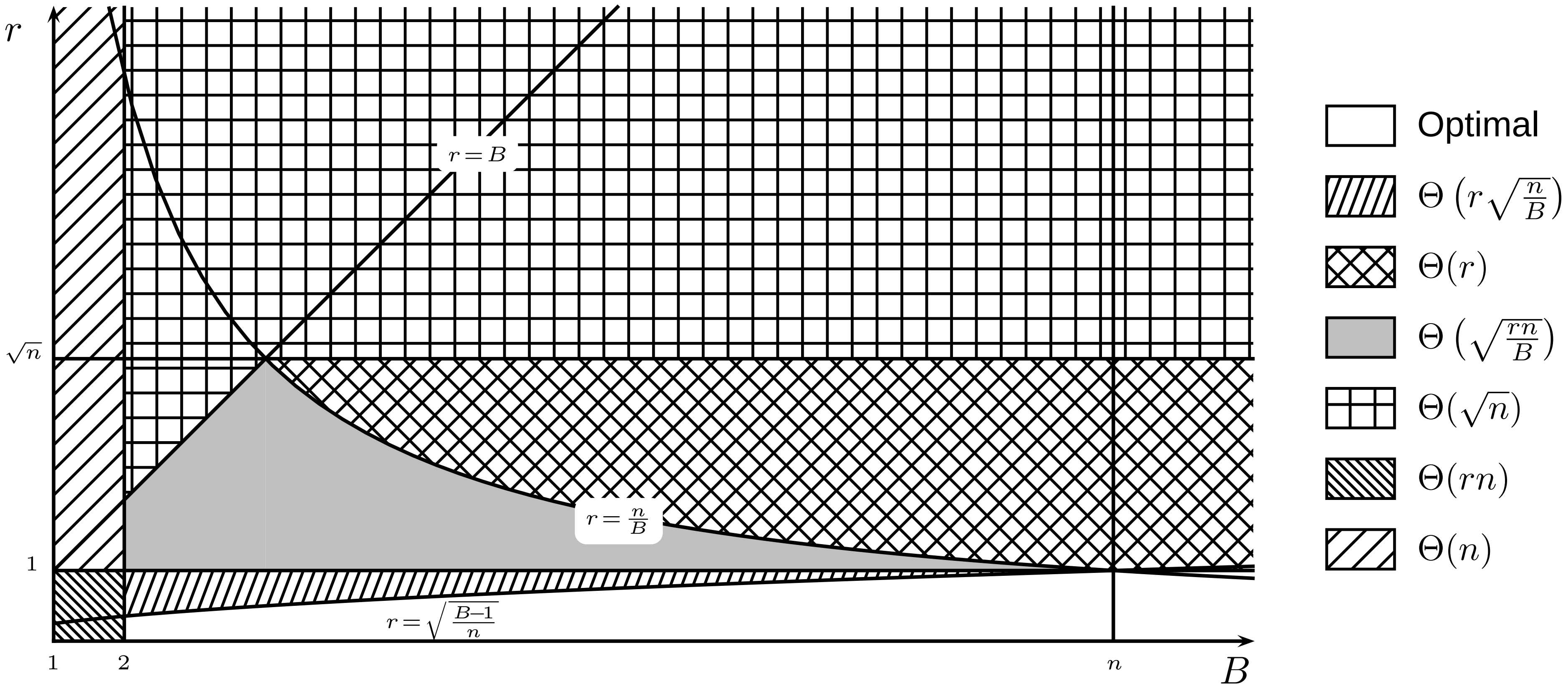
Outline of Results

- $B = 1$, $r \leq 1$: $\Theta(rn)$-competitive.

