Advanced Computer Networking (ACN)

Exercise 6 – Solution

Prof. Dr.-Ing. Georg Carle

Sebastian Gallenmüller, Max Helm, Benedikt Jaeger,
Marcel Kempf, Patrick Sattler, Johannes Zirngibl

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

Outline

Announcements

Tutorial6 - Problem 1: Network Calculus

Tutorial6 – Problem 2: Content Delivery Networks

Announcements

Deadline first version:

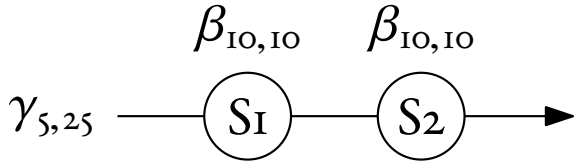
- The deadline for the first version of Tutorial 6 has passed 15 minutes ago
- Commit and push your solution, if you haven't already

Next week:

- Tuesday: Internet measurements lecture
- Thursday: Exam question session
  - Send us your questions until Tuesday EoD
  - We will answer them in the lecture on Thursday
  - Send questions **either** to `acn@net.in.tum.de` and use [FAQ Session] in the subject line **or** ask on Moodle

One flow with arrival curve $\gamma_{5,25}$ traverses two servers with service curves $\beta_{10,10}$.

Goal: Apply network calculus to give guarantees for the studied flow



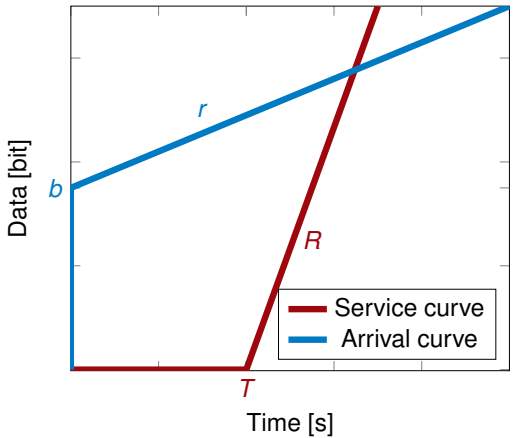
Reminder on notation

- $\beta_{R,T}$: rate latency curve with rate R and latency T
- $\gamma_{r,b}$: token bucket curve with rate r and burst b

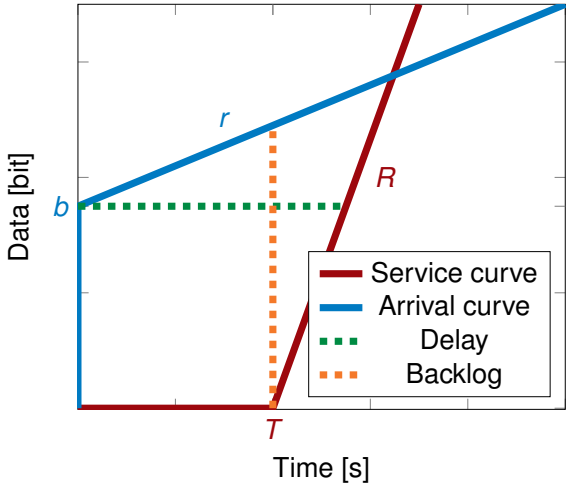
Tutorial6 - Problem 1: Network Calculus

1a) Represent the curves $\beta_{10,10}$ and $\gamma_{5,25}$ on the same figure.

- Generalized to $\beta_{R,T}$ and $\gamma_{r,b}$
- $\beta_{R,T}(t) = [R \cdot (t - T)]^+$
- $\gamma_{r,b}(t) = r \cdot t + b, \forall t \geq 0$



1b) Represent the backlog and the delay on the figure.



1c) What is the latency/delay bound of the flow after S1?

