Network QoS
371-2-0213

Lecture 7

Gabriel Scalosub

Outline

1. Competitive Bounded-Delay Buffer Management & Scheduling
   - Basic Model
   - Upper Bounds
   - Lower Bounds
2. Aftermath and Other Models

Model

- A scheduling problem
  - non-FIFO
  - doesn't focus on buffer management / overflows
- Delay-minded model
  - managing multiple AF PHBs / EF classes
    - packets have weights
  - delay considerations affect schedule
    - packets have deadlines

Model

- A scheduling problem
  - non-FIFO
  - doesn't focus on buffer management / overflows
- Delay-minded model
  - managing multiple AF PHBs / EF classes
    - packets have weights
  - delay considerations affect schedule
    - packets have deadlines
- System model: slotted time
- Traffic model
  - uniform size packets (WLOG, size is 1)
  - for each packet p we have
    - $a(p)$: arrival time
    - $w(p)$: packet *weight*
    - $\ell(p)$: packet *slack*
    - $s_A(p)$: forwarding/delivery time by algorithm A
  - $a(p) + \ell(p) = d(p)$: packet *deadline*

Model

- Queue model
  - unbounded buffer space
  - service rate r
  - deadline of an enqueued packet *expires*
    - if p is not forwarded by $d(p)$, p is dropped at $d(p) + 1$
  - cut-through:
    - packet p can be scheduled at $a(p)$

Model

- Queue model
  - unbounded buffer space
  - service rate r
  - deadline of an enqueued packet *expires*
    - if p is not forwarded by $d(p)$, p is dropped at $d(p) + 1$
  - cut-through:
    - packet p can be scheduled at $a(p)$

Goal

Maximize overall weight of forwarded packets

Model

- Queue model
  - unbounded buffer space
  - service rate r
  - deadline of an enqueued packet *expires*
    - if p is not forwarded by $d(p)$, p is dropped at $d(p)+1$
  - cut-through:
    - packet p can be scheduled at $a(p)$

Goal

Maximize overall weight of forwarded packets

- Same question as last time:
  - What algorithm would you pick?

Model

- Queue model
  - unbounded buffer space
  - service rate r
  - deadline of an enqueued packet *expires*
    - if p is not forwarded by $d(p)$, p is dropped at $d(p)+1$
  - cut-through:
    - packet p can be scheduled at $a(p)$

Goal

Maximize overall weight of forwarded packets

- Same question as last time:
  - What algorithm would you pick?
- Answer not so obvious
  - have to balance (possibly) conflicting goals
    - goal 1: deliver highest weight first
    - goal 2: deliver most urgent first
  - very different from managing FIFO
    - or is it?

The Static Priority Algorithm

- Assumptions and notation:
  - $\mathcal{B}_A(t)$: set of packets still available for algorithm A at time t
    - p not yet sent by A
    - $d(p) \leq t$
  - $S_A(t)$: set of packets sent by algorithm A at time t
  - $w(P)$: overall weight of set of packets P
    - i.e., $w(P) = \sum_{p \in P} w(p)$
  - for simplicity, assume $r = 1$

Algorithm 1 SP: at time t

1: send $\arg\max \{w(q) \mid q \in \mathcal{B}_{\mathrm{SP}}(t)\}$ ▷ ties broken arbitrarily

- Notation (given some arrival sequence):
  - SP: set of packets forwarded by SP
  - $\textsc{Opt}$: set of packets forwarded by some optimal schedule

Competitive Analysis of Static Priority - Upper Bound

Theorem

Algorithm SP *is 2-competitive*

Competitive Analysis of Static Priority - Upper Bound

Theorem

Algorithm SP *is 2-competitive*

Proof.

