Balanced fairness

Adapted from a poster presentation by T. Bonald and A. Proutière

J. Virtamo

Helsinki University of Technology

Networking Laboratory
Outline

- Introduction, bandwidth sharing, insensitivity . . .
- Network models: physical network, Whittle queueing network
- Balance property, balance function, balanced fairness
- Examples: lines, grids, trees, cycles
- How to calculate the balance function: parking lot
- Normalization constant and performance measures
- Recursive algorithm for the normalization constant and throughputs
- Examples: line, parking lot, trees; comparison with bounds
- Limited access rate
- State dependent arrival rates, adaptive routing
- Open problems, future research
Introduction

- Traffic theory and network design
  - The objective is to characterize the demand–capacity–performance relationship accounting random flow arrivals.
  - Can we operate the network so that this relationship depends only on simple traffic parameters – like the Erlang formula in telephone networks?
Bandwidth sharing and the notion of utility

- Static bandwidth sharing
  - Max-min fairness (Bertsekas & Gallager, 1987)
  - Proportional fairness (Kelly, 1998)
  - More general utility-based allocations (Mo & Walrand, 2000)
Bandwidth sharing (continued . . .)

- Dynamic bandwidth sharing with random traffic
  - A single link: explicit results for fair sharing (Ben Fredj & al., 2001)
    * insensitive to any traffic characteristics
  - A homogeneous line, grid or hypercube (Bonald & Massoulié, 2001)
    * explicit results for proportional fairness only
    * also insensitive to any traffic characteristics
Questions posed by Bonald and Proutière

*Can we characterize the set of allocations leading to insensitivity?*

*Can we approximate the performance of usual allocations (max-min, TCP, ...)?*
Insensitivity in stochastic networks

- **Insensitivity**: the state distribution does not depend on the service time distributions.

- **Literature**: Barbour, Kelly, Baccelli, Baskett et al., Schassberger, Whittle, Miyazawa, Henderson, Burman, ...
Insensitivity (continued . . .)

- Insensitivity in GSMP (Schassberger, 1977)
  - Partial balance is equivalent to insensitivity.
  - But this does not cover simple stochastic networks – not even the M/GI/PS queue.
  - Usual networks are covered by more general processes, R-GSMP (Schassberger, 1986, Miyazawa, 1993),
    * partial balance is equivalent to product form decomposability, which implies insensitivity,
    * the converse is not true in general.
Insensitivity (continued . . .)

- Bonald and Proutière
  - proved that for state dependent processor sharing networks partial balance is equivalent to insensitivity,
  - characterized insensitive allocations in data networks.

- Balanced fairness
  - Flow level performance metrics are independent of detailed traffic characteristics: flow size distribution, distribution of the number of flows per session, correlation between successive flow sizes and think-time durations . . . as far as session arrivals are Poisson.
Physical network model

- Network: $L$ links
  - capacities $C_1, \ldots, C_L$

- Flows: $K$ flow types (classes)
  - route $r_i$ (a set of links)
  - arrival rate $\lambda_i$
  - mean flow size $1/\mu_i$
  - network state $x = (x_1, \ldots, x_K)$: number of flows in each class
Whittle queueing network model (session/flow level)

- Network: $K$ processor sharing nodes
  - Each node represents a flow class (route) or a think time.
  - Node is served with capacity $\phi_i(x)$ (depends on the network state).
  - External Poisson arrival rate to each node $\nu_i$
    * represents session arrivals starting with flow $i$.
  - After completion of flow $i$ the session moves to another node $j$
    * probability $p_{ij}$: session generates another flow or goes idle
Constraints

Total arrival rate of type $i$ flows determined by

$$\lambda_i = \nu_i + \sum_j p_{ij} \lambda_j, \quad \rho_i = \lambda_i / \mu_i$$

Bandwidth constraint

$$\sum_{i \in \mathcal{F}_l} \phi_i(x) \leq C_l, \quad \forall l,$$

Stability condition

$$\sum_{i \in \mathcal{F}_l} \rho_i < C_l, \quad \forall l,$$

where $\mathcal{F}_l$ = the set of flow classes going through link $l$. 
Example 1

- A data network represented as a processor sharing network
Example 2

- From Poisson arrivals, i.i.d. exponential services and Bernoulli routing to very general traffic characteristics: one can represent any type of session (succession of correlated flows and think times) by adding new nodes.

- For example, a session on a route with 2 flows types with Erlang-2 and exponential sizes. The exponential flow is repeated separated by an exponential think time:

\[ \phi_i^1(x) = \frac{x_i^1}{x_i^1 + x_i^2} \phi(x) \]
\[ \phi_i^2(x) = \frac{x_i^2}{x_i^1 + x_i^2} \phi(x) \]
\[ \phi_i^3(x) = x_i^3 \]
Insensitive PS networks: balance property

- Whittle network: PS network satisfying the balance property

\[
\frac{\phi_i(x - e_j)}{\phi_i(x)} = \frac{\phi_j(x - e_i)}{\phi_j(x)}, \quad \forall i, j, x_i > 0, x_j > 0,
\]

\[
\phi_i(x) \phi_j(x - e_i) = \phi_j(x) \phi_i(x - e_j)
\]

“left path” = “right path”
Balance property (continued . . .)

- Remark: if for all classes $i$ the allocation $\phi_i$ depends only on $x_i$ and not on the number of flows in other classes, the system is balanced;
  
  - because then $\phi_i(x - e_j) = \phi_i(x)$
  
  - e.g., multibitrate systems
Balance function

- Let $p$ be a path of length $n$ from state 0 to state $x$ with $n$ flows

\[ p = (x, x - e_{k_1}, x - e_{k_1} - e_{k_2}, \ldots, x - e_{k_1} - \ldots - e_{k_{n-1}}, 0) \]

- When the balance property holds the product

\[ \phi_{k_1}(x)\phi_{k_2}(x - e_{k_1}) \cdots \phi_{k_n}(x - e_{k_1} - \ldots - e_{k_{n-1}}) \]

is independent of the path between 0 and $x$.

- The inverse of the product is called the balance function

\[ \Phi(x) = \frac{1}{\phi_{k_1}(x)\phi_{k_2}(x - e_{k_1}) \cdots \phi_{k_n}(x - e_{k_1} - \ldots - e_{k_{n-1}})} \]
Balance function (continued . . .)

- Any positive function $\Phi(x)$ will do as a balance function and defines a balanced allocation.

- Given the balance function, the capacity allocated to class $i$ is

$$
\phi_i(x) = \frac{\Phi(x - e_i)}{\Phi(x)}, \quad \forall i, \ x_i > 0.
$$

- An acceptable allocation has to satisfy the capacity constraints

$$
\sum_{i \in F_l} \phi_i(x) \leq C_l \Rightarrow \sum_{i \in F_l} \frac{\Phi(x-e_i)}{\Phi(x)} \leq C_l \Rightarrow \Phi(x) \geq \frac{1}{C_l} \sum_{i \in F_l} \Phi(x - e_i), \ \forall l
$$

- Even then there are an infinity of different balanced allocations.
Balanced fairness

- *Balanced fairness* refers to the balanced allocation with the property that in each state *at least one link is saturated*.

