Optimization in Communication Networks Lecture 7: Internet Bandwidth Sharing (Flow-level Performance)

Prof. Yung Yi, yiyung@ee.kaist.ac.kr,
http://lanada.kaist.ac.kr
EE, KAIST

Lecture Outline

- Motivation
- Session-level (flow-level) dynamics
  - NUM is used as a resource allocation scheme
- (Flow-level) Performance of α-fairness: bad or good?
- Time-scale separation
- Reading: [Bonald and Massoulie, 2001]
- Technically, this lecture note = Lyapunov-based stability + Embedded MC.
- In [Bonald and Massoulie, 2001], another technique called "fluid limit" is used to prove stability.

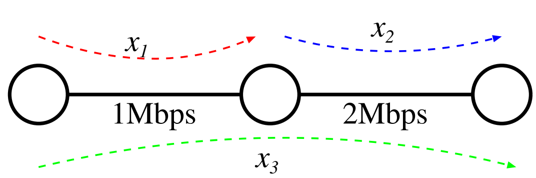
Model and NUM's Assumption

- Given a network of node with network connectivity graph $G = (V, E)$. Each edge $e \in E$ has capacity $c_e \geq 0$.

- Let R be set of possible routes that is pre-determined. Let M be routing matrix, $M = [M_{ei}]_{|E|\times|R|}$, where $M_{ei} = 1$ if $i \in R$ passes through $e \in E$, and 0 otherwise.

- Changed the notation a little bit. Sorry!