- Consider any $p \in \textsc{Opt} \setminus \textsc{SP}$
- Let $t = s_{\textsc{Opt}}(p)$ and $p' = S_{\textsc{SP}}(t)$
  - i.e., the packet forwarded by SP when OPT forwards p
- Definition of t + feasibility of OPT: p is available for SP at t

  $\Rightarrow$ $w(p) \leq w(p')$
- Implies a 1-1 mapping from $\textsc{Opt} \setminus \textsc{SP}$ to SP

  $\Rightarrow$ $w(\textsc{Opt} \setminus \textsc{SP}) \leq w(\textsc{SP})$
- Hence

$$w(\textsc{Opt}) \leq w(\textsc{Opt} \setminus \textsc{SP}) + w(\textsc{SP}) \leq 2w(\textsc{SP})$$

$\square$

Improved Upper Bounds

- The analysis of SP is essentially tight:
    - let $\varepsilon > 0$ and consider 2 packets arriving at time $t = 0$:
        - p: $\ell(p) = 0$, $w(p) = 1$
        - p': $\ell(p') = 1$, $w(p') = 1 + \varepsilon$
    - SP would only schedule p'
    - $\textsc{Opt}$ would schedule both packets
    - ratio:
        - $\frac{w(\textsc{Opt})}{w(\mathrm{SP})} = \frac{2+\varepsilon}{1+\varepsilon}$
        - arbitrarily close to 2

Improved Upper Bounds

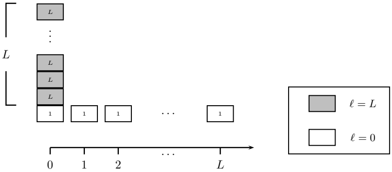
- The analysis of SP is essentially tight:
  - let $\varepsilon > 0$ and consider 2 packets arriving at time $t = 0$:
    - p: $\ell(p) = 0$, $w(p) = 1$
    - p': $\ell(p') = 1$, $w(p') = 1 + \varepsilon$
  - SP would only schedule p'
  - $\textsc{Opt}$ would schedule both packets
  - ratio:
    - $\frac{w(\textsc{Opt})}{w(\mathrm{SP})} = \frac{2+\varepsilon}{1+\varepsilon}$
    - arbitrarily close to 2
- Can one do better than SP?

Improved Upper Bounds

- The analysis of SP is essentially tight:
  - let $\varepsilon > 0$ and consider 2 packets arriving at time $t = 0$:
    - p: $\ell(p) = 0$, $w(p) = 1$
    - p': $\ell(p') = 1$, $w(p') = 1 + \varepsilon$
  - SP would only schedule p'
  - $\textsc{Opt}$ would schedule both packets
  - ratio:
    - $\frac{w(\textsc{Opt})}{w(\mathrm{SP})} = \frac{2+\varepsilon}{1+\varepsilon}$
    - arbitrarily close to 2
- Can one do better than SP?
- Candidate algorithms:
  - prefer urgent packets
  - Earliest-Deadline-First (EDF)

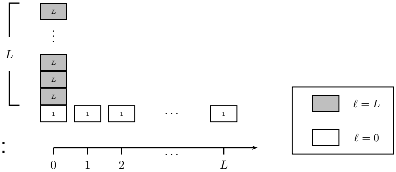
Improved Upper Bounds

- The analysis of SP is essentially tight:
  - let $\varepsilon > 0$ and consider 2 packets arriving at time $t = 0$:
    - p: $\ell(p) = 0$, $w(p) = 1$
    - p': $\ell(p') = 1$, $w(p') = 1 + \varepsilon$
  - SP would only schedule p'
  - OPT would schedule both packets
  - ratio:
    - $\frac{w(\mathrm{OPT})}{w(\mathrm{SP})} = \frac{2+\varepsilon}{1+\varepsilon}$
    - arbitrarily close to 2
- Can one do better than SP?
- Candidate algorithms:
  - prefer urgent packets
  - Earliest-Deadline-First (EDF)



Improved Upper Bounds

- The analysis of SP is essentially tight:
  - let $\varepsilon > 0$ and consider 2 packets arriving at time $t = 0$:
    - p: $\ell(p) = 0$, $w(p) = 1$
    - p': $\ell(p') = 1$, $w(p') = 1 + \varepsilon$
  - SP would only schedule p'
  - $\textsc{Opt}$ would schedule both packets
  - ratio:
    - $\frac{w(\textsc{Opt})}{w(\mathrm{SP})} = \frac{2+\varepsilon}{1+\varepsilon}$
    - arbitrarily close to 2
- Can one do better than SP?
- Candidate algorithms:
  - prefer urgent packets
  - Earliest-Deadline-First (EDF)
    - cannot be completely "local": $2L$ vs. L^2
    - make best plan possible