- Balanced fair allocation is *uniquely* defined by the recursion

\[
\Phi(x) = \max_l \left\{ \frac{1}{C_l} \sum_{k \in F_l} \Phi(x - e_k) \right\}
\]

...that is, \(\Phi(x)\) is reduced until a capacity constraint is encountered.

- Balanced fairness defines the most efficient balanced allocation (not necessarily Pareto efficient).
Balanced fairness: example of the basic recursion

- The recursion
  \[ \Phi(x) = \max_l \left\{ \frac{1}{C_{l_k}} \sum_{l \in r_k} \Phi(x - e_k) \right\} \]

- In the case of a 2-branch tree the maximum of these:
Balanced fairness: a trivial example

- A single link of capacity $C$ shared by two classes.

- The balance function
  \[
  \Phi(x) = \text{(number of paths)} \times \frac{1}{C(\text{path length})} = \left(\frac{x_1 + x_2}{x_1} \right) \frac{1}{Cx_1 + x_2}
  \]

- Capacity allocation: \[\phi_i(x) = \frac{\Phi(x-e_i)}{\Phi(x)} = \frac{x_i}{x_1 + x_2}C, \ i = 1, 2.\]

- Each flow is allocated the capacity $\frac{C}{x_1 + x_2}$: a single PS system.
How to determine the balance function: parking lot

- Parking lot topology

- Special case of a multi-level tree
How to determine the balance function (continued . . . )

\[
\Phi^{(1)}(x_1) = \frac{1}{C_1^{x_1}}
\]

\[
\Phi^{(2)}(x_1, x_2) = \sum_{y_1 \leq x_1} \left( y_1 + x_2 - 1 \right) \frac{\Phi^{(1)}(x_1 - y_1)}{C_2^{y_1 + x_2}}
\]
How to determine the balance function (continued . . . )

\[
\Phi^{(n)}(x_1, \ldots, x_n) = \sum_{y_1 \leq x_1, \ldots, y_{n-1} \leq x_{n-1}} \left( y_1 + \cdots + y_{n-1} + x_n - 1 \right) \frac{\Phi^{(n-1)}(x_1 - y_1, \ldots, x_{n-1} - y_{n-1})}{C_n^{y_1 + \cdots + y_{n-1} + x_n}}
\]
State distribution under balanced fairness

- In a dynamic setting where flows arrive and are transferred across the network until completed the network state $x$ is a random variable.

- The invariant measure (unnormalized distribution) is

$$\pi(x_1, \ldots, x_K) = \Phi(x_1, \ldots, x_K)\rho_1^{x_1} \cdots \rho_K^{x_K}$$

- $\rho_i = \lambda_i/\mu_i$ is the load of flow class $i$ ($1/\mu_i$ is the mean flow size),

- $\rho_i$ has the dimension of rate, e.g. kbit/s.
State distribution (continued . . .)

- If $p_{ij} = 0$ for $i \neq j$, balance property implies detailed balance and the form of the invariant measure is obvious but it holds also for the queueing network.

- The invariant measure is insensitive to any traffic characteristics.
  - Whittle networks are the only insensitive PS networks.
Insensitivity

- From exponential service times (flow sizes) to general distribution.

- Replace node $i$ by a set of sub-nodes $S_i$ such that each node $i \in S_i$ represents an exponential phase of a phase type service time distribution;
  
  - let $y_i$ be the number of customers in sub-node $i$, $\sum_{i \in S_i} y_i = x_i$,
  
  - total capacity $\phi_i(x)$ allocated to node $i$ is equally shared by all customers in the node no matter which phase (sub-node) they are in,
  
  - the capacity $\psi_i(y)$ allocated to sub-node $i$ is then $\psi_i(y) = \frac{y_i}{x_i} \phi_i(x)$. 
Insensitivity (continued . . .)

- This allocation is still balanced.

- If the $\phi_i(x)$ are balanced by $\Phi(x)$ then the $\psi_i(y)$ are balanced by $\Psi(y)$,

\[ \psi(y) = \prod_{i=1}^{K} \left( \frac{x_i}{y_{i}, i \in S_i} \right) \Phi(x) \]

- It is readily verified that $\psi_i(y) = \Psi(y - e_i)/\Psi(y)$.

- The system is another Whittle network with the invariant measure

\[ \chi(y) = \Psi(y) \prod_{i=1}^{K} \prod_{i \in S_i} \rho_i^y \]
Insensitivity (continued . . .)

- The aggregate invariant measure of the node states is obtained by summing over the states of the sub-nodes with the constraints $\sum_{\nu \in S_i} y_{\nu} = x_i$,

$$
\pi(x) = \Phi(x) \prod_{i=1}^{K} \sum_{y_{\nu}, \nu \in S_i} \left( x_i \right) \prod_{\nu \in S_i} \rho_i^{y_{\nu}} = \Phi(x) \prod_{i=1}^{K} \left( \sum_{\nu \in S_i} \rho_i \right)^{x_i} = \Phi(x) \prod_{i=1}^{K} \rho_i^{x_i}
$$

- This is the same as in the original network.

- The invariant measure is insensitive to the replacement of exponential service times by phase type distributions $\Rightarrow$ general insensitivity.

- Remark: the above is valid even if node-$i$ service time is not exponential: the aggregate invariant measure is insensitive to the division of any node $i$ into sub-nodes equally sharing the capacity $\phi_i(x)$. 
Throughput

- The most important performance measure for elastic traffic is the flow throughput, defined as \((\text{mean flow size})/(\text{mean flow response time})\)

\[
\gamma_i = \frac{1}{\mu_i} = \frac{\rho_i}{E[T_i]} = \frac{\rho_i}{E[x_i]}
\]

- The latter form follows from expanding the expression by \(\lambda_i\) and then applying Little’s theorem.

- A direct calculation using \(\chi(y)\) shows that if node \(i\) is divided into a set \(S_i\) of sub-nodes equally sharing the capacity \(\phi_i(x)\), all the sub-nodes have the same throughput

\[
\frac{\rho_i}{E[x_i]} = \frac{\rho_{i'}}{E[x_{i'}]} \quad (= \frac{\sum_{i \in S_i} \rho_i}{E[\sum_{i \in S_i} x_i]} = \frac{\rho_i}{E[x_i]} = \gamma_i) \quad \forall i, i' \in S_i
\]
Linearity of the conditional response time

- Denote by $\sigma_i$ the amount of service required at node $i$. Then,
  \[
  \gamma_i = \frac{s}{\mathbb{E}[T_i | \sigma_i = s]}
  \]

  \textit{Proof}: Replace node $i$ by a set of nodes equally sharing the capacity $\phi_i(x)$, each node corresponding to a different value of $\sigma_i$. All the sub-nodes, including the one with $\sigma_i = s$, have a throughput equal to $\gamma_i$.

