Aalto University
School of Electrical Engineering

ELEC-E7450
Performance Analysis

Processor sharing networks

Samuli Aalto
Department of Communications and Networking

Contents

- Processor sharing networks
- Fairness concepts
- Whittle networks
- Balanced fairness
- Performance

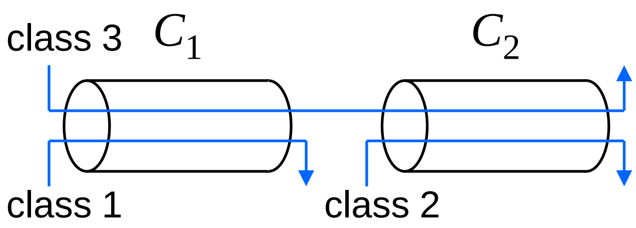
Processor sharing network

- Processor sharing network is a flow-level model of a data network loaded with elastic traffic.
- Elastic traffic consists of flows, such as file transfers using TCP.
- Elasticity refers to the property that the transmission rate of a flow is not fixed but it is adjusted according to the congestion state of the system.
- The network consists of J links with capacities C_j (in bits/sec)
- There are K traffic classes
- All flows in class k follow the same route:

$$a_{k,j} = \begin{cases} 1, & \text{if link } j \text{ belongs to the route of class } k; \\ 0, & \text{otherwise.} \end{cases}$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

Necessary stability condition

- New flows of class k arrive according to a Poisson process at rate λ_k (in 1/sec)
- Flow sizes B_k in class k (in bits) are exponentially distributed with mean $E[B_k] = 1/\beta_k$
- Load of class k (in bits/sec) is defined by

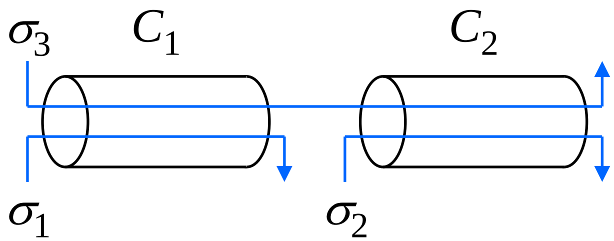
$$\sigma_k := \lambda_k E[B_k]$$

- Link capacity C_j is shared by the flows of classes k for which $a_{k,j} = 1$
- Necessary stability conditions are thus as follows: for each link j, we have constraint

$$\sum_{k=1}^{K} \sigma_k a_{k,j} < C_j$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$\sigma_1 + \sigma_3 < C_1$$
$$\sigma_2 + \sigma_3 < C_2$$

State description

- The network state is described by vector

$$N(t) = (N_1(t), \ldots, N_K(t))$$

where $N_k(t)$ refers to the total number of flows in class k at time t

- Process $N(t)$ is an irreducible Markov process with state space

$$S = \{n = (n_1, \ldots, n_K) \mid n_k \in \{0,1,2,\ldots\}\}$$

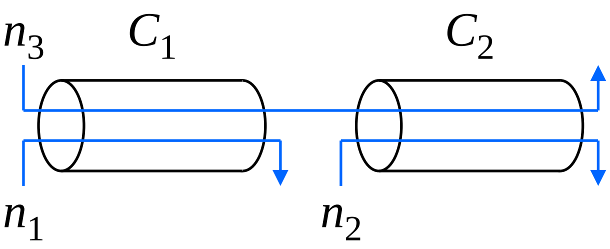
- The transition rates depend on the arrival rates λ_k, mean flow sizes $1/\beta_k$, and inter-class allocations $\phi_k(n)$ (defined in the following slide) as follows:

$$q(n, n+e_k) = \lambda_k$$

$$q(n+e_k, n) = \beta_k \phi_k(n+e_k)$$

Example: Linear network

$$K = 3, \quad J = 2$$



Resource allocation

- Inter-class allocations $\phi_k(n)$ may depend on the network state n, and they specify how the link capacities are shared among the flow classes
- These inter-class allocations $\phi_k(n)$ are feasible if, for all links j,