- Can be determined graphically (see previous slide)
- Or also mathematically:

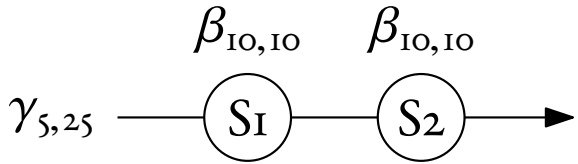
$$\sup_{t \geq 0} \left\{ \inf_{s \geq 0} \{\gamma_{r,b}(t) \leq \beta_{R,T}(t+s)\} \right\} \leftarrow \text{Definition of maximum horizontal distance}$$

$$= \sup_{t \geq 0} \left\{ \inf_{s \geq 0} \{r \cdot t + b \leq R \cdot (t + s - T)\} \right\}$$

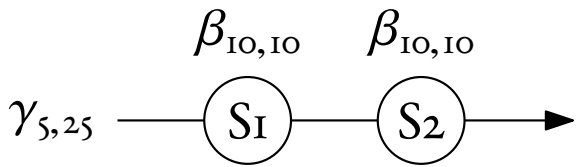
$$= T + \frac{b}{R}$$

- This simplification is **not** valid for arbitrary curves, only for $\beta_{R,T}$ and $\gamma_{r,b}$!
- Additional requirement: $r \leq R$

1d) What is the curve of the flow after it has traversed the server S1?



1d) What is the curve of the flow after it has traversed the server S1?



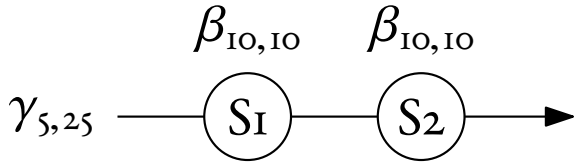
- Use the definition of the output envelope:

$$\alpha^*(t) = (\alpha \oslash \beta)(t)$$

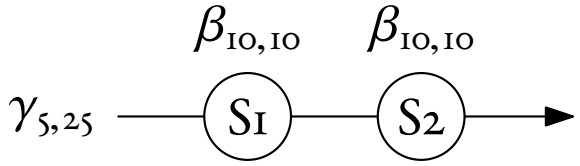
- Application of Slide 26 from the Network Calculus slideset:

$$(\gamma_{r,b} \oslash \beta_{R,T})(t) = \gamma_{r,b+r\cdot T}(t)$$

1e) Using the two previous results, what is the end-to-end latency/delay bound of the flow?

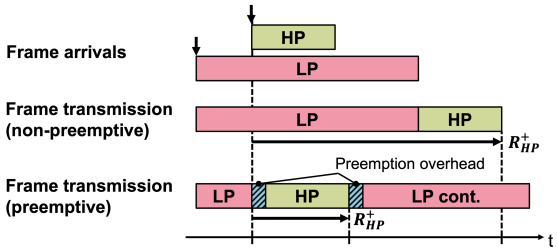


1f) Apply the concatenation theorem to compute the end-to-end latency/delay bound of the flow and interpret the result.



1g) Explain the difference between preemptive and non-preemptive scheduling.

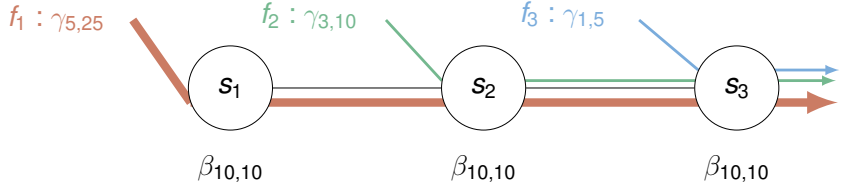
- Transmission of frames can be interrupted and resumed under preemptive scheduling
- Important for the left-over service curve calculation (non-preemptive scheduling needs to take maximum frame sizes per priority into account)



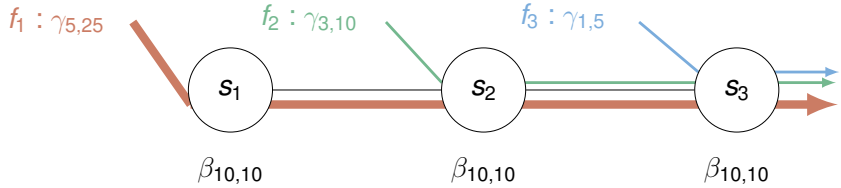
source[1]

[1] D. Thiele and R. Ernst, "Formal worst-case performance analysis of time-sensitive Ethernet with frame preemption," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 2016, pp. 1-9, doi: 10.1109/ETFA.2016.7733740.