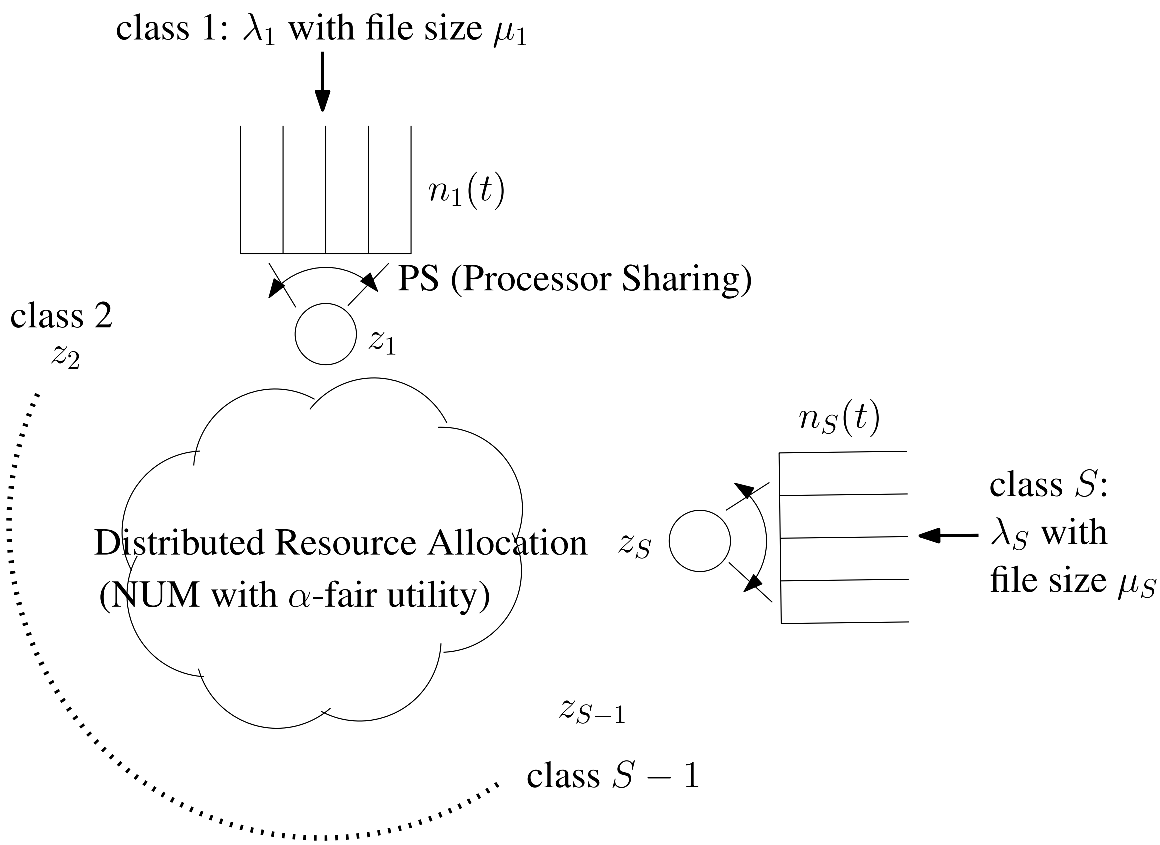
- Example



- What are the assumptions behind NUM in the earlier class?
  - Infinite backlog assumption
  - Focuses on how the network should allocate resources to each flows, given that they are fixed.

- In this lecture, we are interested in
  - What happens if flow configurations changes?
  - What would be the good performance metric?
  - etc ...

Model and Problem Formulation



- Traffic Model
  - Flows of type $i \in R$ arrive according to a Poisson process of rate λ_i.
  - They bring in amount of work (i.e., file-size) which is exponentially distributed with mean μ_i.
  - Let $\rho_i = \lambda_i \mu_i$: average amount of "workload" coming to flow type $i \in R$, called traffic intensity or load in the queuing community.
  - "Type": can be defined variously. Here we assume that "type" is defined in terms of their src, dst, and its routing path.

- (Resource allocation rule: α-fair) Let $n_i(t)$ be number of flows of type i that are in the system at time t. Then, each flow of type i is allocated rate $x_i(t)$ that is the solution to the following optimization problem:

$$\begin{array}{ll} \max & \sum n_i(t) U(x_i(t)) \\ s.t. & z_i(t) = n_i(t) \times x_i(t) \\ & y(t) = Mz(t) \leq C, \\ & x(t) \leq 0. \end{array}$$

  What is $x(t)$ and $y(t)$ intuitively? What we are assuming here?

  Note that you can imagine other allocation rule.

- Time-scale separation: Resource allocation (packet-level dynamics) is faster than flow-level dynamics. Good assumption, why?

- Problem Formulation

  First Goal: Flow-level stability

  Under a given resource allocation A, what are the precise conditions on $\rho = (\rho_i)$ (or equivalently the set of $\rho = (\rho_i)$ "supported" by A) that allows for $n(t)$ to remain finite with probability 1.

  What does flow-level stability mean physically?

  Second Goal: Delay, i.e., how long does it take for each flow to be served?

- Why is this problem challenging? Complex coupling

  Completely depends on (packet-level) resource allocation algorithm (i.e., solution from NUM)

Necessary Condition

- "Obvious" necessary condition

$$M\rho < C, \quad \text{or} \quad \sum_{i:i\in e} \rho_i < C_e$$

- Any allocation that guarantees flow-level stability should satisfy the above.

- Why obvious? Think, using contradiction.

- The proof starts with: Consider ρ, where $M\rho \geq C$. Then, we have to find a contradiction.

- Refer to the paper for the proof (homework).

Sufficient Condition

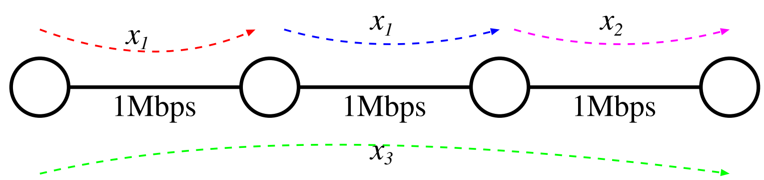
- The first question: if we allocate bandwidth to flows maximally (i.e., pareto-efficiently),

  Definition. A bandwidth allocation λ is said to be Pareto-efficient if for any bandwidth allocation x such that $y \geq x$, it then holds that $x = y$.

- Is the necessary condition sufficient under the Pareto-efficient allocation?