$$\sum_{k=1}^{K} \phi_k(n) a_{k,j} \le C_j$$

- The family of feasible allocations is called the capacity set C,

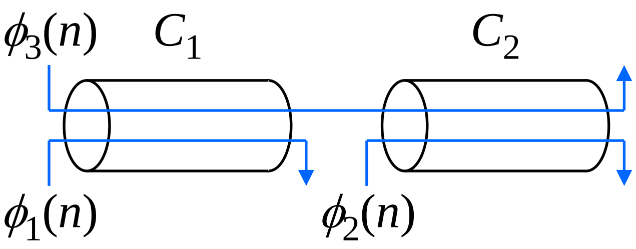
$$C = \{c = (c_1, \dots, c_K) : \sum_{k=1}^{K} c_k a_{k,j} \le C_j \quad \forall j\}$$

- Intra-class flow allocations $\psi_k(n)$ are assumed to be fair so that each flow in class k gets an equal share denoted by

$$\psi_k(n) := \phi_k(n) / n_k$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$\phi_1(n) + \phi_3(n) \le C_1$$
$$\phi_2(n) + \phi_3(n) \le C_2$$

- **Note**: If n is such that $n_k = 0$, we assume that $\phi_k(n) = 0$

Contents

- Processor sharing networks
- Fairness concepts
- Whittle networks
- Balanced fairness
- Performance

Max-min fairness

- Aims at maximizing the minimum intra-class flow allocation
- Max-min fair intra-class flow allocations $\psi_k(n)$ satisfy, for any $c \in C$,

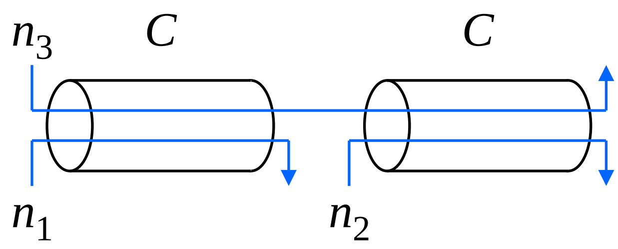
$$\frac{c_k}{n_k} > \psi_k(n) \Rightarrow \exists l: \frac{c_l}{n_l} < \psi_l(n) \leq \psi_k(n)$$

Waterfilling algorithm:

1. Set all intra-class flow allocations to zero
2. Increase the allocations for all non-freezed flows evenly until the capacity of some link is fully used
3. Freeze the allocations for the classes that use this link
4. If there are still non-freezed classes, go back to 2
5. Stop

Example: Linear network

$$K = 3, \quad J = 2$$



$$\phi_1(n) = \frac{\max\{n_1, n_2\}}{\max\{n_1, n_2\} + n_3} C$$

$$\phi_2(n) = \frac{\max\{n_1, n_2\}}{\max\{n_1, n_2\} + n_3} C$$

$$\phi_3(n) = \frac{n_3}{\max\{n_1, n_2\} + n_3} C$$

Proportional fairness

- Utility based fairness concept
- Utility of bit rate b_k for a class-k flow:

$$U(b_k) = \log b_k$$

- Optimization:

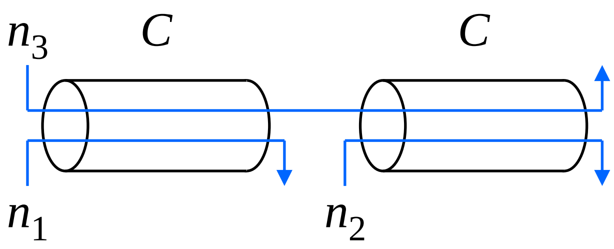
$$\psi(n) = \underset{b:(n_1 b_1,\ldots,n_K b_K)\in C}{\arg\max} \sum_{k=1}^{K} n_k U(b_k)$$

- Inter-class allocations

$$\phi_k(n) = n_k \psi_k(n)$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$\phi_1(n) = \frac{n_1+n_2}{n_1+n_2+n_3} C$$

$$\phi_2(n) = \frac{n_1+n_2}{n_1+n_2+n_3} C$$

$$\phi_3(n) = \frac{n_3}{n_1+n_2+n_3} C$$

Alpha fairness

- Utility based fairness concept with parameter $\alpha \geq 0$
- Utility of bit rate b_k for a class-k flow:

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

- Optimization:

$$\psi(n) = \underset{b:(n_1 b_1, \ldots, n_K b_K) \in C}{\arg\max} \sum_{k=1}^{K} n_k U(b_k)$$

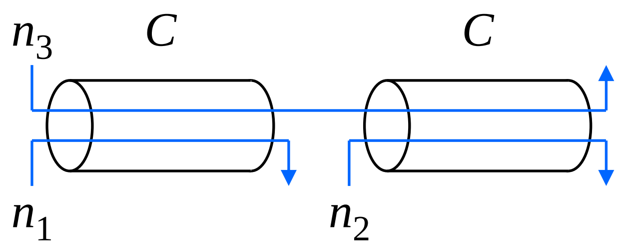
- Inter-class allocations

$$\phi_k(n) = n_k \psi_k(n)$$

- Special cases:
  - $\alpha = 0$: Maximum total bit rate
  - $\alpha = 1$: Proportional fairness
  - $\alpha \to \infty$: Max-min fairness

Example: Linear network

$$K = 3, \quad J = 2$$



$$\phi_1(n) = \frac{(n_1^\alpha + n_2^\alpha)^{1/\alpha}}{(n_1^\alpha + n_2^\alpha)^{1/\alpha} + n_3} C$$

$$\phi_2(n) = \frac{(n_1^\alpha + n_2^\alpha)^{1/\alpha}}{(n_1^\alpha + n_2^\alpha)^{1/\alpha} + n_3} C$$

$$\phi_3(n) = \frac{n_3}{(n_1^\alpha + n_2^\alpha)^{1/\alpha} + n_3} C$$

Contents

Whittle network

- A processor sharing network is a Whittle network if
  - (i) the inter-class allocations $\phi_k(n)$ are feasible for any state n, and
  - (ii) there is a positive function $\Phi(n)$ such that $\Phi(0) = 1$ and

$$\phi_k(n+e_k) = \frac{\Phi(n)}{\Phi(n+e_k)}$$

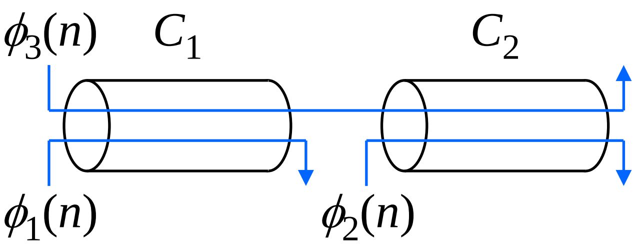
- Function $\Phi(n)$ is called the corresponding balance function.

- **Proposition**:
  For a Whittle network the inter-class allocations are balanced as follows:

$$\frac{\phi_k(n+e_k+e_{k'})}{\phi_{k'}(n+e_k+e_{k'})} = \frac{\phi_k(n+e_k)}{\phi_{k'}(n+e_{k'})}$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$\phi_1(n) + \phi_3(n) \le C_1$$
$$\phi_2(n) + \phi_3(n) \le C_2$$

Steady-state distribution

- **Theorem**:
  Consider a stable Whittle network. The steady-state distribution of process $N(t)$ is

$$P\{N=n\}=\frac{\Phi(n)}{G}\prod_{k=1}^{K}\sigma_k^{n_k}$$

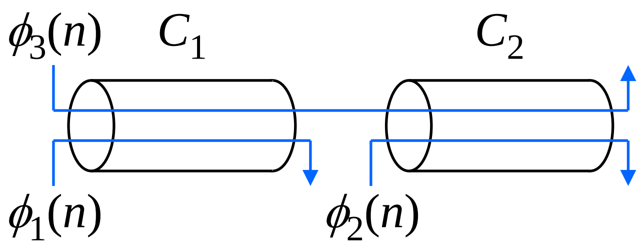
where

$$G:=\sum_{n'\in S}\Phi(n')\prod_{k=1}^{K}\sigma_k^{n_k'}$$

- **Note**: As in a single-server M/G/1-PS queue, it can be shown that the steady-state distribution of a stable Whittle network is insensitive to the flow size distributions (as long as the mean flow sizes $1/\beta_k$ remain the same for all classes k).