1h) Use the three steps of the Separate Flow Analysis to calculate the latency/delay bound for f_1. Assume all servers use strict priority queuing and use preemptive scheduling. Furthermore, assume f_1 has the lowest priority and f_2 and f_3 have the highest priority. Clearly differentiate between the three steps in your answer.

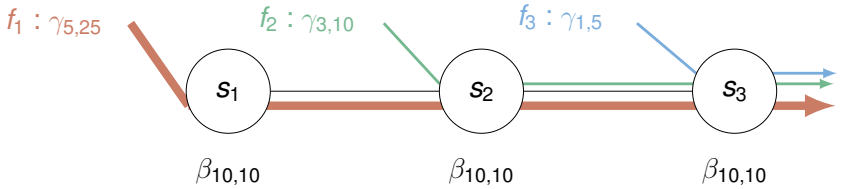


1h) Use the three steps of the Separate Flow Analysis to calculate the latency/delay bound for f_1. Assume all servers use strict priority queuing and use preemptive scheduling. Furthermore, assume f_1 has the lowest priority and f_2 and f_3 have the highest priority. Clearly differentiate between the three steps in your answer.



1. Calculate the left-over service curves of f_1 at each server
2. Concatenate all servers on the path of f_1
3. Use the resulting service curve and the arrival curve of f_1 to calculate the latency/delay bound

1h) Step 1: Compute the left-over service curves for f_1 at s_1, s_2, and s_3.

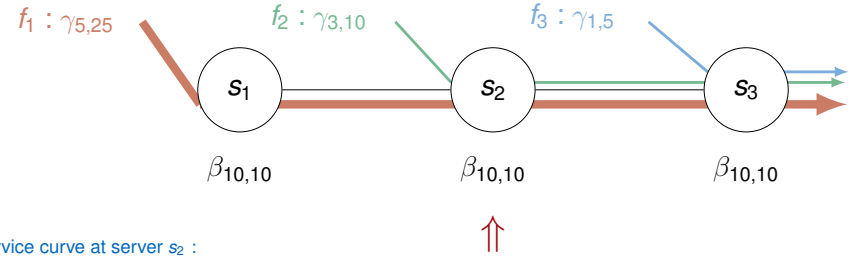


Left-over service curve at server s_1 : ⇑

- Only one flow f_1 traverses the server
- Residual service curve for f_1 is equal to the service curve, since there are no other flows present at s_1

Tutorial6 - Problem 1: Network Calculus

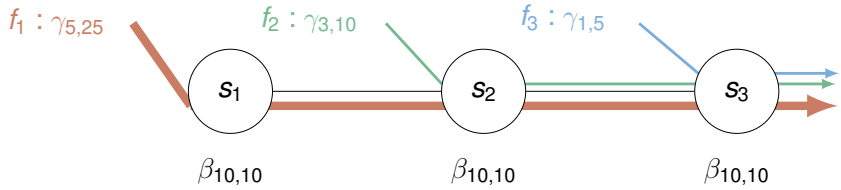
1h) Step 1: Compute the left-over service curves for f_1 at s_1, s_2, and s_3.



