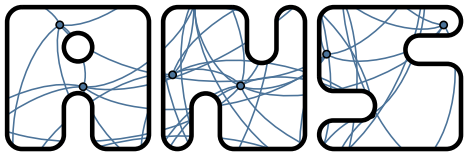


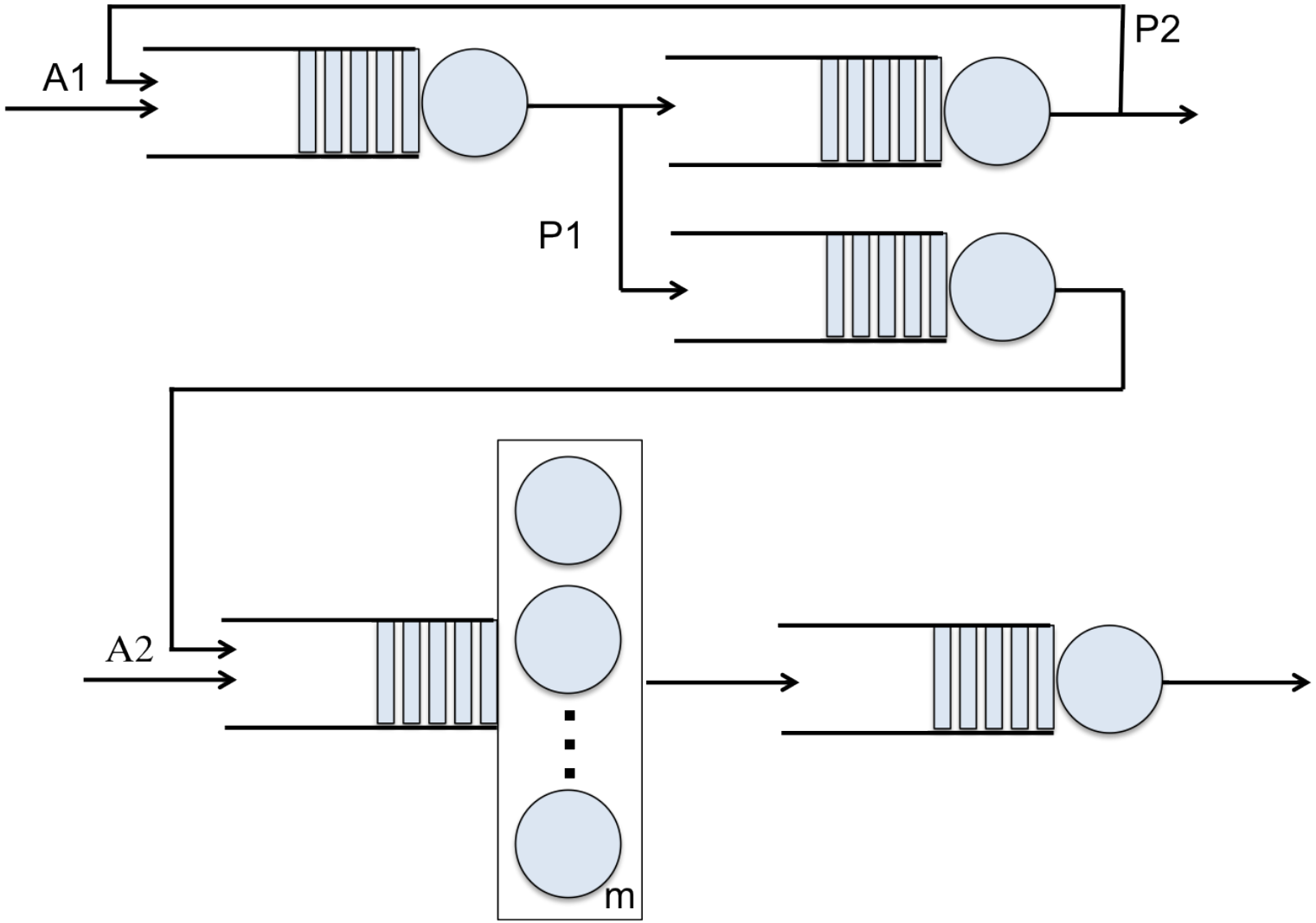
UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Queuing Networks