Restricting Traffic and Observations

- Restricted adversaries/models
  - s-bounded model
    - for every packet p, $\ell(p) \in \{0, \ldots, s-1\}$
  - s-uniform model
    - for every packet p, $\ell(p) = s - 1$
  - agreeable deadlines
    - if $a(p) \leq a(p')$ then $d(p) \leq d(p')$
- In what follows, we focus on the 2-bounded model

Restricting Traffic and Observations

- Restricted adversaries/models
  - s-bounded model
    - for every packet p, $\ell(p) \in \{0, \ldots, s-1\}$
  - s-uniform model
    - for every packet p, $\ell(p) = s - 1$
  - agreeable deadlines
    - if $a(p) \leq a(p')$ then $d(p) \leq d(p')$
- In what follows, we focus on the 2-bounded model
- Observations (P: set of packets)
  - the offline problem: solvable in polynomial time (matching)

Restricting Traffic and Observations

- Restricted adversaries/models
  - s-bounded model
    - for every packet p, $\ell(p) \in \{0, \ldots, s-1\}$
  - s-uniform model
    - for every packet p, $\ell(p) = s-1$
  - agreeable deadlines
    - if $a(p) \leq a(p')$ then $d(p) \leq d(p')$
- In what follows, we focus on the 2-bounded model
- Observations (P: set of packets)
  - the offline problem: solvable in polynomial time (matching)
  - P is scheduleable iff it is EDF-schedulable
    - there exists an EDF-schedule of P

Restricting Traffic and Observations

- Restricted adversaries/models
  - s-bounded model
    - for every packet p, $\ell(p) \in \{0, \dots, s-1\}$
  - s-uniform model
    - for every packet p, $\ell(p) = s-1$
  - agreeable deadlines
    - if $a(p) \leq a(p')$ then $d(p) \leq d(p')$
- In what follows, we focus on the 2-bounded model
- Observations (P: set of packets)
  - the offline problem: solvable in polynomial time (matching)
  - P is scheduleable iff it is EDF-schedulable
    - there exists an EDF-schedule of P
  - P is EDF-scheduleable $\Rightarrow$ P is EDF-SP-schedulable
    - for identical deadlines, use highest-weight-first

Restricting Traffic and Observations

- Restricted adversaries/models
  - s-bounded model
    - for every packet p, $\ell(p) \in \{0, \ldots, s-1\}$
  - s-uniform model
    - for every packet p, $\ell(p) = s-1$
  - agreeable deadlines
    - if $a(p) \leq a(p')$ then $d(p) \leq d(p')$
- In what follows, we focus on the 2-bounded model
- Observations (P: set of packets)
  - the offline problem: solvable in polynomial time (matching)
  - P is scheduleable iff it is EDF-schedulable
    - there exists an EDF-schedule of P
  - P is EDF-scheduleable $\Rightarrow$ P is EDF-SP-schedulable
    - for identical deadlines, use highest-weight-first
  - WLOG, $\textsc{Opt}$ is an EDF-SP-schedule
    - ties broken as we want

The β-EDF Algorithm

- Notation:
  - $E_A(t)$: set of *eligible* packets for algorithm A at t
    - non-expired, not yet sent
  - $\text{OFF}(X)$: optimal offline schedule given X
    - X: set of packets
- Let $\beta \in [0, 1]$

Algorithm 2 β-EDF: at time t

1: $F \leftarrow \text{OFF}(E_{\beta\text{-EDF}}(t))$ ▷ optimal offline at t
2: $p \leftarrow S_F(t)$ ▷ packet sent at t by OFF
3: $p' \leftarrow S_F(t+1)$ ▷ packet sent at $t+1$ by OFF
4: **if** $w(p) < \beta \cdot w(p')$ **then**
5:     forward p' ▷ *p' pushes out p*
6: **else**
7:     forward p
8: **end if**

Competitive Analysis of β-EDF - Upper Bound

- Observations:
  - 0-EDF: follow $\textsc{Off}$ locally
  - 1-EDF: SP

Competitive Analysis of β-EDF - Upper Bound

- Observations:
  - 0-EDF: follow $\textsc{Off}$ locally
  - 1-EDF: SP

Theorem

Algorithm $\frac{1}{\phi}$-EDF *has competitive ratio* $\phi \approx 1.618$ *(Golden Ratio)*