- Conversely,
  \[
  \mathbb{E}[T_i | \sigma_i = s] = \frac{s}{\gamma_i}
  \]
Comparison with utility-based allocations

- Allocations considered so far are based on the notion of utility
  - the allocation is defined by the solution of an optimization problem

\[
\max \sum_{i} x_i U(\phi_i/x_i) \quad \text{subject to} \quad \sum_{i \in \mathcal{F}_l} \phi_i \leq C_l
\]

- Balanced fairness coincides with proportional fairness on homogeneous hypercubes.

\[
\phi_1(x) = \phi_2(x) = \frac{x_1 + x_2}{x_1 + x_2 + x_3 + x_4}, \quad \phi_3(x) = \phi_4(x) = \frac{x_3 + x_4}{x_1 + x_2 + x_3 + x_4}.
\]
Application to specific network topologies

- Utility-based allocations are always sensitive except for proportional fairness in homogeneous hypercubes.
  - For max-min fairness, performance is sensitive.

- Balanced fairness is compared with max-min fairness on different network topologies: lines, grids, hypercubes, trees, hypercycles.
  - User performance is determined by the mean flow throughput = mean flow size / mean flow duration.

- Balanced fairness provides a good approximation of max-min fairness!
Definition of some topologies

- A *hypercube* of dimension $K$ is a network made of $K$ classes of routes (called directions) such that the set of links is the set of intersections of $K$ routes of different classes.

- A *hypercycle* is a network made of $N$ routes such that the set of links is the set of intersections of $N - 1$ routes.
Homogeneous hypercubes

- $K$ directions; sets of routes in each direction $\mathcal{D}_1, \ldots, \mathcal{D}_K$.

- Contains as special cases homogeneous lines and grids.

- Balance function and capacity allocation

$$
\Phi(x) = \left( \sum_{i: r_i \in \mathcal{D}_1} x_i, \ldots, \sum_{i: r_i \in \mathcal{D}_K} x_i \right) \frac{1}{C \sum_i x_i},
\phi_j(x) = \frac{\Phi(x - e_j)}{\Phi(x)} = \frac{\sum_{i: r_i \in \mathcal{D}_j} x_i}{\sum_i x_i} C
$$
Heterogeneous lines

- Heterogeneous lines (minimum capacity as a unit of bandwidth):

\[
\Phi(x) = \sum_{y_1 + y_2 \leq x_3} \left( \frac{x_1 + y_1 - 1}{y_1} \right) \left( \frac{x_2 + y_2 - 1}{y_2} \right) \frac{1}{C_1^{x_1 + y_1} C_2^{x_2 + y_2}}
\]

![Diagram of heterogeneous lines with flow throughput and link load plots showing balanced fairness and max-min fairness](image)
Trees

- The balance function is known for a general tree.
- In particular, consider a two-branch tree:

\[
\Phi(x) = \sum_{y_1 < x_1} \left( y_1 + x_2 - 1 \right) \frac{1}{C_1^{x_1 - y_1}} + \sum_{y_2 < x_2} \left( y_2 + x_1 - 1 \right) \frac{1}{C_1^{x_2 - y_2}}
\]

\[\text{Flow throughput}\]

\[\text{Link load}\]

\[C_1 = 0.5, C_2 = 1\]
Cycles

- There is no closed expression for the balance function
  - but it can be computed recursively.
- Balanced fairness is not Pareto-efficient on hypercycles
  - but the performance is not deteriorated.
Some bounds for the throughput

- Bonald and Prouti`ere have shown that a store-and-forward network provides a simple lower bound for the throughput:
  - flows are thought to be sent as “packets”; the nodes are PS servers
  - inverse of the sum of the inverse throughputs \((C_l - \rho_l)^{-1}\)

- Parking lot provides another lower bound
  - all the capacity constraints outside the considered route are relaxed
  - the cross traffic streams are made less constrained than in reality

- Deterministic approximation gives an upper bound
  - all constraints outside the route are made as tight as stability allows
  - cross traffic streams become deterministic: subtract their load from the capacities on the main route; the bottleneck determines the throughput
Normalization constant

- An important role is played by the normalization constant

\[ G(\rho) = G(\rho_1, \ldots, \rho_K) = \sum_{x_1=0}^{\infty} \cdots \sum_{x_K=0}^{\infty} \Phi(x_1, \ldots, x_K) \rho_1^{x_1} \cdots \rho_K^{x_K} \]

- Generating function of the balance function,
  - contains the same information as \( \Phi(x) \).

- Performance measures can be derived from the normalization constant.

- Flow \( k \) throughput:

\[ \gamma_k = \frac{\rho_k}{E[x_k]} = \frac{G(\rho)}{\frac{\partial}{\partial \rho_k} G(\rho)} = \frac{1}{\frac{\partial}{\partial \rho_k} \log G(\rho)} \]
Recursive algorithm for the normalization constant

Definitions

- Let $\mathcal{I}_k = \{i_1, \ldots, i_k\}$, $1 \leq i_1 < \ldots < i_k \leq K$.

- Partial sets of states
  \[ \Omega_{\mathcal{I}_k} = \{ x : x_j > 0 \text{ if and only if } j \in \mathcal{I}_k \} \]

- $\mathcal{I}_0$ means the empty set $\emptyset$ and
  \[ \Omega_{\emptyset} = \{(0, \ldots, 0)\} \]
Recursive algorithm – state space decomposition

- The whole state space is decomposed as

\[
\Omega = \sum_{k=0}^{K} \sum I_k \Omega I_k.
\]

- For instance \( \Omega = \Omega_0 + \Omega_1 + \Omega_2 + \Omega_3 + \Omega_{1,2} + \Omega_{1,3} + \Omega_{2,3} + \Omega_{1,2,3} \)
Recursive algorithm – state space decomposition

- Partial sums over the sets $\Omega_{\mathcal{I}_k}$

$$
G_{\mathcal{I}_k} = \sum_{x \in \Omega_{\mathcal{I}_k}} \Phi(x_1, \ldots, x_n) \rho_1^{x_1} \cdots \rho_n^{x_n}
$$

- The normalization constant $G(\rho)$ is decomposed as

$$
G(\rho) = \sum_{k=0}^{K} \sum_{\mathcal{I}_k} G_{\mathcal{I}_k}(\rho)
$$
Recursion

- Assumption: for each $\mathcal{I}_k$ a given link or links are saturated in all the states of $\Omega_{\mathcal{I}_k}$ (true for lines, trees, ...).

- Under this assumption we can derive a recursion expressing $G_{\mathcal{I}_k}(\rho)$ in terms of the $G_{\mathcal{I}_{k-1}}(\rho)$.

- For a reasonable number of flow classes the number of different sets $\mathcal{I}_k$ is manageable ($2^k$).
Recursion (continued ...)

- Consider a 2-dimensional set $\Omega_{i,j}$:

- State $x$ in the set $\Omega_i$ contributes as a “source of recursion” an amount

$$\frac{\rho_j}{C} \cdot \frac{1}{1 - (\frac{\rho_i}{C} + \frac{\rho_j}{C})}$$

times its own measure $\pi(x)$. 
Recursion (continued . . .)

- The first factor comes from the “bridge”.