Left-over service curve at server s_2 :

- Two flows f_1 and f_2 traverse the server
- Residual service curve β^{f_1} for f_1 is service curve of s_2 minus the arrival curves of all higher priority flows at this server (in this case only f_2)
- $\beta^{f_i} = \left[\beta - \sum_{k=i+1}^{F} \alpha_k\right]^+$ with F: number of flows
  Note: difference to formula on lecture slides is due to the fact that in this case flows with lower ID have a lower priority → The sum always contains all higher priority flows
- $\beta^{f_1}(t) = \left[\beta_{R_2,T_2} - \gamma_{r_2,b_2}\right]^+$
- Use formula (11) from the Network Calculus slideset: $\beta^{l.o.} = \beta_{R-r,\frac{b+R\cdot T}{R-r}}$ and apply it to β^{f_1}

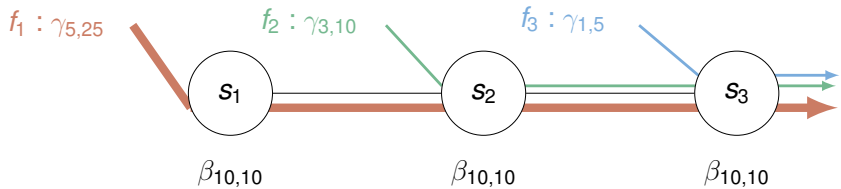
1h) Step 1: Compute the left-over service curves for f_1 at s_1, s_2, and s_3.



Left-over service curve at server s_3 :

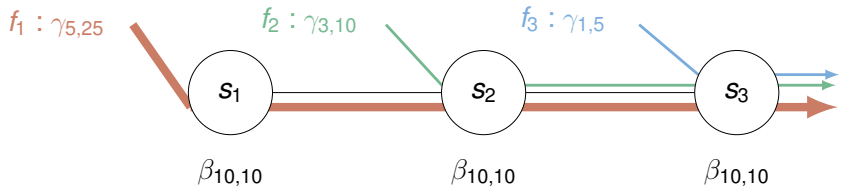
- Three flows f_1, f_2, and f_3 traverse the server
- Residual service curve β^{f_1} is service curve of s_3 minus all higher priority flows
- $\beta^{f_1}(t) = \left[\beta_{R_3,T_3} - \left(\gamma_{r_2^\star,b_2^\star} + \gamma_{r_3,b_3}\right)\right]^+$
- Careful: The flow f_2 traversed s_2 and has a different arrival curve at s_3! The new arrival curve is characterized by $r_2^\star$ and $b_2^\star$ which are the rate and burst of f_2 after traversing s_2. They can be calculated using formula (9) from the slideset

1h) Step 2: Concatenate all servers on the path of f_1.



- $\beta_{e2e}^{f_1} = \beta_{s_1}^{l.o.<f_1>} \otimes \beta_{s_2}^{l.o.<f_1>} \otimes \beta_{s_3}^{l.o.<f_1>}$
- $\beta_{min(R_1^{l.o.}, R_2^{l.o.}, R_3^{l.o.}), T_1^{l.o.} + T_2^{l.o.} + T_3^{l.o.}}$
- Result is $\beta_{e2e}^{f_1} = \beta_{6, \frac{2095}{42}}$

1h) Step 3: Calculate the latency/delay bound.



- The service curve is the left-over end-to-end service curve calculated in the last step
- The arrival curve is the arrival curve of f_1
- Use the simplified formula from **1c)** to calculate the latency/delay bound
- Result is $\frac{1135}{21} \approx 54.05$

Tutorial6 - Problem 1: Network Calculus

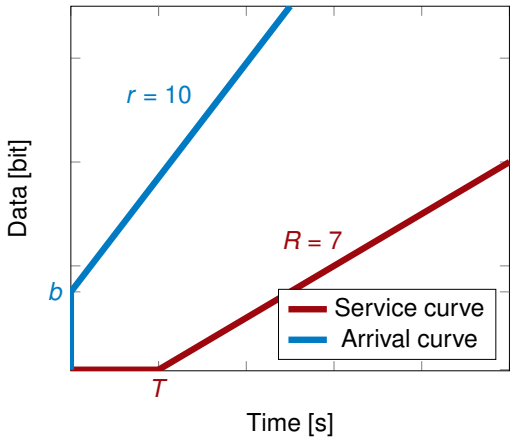
1i) Assume the priorities of the flows are switched (i.e. f_1 is the highest priority and f_2 and f_3 are the lowest priority). Use the Separate Flow Analysis to calculate the latency/delay bound of f_1 traversing the network.

