Analyzing the M/G/1 Queue with a Branching Process

Céline Comte
Nokia Bell Labs France - Télécom ParisTech

Reading Group ”Network Theory”
December 18, 2017

M/G/1 queue

Branching process

Stability

- Recurrence and transcience
- Positive recurrence

Related work and references

M/G/1 queue

Branching process

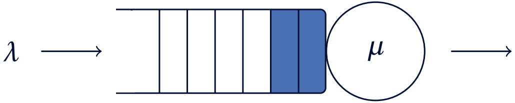
Stability

Recurrence and transcience

Positive recurrence

Related work and references

The M/G/1 queue with FCFS service discipline



M The arrival process is Poisson

→ Arrival rate $0 < \lambda < +\infty$

G The service times are i.i.d. with a general distribution

→ Mean service time $0 < \frac{1}{\mu} < +\infty$

1 A single server

? Infinite queue length

FCFS Customers leave in their arrival order

   NOKIA Bell Labs TELECOM ParisTech

More on Poisson processes

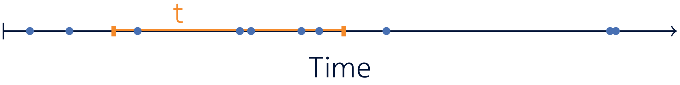
- Exponentially distributed inter-arrival times

More on Poisson processes

- Exponentially distributed inter-arrival times
- The number of arrivals during a time interval of length t has a Poisson distribution with mean λt.

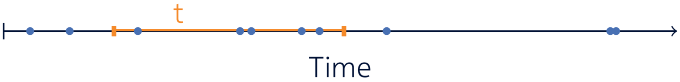
  Public NOKIA Bell Labs TELECOM ParisTech

More on Poisson processes



- Exponentially distributed inter-arrival times
- The number of arrivals during a time interval of length t has a Poisson distribution with mean λt.

More on Poisson processes

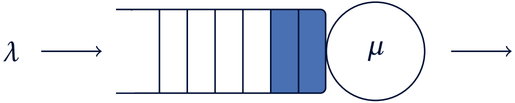


- Exponentially distributed inter-arrival times
- The number of arrivals during a time interval of length t has a Poisson distribution with mean λt.
- The numbers of arrivals during two disjoint intervals are independent.

More on Poisson processes

- Exponentially distributed inter-arrival times
- The number of arrivals during a time interval of length t has a Poisson distribution with mean λt.
- The numbers of arrivals during two disjoint intervals are independent.

The M/G/1 queue with FCFS service discipline



M The arrival process is Poisson

→ Arrival rate $0 < \lambda < +\infty$

G The service times are i.i.d. with a general distribution

→ Mean service time $0 < \frac{1}{\mu} < +\infty$

1 A single server

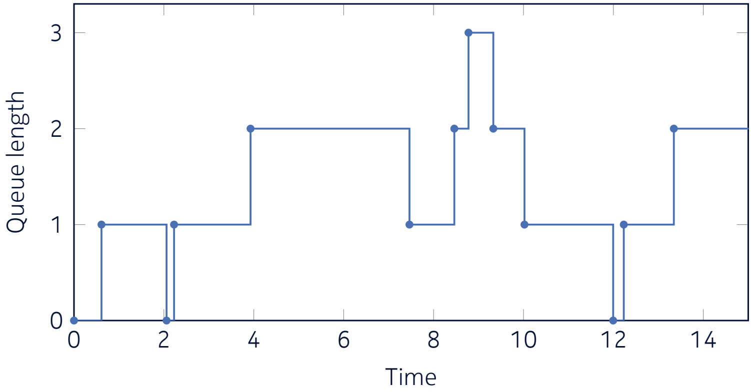
? Infinite queue length

FCFS Customers leave in their arrival order

NOKIA Bell Labs TELECOM ParisTech

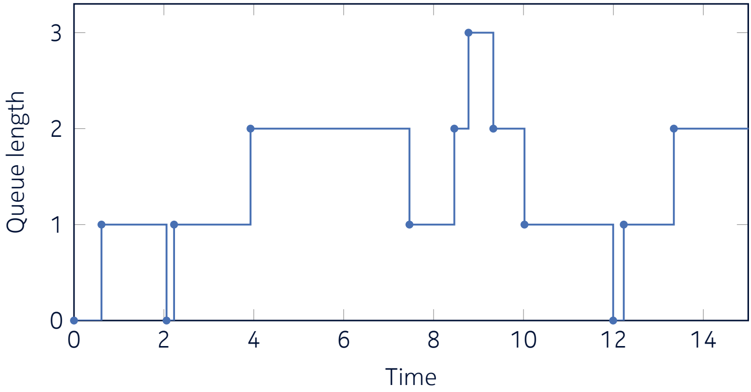
Queue state

- X_t = number of customers in the queue at time t
- $(X_t)_{t\geq 0}$ is not a Markov process (in general)