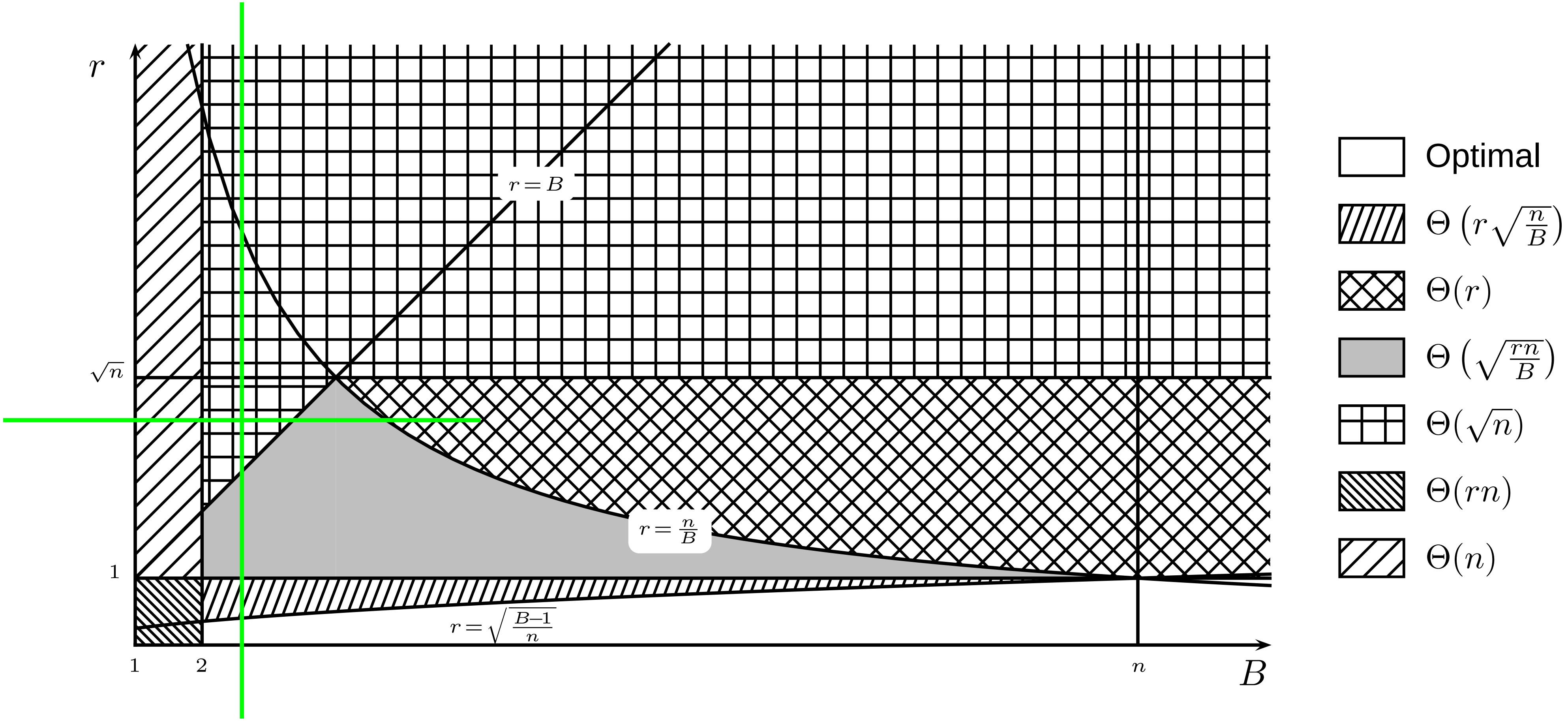
- $B \geq 2$:

Range of r	Subrange of r	Result
$r \leq 1$	$r < \sqrt{\frac{B-1}{n}}$	Optimal
	$r \geq \sqrt{\frac{B-1}{n}}$	$\Theta\left(\max\left\{1, r\sqrt{\frac{n}{B}}\right\}\right)$
$1 < r < \min\{B, \sqrt{n}\}$	$r \leq \frac{n}{B}$	$\Theta\left(\sqrt{\frac{rn}{B}}\right)$
	$\frac{n}{B} < r < \min\{B, \sqrt{n}\}$	$\Theta(r)$
$r \geq \min\{B, \sqrt{n}\}$		$\Theta(\sqrt{n})$