Renato Lo Cigno

Simulation and Performance Evaluation 2017-18

Moving between Queues



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Customers serviced from a queue can enter another queue before they exit the system
- A system of interconnected queues is called a queuing network
- Underlying a queuing network we always find a Markov Chain (DT, or CT, or Semi-Markov), whose state is represented by the vector of the number of customers in each queue

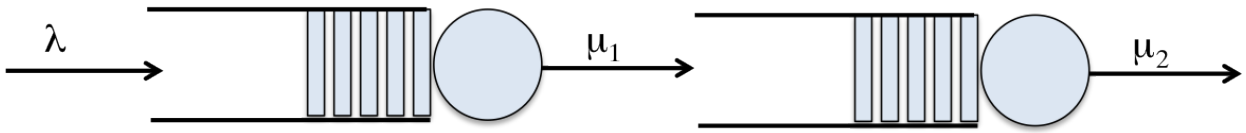
$$S = \{n_1, n_2, \cdots, n_k\}$$

where n_i is the number of customers in the i-th queue of the network

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

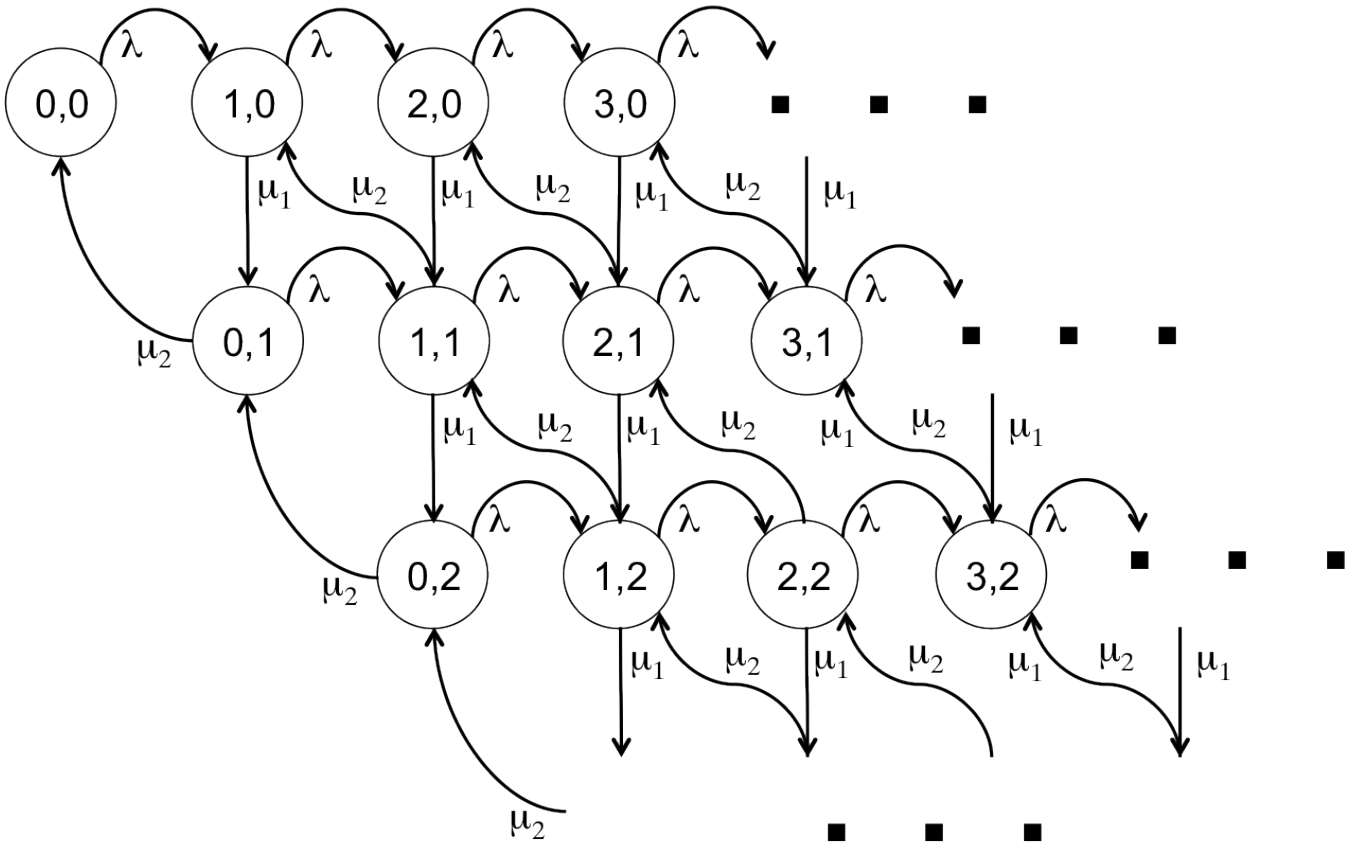
- The queuing network has k queues, but only one customer at a time can arrive, leave or change queue
- All transitions are in one of the following forms:
  $\{\cdots, n_k, \cdots\} \rightarrow \{\cdots, n_k + 1, \cdots\}$
  if queue k is an entry queue
  $\{\cdots, n_k, \cdots\} \rightarrow \{\cdots, n_k - 1, \cdots\}$
  if queue k is an exit queue
  $\{\cdots, n_i, \cdots, n_j, \cdots\} \rightarrow \{\cdots, n_i - 1, \cdots, n_j + 1, \cdots\}$
  if there is the possibility that a customer moves from queue i to queue j
- In practice we have a k dimensional Birth-Death process
- If all the transition rates (probabilities) can be found then, properly ordering the states, the solution can be found