Departure time D_n

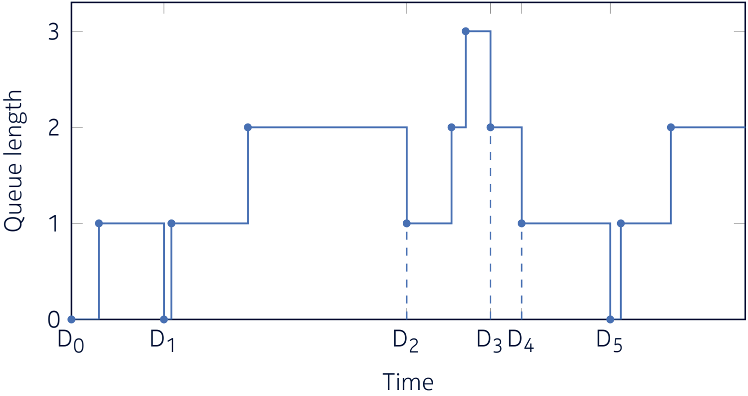
- D_n = departure time of the n-th customer
- D_n is a regeneration point of $(X_t)_{t \geq 0}$



  Public NOKIA Bell Labs TELECOM ParisTech

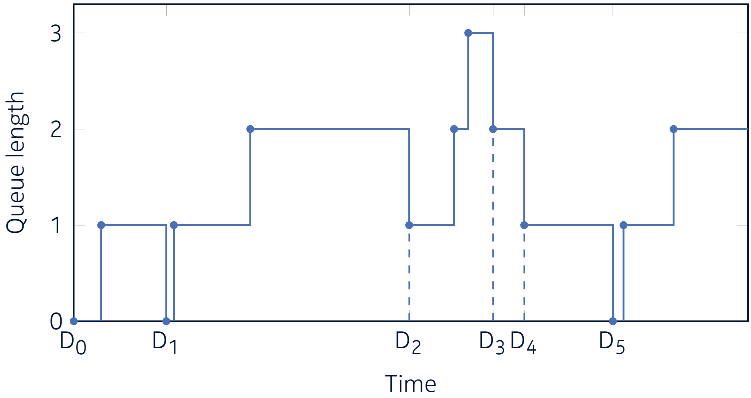
Departure time D_n

- D_n = departure time of the n-th customer
- D_n is a regeneration point of $(X_t)_{t \geq 0}$



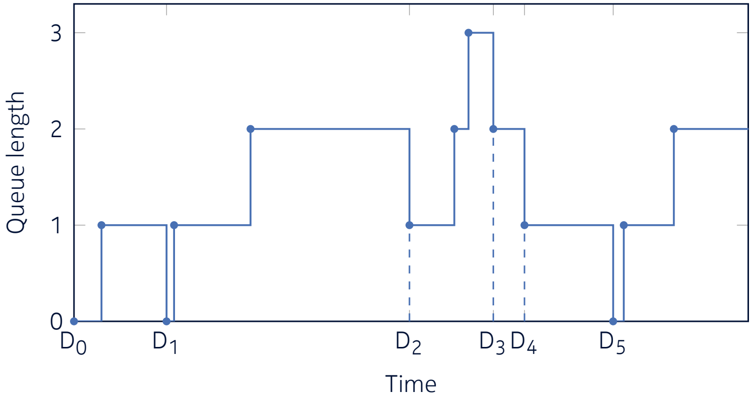
Chain embedded at departure times

- $Y_n = X_{D_n}$ number of customers left behind customer n
- $(Y_n)_{n \in \mathbb{N}}$ is a Markov chain



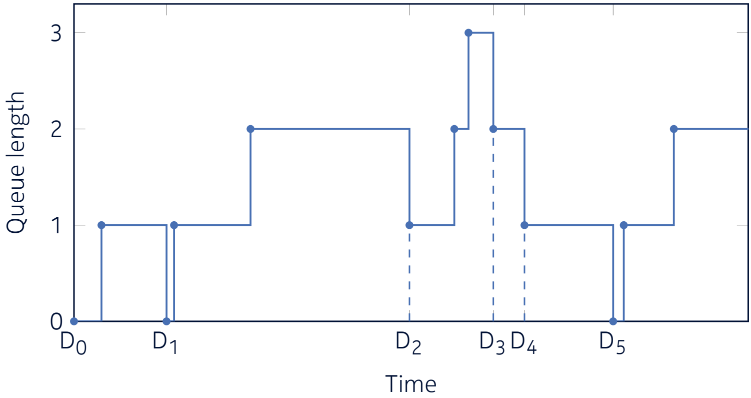
Chain embedded at departure times

- Because of the FCFS assumption,
  Y_n = number of customers that arrived since customer n entered the queue



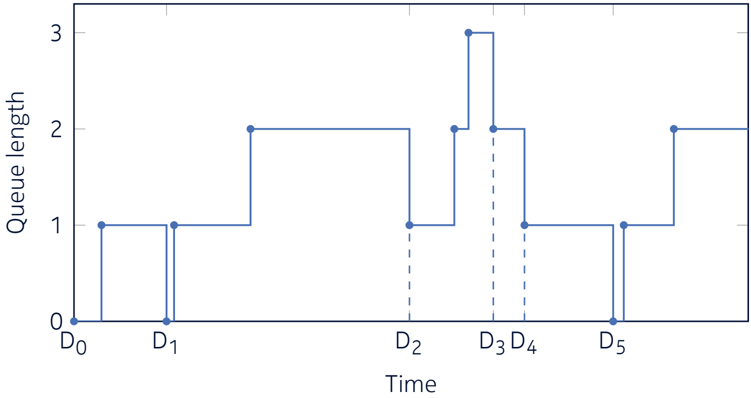
Chain embedded at departure times

- Because of the FCFS assumption,
  $Y_n - Y_{n-1} + 1$ = number of customers that arrived between the departures of customers $n-1$ and n