- Intuitively yes, but unfortunately no. Thus, we should find an allocation (if exists) better than the Pareto-efficient allocation.

- Counter example.



  - 3 links, unit capacity, 4 flow types: 0,1,2,3.
  - Necessary condition: $\rho_0 + \rho_i < 1$ for $i = 1, 2, 3$ - **C1**
  - Consider the following priority allocation: if flow $i \in \{1, 2, 3\}$ is present, it gets all of link i. When all the three links are available, flow 0 gets the links 1,2,3 to itself.
    Pareto-optimal, right? No waste of resource
  - But, priority allocation requires that $\rho_0 < (1 - \rho_1)(1 - \rho_2)(1 - \rho_3)$ - **C2**
  - Consider $\rho_0 = \rho_1 = \rho_2 = \rho_3 = 1/3$.
  - Satisfies **C1**, but not **C2**.
  - Thus, priority allocation is pareto-efficient, where, however, the necessary condition is not equal to the sufficient condition.

- There exists a maximal (i.e., pareto-efficient) resource allocation, where the necessary condition is NOT the sufficient condition.

- We need non-trivial resource allocation ...

What About α-Fair Allocations?

- For any $\alpha \in (0, \infty)$, the α-fair allocation policy induces flow numbers to be finite with probability 1 as long as the necessary condition is satisfied. Surprising!

- α-fair allocation is also very good in terms of session-level performance.

- $(n_i(t))_{i \in R}$: the number of flows at time t.

- $(X_i(t))_{i \in R}$: rate allocated to flow type i, which is the solution of the following:

$$\begin{aligned} \max \quad & \sum_{i \in R} U^\alpha(x_i(t)) n_i(t) \\ s.t. \quad & \sum_{i : i \in E} n_i(t) x_i(t) \le C_e, \quad x_i(t) \ge 0, \end{aligned}$$

where when $\alpha \neq 1, \alpha > 0$,

$$U^\alpha(x) = \frac{x^{1-\alpha}}{1-\alpha},$$

and when $\alpha = 1$,

$$U^\alpha(x) = \log(x).$$

- The above defines Markov chain $n(t) = (n_i(t) : i \in R)$, because
  - Arrival process of flow type i is a Poisson of rate λ_i
  - Departure of a type i happens when it is served completely. But, the service requirement is exponentially distributed; hence memory-less. The exponential service has mean μ_i or rate $1/\mu_i$.
  - The rate $X_i(t)$ remains the same until next transition. Thus, the effective departure "rate" is $X_i(t)/\mu_i$

- Thus, Markov chain transition rates are: for all $i \in R$,

$$\begin{aligned} n_i(t) &\to n_i(t) + 1 && \text{at rate } \lambda_i \\ n_i(t) &\to (n_i(t) - 1)^+ && \text{at rate } \frac{n_i(t)X_i(t)}{\mu_i} \end{aligned}$$

Think! Can you write down the above transition rates for yourself?: Start of this research!

Question again: Is this homogeneous MC or not?

- More systematic method of computing the transition rate: Over the short-time interval h,

- $n_i(t) \to n_i(t) + 1$: What is the probability that we have one arrival over h in the Poisson process? That is, the probability that $\mathbb{P}[A \leq h]$, where A is an exponential r.v. with rate λ_i.

- $n_i(t) \to (n_i(t) - 1)^+$: What is the probability that one of $n_i(t)$ session will finish the service over h? That is, the probability that $\mathbb{P}[\min(A_1, A_2, \ldots, A_{n(t)}) \leq h]$, where A_i are the exponential r.v. with rate $1/\mu_i$.

Question: What is the minimum of two exponential random variables?

Do we need to analyze with continuous HMC?

- Enough to consider a "discretized" version (embedded MC) of the original continous MC.

- Remember that what we want to show is

$$\max_{i \in R} n_i(t) < \infty, \quad \text{w.p. } 1.$$

By the way, why do we keep needing "w.p. 1"?

- In the embedded MC, a transition happens every time.

- As long as the rate of transitions is *bounded* above in the original MC in continuous time, it is sufficient to prove the finiteness of $n(\cdot)$ w.r.t. discrete time.