Example: Linear network

$$K=3,\quad J=2$$



$$\phi_1(n)+\phi_3(n)\le C_1$$
$$\phi_2(n)+\phi_3(n)\le C_2$$

Contents

Balanced fairness (1)

- Consider a Whittle network. If $n - e_k \in S$, then

$$\phi_k(n) = \frac{\Phi(n-e_k)}{\Phi(n)}$$

- Thus, to get feasible balanced allocations, we have to require that

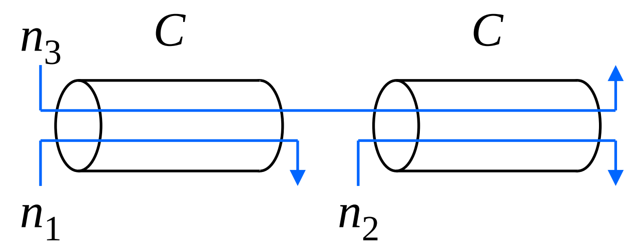
$$\left(\frac{\Phi(n-e_1)}{\Phi(n)}, \ldots, \frac{\Phi(n-e_K)}{\Phi(n)} \right) \in C$$

- Balanced fairness (BF) refers to the case where these feasible balanced allocations are maximized:

$$\Phi(n) = \min\left\{ \alpha > 0 : \left(\frac{\Phi(n-e_1)}{\alpha}, \ldots, \frac{\Phi(n-e_K)}{\alpha} \right) \in C \right\}$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$\Phi(n) = \binom{n_1 + n_2 + n_3}{n_3} \left(\frac{1}{C} \right)^{n_1+n_2+n_3}$$

Balanced fairness (2)

- **Proposition**:
  Consider a Whittle network with balanced fair allocations. The system is stable if and only if the necessary stability conditions

$$\sum_{k=1}^{K} \sigma_k a_{k,j} < C_j$$

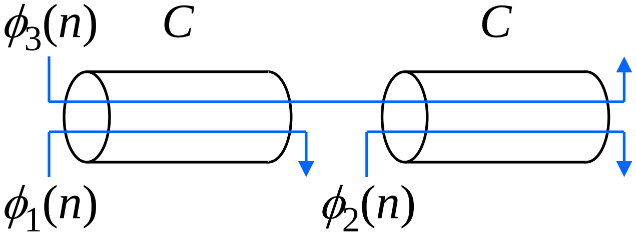
are satisfied that for all links j.

- **Proposition**:
  Consider a Whittle network with balanced fair allocations. Its balance function $\Phi(n)$ satisfies the following recursion:

$$\Phi(n) = \max_j \frac{1}{C_j} \sum_{k=1}^{K} \Phi(n - e_k) a_{k,j}$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$\phi_1(n) = \frac{n_1 + n_2}{n_1 + n_2 + n_3} C$$

$$\phi_2(n) = \frac{n_1 + n_2}{n_1 + n_2 + n_3} C$$

$$\phi_3(n) = \frac{n_3}{n_1 + n_2 + n_3} C$$

Contents

Performance (1)

- Consider a stable Whittle network with balanced fair allocations.
- Let N_k denote the steady-state variable for the number of flows in class k, and T_k the steady-state flow-level delay, i.e., the total time needed for transferring all the bits of a class-k flow.
- Define the normalization constant G as a function

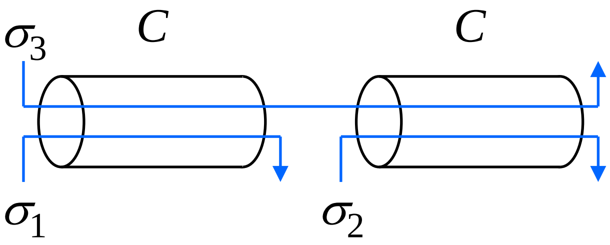
$$G(\sigma) := \sum_{n \in S} \Phi(n) \prod_{k=1}^{K} \sigma_k^{n_k}$$

where σ is the load vector defined by

$$\sigma = (\sigma_1, \ldots, \sigma_K)$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$G(\sigma) = \frac{C(C - \sigma_3)}{(C - \sigma_1 - \sigma_3)(C - \sigma_2 - \sigma_3)}$$