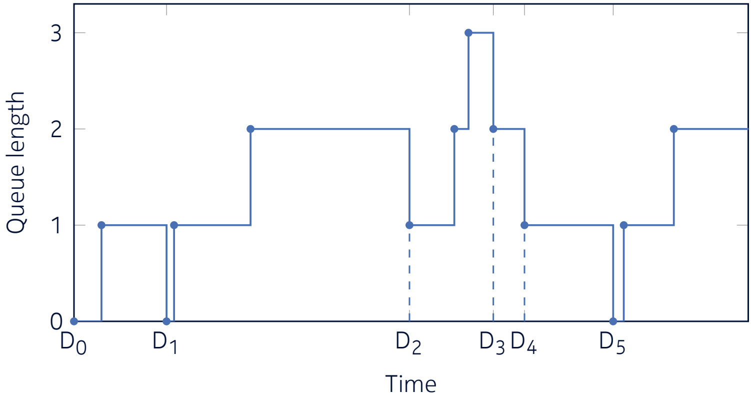
Chain embedded at departure times

- Because of the FCFS assumption, $Y_n - Y_{n-1} + 1$ = number of customers that arrived during the service of customer n



Chain embedded at departure times

- Because of the FCFS assumption, $Y_n - (Y_{n-1} - 1)_+$ = number of customers that arrived during the service of customer n



Why is it a Markov chain ?

- p_k = probabily that k customers arrive during the service time of a customer

Why is it a Markov chain ?

- p_k = probabily that k customers arrive during the service time of a customer
- The number of arrivals during a service time of length t has a Poisson distribution with mean λt.

Why is it a Markov chain ?

- p_k = probabily that k customers arrive during the service time of a customer
- The number of arrivals during a service time of length t has a Poisson distribution with mean λt.
- Letting F denote the CDF of the service time distribution,

$$p_k = \int_0^{+\infty} e^{-\lambda t} \frac{(\lambda t)^k}{k!} dF(t).$$

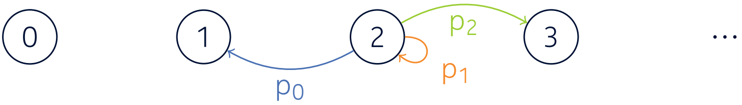
Why is it a Markov chain ?

- p_k = probabily that k customers arrive during the service time of a customer
- The number of arrivals during a service time of length t has a Poisson distribution with mean λt.
- Letting F denote the CDF of the service time distribution,

$$p_k = \int_0^{+\infty} e^{-\lambda t} \frac{(\lambda t)^k}{k!} dF(t).$$

- Independence

Why is it a Markov chain ?

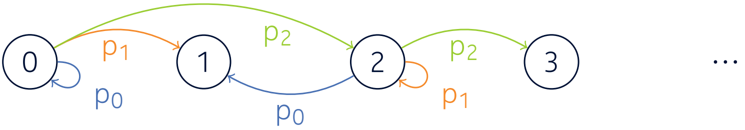


- p_k = probabily that k customers arrive during the service time of a customer
- The number of arrivals during a service time of length t has a Poisson distribution with mean λt.
- Letting F denote the CDF of the service time distribution,

$$p_k = \int_0^{+\infty} e^{-\lambda t} \frac{(\lambda t)^k}{k!} dF(t).$$

- Independence

Why is it a Markov chain ?

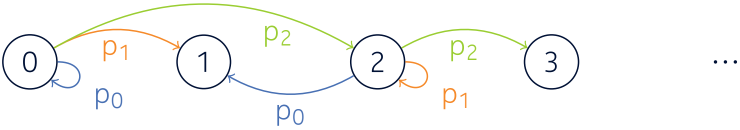


- p_k = probabily that k customers arrive during the service time of a customer
- The number of arrivals during a service time of length t has a Poisson distribution with mean λt.
- Letting F denote the CDF of the service time distribution,

$$p_k = \int_0^{+\infty} e^{-\lambda t} \frac{(\lambda t)^k}{k!} dF(t).$$

- Independence

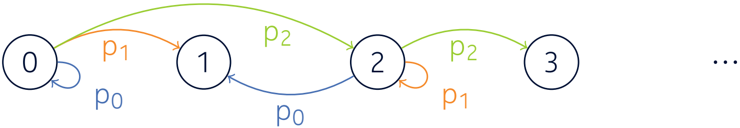
Why is it a Markov chain ?