- The second factor from the infinite sum $S$ over the area in dark blue:

\[
\sum_{x_1=0}^{\infty} \cdots \sum_{x_k=0}^{\infty} \left( \frac{x_1 + \cdots + x_k}{x_1, \ldots, x_k} \right) y_{i_1}^{x_1} \cdots y_{i_k}^{x_k} = \sum_{x=0}^{\infty} \sum_{x_1 + \cdots + x_k=x}^{\infty} \left( \frac{x_1 + \cdots + x_k}{x_1, \ldots, x_k} \right) y_{i_1}^{x_1} \cdots y_{i_k}^{x_k}
\]

\[
= \sum_{x=0}^{\infty} (y_{i_1} + \cdots + y_{i_k})^x
\]

\[
= \frac{1}{1 - (y_{i_1} + \cdots + y_{i_k})}
\]

- Alternatively: $S = 1 + y_{i_1}S + \cdots + y_{i_k}S$
Recursion (continued . . .)

- Since the contribution is the same for each state in $\Omega_i$, in total the set $\Omega_i$ gives to $G_{i,j}(\rho)$ a contribution

\[
\frac{\rho_j}{C} \cdot \frac{1}{1 - (\frac{\rho_i}{C} + \frac{\rho_j}{C})} \cdot G_i(\rho)
\]

- Similar contribution comes from the set $\Omega_j$ and the sought for recursion is

\[
G_{i,j}(\rho) = \frac{\rho_j G_i(\rho) + \rho_i G_j(\rho)}{C - (\rho_i + \rho_j)}
\]
General recursion for the normalization constant

- In general

\[
G_I(\rho) = \frac{\sum_{j \in I'} \rho_j G_{I \setminus j}(\rho)}{C_{\sigma(I)} - \sum_{j \in I'} \rho_j}
\]

\(\sigma(I)\) is the link which is saturated when \(x_j > 0\) iff \(j \in I\)

\(I' \subset I\) stands for those classes \(j \in I\) for which \(\sigma(I) \in r_j\)

- If \(\sigma(I)\) is not unique any of the saturated links can be used as the basis for the recursion.
Recursion for the throughput

- Direct recursion for the throughput:

\[ \gamma_i = \frac{G(\rho)}{\partial \rho_i G(\rho)} \]

- The denominator is decomposed as

\[ \frac{\partial}{\partial \rho_i} G(\rho) = \sum_{k=0}^{K} \sum_{\mathcal{I}_k} \frac{\partial}{\partial \rho_i} G_{\mathcal{I}_k}(\rho) \equiv \sum_{k=0}^{K} \sum_{\mathcal{I}_k} H^{(i)}_{\mathcal{I}_k}(\rho) \]

\[ H^{(i)}_{\mathcal{I}_k}(\rho) = \frac{1_{i \in \mathcal{I}'_k}(G_{\mathcal{I}_k}(\rho) + G_{\mathcal{I}'_k}(\rho)) + \sum_{j \in \mathcal{I}'_k} \rho_j H^{(i)}_{\mathcal{I}'_k}(\rho)}{C_{\sigma(\mathcal{I}_k)} - \sum_{j \in \mathcal{I}'_k} \rho_j} \]
Examples: Inhomogeneous line

- Redefine $\Omega_i = \{x : x_j = 0 \text{ for } j > i\}$
- Then the recursion reads

$$
\begin{align*}
G_0(\rho) &= \frac{1}{1-\rho_0 C_i} \\
G_i(\rho) &= \left(1 + \frac{\rho_i}{1-\rho_0+\rho_i} \right) \cdot G_{i-1}(\rho) = \frac{1-\rho_0}{1-\rho_0+\rho_i} \cdot G_{i-1}(\rho)
\end{align*}
$$

$$
G(\rho) = \frac{1}{1-\rho_0 C_i} \cdot \frac{1-\rho_0 C_1}{1-\rho_0+\rho_1} \cdots \frac{1-\rho_0 C_n}{1-\rho_0+\rho_n}
$$

- The throughput

$$
\gamma_0 = \left(\frac{1}{C-\rho_0} + \sum_{l=1}^{n} \left(\frac{1}{C_l-\rho_l-\rho_0} - \frac{1}{C_l-\rho_0} \right) \right)^{-1}
$$
Examples: Parking lot

- The recursion reads

\[
\begin{align*}
G_1(\rho) &= \frac{1}{1 - \frac{\rho_1}{C_1}} \\
G_i(\rho) &= \left(1 + \frac{\rho_i}{\frac{C_i}{1 - \rho_1 + \cdots + \rho_i}}\right) \cdot G_{i-1}(\rho) = \frac{1 - \frac{\rho_1 + \cdots + \rho_{i-1}}{C_i}}{1 - \frac{\rho_1 + \cdots + \rho_i}{C_i}} \cdot G_{i-1}(\rho)
\end{align*}
\]

- By denoting the link \( i \) load by \( R_i = \sum_{j=1}^{i} \rho_j \)

\[
G(\rho) = \frac{1}{1 - \frac{R_1}{C_1}} \cdot \frac{1 - \frac{R_1}{C_2}}{1 - \frac{R_2}{C_2}} \cdots \frac{1 - \frac{R_{n-1}}{C_n}}{1 - \frac{R_n}{C_n}}
\]

- The throughput

\[
\gamma_i = \left(\frac{1}{C_i - R_i} + \sum_{l=i+1}^{n} \left(\frac{1}{C_l - R_l} - \frac{1}{C_l - R_{l-1}}\right)\right)^{-1}
\]
Examples: 2-branch tree

\[
\begin{align*}
G_0(\rho) & = 1 \\
G_1(\rho) & = \frac{\rho_1}{C_1} \cdot G_0(\rho) \left(1 - \frac{\rho_1}{C_1}\right) \\
G_2(\rho) & = \frac{\rho_2}{C_2} \cdot G_0(\rho) \left(1 - \frac{\rho_2}{C_2}\right) \\
G_{1,2}(\rho) & = \frac{\rho_1}{C_0} \cdot G_2(\rho) + \frac{\rho_2}{C_0} \cdot G_1(\rho) \left(1 - \frac{\rho_1 + \rho_2}{C_0}\right) \\
G(\rho) & = \frac{1}{1 - \frac{\rho_1 + \rho_2}{C_0}} \cdot \left(1 - \frac{\rho_1}{C_0} + \frac{1 - \rho_1}{C_1} + \frac{1 - \rho_2}{C_2} - 1\right)
\end{align*}
\]

\[
= \frac{1 - \frac{\rho_1 + \rho_2}{C_0} + \frac{\rho_1 \rho_2}{C_1 C_2} (\frac{C_1 + C_2}{C_0} - 1)}{(1 - \frac{\rho_1 + \rho_2}{C_0})(1 - \frac{\rho_1}{C_1})(1 - \frac{\rho_2}{C_2})}
\]
Examples: 3-branch tree