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science



- We have two dimensions, the state is simply $S = \{n_1, n_2\}$
- The transitions between states are governed by λ, μ_1, μ_2
- Q: Is the 2-dimensional B-D process still a Markov Chain?

Example: the tandem queue



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We can easily show that this process is still Markovian, because all processes are exponential
- Finding the global balance equation (the infinitesimal generator deriving from the Chapman-Kolmogorov differential equations) is not difficult, but . . .
- . . . this is the simplest queuing network we can imagine!
- Can we find a more general and simpler way to solve this system?
- We know that the steady-state solution of an MC is unique, thus to find the solution of an MC, we only have to find *one* solution

Product Form Solution

- . . . yes, but guessing a solution at random . . .
- Indeed, we can observe that the first queue is entirely independent from the second, thus its behavior should be like a normal M/M/1
- But at steady state, jobs arrive randomly, and they are served by the first queue with exponential time services, thus the second queue "sees" as arrival process, the service process of the first one
- We can guess (not really at random) that this second queue behaves like another M/M/1 with some ρ_2

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The two queues are clearly not independent (the input to the second one is the output of the first one!!)
- But what if they behave "as if they were" independent?
- Then the solution would be the product of the two solutions

$$\pi_{i,j} = (1-\rho_1)\rho_1^i\,(1-\rho_2)\rho_2^j$$

- Plugging this simple solution in the global balance equations solves them . . . Bingo!
- Indeed, we still don't know what is ρ_2