Competitive Analysis of β-EDF - Upper Bound

- Observations:
  - 0-EDF: follow OFF locally
  - 1-EDF: SP

Theorem

Algorithm $\frac{1}{\phi}$-EDF *has competitive ratio* $\phi \approx 1.618$ *(Golden Ratio)*

Define a mapping $m : S_{\mathrm{OPT}} \to S_{\beta\text{-EDF}}$ (for $\beta = \frac{1}{\phi}$)

Let $p = S_{\mathrm{OPT}}(t)$

- If $p = S_{\beta\text{-EDF}}(t-1)$ or $p = S_{\beta\text{-EDF}}(t)$
  - $m(p) = p$
- Else $\quad \triangleright p = S_{\beta\text{-EDF}}(t+1)$ or $p \in S_{\mathrm{OPT}} \setminus S_{\beta\text{-EDF}}$
  - $m(p) = S_{\beta\text{-EDF}}(t)$

Competitive Analysis of β-EDF - Upper Bound

- Observations:
  - 0-EDF: follow $\textsc{Off}$ locally
  - 1-EDF: SP

Theorem

Algorithm $\frac{1}{\phi}\text{-EDF}$ has competitive ratio $\phi \approx 1.618$ (Golden Ratio)

Define a mapping $m : S_{\text{OPT}} \to S_{\beta\text{-EDF}}$ (for $\beta = \frac{1}{\phi}$)

Let $p = S_{\text{OPT}}(t)$

- If $p = S_{\beta\text{-EDF}}(t-1)$ or $p = S_{\beta\text{-EDF}}(t)$ $\boxed{\text{Type 1}}$
  - $m(p) = p$
- Else $\triangleright p = S_{\beta\text{-EDF}}(t+1)$ or $p \in S_{\text{OPT}} \setminus S_{\beta\text{-EDF}}$
  - $m(p) = S_{\beta\text{-EDF}}(t)$ $\boxed{\text{Type 2}}$

Competitive Analysis of $\beta\text{-}\mathrm{EDF}$ - Upper Bound

- The mapping is well defined
  - In type 2, $S_{\beta\text{-}\mathrm{EDF}}(t)$ exists
    - p is eligible for $\beta\text{-}\mathrm{EDF}$ at t but $p \neq S_{\beta\text{-}\mathrm{EDF}}(t)$
    - $\exists p' \neq p$ s.t. $p' = S_{\beta\text{-}\mathrm{EDF}}(t)$
- The mapping is a bijection over $\{p \in S_{\mathrm{OPT}} \mid m(p) \neq p\}$

Competitive Analysis of β-EDF - Upper Bound

- The mapping is well defined
  - In type 2, $S_{\beta\text{-EDF}}(t)$ exists
    - p is eligible for β-EDF at t but $p \neq S_{\beta\text{-EDF}}(t)$
    - $\exists p' \neq p$ s.t. $p' = S_{\beta\text{-EDF}}(t)$
- The mapping is a bijection over $\{p \in S_{\mathrm{OPT}} \mid m(p) \neq p\}$

Lemma

if $p = S_{\mathrm{OPT}}(t)$ is mapped to $p' \neq p$, then $w(p) \leq \frac{1}{\phi} \cdot w(p')$

Competitive Analysis of β-EDF - Upper Bound

- The mapping is well defined
  - In type 2, $S_{\beta\text{-EDF}}(t)$ exists
    - p is eligible for β-EDF at t but $p \neq S_{\beta\text{-EDF}}(t)$
    - $\exists p' \neq p$ s.t. $p' = S_{\beta\text{-EDF}}(t)$
- The mapping is a bijection over $\{p \in S_{\mathrm{OPT}} \mid m(p) \neq p\}$

Lemma

if $p = S_{\mathrm{OPT}}(t)$ is mapped to $p' \neq p$, then $w(p) \leq \frac{1}{\phi} \cdot w(p')$

Proof of Theorem.