1i) Assume the priorities of the flows are switched (i.e. f_1 is the highest priority and f_2 and f_3 are the lowest priority). Use the Separate Flow Analysis to calculate the latency/delay bound of f_1 traversing the network.

- Flow f_1 has the highest priority and scheduling is preemptive
- Residual service curves are equal to the service curves of the servers

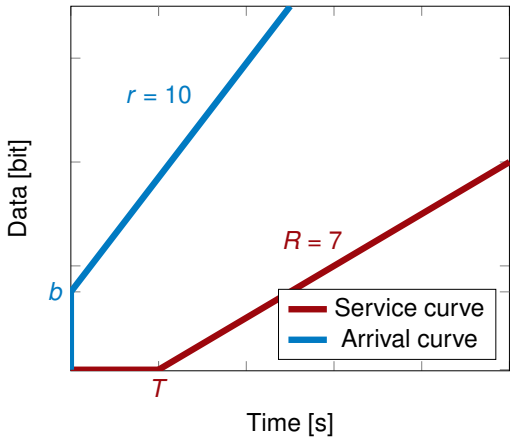
Tutorial6 - Problem 1: Network Calculus

1j) Assume a flow with an arrival curve $\gamma_{10,5}$ traverses a server with a service curve $\beta_{7,3}$. Give an explanation why the latency/delay bound is infinity.



Tutorial6 - Problem 1: Network Calculus

1j) Assume a flow with an arrival curve $\gamma_{10,5}$ traverses a server with a service curve $\beta_{7,3}$. Give an explanation why the latency/delay bound is infinity.



- Try to find the maximum horizontal distance

General approach to calculating latency/delay bounds in networks with multiple flows and multiple servers using the Separate Flow Analysis:

1. Calculate the left-over service curves at each server for the flow you want to derive the delay bound for (Consider changes of the arrival curve of other flows as they traverse servers)
2. Use the concatenation theorem to combine all left-over service curves of servers into one server with a single left-over service curve $\beta_{e2e}^{l.o.}$
3. Use the arrival curve of the flow you want to derive the latency/delay bound for in combination with the service curve $\beta_{e2e}^{l.o.}$ to derive the delay bound (maximal horizontal distance)

Note: You always calculate a latency/delay bound for **exactly one** flow at a time. You can repeat the steps to calculate it for each flow.

Tutorial6 – Problem 2: Content Delivery Networks

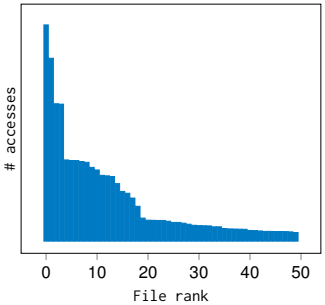
2a)

Create the access histogram from the access log.

- List of tuples, e.g. (’/images/NASA-logosmall.gif’, 1662)
- Sort descending by number of accesses
- Only keep 50 most accessed files

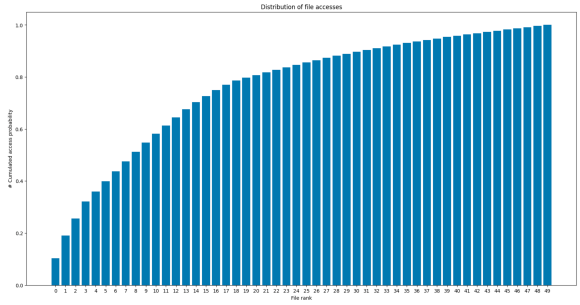
2b)

Use a bar plot to visualize the access histogram.



2c)

Plot a CDF.



2d)

Using the visualization of the previous subproblem. How many of the most popular files must be cached to answer >50% of the requests.

- Utilize the CDF (Cumulative Distribution Function) you plotted

2e)

NASA currently distributes its users to three web caches. Create the function resulting in a dictionary mapping every user to the respective cache.