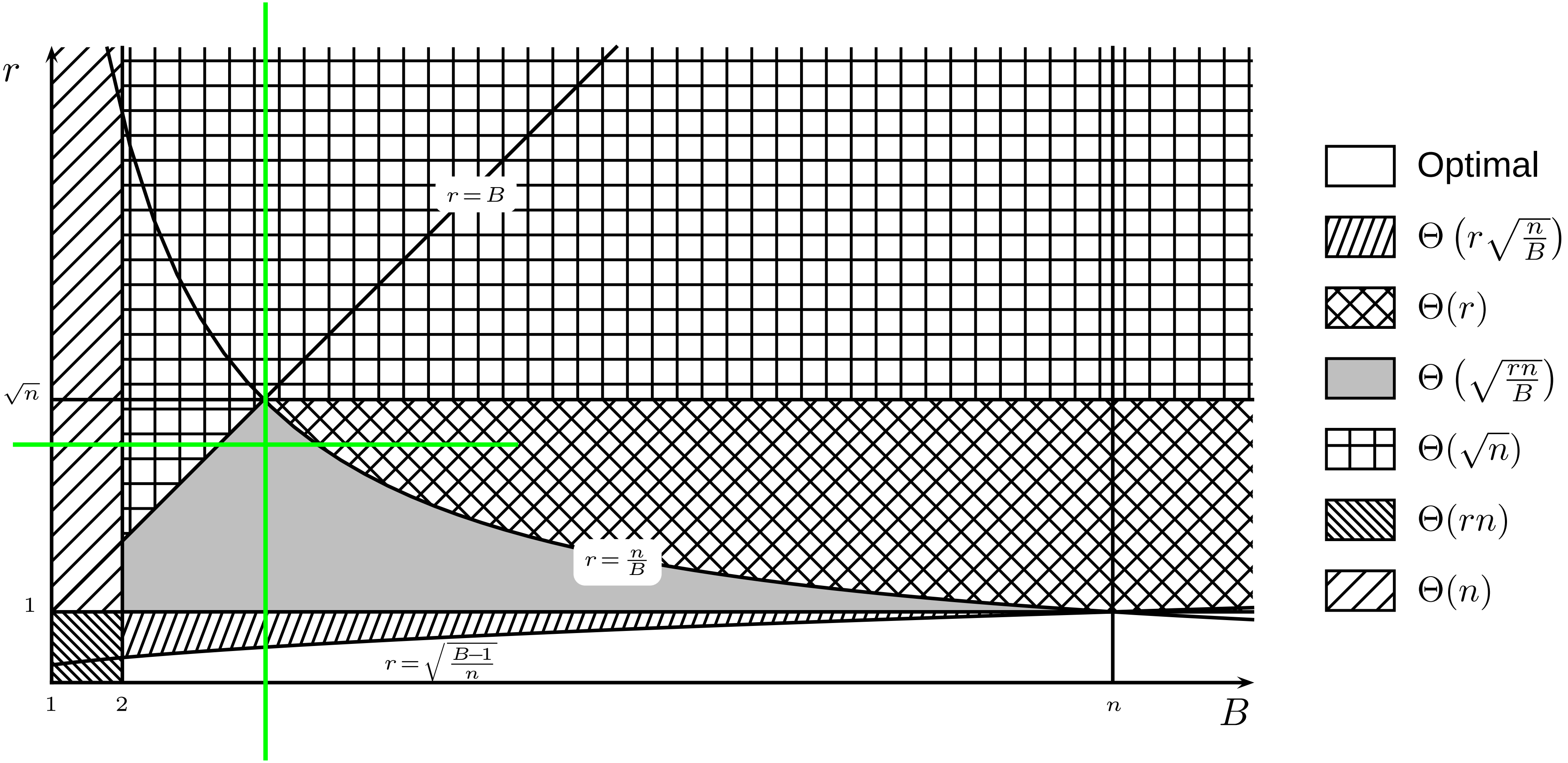
Outline of Results (Graphically...)