$$
\begin{aligned}
w(\mathrm{OPT}) &\leq w(S_{\mathrm{OPT}} \setminus S_{\frac{1}{\phi}\text{-EDF}}) + w(S_{\frac{1}{\phi}\text{-EDF}}) \\
&\leq w(\{p \in S_{\mathrm{OPT}} \mid m(p) \neq p\}) + w(S_{\frac{1}{\phi}\text{-EDF}}) \\
\boxed{\text{by lemma}} \quad &\leq \frac{1}{\phi} \cdot w(S_{\frac{1}{\phi}\text{-EDF}}) + w(S_{\frac{1}{\phi}\text{-EDF}}) \\
&= (1 + \frac{1}{\phi}) w(S_{\frac{1}{\phi}\text{-EDF}}) \\
&= \phi \cdot w(S_{\frac{1}{\phi}\text{-EDF}})
\end{aligned}
$$

$\square$

Competitive Analysis - Deterministic Lower Bounds

- Lower bounds use 2-bounded adversaries
  - actually an *infinite series* of adversaries

Competitive Analysis - Deterministic Lower Bounds

- Lower bounds use 2-bounded adversaries
  - actually an *infinite series* of adversaries
- Each adversary proves increasingly stronger lower bounds

Lemma (weak lower bound)

No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Competitive Analysis - Deterministic Lower Bounds

- Lower bounds use 2-bounded adversaries
  - actually an *infinite series* of adversaries
- Each adversary proves increasingly stronger lower bounds

Lemma (weak lower bound)

No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Lemma (stronger lower bound)

No det. algorithm has competitive ratio better than $\frac{3}{2} = 1.5$

Competitive Analysis - Deterministic Lower Bounds

- Lower bounds use 2-bounded adversaries
  - actually an *infinite series* of adversaries
- Each adversary proves increasingly stronger lower bounds

Lemma (weak lower bound)

No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Lemma (stronger lower bound)

No det. algorithm has competitive ratio better than $\frac{3}{2} = 1.5$

- How far can we go?

Lower Bound Proofs – $\sqrt{2}$

Lemma (weak lower bound)

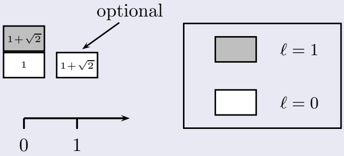
No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Lower Bound Proofs – $\sqrt{2}$

Lemma (weak lower bound)

No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Proof.



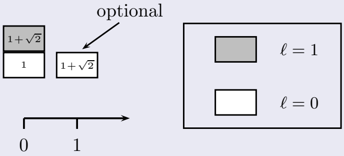
Lower Bound Proofs – $\sqrt{2}$

Lemma (weak lower bound)

No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Proof.

- Consider time $t = 0$



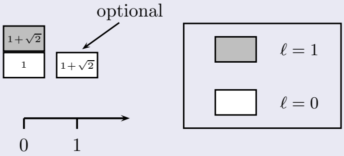
Lower Bound Proofs – $\sqrt{2}$

Lemma (weak lower bound)

No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Proof.

- Consider time $t = 0$
- If $\textsc{Alg}$ forwards high-weight packet
  - sequence ends
  - $w(\textsc{Alg}) = 1 + \sqrt{2}$
  - $w(\textsc{Opt}) = 2 + \sqrt{2}$
  - ratio:
    - $\frac{w(\textsc{Opt})}{w(\textsc{Alg})} = \frac{2+\sqrt{2}}{1+\sqrt{2}} = \sqrt{2}$



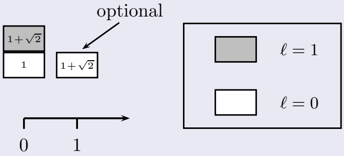
Lower Bound Proofs – $\sqrt{2}$

Lemma (weak lower bound)

No det. algorithm has competitive ratio better than $\sqrt{2} \approx 1.414$

Proof.

- Consider time $t = 0$
- If $\textsc{Alg}$ forwards low-weight packet
  - high-weight packet arrives at $t = 1$
  - at $t = 1$ at least one high-weight packet expires
  - $w(\textsc{Alg}) = 2 + \sqrt{2}$
  - $w(\textsc{Opt}) = 2(1 + \sqrt{2})$
  - ratio:
    - $\frac{w(\textsc{Opt})}{w(\textsc{Alg})} = \frac{2(1+\sqrt{2})}{2+\sqrt{2}} = \sqrt{2}$



$\square$

Lower Bound Proofs – 1.5

Lemma (stronger lower bound)

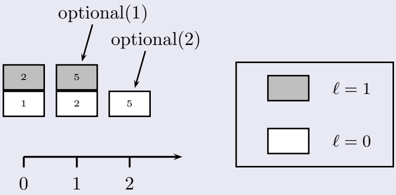
No det. algorithm has competitive ratio better than 1.5

Lower Bound Proofs – 1.5

Lemma (stronger lower bound)

No det. algorithm has competitive ratio better than 1.5

Proof.