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

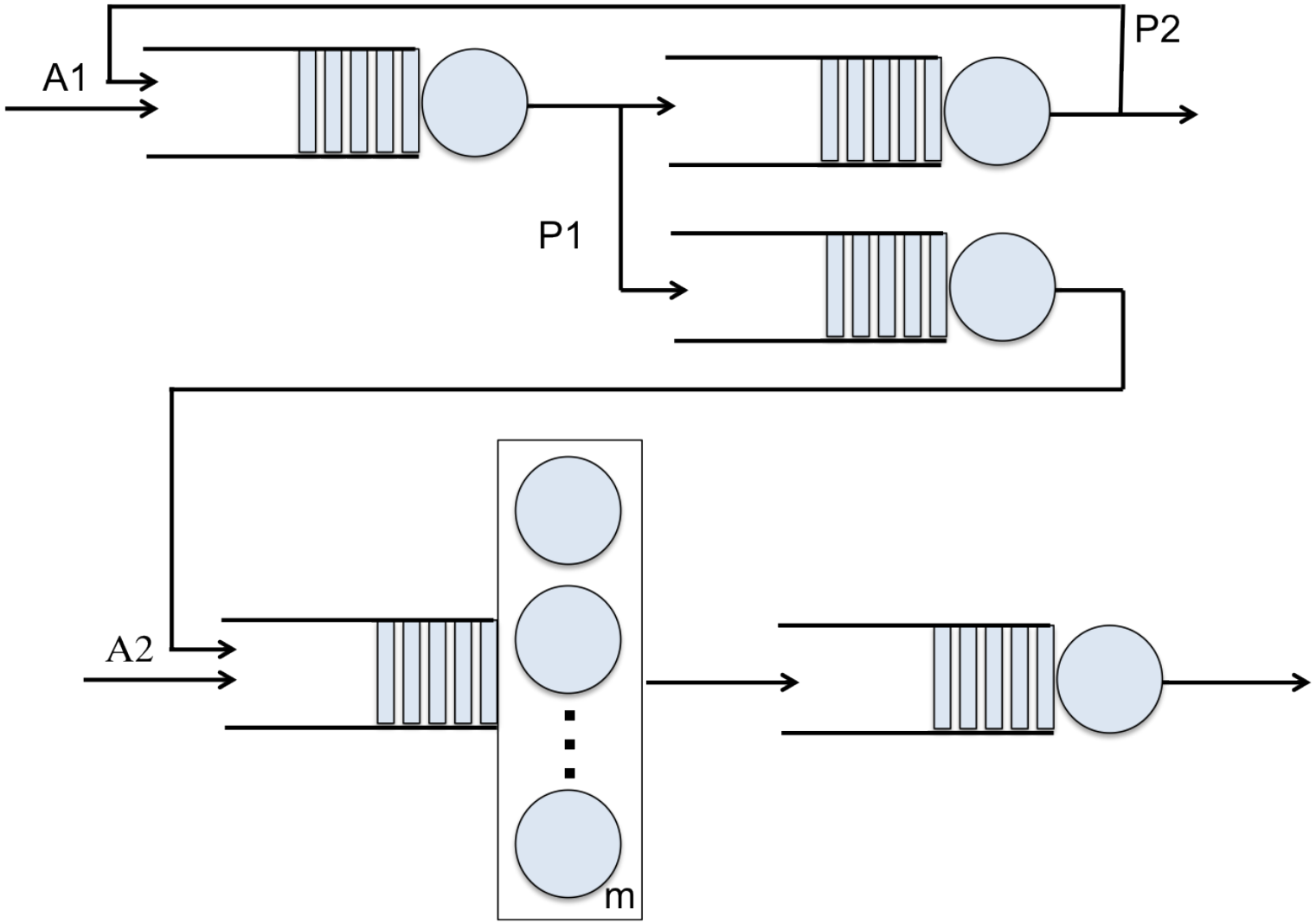
- In 1957 Jackson demonstrated that any network of Markovian queues without losses and FCFS serving discipline admits a product form solution
- Feedback among queues is admitted
- The system must be stable, but this can be checked verifying that every queue is stable
- This only requires to solve a simple k-dimensional linear system that balances the arrivals at all queues

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- What are the arrival rates to all the k queues in the network?
- If the external arrivals
  $\lambda_{ei},\ i = 1, 2, \ldots, k$ are known, and
- the routing probabilities
  $P_{ij},\ i, j = 1, 2, \ldots, k$ between all queues are known
- then all the
  $\lambda_i,\ i = 1, 2, \ldots, k$ are known solving a simple linear system of k equations
- The *local* balance is independent from the service rate of the queues and is given by

$$\lambda_i = \lambda_{ei} + \sum_{j=0}^{k} P_{ji}\lambda_j;\ \text{for } i = 1, 2, \ldots, k$$

Example of Local Balance



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- For the tandem network, we have
  - $\lambda_1 = \lambda_2 = \lambda$
  - $\rho_1 = \frac{\lambda}{\mu_1}$, $\rho_2 = \frac{\lambda}{\mu_2}$,
- In general $\rho_i = \frac{\lambda_i}{\mu_i}$ and the global solution is

$$\pi_{n_1,n_2,\ldots,n_k} = \pi_{n_1}\pi_{n_2}\cdots\pi_{n_k}$$

$$\sum_{n_1,n_2,\ldots,n_k} \pi_{n_1,n_2,\ldots,n_k} = 1$$

- Be careful with multi-server queues, the π_j are not necessarily always those of an M/M/1