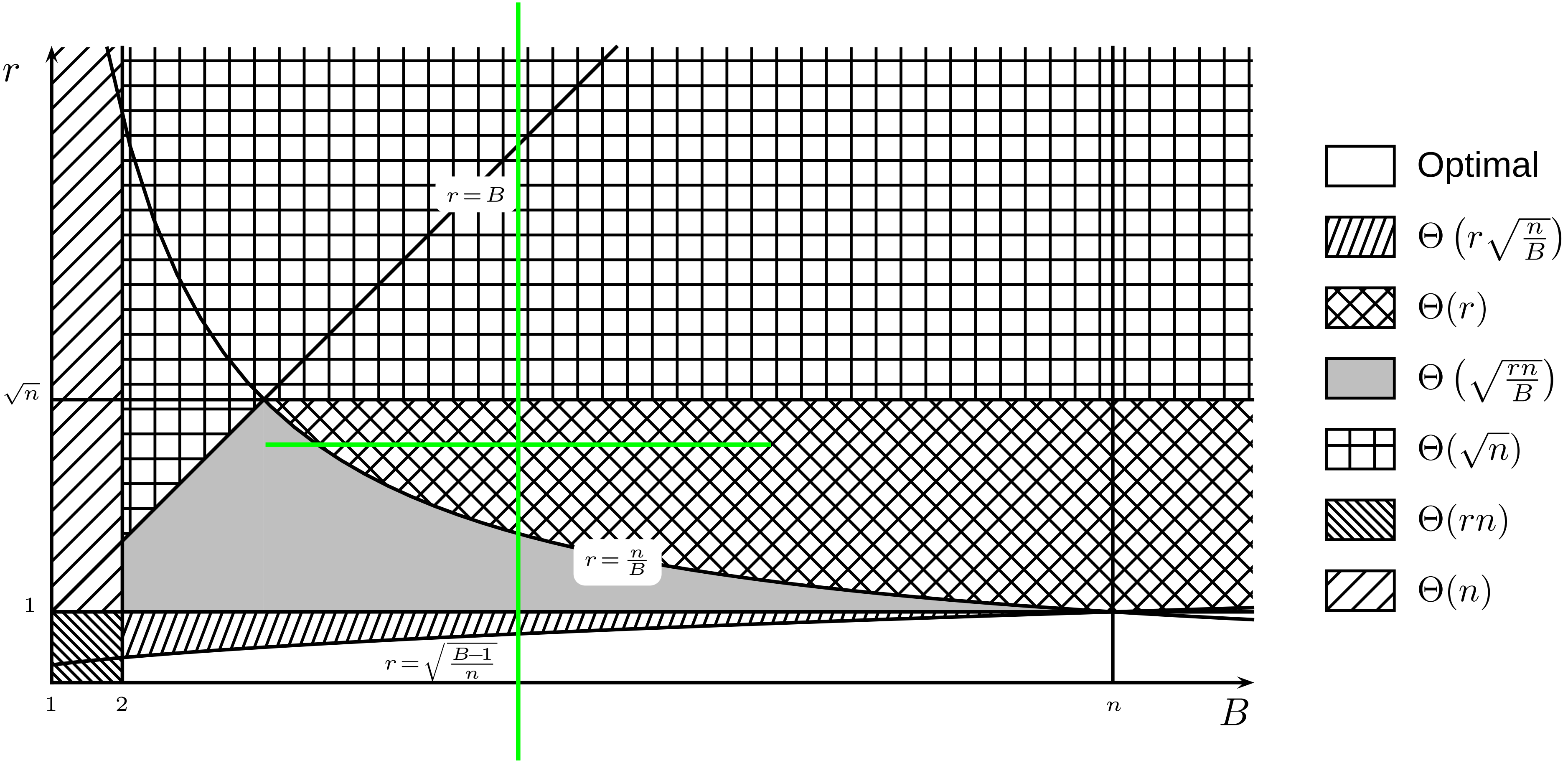
Outline of Results (Graphically...)