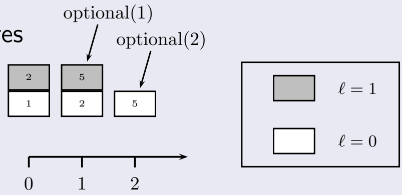
Lower Bound Proofs – 1.5

Lemma (stronger lower bound)

No det. algorithm has competitive ratio better than 1.5

Proof.

- Consider $t = 0$
- If $\textsc{Alg}$ forwards high-weight packet (2)
  - sequence ends
  - low-weight (1) packet expires
  - $w(\textsc{Alg}) = 2$
  - $w(\textsc{Opt}) = 1 + 2 = 3$
  - ratio:
    - $\frac{w(\textsc{Opt})}{w(\textsc{Alg})} = \frac{3}{2}$



- Assume $\textsc{Alg}$ forwards low-weight packet (1)

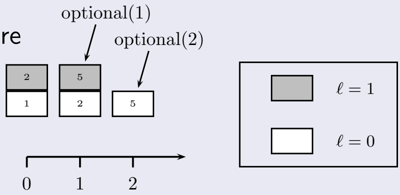
Lower Bound Proofs – 1.5

Lemma (stronger lower bound)

No det. algorithm has competitive ratio better than 1.5

Proof.

- At $t = 1$: optional(1) arrives
- If ALG forwards high-weight packet (5)
  - sequence ends
  - low-weight (2) packets expire
  - $w(\text{ALG}) = 1 + 5 = 6$
  - $w(\text{OPT}) = 2 + 2 + 5 = 9$
  - ratio:
    - $\frac{w(\text{OPT})}{w(\text{ALG})} = \frac{9}{6} = \frac{3}{2}$



- Assume ALG forwards low-weight packet (2)

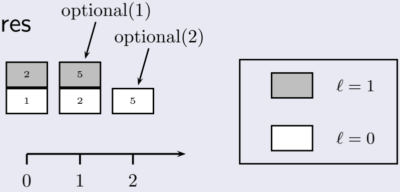
Lower Bound Proofs – 1.5

Lemma (stronger lower bound)

No det. algorithm has competitive ratio better than 1.5

Proof.

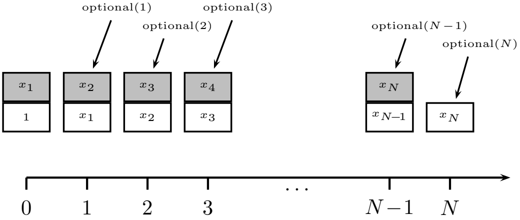
- Consider $t = 2$
  - optional(2) arrives
  - at $t = 2$ at least one high-weight (5) packet expires
  - $w(\textsc{Alg}) = 1 + 2 + 5 = 8$
  - $w(\textsc{Opt}) = 2 + 5 + 5 = 12$
  - ratio:
    - $\frac{w(\textsc{Opt})}{w(\textsc{Alg})} = \frac{12}{8} = \frac{3}{2}$



□

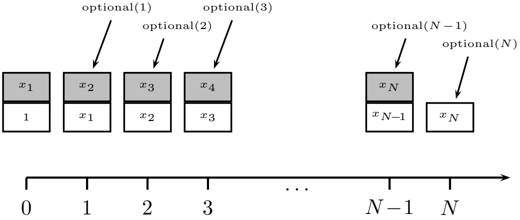
Competitive Analysis - Strong Lower Bound

- Proof generalizes the previous two lower bounds



Competitive Analysis - Strong Lower Bound

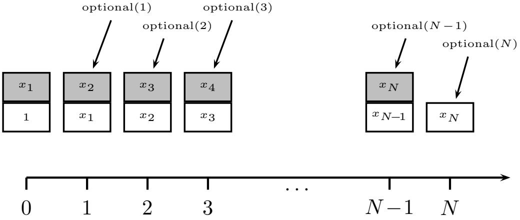
- Proof generalizes the previous two lower bounds