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

At the blackboard

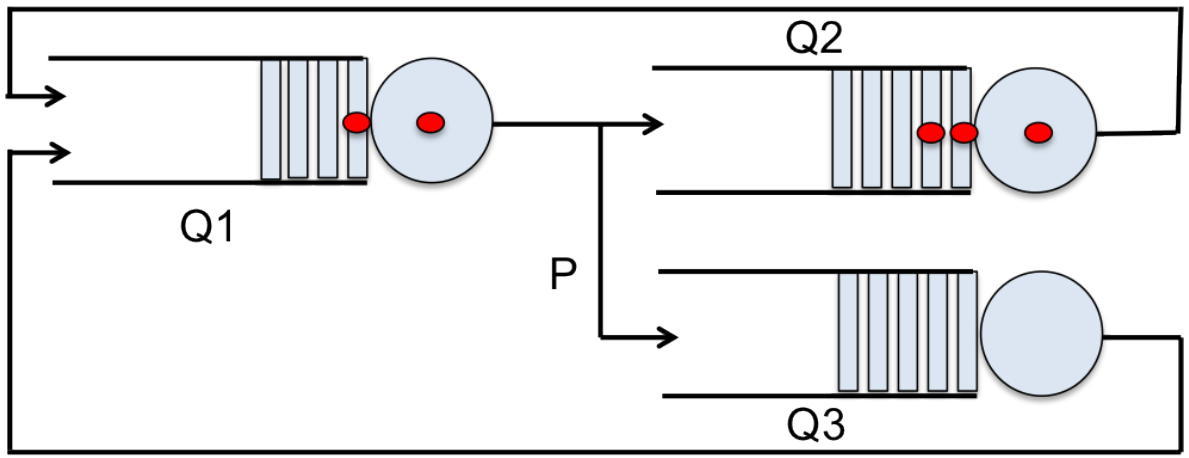
UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

1. For all queues in the network compute the arrival rates λ_i solving the local balance equations
2. For every queue j, consider it as an M/M/m station, check its stability, and compute the steady state probabilities π_{n_j}
3. Using the product form compute the global steady state probabilities $\pi_{n_1,n_2,\ldots,n_k}$

Note that as the M/M/m solutions are known in closed form, the solution of networks of arbitrary topology and dimension requires only to solve a linear system (the local balance), empowering also "what-if" analysis changing the parameters or the queues

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Networks of queues with a constant number of customers H that continuously cycle around the network
- Model pretty well any system where a new customer enters the system as soon as an old one leaves
  - Notice how this can be used to model arrivals correlated to services, which are otherwise intractable
- E.g., production systems, systems in saturation, ...



The network has a total of $H = 5$ customers and queue 3 is empty (in the snapshot)

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Closed networks with Markovian (exponential) services and FIFO service discipline
- They admit a product form solution subject to a constant number H of customers in the system
- We have the problem of defining the arrival rates ...

- As there are no arrivals from the external the system of linear equations

$$\lambda_i = \lambda_{ei} + \sum_{j=0}^{k} P_{ji}\lambda_j; \text{ for } i = 1, 2, \ldots, k$$

reduces to a homogeneous system

$$\lambda_i = \sum_{j=0}^{k} P_{ji}\lambda_j; \text{ for } i = 1, 2, \ldots, k$$

which has infinite solutions ...

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We can compute normalized arrivals setting $\lambda_1 = 1$ (any other choice is fine) and find a normalized solution
- It is customary to call these *relative arrival rates* e_i
- They are also called *visit ratios*

$$e_1 = 1; \;\; e_i = \sum_{j=0}^{k} P_{ji} e_j; \;\; \text{for } i = 2, \ldots, k$$

- And we define for each queue the relative utilization $x_i = \frac{e_i}{\mu_i}$ instead of the load ρ_i

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Recall that we have $H = \sum_{i=1}^{k} n_k$
- The total number of states is finite, even if the queues have infinite positions

$$N_S = \binom{k + H - 1}{k - 1}$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The product form can be written as

$$\pi_{n_1,n_2,\ldots,n_k} = \frac{1}{G(H)} \pi_{n_1} \pi_{n_2} \cdots \pi_{n_k}; \quad \sum_{j=1}^{k} n_j = H$$