- p_k = probabily that k customers arrive during the service time of a customer
- Transition matrix :

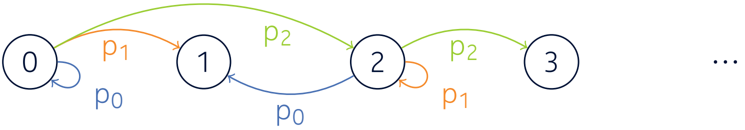
$$\begin{pmatrix} p_0 & p_1 & p_2 & p_3 & \dots & p_k & \dots & & & \\ p_0 & p_1 & p_2 & p_3 & \dots & p_k & \dots & & & \\ & p_0 & p_1 & p_2 & \dots & & p_k & \dots & & \\ & & p_0 & p_1 & \dots & & & p_k & \dots & \\ & & & p_0 & \dots & & & & p_k & \dots \end{pmatrix}$$

Properties of $(Y_n)_{n \in \mathbb{N}}$



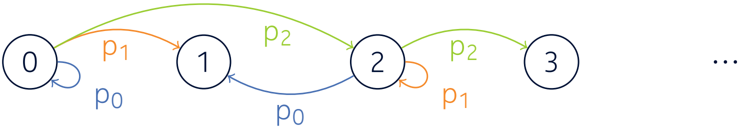
- Irreducible
  - → All the states of $(Y_n)_{n \in \mathbb{N}}$ have the same nature (positive recurrent, null recurrent or transcient)
  - → We just need to know the nature of state 0

Properties of $(Y_n)_{n \in \mathbb{N}}$



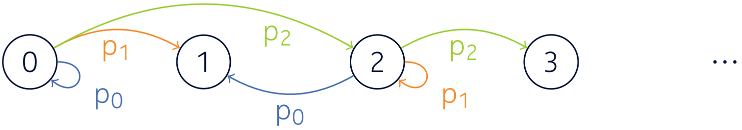
- Irreducible
  - → All the states of $(Y_n)_{n \in \mathbb{N}}$ have the same nature (positive recurrent, null recurrent or transcient)
  - → We just need to know the nature of state 0
- Aperiodic
  - → If $(Y_n)_{n \in \mathbb{N}}$ is positive recurrent, then it is also ergodic

Stability of $(Y_n)_{n \in \mathbb{N}}$



- State 0 is recurrent
  - ⇔ If the queue starts empty, then with probability 1 it empties out an infinite number of times
  - ⇔ If the queue starts empty, then with probability 1 it eventually empties out

Stability of $(Y_n)_{n\in\mathbb{N}}$

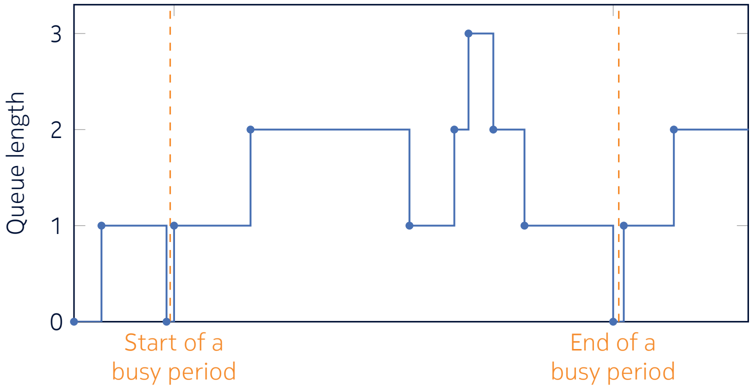


Assume that state 0 is recurrent.

- State 0 is positive recurrent
  - ⇔ If the queue starts empty, then the mean time until it empties out again is finite
- State 0 is null recurrent
  - ⇔ If the queue starts empty, then the mean time until it empties out again is infinite

Busy period

- We just need to look at one busy period of the queue.



M/G/1 queue

Branching process

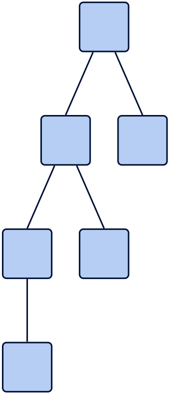
Stability

- Recurrence and transcience
- Positive recurrence

Related work and references

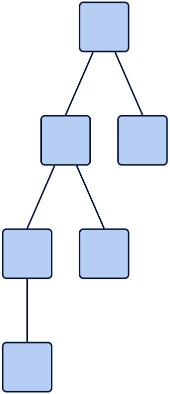
Definition

- Random tree



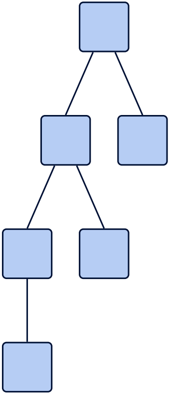
Definition

- Random tree
- One node at generation 0



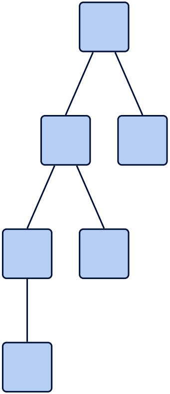
Definition

- Random tree
- One node at generation 0
- The children of the nodes of generation n belong to generation $n+1$



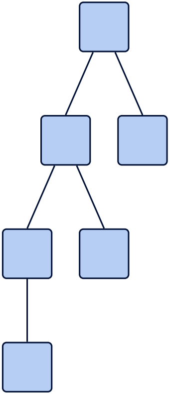
Definition

- Random tree
- One node at generation 0
- The children of the nodes of generation n belong to generation $n+1$
- The number of children of a node is



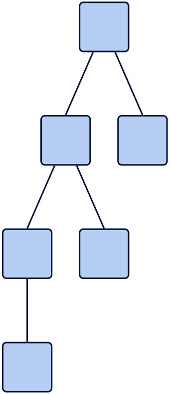
Definition

- Random tree
- One node at generation 0
- The children of the nodes of generation n belong to generation n + 1
- The number of children of a node is
  - random,