- Solve $N+1$ equations (maximal c_N, $x_0 = 1$)
  - $x_0 + x_1 \geq c_N \cdot x_1$
  - $x_1 + \ldots + x_{k-1} + x_k + x_k + x_{k+1} \geq c_N \cdot (x_0 + \ldots + x_{k-1} + x_{k+1})$
  - $x_1 + \ldots + x_{N-1} + x_N + x_N \geq c_N \cdot (x_0 + \ldots + x_{N-1} + x_N)$
- Lower bound obtained by having $N \to \infty$

Competitive Analysis - Strong Lower Bound

- Proof generalizes the previous two lower bounds



- Solve $N+1$ equations (maximal c_N, $x_0 = 1$)
  - $x_0 + x_1 \geq c_N \cdot x_1$
  - $x_1 + \ldots + x_{k-1} + x_k + x_k + x_{k+1} \geq c_N \cdot (x_0 + \ldots + x_{k-1} + x_{k+1})$
  - $x_1 + \ldots + x_{N-1} + x_N + x_N \geq c_N \cdot (x_0 + \ldots + x_{N-1} + x_N)$
- Lower bound obtained by having $N \to \infty$
- Some values of c_N (for the "right" values of x_i-s):

N	1	2	3	4	...	20	...	40	...
c_N	1.414	1.5	1.539	1.562	...	1.617	...	1.618	...

Competitive Analysis - Strong Lower Bound

Theorem

No det. algorithm has competitive ratio better than $\phi \approx 1.618$

Competitive Analysis - Strong Lower Bound

Theorem

No det. algorithm has competitive ratio better than $\phi \approx 1.618$

- Somewhat simpler proofs exist
  - knowing *c* in advance helps finding simpler proofs
  - "guessing" is useful!!
- Requires multiple values of packets
  - not just 1 and $\alpha > 1$

Competitive Analysis - Strong Lower Bound

Theorem

No det. algorithm has competitive ratio better than $\phi \approx 1.618$

- Somewhat simpler proofs exist
  - knowing *c* in advance helps finding simpler proofs
  - "guessing" is useful!!
- Requires multiple values of packets
  - not just 1 and $\alpha > 1$
- Corollaries (for 2-bounded model)
  - the analysis of β-EDF is tight
  - β-EDF is optimal for the 2-bounded model

Aftermath

- Above results are due to:
  - SP & lower bounds: Hajek (2001)
  - β-EDF: Kesselman, Lotker, Mansour, Patt-Shamir, Schieber, Sviridenko (2001)

Aftermath

- Above results are due to:
  - SP & lower bounds: Hajek (2001)
  - β-EDF: Kesselman, Lotker, Mansour, Patt-Shamir, Schieber, Sviridenko (2001)
- Many papers addressed the question
  - many algorithms!!
  - mostly in restricted models
  - still a very active field of research!!
- What about randomized algorithms?
  - strictly better than deterministic...

	deterministic		randomized	
restriction	upper	lower	upper	lower
General	2			
			1.582	
	1.939			
	1.854			
	1.893			
	1.828			
		1.618		1.25
				1.33
s-bounded	$2-\frac{2}{s}+o(\frac{1}{s})$			
4-bounded	1.732			
3-bounded	1.618			
2-bounded	1.618			
		1.17†		
		1.414		
		1.618		
				1.25
			1.25	
				1.33
			1.33	
agreeable		1.618		1.25
deadlines				1.33
	1.838			
	1.618			
			1.33	
s-uniform	1.618		1.33	
		1.377		
				1.25
2-uniform	1.434			
		1.11		
		1.25		
	1.414			
		1.366		
		1.414		1.17
	1.377			
		1.377		
				1.2
			1.25	
			1.33	

Related Models

- Resource augmentation
  - comparing online with capacity W to offline with capacity 1
- limited information
  - *order* of deadlines is known
  - exact deadlines are unknown
  - generalizes the bounded-delay model
- bounded-delay + bounded-buffer
  - combined scheduling / buffer management problem
- Latency-sensitive FIFO
  - FIFO model where value decreases as packet is delayed
- And more... (later in the course)