- Where $G(H)$ is a normalization constant that takes into account the finite number of total customers, i.e., the total number of possible states
- Unfortunately it is computationally heavy ad H grows

$$G(K) = \sum_{j}^{N_S} \prod_{i=1}^{k} F_i(n_i)$$

$$F_i(n_i) = \frac{x_i^{n_i}}{\beta_i(n_i)}; \quad x_i = \frac{e_i}{\mu_i}$$

$$\beta_i(n_i) = \begin{cases} n_i!; & n_i \leq m_i \\ m_i! \cdot m_i^{n_i - m_i}; & n_i \geq m_i \\ 1; & m_i = 1 \end{cases}$$

where m_i is the number of servers in the i-th queue, i.e., queue i is an M/M/m and not a simple M/M/1. Indeed in G-N networks (an in BCMP ones) arrivals are not Poisson, so it is more appropriate to indicate the queues as –/M/m to show that arrivals are "undefined"

Steps to solve G-N Networks

1. For all queues in the network compute the relative arrival rates or visit ratios e_i solving the local balance equations
2. For every queue i, consider it as an M/M/m station, and compute the functions $F_i(n_i)$
   - Stability check is not needed as there is a finite number of customers
3. Compute the normalization factor $G(K)$
4. Using the product form compute the global steady state probabilities $\pi_{n_1,n_2,\ldots,n_k}$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

In 1975 Baskett, Chandy, Muntz and Palacios extended the queuing networks that admit product form solution to include a large number of queuing systems

- Four different types or queues are comprised
- Customers can belong to different classes
  - They can even switch between classes
  - This extension is non-trivial, and it always required complex matrix manipulations
  - I classes are present, then the solution of the queue is referred to a vector of customers present
- Networks can be open (like Jackson), closed (like G-N) or mixed (e.g., one class of customers come and go and another one remains always in the network)

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Type-1: –/M/m/FCFS

- Is the same type of queues found in Jackson and G-N networks
- Can be used to mode I/O devices, disks, ...

Type-2: –/G/1/PS

- CPUs or cores, ALUs, ...
- In general (sub)systems where the service is round-robin with a fast round pace

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Type-3: $-/G/\infty$

- Terminals can be modeled with Type-3
- "Pools" of customers moving from one service to another

Type-4: $-/G/1/LCFS\ PR$

- PR: Preemptive Resume
- Interrupt-based systems
- Stacks and accumulators

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

1. For all queues in the network compute the relative arrival rates or visit ratios e_i solving the local balance equations
2. Check the stability of every queue
3. For every queue i, compute the generic non-normalized solutions $F_i(n_i)$
   - We know the solution of –/M/m, but we can find in books the others
4. If needed compute a normalization constant $G(K)$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

5. Using the product form compute the global steady state probabilities

$$\pi_{n_1,n_2,\ldots,n_k} = \frac{1}{G(K)} \prod_{i=1}^{k} F_i(n_i)$$

6. If there are r customer classes, then all n_i are r-dimensional vectors

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Even if BCMP theorem is extremely powerful, the majority of queuing networks cannot be solved in product form
  - Finite queues with losses
  - General arrival times from the external
  - Correlated arrival/services that cannot be modeled as a closed network
  - ...
- In any case a queuing network solution can be found with a Monte-Carlo integration of the Chapman Kolmogorov equations, which are very often extremely easy to write *given the state*
  - Once the state is defined, deciding what (stochastically) happens next is defined
  - This is specially true for man-made systems
- This is nothing else than an Event-Driven simulation ...

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- As mentioned several times a Discrete Event Simulation (just simulation from now on) is a random walk on the space of a DTMC
- As this random walk is governed (or described if you prefer) by differential equations (finite differences to be more precise) this is equivalent to a Monte-Carlo integration of the model
- As queuing networks can always be mapped to a DTMC we can model a system with queues and then simulate it
- If we can describe our simulator as a DTMC, possibly using queues, Petri nets, or other high-level description language, then it is easier to evaluate results and control "outliers"