Definition

- Random tree
- One node at generation 0
- The children of the nodes of generation n belong to generation n + 1
- The number of children of a node is
  - random,
  - **with a probability distribution that is the same for all nodes**,

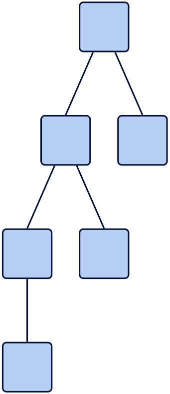


  Public

NOKIA Bell Labs TELECOM ParisTech

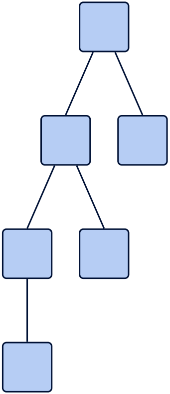
Definition

- Random tree
- One node at generation 0
- The children of the nodes of generation n belong to generation n + 1
- The number of children of a node is
  - random,
  - **with a probability distribution that is the same for all nodes**,
  - **independent of the number of children of other nodes**,



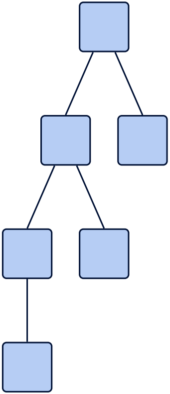
Definition

- Random tree
- One node at generation 0
- The children of the nodes of generation n belong to generation n + 1
- The number of children of a node is
  - random,
  - **with a probability distribution that is the same for all nodes**,
  - **independent of the number of children of other nodes**,
- Mean number of children per node
  $\rho < +\infty$



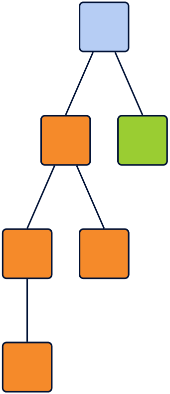
Definition

⚠ Given the number of children of the root, the subtrees rooted at these nodes are independent and have the same distribution as the initial tree

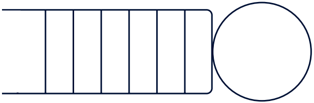


Definition

⚠ Given the number of children of the root, the subtrees rooted at these nodes are independent and have the same distribution as the initial tree