Case 1. All pairs add to more than 1, e.g., \( C_1 = 0.9, C_2 = 0.6, C_3 = 0.5 \)

\[
\begin{align*}
G_0(\rho) &= 1 \\
G_1(\rho) &= \frac{\frac{\rho_1}{C_1} \cdot G_0(\rho)}{1 - \frac{\rho_1}{C_1}} \\
G_2(\rho) &= \frac{\frac{\rho_2}{C_2} \cdot G_0(\rho)}{1 - \frac{\rho_2}{C_2}} \\
G_3(\rho) &= \frac{\frac{\rho_3}{C_3} \cdot G_0(\rho)}{1 - \frac{\rho_3}{C_3}} \\
G_{1,2}(\rho) &= \frac{\frac{\rho_1}{C_0} \cdot G_2(\rho) + \frac{\rho_2}{C_0} \cdot G_1(\rho)}{1 - \frac{\rho_1 + \rho_2}{C_0}} \\
G_{1,3}(\rho) &= \frac{\frac{\rho_1}{C_0} \cdot G_3(\rho) + \frac{\rho_3}{C_0} \cdot G_1(\rho)}{1 - \frac{\rho_1 + \rho_3}{C_0}} \\
G_{2,3}(\rho) &= \frac{\frac{\rho_2}{C_0} \cdot G_3(\rho) + \frac{\rho_3}{C_0} \cdot G_2(\rho)}{1 - \frac{\rho_2 + \rho_3}{C_0}} \\
G_{1,2,3}(\rho) &= \frac{\frac{\rho_1}{C_0} \cdot G_{2,3}(\rho) + \frac{\rho_2}{C_0} \cdot G_{1,3}(\rho) + \frac{\rho_3}{C_0} \cdot G_{1,2}(\rho)}{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0}}
\end{align*}
\]
Examples: 3-branch tree

Case 1. All pairs add to more than 1, e.g., $C_1 = 0.9$, $C_2 = 0.6$, $C_3 = 0.5$

The sum of these can be simplified to

$$G(\rho) = \frac{1}{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0}} \left( \frac{1 - \frac{\rho_1}{C_0}}{1 - \frac{\rho_1}{C_1}} + \frac{1 - \frac{\rho_2}{C_0}}{1 - \frac{\rho_2}{C_2}} + \frac{1 - \frac{\rho_3}{C_0}}{1 - \frac{\rho_3}{C_3}} - 2 \right)$$

$$= 1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0} + \frac{\rho_1}{C_1} \frac{\rho_2}{C_2} \left( \frac{C_1 + C_2}{C_0} - 1 \right) + \frac{\rho_2}{C_2} \frac{\rho_3}{C_3} \left( \frac{C_2 + C_3}{C_0} - 1 \right) + \frac{\rho_3}{C_3} \frac{\rho_1}{C_1} \left( \frac{C_3 + C_1}{C_0} - 1 \right) + \frac{\rho_1}{C_1} \frac{\rho_2}{C_2} \frac{\rho_3}{C_3} \left( 2 - \frac{C_1 + C_2 + C_3}{C_0} \right)$$

with an obvious generalization to any number of branches.
Examples: 3-branch tree

Case 2. Two pairs add to more than 1, e.g., $C_1 = 0.9$, $C_2 = 0.6$, $C_3 = 0.2$

\[
\begin{align*}
G_0(\rho) &= 1 \\
G_1(\rho) &= \frac{\rho_1}{c_1} \cdot G_0(\rho) \\
G_2(\rho) &= \frac{\rho_2}{c_2} \cdot G_0(\rho) \\
G_3(\rho) &= \frac{\rho_3}{c_3} \cdot G_0(\rho) \\
G_{1,2}(\rho) &= \frac{\rho_1}{c_0} \cdot G_2(\rho) + \frac{\rho_2}{c_0} \cdot G_1(\rho) \\
G_{1,3}(\rho) &= \frac{\rho_1}{c_0} \cdot G_3(\rho) + \frac{\rho_3}{c_0} \cdot G_1(\rho) \\
G_{2,3}(\rho) &= \frac{\rho_2}{c_0} \cdot G_3(\rho) = \frac{\rho_3}{c_0} \cdot G_2(\rho) \\
G_{1,2,3}(\rho) &= \frac{\rho_1}{c_0} \cdot G_{2,3}(\rho) + \frac{\rho_2}{c_0} \cdot G_{1,3}(\rho) + \frac{\rho_3}{c_0} \cdot G_{1,2}(\rho)
\end{align*}
\]
Examples: 3-branch tree

Case 2. Two pairs add to more than 1, e.g., $C_1 = 0.9$, $C_2 = 0.6$, $C_3 = 0.2$

After simplification

$$G(\rho) = \frac{1}{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0}} \left( \frac{1 - \frac{\rho_1}{C_0}}{1 - \frac{\rho_1}{C_1}} + \frac{1 - \frac{\rho_2 + \rho_3}{C_0}}{1 - \frac{\rho_2}{C_2} \left(1 - \frac{\rho_3}{C_3}\right)} - 1 \right)$$

$$= \frac{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0} + \frac{\rho_1}{C_1} \frac{\rho_2}{C_2} \left(\frac{C_1 + C_2}{C_0} - 1\right) + \frac{\rho_1}{C_1} \frac{\rho_3}{C_3} \left(\frac{C_1 + C_3}{C_0} - 1\right) + \frac{\rho_1}{C_1} \frac{\rho_2}{C_2} \frac{\rho_3}{C_3} \left(1 - \frac{C_1}{C_0}\right)}{(1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0})(1 - \frac{\rho_1}{C_1})(1 - \frac{\rho_2}{C_2})(1 - \frac{\rho_3}{C_3})}$$
Examples: 3-branch tree

Case 3. One pair adds to more than 1, e.g., $C_1 = 0.7$, $C_2 = 0.4$, $C_3 = 0.2$

\[
\begin{align*}
G_0(\rho) &= 1 \\
G_1(\rho) &= \frac{\rho_1 C_1}{C_1 - \rho_1} \cdot G_0(\rho) \\
G_2(\rho) &= \frac{\rho_2 C_2}{C_2 - \rho_2} \cdot G_0(\rho) \\
G_3(\rho) &= \frac{\rho_3 C_3}{C_3 - \rho_3} \cdot G_0(\rho) \\
G_{1,2}(\rho) &= \frac{\rho_1 C_2}{C_2 - \rho_1} \cdot G_2(\rho) + \frac{\rho_2 C_2}{C_2 - \rho_2} \cdot G_1(\rho) \\
G_{1,3}(\rho) &= \frac{\rho_1 C_3}{C_3 - \rho_1} \cdot G_3(\rho) = \frac{\rho_3 C_3}{C_3 - \rho_3} \cdot G_1(\rho) \\
G_{2,3}(\rho) &= \frac{\rho_2 C_3}{C_3 - \rho_2} \cdot G_3(\rho) = \frac{\rho_3 C_3}{C_3 - \rho_3} \cdot G_2(\rho) \\
G_{1,2,3}(\rho) &= \frac{\rho_1 C_2}{C_2 - \rho_1} \cdot G_2,3(\rho) + \frac{\rho_2 C_2}{C_2 - \rho_2} \cdot G_{1,3}(\rho) + \frac{\rho_3 C_2}{C_2 - \rho_3} \cdot G_{1,2}(\rho) \\
\end{align*}
\]
Examples: 3-branch tree