- Indeed, the arrival rate is bounded, and the service rate is also bounded, since

$$\frac{X_i(t) n_i(t)}{\mu_i} \leq (\max_e C_e)(\max_i 1/\mu_i) < \infty.$$

Transition Matrix of Embedded MC

- In each time, exactly one transition happens. The transition probabilities are as follows:

$$\begin{aligned} P_{n_i,n_i+1} &\propto \lambda_i \\ P_{n_i,n_i-1} &\propto \frac{X_i n_i}{\mu_i}, \end{aligned}$$

  where X_i is the solution of NUM.

- As we know, it is sufficient to establish that the discrete MC $n(t) = (n_i(t) : i \in R)$ is *positive recurrent.*

Main Results

- Theorem. Under α-fair allocation, the HMC $n(t)$ is positive recurrent if $M\rho < C$.

  Proof. . We will use Lyapunov function-Foster's criteria to prove this theorem.

  Note that the rate $X(t) = (X_i(t))$ is determined by the following solution:

$$\begin{aligned} \max_{x \geq 0} \quad & \sum_{i \in R} n_i(t) \left(\frac{(x_i)^{1-\alpha}}{1-\alpha} \right) \\ s.t. \quad & \sum_{i : i \in E} n_i(t) x_i \leq C_e. \end{aligned} \tag{1}$$

- Step 1: Re-express (1) in terms of $n_i(t) \cdot x_i$, i.e., the total rate allocated to type i.

$$\begin{aligned} \sum_{i \in R} n_i(t) \left(\frac{(x_i)^{1-\alpha}}{1-\alpha} \right) &= \sum_{i \in R} n_i(t) \frac{n_i(t)^{1-\alpha}}{n_i(t)^{1-\alpha}} \left(\frac{(x_i)^{1-\alpha}}{1-\alpha} \right) \\ &= \sum n_i^{\alpha}(t) \frac{1}{1-\alpha} \left(\frac{x_i}{n_i(t)} \right)^{1-\alpha} \end{aligned} \tag{2}$$

Using (2), define

$$G(u) = \sum n_i^{\alpha}(t) \frac{u_i^{1-\alpha}}{1-\alpha}.$$

Note that $G(u)$ is a concave function. Now, we change our interest in terms of $\Lambda(t)$. Then, we can prove that solving (1) is equivalent to solving

$$\max_{u \geq 0} G(u),$$

and subject to the convex constraint. Let

$$\Lambda(t) = \arg\max_{u \geq 0} G(u),$$

Simple representation! Note that the solution $\Lambda(t)$ is unique due to convex optimization formulation.

- Step 2. Under the condition $M\rho < C$, $\exists \epsilon > 0$, such that $(1+\epsilon)\rho M < C$, i.e., $(1+\epsilon)\rho$ is also a feasible vector.

  We will prove that $(1+\epsilon)\rho$ can be stabilized by α-fair allocation. Then we are done.

- Step 3. Consider a Lyapunov function:

$$V(n(\tau)) = \sum_i \frac{n_i^{\alpha+1}(\tau)}{\alpha+1} \rho_i^{-\alpha} \mu_i.$$

For ease of writing, let's understand "differentiation" of $V(n(\tau))$ on "average":

$$\frac{dV(n(\tau))}{d\tau} = \sum_i n_i^{\alpha}(\tau)\rho_i^{-\alpha}\mu_i \frac{dn_i(\tau)}{d\tau}.$$

Note that

$$\mathbb{E}\Big[\frac{dn_i(\tau)}{d\tau}\Big] = \phi(\lambda_i - \Lambda_i/\mu_i),$$

for some constant $\phi > 0$.

Thus, we have:

$$\begin{aligned}\mathbb{E}\Big[\frac{dV(n(\tau))}{d\tau}\Big] &= \phi \sum_i n_i^{\alpha}(\tau)\rho_i^{-\alpha}\mu_i(\lambda_i - \Lambda_i/\mu_i) \\ &= \phi \sum_i n_i^{\alpha}(\tau)\rho_i^{-\alpha}(\rho_i - \Lambda_i). \qquad (3)\end{aligned}$$

Now, using the optimality of $\Lambda = (\Lambda_i)$, we need to prove that (3) has a negative drift.