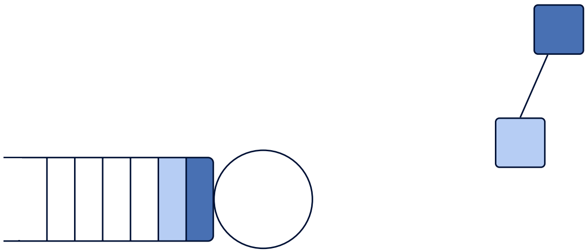
Build the tree



Build the tree

  Public

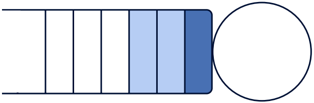
Build the tree

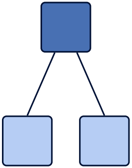


  Public

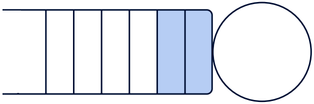
NOKIA Bell Labs TELECOM ParisTech

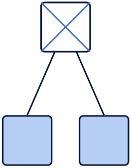
Build the tree



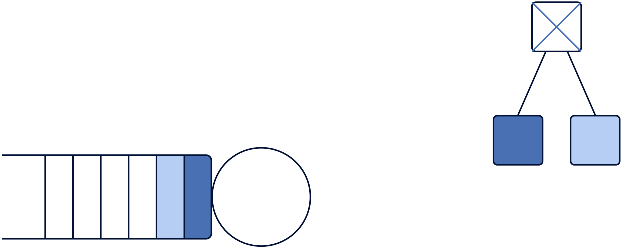


Build the tree

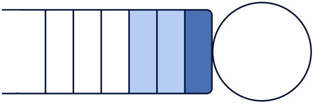


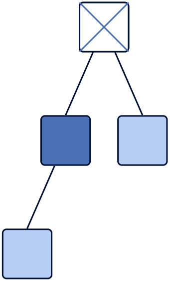


Build the tree



Build the tree

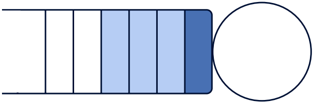


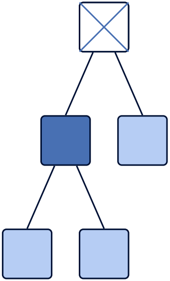


  Public

NOKIA Bell Labs TELECOM ParisTech

Build the tree

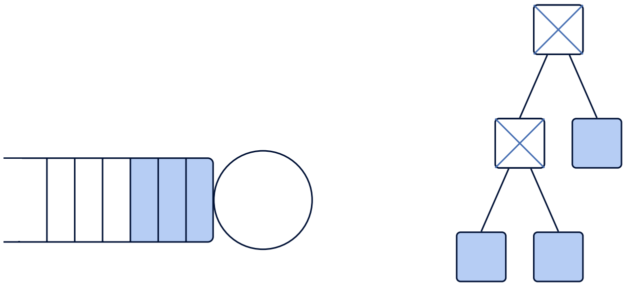




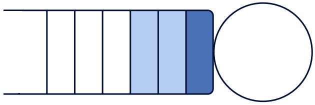
  Public

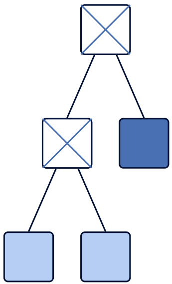
NOKIA Bell Labs TELECOM ParisTech

Build the tree



Build the tree

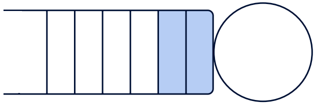


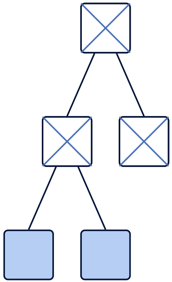


  Public