Case 3. One pair adds to more than 1, e.g., $C_1 = 0.7$, $C_2 = 0.4$, $C_3 = 0.2$

After simplification

\[
G(\rho) = \frac{1}{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0}} \cdot \frac{1}{1 - \frac{\rho_3}{C_3}} \left( \frac{1 - \frac{\rho_1 + \rho_3}{C_0}}{1 - \frac{\rho_1}{C_1}} + \frac{1 - \frac{\rho_2 + \rho_3}{C_0}}{1 - \frac{\rho_2}{C_2}} - (1 - \frac{\rho_3}{C_0}) \right)
\]

\[
= \frac{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0} + \frac{\rho_1 \rho_2}{C_1 C_2 (C_1 + C_2 - 1)} + \frac{\rho_1 \rho_2 \rho_3}{C_1 C_2 C_3}}{(1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0})(1 - \frac{\rho_1}{C_1})(1 - \frac{\rho_2}{C_2})(1 - \frac{\rho_3}{C_3})}
\]
Examples: 3-branch tree

Case 4. None of the pairs adds to more than 1, e.g., \( C_1 = 0.5 \), \( C_2 = 0.4 \), \( C_3 = 0.2 \)

\[
\begin{align*}
G_0(\rho) &= 1 \\
G_1(\rho) &= \frac{\rho_1}{C_1} G_0(\rho) \frac{1}{1 - \frac{\rho_1}{C_1}} \\
G_2(\rho) &= \frac{\rho_2}{C_2} G_0(\rho) \frac{1}{1 - \frac{\rho_2}{C_2}} \\
G_3(\rho) &= \frac{\rho_3}{C_3} G_0(\rho) \frac{1}{1 - \frac{\rho_3}{C_3}} \\
G_{1,2}(\rho) &= \frac{\rho_1}{C_1} G_2(\rho) \frac{1 - \frac{\rho_1}{C_1}}{1 - \frac{\rho_2}{C_2}} \\
G_{1,3}(\rho) &= \frac{\rho_1}{C_1} G_3(\rho) \frac{1 - \frac{\rho_1}{C_1}}{1 - \frac{\rho_3}{C_3}} \\
G_{2,3}(\rho) &= \frac{\rho_2}{C_2} G_3(\rho) \frac{1 - \frac{\rho_2}{C_2}}{1 - \frac{\rho_3}{C_3}} \\
G_{1,2,3}(\rho) &= \frac{\rho_1}{C_1} G_{2,3}(\rho) + \frac{\rho_2}{C_2} G_{1,3}(\rho) + \frac{\rho_3}{C_3} G_{1,2}(\rho) \frac{1 - \rho_1 + \rho_2 + \rho_3}{c_0}
\end{align*}
\]
Examples: 3-branch tree

Case 4. None of the pairs adds to more than 1, e.g., $C_1 = 0.5$, $C_2 = 0.4$, $C_3 = 0.2$

After simplification

$$G(\rho) = \frac{1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0}}{(1 - \frac{\rho_1 + \rho_2 + \rho_3}{C_0})(1 - \frac{\rho_1}{C_1})(1 - \frac{\rho_2}{C_2})(1 - \frac{\rho_3}{C_3})} + \frac{\rho_1 \rho_2 \rho_3}{C_1 C_2 C_3} \left( \frac{C_1 + C_2 + C_3}{C_0} - 1 \right)$$
Examples: Homogeneous $n$-branch tree

- All branches have equal capacity and equal load, $C_i = C$, $\rho_i = \rho$. The capacity of the trunk is $C_0$.

- Denote by $m$ the largest integer such that $m \times C < C_0$.

- Applying the recursion on any set of states where given $k$ branches are saturated we have

$$G_k(\rho) = \left( \frac{\rho}{C - \rho} \right)^k, \quad k = 0, \ldots, m$$

- There are \( \binom{n}{k} \) sets of this type for any $k$. 
Examples: Homogeneous $n$-branch tree (continued . . .)

- Applying the recursion to the set $\Omega$ where the trunk is saturated, we get

$$\tilde{G}(\rho) = \frac{(n-m)\rho}{C_0 - n\rho} \binom{n}{m} G_m$$

- Thus

$$G(\rho) = \sum_{k=0}^{m} \binom{n}{k} \left( \frac{\rho}{C - \rho} \right)^k + \frac{(n-m)\rho}{C_0 - n\rho} \binom{n}{m} \left( \frac{\rho}{C - \rho} \right)^m$$
Examples: Homogeneous $n$-branch tree (continued . . . )

- $n = 20$ branches, $m = 1, \ldots, 20$
Examples: 3-level tree

With the capacities given in the figure one infers

\[
I' = \{1\} \text{ or } \{2\} \quad \text{with links 1 and 2 saturated} \quad \text{for } I = \{1, 2\}
\]
\[
I' = \{1, 3\} \quad \text{with link 0 saturated} \quad \text{for } I = \{1, 3\}
\]
\[
I' = \{2, 3\} \quad \text{with link 4 saturated} \quad \text{for } I = \{2, 3\}
\]
\[
I' = \{1, 2, 3\} \quad \text{with link 0 saturated} \quad \text{for } I = \{1, 2, 3\}
\]
Examples: 3-level tree (continued . . .)

\[
\begin{align*}
G_0(\rho) &= 1 \\
G_1(\rho) &= \frac{\rho_1}{c_1} \cdot G_0(\rho) \\
G_2(\rho) &= \frac{\rho_2}{c_2} \cdot G_0(\rho) \\
G_3(\rho) &= \frac{\rho_3}{c_3} \cdot G_0(\rho) \\
G_{1,2}(\rho) &= \frac{\rho_1}{c_0} \cdot G_2(\rho) + \frac{\rho_3}{c_0} \cdot G_1(\rho) \\
G_{1,3}(\rho) &= \frac{\rho_2}{c_0} \cdot G_3(\rho) + \frac{\rho_3}{c_0} \cdot G_1(\rho) \\
G_{2,3}(\rho) &= \frac{\rho_2}{c_0} \cdot G_3(\rho) + \frac{\rho_3}{c_0} \cdot G_2(\rho) \\
G_{1,2,3}(\rho) &= \frac{\rho_1}{c_0} \cdot G_{2,3}(\rho) + \frac{\rho_2}{c_0} \cdot G_{1,3}(\rho) + \frac{\rho_3}{c_0} \cdot G_{1,2}(\rho)
\end{align*}
\]
Examples: 3-level tree (continued . . .)