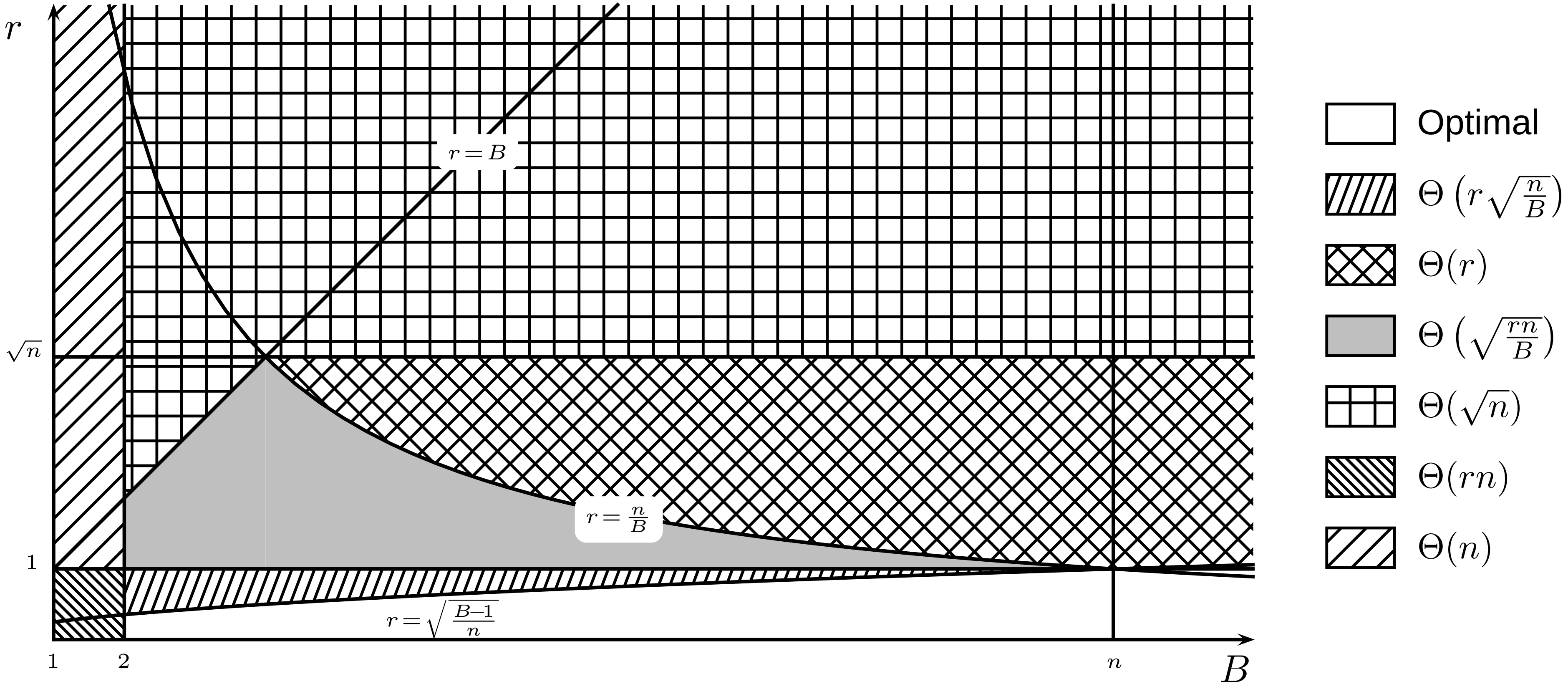
Outline of Results (Graphically...)