- Hash function: $u \bmod |S|$ with u: User, S: Set of servers
- Hash function returns server assigned to user u

2f)

NASA upgrades their server infrastructure and now has four web caches. How many users must be reassigned to a new web cache?

- Compare remapped users in the two mappings (once with three and once with four web caches)

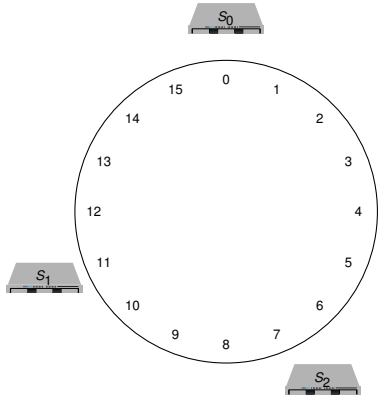
Tutorial6 – Problem 2: Content Delivery Networks

2g)

Compare the value calculated in the previous subproblem to the potential reallocation when using consistent hashing. How many clients must be remapped when using consistent hashing?

2g)

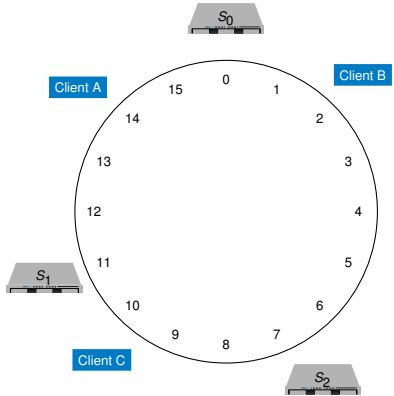
Compare the value calculated in the previous subproblem to the potential reallocation when using consistent hashing. How many clients must be remapped when using consistent hashing?



- The three cache servers mapped to a ring via a hash function

2g)

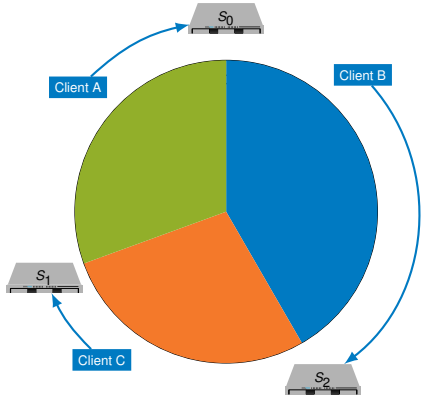
Compare the value calculated in the previous subproblem to the potential reallocation when using consistent hashing. How many clients must be remapped when using consistent hashing?



- Clients are also are also hashed and placed on the ring

2g)

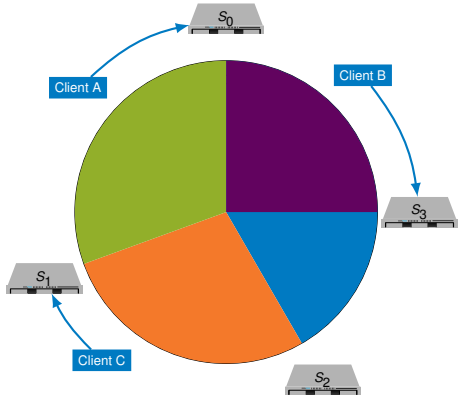
Compare the value calculated in the previous subproblem to the potential reallocation when using consistent hashing. How many clients must be remapped when using consistent hashing?



- Clients are assigned to their nearest cache server

2g)

Compare the value calculated in the previous subproblem to the potential reallocation when using consistent hashing. How many clients must be remapped when using consistent hashing?



- Introduction of new cache server
- Most clients are assigned to the same cache server as before

Tutorial6 – Problem 2: Content Delivery Networks

2g)

Compare the value calculated in the previous subproblem to the potential reallocation when using consistent hashing. How many clients must be remapped when using consistent hashing?

- Consistent hashing also allows to map cache servers to several positions

→ Leads to better average distribution of users to servers

- Formula from the lecture: $\frac{K}{N}$,
- K is the number of clients and N is the number of cache servers
- Our example has 10 users and increases from 3 servers to 4 servers