For the capacities given in the figure one gets with the aid of Mathematica

\[
G(\rho) = \frac{720 - 120 \rho_1 - 264 (\rho_2 + \rho_3) + 24 (\rho_1 \rho_2 + \rho_2^2 + \rho_3^2) + 3 \rho_2 \rho_3 (22 - (\rho_2 + \rho_3)) + \rho_1 \rho_3 (34 + (2 + \rho_2)(\rho_2 - \rho_3))}{(3 - \rho_1)(2 - \rho_2)(4 - \rho_3)(5 - \rho_2 - \rho_3)(6 - \rho_1 - \rho_2 - \rho_3)}
\]

As an example of the throughputs, with \(\rho_2 = \rho_3 = 1\), one gets

\[
\begin{align*}
\gamma_1 &= \frac{(4 - \rho_1)(3 - \rho_1)(75 - 16 \rho_1)}{333 - 150 \rho_1 + 16 \rho_1^2} \\
\gamma_2 &= \frac{12 (4 - \rho_1)(75 - 16 \rho_1)}{3792 - 1663 \rho_1 + 187 \rho_1^2} \\
\gamma_3 &= \frac{12 (4 - \rho_1)(75 - 16 \rho_1)}{1392 - 479 \rho_1 + 41 \rho_1^2}
\end{align*}
\]
Examples: A larger tree

- We study the throughput of flow 10 going through links 1, 3, 6, and 10 as a function of its own load $\rho_{10}$.

- The other classes are assumed to have fixed loads as follows: $\rho_1 = \rho_2 = \rho_4 = \rho_6 = \rho_7 = 2$ and $\rho_5 = \rho_8 = \rho_9 = 1$.

- With these loads all the four links on the route of flow 10 have the average residual capacity of 3 units.
Examples: A larger tree (continued . . . )

- As before, one derives the normalization constant

\[
G(\rho_{10}) = \frac{6(5-\rho_{10})(951-411\rho_{10}+46\rho_{10}^2)}{(3-\rho_{10})^4}
\]

- Comparison of the exact throughput with simple bounds:

From bottom up: store-and-forward, parking lot, exact, deterministic.
Limited access rates

- A single link with capacity $C$ offered flows from $K$ different classes.

- All flows in each class $i$ are limited by an access rate $a_i$.

- The capacity constraints must then be supplemented by the requirement $\phi_i(x) \leq x_i a_i$ for all $x$ and $i$ and the basic recursion has to be modified accordingly.

- As long as the sum of the access rates of the active flows is less than $C$ each flow is allocated a rate equal to its access rate. Only when the sum exceeds $C$ becomes the capacity sharing effective.
Limited access rates (continued . . .)

- For convenience think in terms of integer valued rates $a_i$ and capacity $C$.

- Denote $\Omega_{\emptyset} = \{ x : x \cdot a \leq C \}$ and $G_{\emptyset} = \sum_{x \in \Omega_{\emptyset}} \pi(x)$.

- The iman-Roberts recursion can be used to obtain

$$G_{\emptyset} = \sum_{c=0}^{C} g(c), \quad g(c) = \sum_{k=0}^{K} \frac{\rho_k}{c} g(c - a_k), \quad c = 1, 2, \ldots$$

with $g(1) = 1$ and $g(c) = 0$ for $c < 0$. 
Limited access rates (continued . . .)

- In addition it gives the measures of the “blocking sets” (referring to a virtual multi-bitrate system, where the flows are inelastic having fixed bandwidth requirements of $a_k$),

\[ B_k = \sum_{c=C-a_k+1}^{C} g(c) \]

- The blocking sets form the boundary between inelastic and elastic capacity sharing:
Limited access rates (continued . . .)

• As before, each state $x$ in the blocking set of class $i$ contributes as a “source of recursion” an amount

$$\frac{\rho_i}{C} \cdot \frac{1}{1 - \frac{\rho}{C}} \cdot \pi(x)$$

where $\rho = \sum_k \rho_k$ is the total load.

• Thus, the normalization constant is

$$G = \sum_{c=0}^C g(c) + \sum_{k=1}^K \frac{B_k \rho_k}{C - \rho}$$
Limited access rates (continued . . .)

- A similar idea can be applied also in the case of a tree with access rates.

- The algorithm becomes, however, more complicated.

\[
G = \sum_{S \in \Sigma(\mathcal{T})} G_S
\]

where \( S \) is a saturation set and \( \Sigma(\mathcal{T}) \) is the set of all feasible saturation sets in tree \( \mathcal{T} \).
Limited access rates (continued ...)

- General reduction:

\[
G_S[T] = G_{\emptyset}[T \setminus \{T_l\}] \prod_{l \in S} G_l[T_l]
\]

- Recursion for a subtree \(T_l\) with saturated root \(l\)

\[
G_l[T_l] = \frac{\sum_{S \in \Sigma(T_l)} \prod_{j \in S} G_j[T_j] \sum_{i \in F_l \cup \bigcup_{j \in S} F_j} \rho_i G_{\emptyset}|_{i,l}[T_l \setminus \{T_j\}] - \sum_{i \in F_l} \rho_i C_l}{C_l - \sum_{i \in F_l} \rho_i}
\]

- Partial class-\(k\) blocking measure “due to the root link \(l\)”

\[
G_{\emptyset}|_{l}[T_l(C)] = G_{\emptyset}[T_l(C)] - G_{\emptyset}[T_l(C - a_i r_i[T_l])] - \sum_{j \in R_i[T_l]} G_{\emptyset}[T_l(C) \setminus T_j(C)] \cdot G_{\emptyset}|_{j}[T_j(C)]
\]
Limited access rates (continued . . .)

Above we have used the following notation:

\[
\begin{align*}
T & = \text{a tree (set of links; a link is indexed by the set of flows going through it)} \\
T(C) & = \text{a tree, emphasizing the capacities of its links (} C \text{ is the capacity vector)} \\
T_l & = \text{subtree of } T \text{ having link } l \text{ as the root (exception: the third bullet)} \\
G_S[T] & = \text{state sum over the state space } \Omega_S \text{ of tree } T \text{ with saturation set } S \\
G_l[T_l] & = \text{as the previous one, the saturation set comprising the root of tree } T_l \text{ only} \\
G_{\emptyset,l}[T_l] & = \text{state sum over the border set } \Omega_{\emptyset,l} \text{ of tree } T_l \\
\mathcal{R}_i[T_l] & = \text{route of flow class } i \text{ in tree } T_l \text{ excluding the root } l \\
r_T^i & = \text{a vector of the same form as the capacity vector of tree } T, \text{ with 1 in each entry corresponding to a link used by route } \mathcal{R}_i[T_l] \text{ and 0 elsewhere} \\
T \setminus T_l & = \text{tree remaining when subtree } T_l \text{ is detached from tree } T \text{ and the capacity } C_l \text{ is subtracted from the capacities of the links on the route from } l \text{ to the root of } T \\
T \setminus \bigcup_{l \in S} T_l & = \text{the above subtraction operation repeated over all subtrees } T_l \text{ with } l \in S
\end{align*}
\]
Limited access rates (continued . . .)

Example

- The graph shows the throughput of each flow class as a function of its own load assuming all the other loads equalling 1.
Closed networks

- Reducible routing: the nodes are partitioned into $K$ subsets of nodes such that routing is irreducible on each subset $c_k$. $M_k$ flows in the routing group $k$.

- Node $i_k$ is the “source” of set $c_k$: the capacity allocation to this node, $\psi_k(x_{i_k})$, is a function of $x_{i_k}$ only. Its load is denoted $\rho_k$.