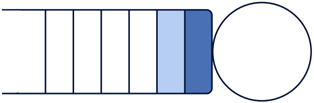
NOKIA Bell Labs TELECOM ParisTech

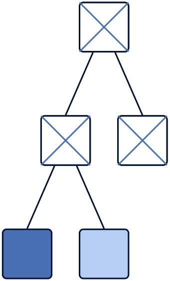
Build the tree



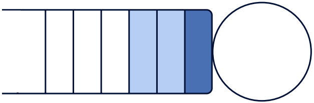


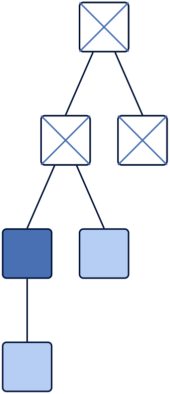
Build the tree



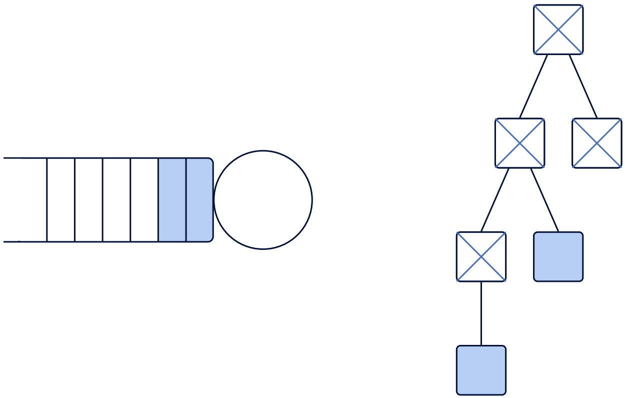


Build the tree

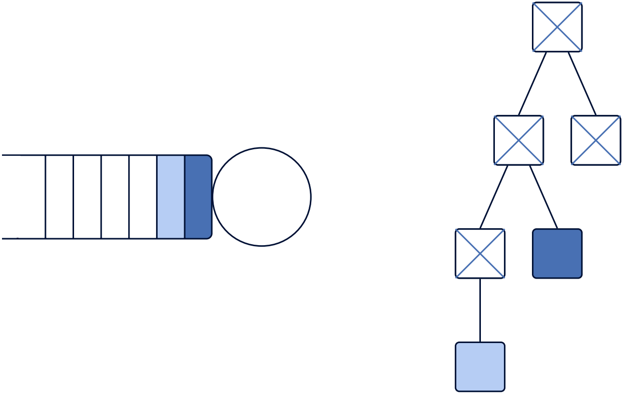




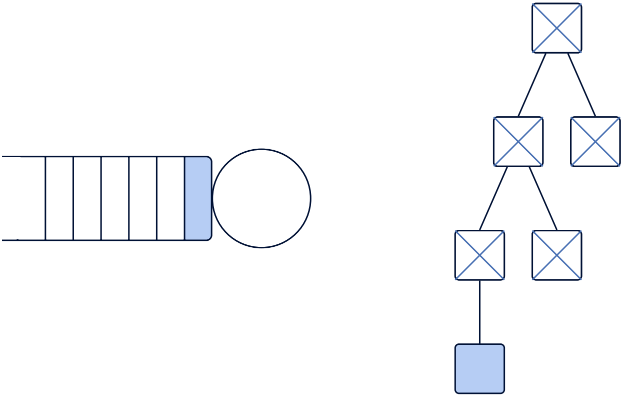
Build the tree



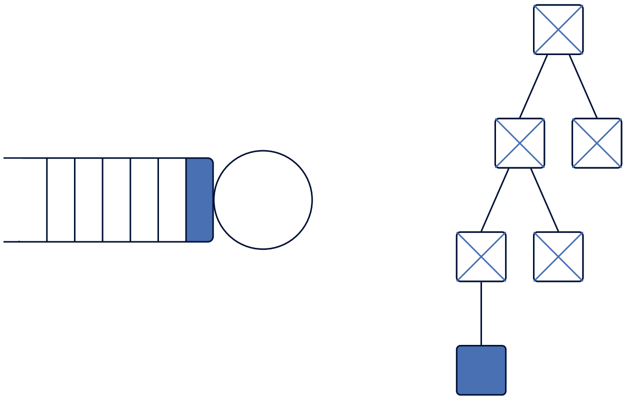
Build the tree



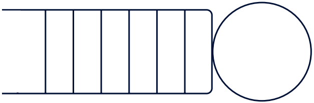
Build the tree

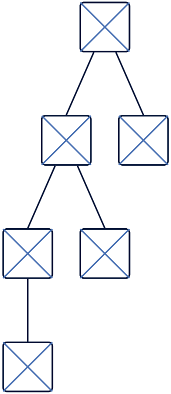


Build the tree

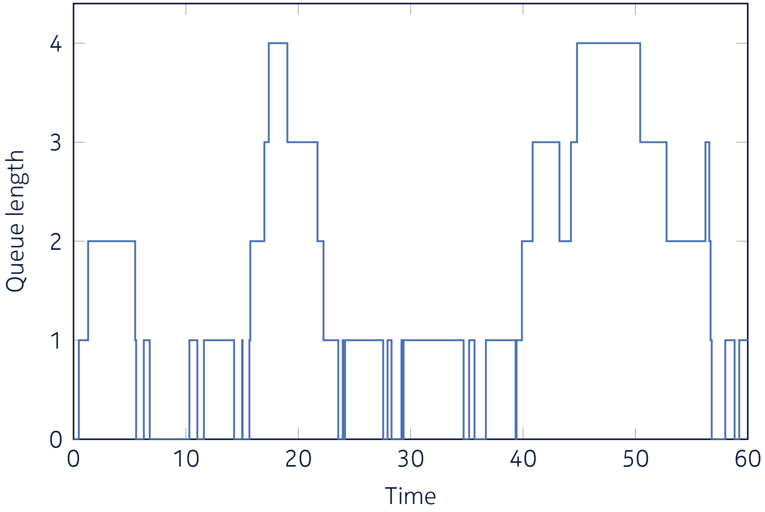


Build the tree





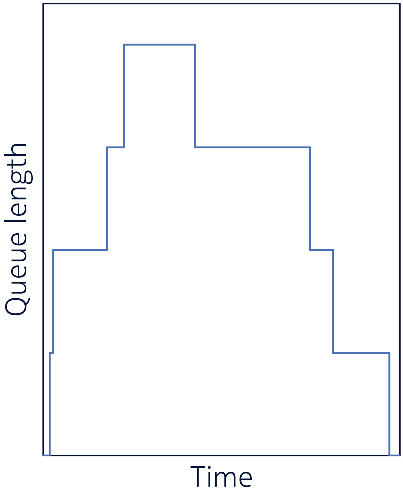
Queue fluctuation with time



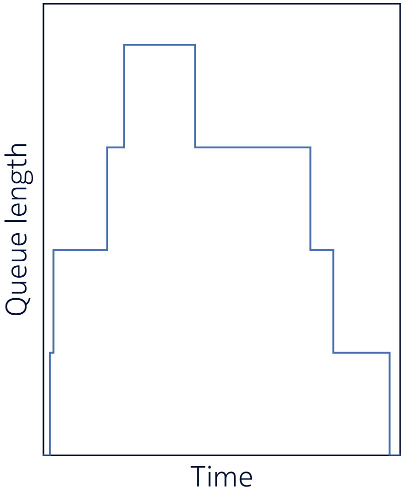
  Public

NOKIA Bell Labs TELECOM ParisTech

Building the tree



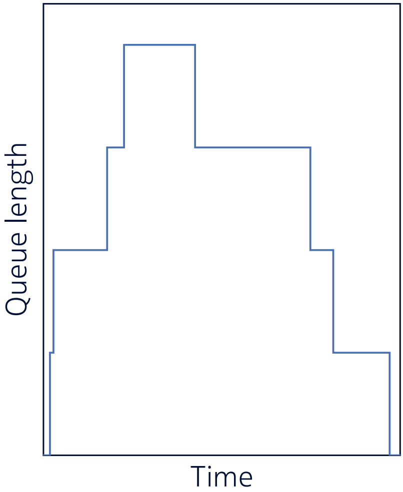
Building the tree

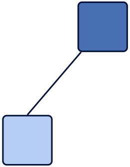


NOKIA Bell Labs

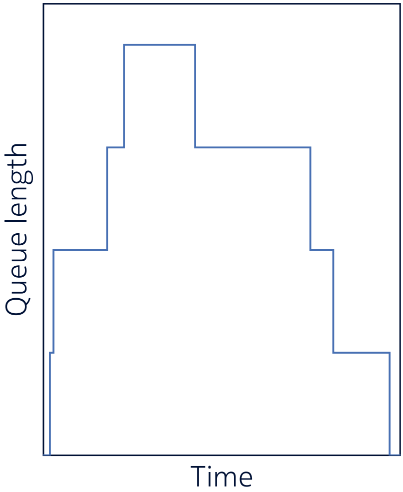