Performance (2)

- From Theorem in slide 13, we get

$$E[N_k] = \frac{\sigma_k}{G(\sigma)} \sum_{n \in S} (n_k + 1)\Phi(n + e_k) \prod_{i=1}^{K} \sigma_i^{n_i}$$

- Thus,

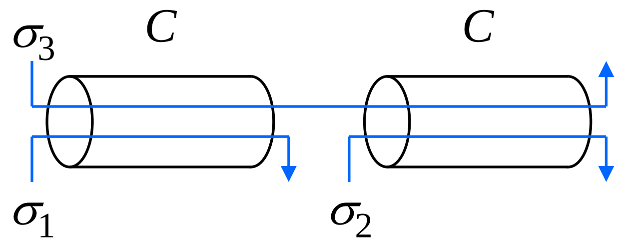
$$E[N_k] = \frac{\sigma_k}{G(\sigma)} \frac{\partial G(\sigma)}{\partial \sigma_k}$$

and by Little's formula

$$E[T_k] = \frac{E[B_k]}{G(\sigma)} \frac{\partial G(\sigma)}{\partial \sigma_k}$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$E[T_1] = \frac{E[B_1]}{C - \sigma_1 - \sigma_3}$$

$$E[T_2] = \frac{E[B_2]}{C - \sigma_2 - \sigma_3}$$

$$E[T_3] = \frac{E[B_3]((C - \sigma_3)^2 - \sigma_1 \sigma_2)}{(C - \sigma_1 - \sigma_3)(C - \sigma_2 - \sigma_3)(C - \sigma_3)}$$

Performance (3)

- The average bit rate γ_k of a class-k flow (in bits/sec) is

$$\gamma_k := \frac{E[B_k]}{E[T_k]} = \frac{G(\sigma)}{\frac{\partial G(\sigma)}{\partial \sigma_k}}$$

- Let X denote the steady-state variable for the total number of flows, and T the steady-state flow-level delay for an arbitrary flow.
- The mean steady-state number of flows is

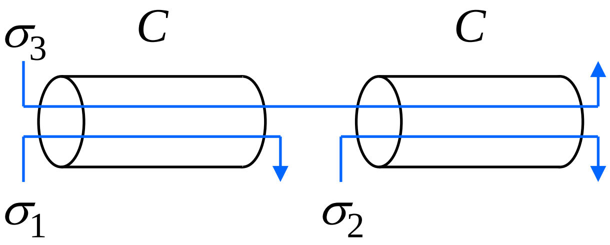
$$E[X] = \sum_{k=1}^{K} \frac{\sigma_k}{\gamma_k}$$

and the mean flow level delay (by Little's formula)

$$E[T] = \frac{1}{\lambda} \sum_{k=1}^{K} \frac{\sigma_k}{\gamma_k}$$

Example: Linear network

$$K = 3, \quad J = 2$$



$$\gamma_1 = C - \sigma_1 - \sigma_3$$

$$\gamma_2 = C - \sigma_2 - \sigma_3$$

$$\gamma_3 = \frac{(C-\sigma_1-\sigma_3)(C-\sigma_2-\sigma_3)(C-\sigma_3)}{(C-\sigma_3)^2 - \sigma_1\sigma_2}$$

Summary

- Processor sharing networks
  - flow-level model, elastic traffic, necessary stability conditions, inter-class allocations, feasible allocations, intra-class allocations
- Fairness concepts
  - max-min fairness, waterfilling algorithm, proportional fairness, logarithmic utility, alpha-fairness, utility function
- Whittle networks
  - balance function, balanced allocations, product-form steady-state distribution, DBE
- Balanced fairness
  - maximal feasible balanced allocations, BF recursion
- Performance
  - flow-level delay, average bit rate