- The arrival frequencies $\lambda_i$ to different nodes are determined up to a multiplicative constant by

$$\lambda_i = \sum_{j \in c_k} \lambda_j p_{ji}, \quad i \in c_k.$$
Closed networks (continued)

- The invariant measure is

\[
\pi(x) = \begin{cases} 
\Phi(x') \prod_{k=1}^{K} \psi_k(x_{i_k}) \prod_{i=1}^{N} \rho_i^{x_i}, & \text{for } \sum_{i \in c_k} x_i = M_k, \forall k \\
0, & \text{otherwise}
\end{cases}
\]

where \( x' \) is the state vector excluding the source nodes, \( \Phi(x') \) is the balance function for those nodes, and \( \psi_k(n) = 1/\psi_k(1) \cdots \psi_k(n) \).

- The state sum with \( M_1, \ldots, M_K \) flows in each class is denoted

\[
G_{M_1, \ldots, M_K}(\rho) = \sum_{\sum_{i \in c_k} x_i = M_k, \forall k} \pi(x)
\]
Closed networks (continued)

- Throughput of node $i$

$$\gamma_i = \frac{G_{M_1, \ldots, M_{k-1}, \ldots, M_K}(\rho)}{\frac{\partial}{\partial \rho_i} G_{M_1, \ldots, M_K}(\rho)}$$

- Generating function

$$G(z_1, \ldots, z_K; \rho) = \sum_{M_1, \ldots, M_K} G_{M_1, \ldots, M_K}(\rho) z_1^{M_1} \cdots z_K^{M_K}$$

$$G(\tilde{\rho}') \prod_{k=1}^{K} H_k(\tilde{\varrho}_k), \quad \tilde{\rho}_i = \rho_i \prod_{k : i \in c_k} z_k$$
Closed networks (continued)

Example: One PS-link, one thinking stage

- The capacity of the link is denoted by $C$.

- The thinking stage is described as an infinite system of $c$-capacity servers.

- The generating function is

$$G(z; \rho, \varrho) = \frac{e^{z\varrho/c}}{1 - z\rho/C}$$
Closed networks (example continued)

- Developing the generating function in Taylor series in $z$ we can identify the $G_M(\rho, \varrho)$ and calculate the mean throughput on the link:

$$\gamma_M = \begin{cases} 
C, & M = 1 \\
\frac{c+C}{2c+C} C, & M = 2 \\
\frac{2c^2+2cC+C^2}{6c^2+4cC+C^2} C, & M = 3 \\
\frac{6c^3+6c^2C+3cC^2+C^3}{24c^3+18c^2C+6cC^2+C^3} C, & M = 4 \\
\frac{24c^4+24c^3C+12c^2C^2+4cC^3+C^4}{120c^4+96c^3C+36c^2C^2+8cC^3+C^4} C, & M = 5
\end{cases}$$

- For $c/C \to 0$ we have $\gamma_M \to C$. For $c/C \to \infty$ we have $\gamma_M \to C/M$. 
State-dependent arrival rates and routing

- Extension to Processor Sharing networks with state dependent capacities, routing and arrival rates $\phi_i(x)$, $p_{ij}(x)$, $\nu_i(x)$.

- Assuming that for any state $x$ the following traffic equations have a unique solution for the $\lambda_i(x)$,

$$
\lambda_i(x) = \nu_i(x) + \sum_j p_{ij}(T^j x) \lambda_j(x) \quad \text{(where } T^j x \equiv x + e_j) \n$$

- Define the traffic intensity at node $i$ in state $x$: $\rho_i(x) = \lambda_i(x) / \mu_i$
State-dependent arrival rates and routing (continued)

- Denote \( \psi_i(x) = \frac{\rho_i(T_ix)}{\phi_i(x)} \) (where \( T_ix = x - e_i \))

- Consider a PS network with state-dependent routing and arrival rates. The corresponding invariant measures are insensitive to the service time distributions at each node if and only if the \( \psi_i(x) \) are balanced. In this case, an invariant measure is

\[
\pi(x) = \psi_{i_1}(x) \psi_{i_2}(T_{i_1}x) \cdots \psi_{i_n}(T_{i_n} \cdots T_{i_1}x)
\]

- If the allocations \( \phi_i(x) \) are balanced then the \( \psi_i(x) \) are balanced iff the \( \lambda_i(x) \) are balanced (i.e., \( \forall x : \lambda_i(x) = u(T^i x) / u(x) \) for some \( u(x) \)).
Parallel links with adaptive routing

- $L$ parallel links
- Link $i$ can serve at most $N_i$ flows
- Adaptive stochastic routing $\nu_i(x)$
- A flow is rejected only if all links are saturated (i.e., for all other states $x$: $\sum_i \nu_i(x) = \lambda$).
- Insensitivity iff $\nu_i(x) = \lambda \frac{N_i - x_i}{\sum_{l=1}^L (N_l - x_l)}$
Adaptive routing (continued . . .)

- Stationary distribution and blocking probability

\[
\pi(x) = \left( \frac{\sum_{l=1}^{L} (N_l - x_l)}{N_1 - x_1, \ldots, N_L - x_L} \right) \prod_{i=1}^{L} \left( \frac{\rho}{C_i} \right)^{x_i}
\]

\[
P = \frac{\pi(N_1, \ldots, N_L)}{\sum_{x_1=0}^{N_1} \ldots \sum_{x_L=0}^{N_L} \pi(x_1, \ldots, x_L)}
\]
Adaptive routing (continued ...)

- Where did the routing formula come from?

- The “shared resource” is the available traffic to be routed on different links
  - “mirror image” of the link sharing

\[
u(x) = \left( \sum_{l=1}^{L} (N_l - x_l) \right) \frac{1}{\lambda \sum_l (N_l - x_l)} \]

\[
\begin{array}{c}
N_j-x_j \\
1/\lambda \\
1/\lambda \\
N_j-x_j \\
x_j \\
x_i \\
N_i-x_i \\
N_j
\end{array}
\]
No efficient adaptive insensitive non-blocking routing

- Using Hausdorff’s theorem M. Jonckheere showed that without any limit on the number of flows on links the general solution of the recursion is

\[ u(x_1, x_2) = \int z^{x_1}(1 - z)^{x_2} d\chi(z) \]

where \( \chi(z) \) is an arbitrary measure on \([0, 1]\).

- For instance if \( d\chi(z)/dz = \delta(z - p) \) then \( u(x_1, x_2) = p^{x_1}(1 - p)^{x_2} \), i.e., static routing with \( \nu_1 = p\lambda \), \( \nu_2 = (1 - p)\lambda \).

- With the above representation Jonckheere proved that there is no solution such that \( \nu_1(x_1, 0) = u(x_1 + 1, 0)/u(x_1, 0) \to 0 \) when \( x_1 \to \infty \).
Open problems – future research

- Can balanced fairness be realized by a distributed algorithm?
- How close is balanced fairness to “classical” allocations?
- How sensitive are allocations that are not insensitive?
- New network topologies
- Background traffic in balanced routing
- Improved bounds