- Step 4. Property from the optimality of $\Lambda = (\Lambda_i)$.

  By "gradient" condition (Remember?), we know that for any feasible rate allocation u,

$$\nabla G(\Lambda)^T (u - \Lambda) \leq 0,$$

and we have

$$\nabla G(u)^T(u-\Lambda) \leq \nabla G(\Lambda)^T(u-\Lambda) \leq 0, \tag{4}$$

where the first inequality comes from concavity of $G(u)$. Why?

[Imagine a single variable function, $f(\cdot)$ which is concave, i.e., $f''(\cdot) \leq 0$. Now

$$f'(x) - f'(y) \approx f''(y) \cdot (x-y)$$

Then, $(f'(x) - f'(y))(x-y) \approx f''(y) \cdot (x-y)^2 \leq 0$.]

In (4), putting $u = (1+\epsilon)\rho$,

$$\nabla G((1+\epsilon)\rho)^T((1+\epsilon)\rho - \Lambda) \leq 0,$$

which implies:

$$\begin{aligned} &\sum_i n_i^\alpha(t) \cdot \rho_i^{-\alpha}(1+\epsilon)^{-\alpha}((1+\epsilon)\rho_i - \Lambda_i) \leq 0 \\ &\rightarrow \sum_i n_i^\alpha(t) \cdot \rho_i^{-\alpha}((1+\epsilon)\rho_i - \Lambda_i) \leq 0 \end{aligned} \tag{5}$$

- Step 5. Final step:

$$(3) \quad = \quad -\phi(\sum_i n_i^\alpha(\tau)\rho_i^{-\alpha+1}) +$$

$$\phi \sum_i n_i^{\alpha}(t) \cdot \rho_i^{-\alpha}((1+\epsilon)\rho_i - \Lambda_i),$$

where the second term is ≤ 0, so we are done.

Trace back the proof to understand how to prove!

Analogy: Scheduling and Bandwidth Allocation

- Always important to understand the big picture and the philosophy
- Resource allocation: Max-Weight vs. NUM
- Packet-level resource allocation vs. Session-level resource allocation
- Packet-level stability vs. Session-level stability
- Start of the proof: Choose a vector that is inside the throughput or rate region, where the vector has ϵ-distance to the boundary of throughput or rate region.
- What to prove: finiteness of $Q(t)$ or $n(t)$.
- Connection to Markov chain: Positive recurrence
- Lyapunov function: Choose a function whose derivative (i.e., Lyapunov drift) is “connected” to Max-Weight or NUM.

- What makes the negative drift? Max-Weight itself or NUM (whose gradient condition)

  Max-Weight: For any feasible λ, $\lambda = \sum_i \alpha_i \boldsymbol{S}_i$, where $\sum_i \alpha_i < 1$, so we can find ϵ, such that

  $$\lambda \cdot Q(t) - W^\star(t) < -(1 - \sum_i \alpha_i) W^\star(t) = -\epsilon W^\star(t)$$

  Bandwidth-sharing: For any feasible ρ, we can find ϵ, such that

  $$\nabla G((1+\epsilon)\rho)^T ((1+\epsilon)\rho - \Lambda) \leq 0.$$

Summary and Key Messages

- NUM's assumption
  - Infinite backlog
  - Static flow populations

- Flow-level dynamics and its performance
  - Pareto-efficient resource allocation policies does not perform very well in some cases.
  - α-fairness allocation: good for fairness quantification, necessary and sufficient flow-level stability condition
  - In other words, rate region = flow-level-stability region, and that is the maximum stability region.

- Sometimes, just for stability (or positive recurrence), enough to consider the discrete embedded MC.

References

[Bonald and Massoulie, 2001] Bonald, T. and Massoulie, L. (2001). Impact of fairness on internet performance. In *Proceedings of ACM Sigmetrics*.