Outline of Results (Graphically...)



Outline of Results (Graphically...)

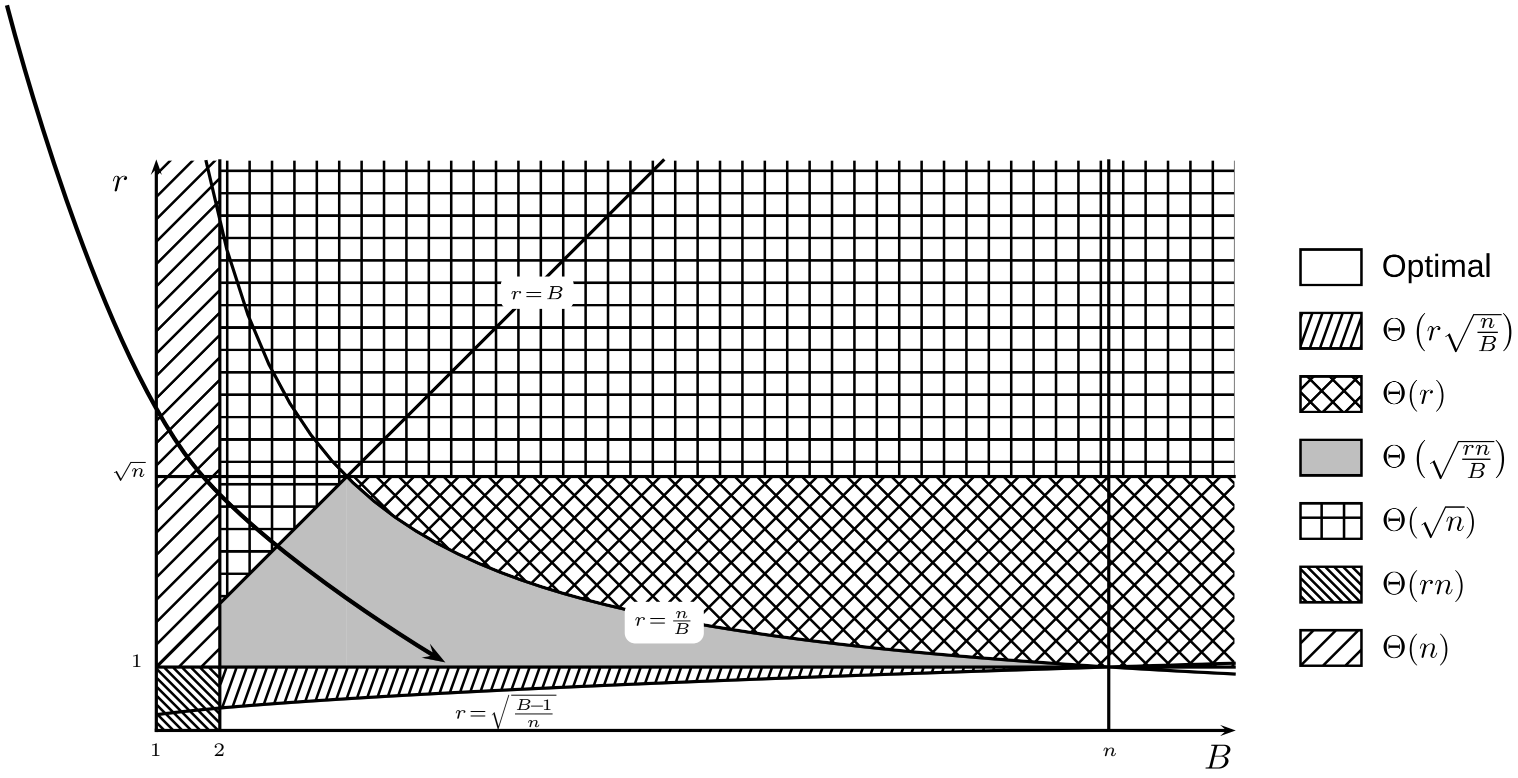


An Upper Bound

Claim. *If* $B \leq n$*, and the adversary injects at most one packet in every time step, then the greedy policy is* $O\left(\sqrt{\frac{n}{B}}\right)$*-competitive.*

An Upper Bound

Claim. *If $B \leq n$, and the adversary injects at most one packet in every time step, then the greedy policy is $O\left(\sqrt{\frac{n}{B}}\right)$-competitive.*



An Upper Bound

Claim. *If $B \leq n$, and the adversary injects at most one packet in every time step, then the greedy policy is $O\left(\sqrt{\frac{n}{B}}\right)$-competitive.*

Proof Idea:

- Divide time into epochs, each with two phases:
    - Phase 1: until the adversary accepts n packets.
    - Phase 2: another $O(n)$ time units.

An Upper Bound

Claim. *If $B \leq n$, and the adversary injects at most one packet in every time step, then the greedy policy is $O\left(\sqrt{\frac{n}{B}}\right)$-competitive.*

Proof Idea:

- Divide time into epochs, each with two phases:
  - Phase 1: until the adversary accepts n packets.
  - Phase 2: another $O(n)$ time units.

- Assign weights to packets accepted by Greedy.
- Overflow $\Longrightarrow$ increase weight of every packet in the node's buffer by 1. Wait B time units before next increase.
- The weight 'pays' for packets in $\mathrm{OPT} \setminus \mathrm{Greedy}$.

An Upper Bound (cont.)

- The best of two worlds:
  - Maximum weight is small $\Longrightarrow$ Few packets were dropped.
  - Maximum weight is large $\Longrightarrow$ Many packets in the system.
- Greedy had stored in phase 1 at least $\Omega(\sqrt{nB})$.
- Phase 2 lasts $O(n)$ time units.
  - Greedy delivers $\Omega(\sqrt{nB})$ packets.
  - Adversary can absorb another $O(n)$ packets in phase 2.
- Competitive Ratio:

$$\frac{|\mathrm{OPT}|}{|\mathrm{Greedy}|} = \frac{O(n)}{\Omega(\sqrt{nB})} = O\left(\sqrt{\frac{n}{B}}\right).$$

A Lower Bound

Claim. *For any $B \leq \frac{n}{16}$, the greedy policy has competitive ratio $\Omega\left(\sqrt{\frac{n}{B}}\right)$, even for adversaries which inject at most one packet in every time step.*

A Lower Bound

Claim. *For any* $B \leq \frac{n}{16}$*, the greedy policy has competitive ratio* $\Omega\left(\sqrt{\frac{n}{B}}\right)$*, even for adversaries which inject at most one packet in every time step.*

Proof Idea:

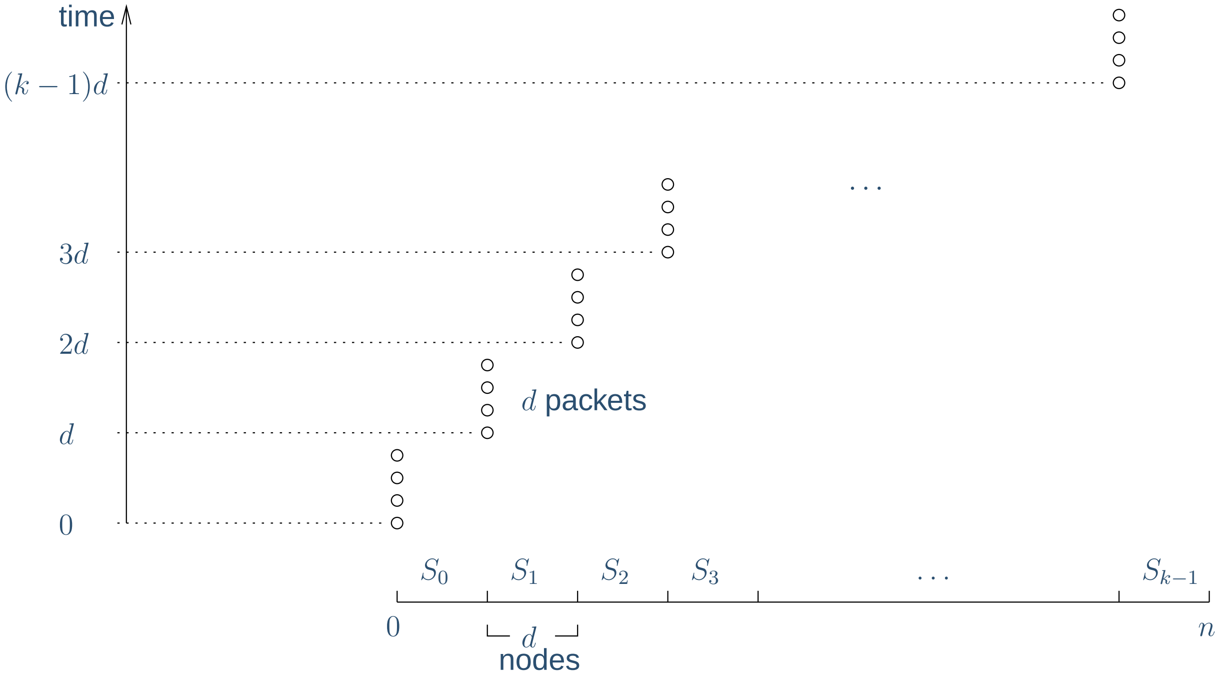
- Divide the line into k segments $S_0, \ldots, S_{k-1}$, each of length d.

A Lower Bound

Claim. *For any* $B \le \frac{n}{16}$*, the greedy policy has competitive ratio* $\Omega\left(\sqrt{\frac{n}{B}}\right)$*, even for adversaries which inject at most one packet in every time step.*

Proof Idea:

- Divide the line into k segments $S_0, \ldots, S_{k-1}$, each of length d.



A Lower Bound

Claim. *For any* $B \leq \frac{n}{16}$*, the greedy policy has competitive ratio* $\Omega\left(\sqrt{\frac{n}{B}}\right)$*, even for adversaries which inject at most one packet in every time step.*

Proof Idea:

- Divide the line into k segments $S_0, \ldots, S_{k-1}$, each of length d.
- Greedy: Forwards packets across segments
  $\Longrightarrow$ delivers $O(d + k \cdot B) = O(d + \frac{n}{d} \cdot B)$ packets.
- Adversary: Does not forward packets across segments until injection is done. Then delivers all packets injected.
  $\Longrightarrow$ delivers n packets.
- For $d = \sqrt{nB}$, the result follows:

$$\frac{n}{O(d + \frac{n}{d} \cdot B)} = \Omega\left(\sqrt{\frac{n}{B}}\right)$$

Summary

- An extension of the CNT model: Adversary's rate.
- Results in terms of network size, and also
  - Adversary's rate, and
  - Buffer size.
- Prior knowledge of adversary's characteristics:
  - Sometimes enables good buffer provisioning.
- Specifically:
  - Low-rate adversaries ($r \leq 1$): buffer size makes *all* the difference.
  - Medium-rate adversaries: buffer size makes some difference.
  - High-rate: competitive ratio independent of buffer size.
- Greedy information gathering on the line:
  - tight results (up to a constant factor).

Future Work

- Other online local control protocols.
  - Not greedy...
- Other topologies. E.g.,
  - General topologies,
  - Specific topologies: line with arbitrary destinations, rings, trees, DAGs, . . .
- Buffer-size aware protocols (?)
  - Non-uniform buffer sizes.

Thank You!