TELECOM ParisTech

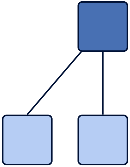
Building the tree





Building the tree

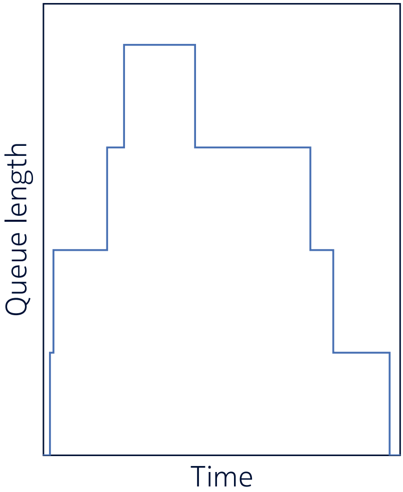


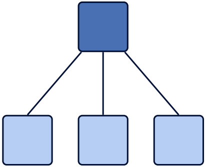


  Public

NOKIA Bell Labs TELECOM ParisTech

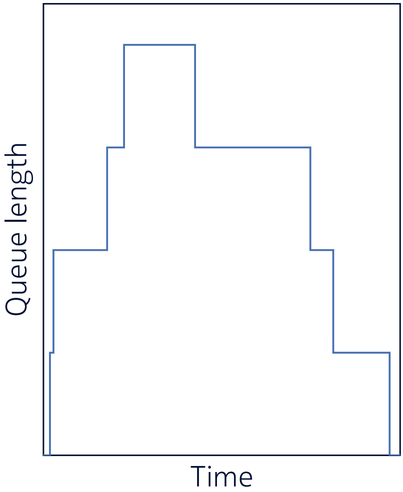
Building the tree

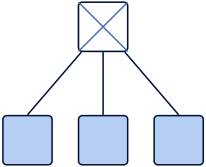




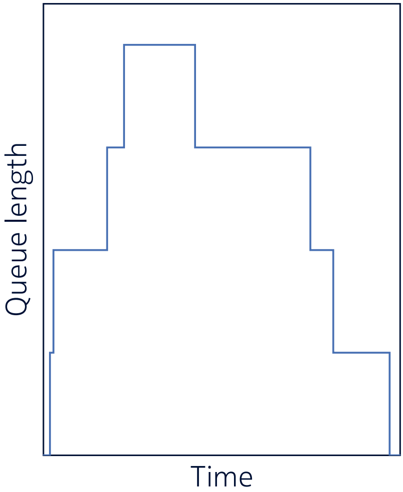
  Public NOKIA Bell Labs TELECOM ParisTech

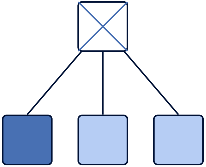
Building the tree



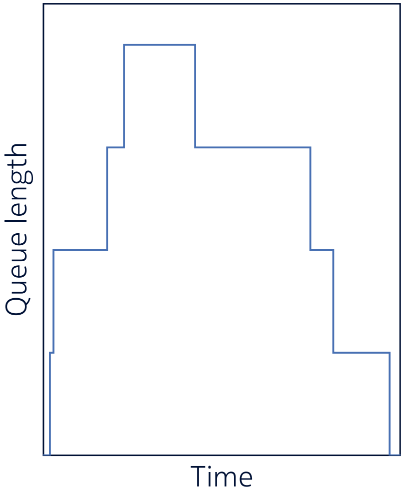


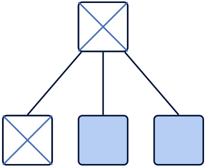
Building the tree





Building the tree

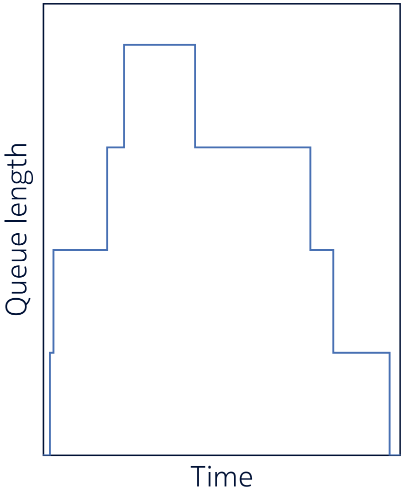


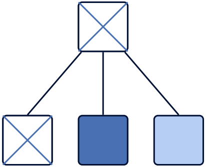


  Public

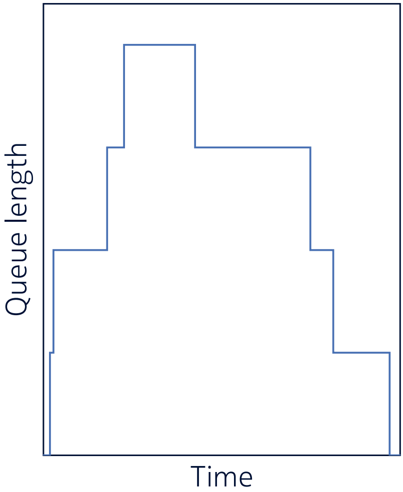
NOKIA Bell Labs TELECOM ParisTech

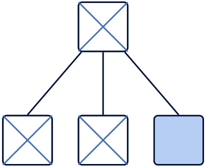
Building the tree





Building the tree

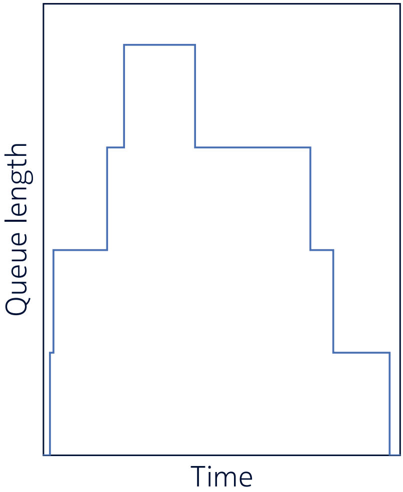


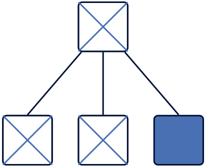


  Public

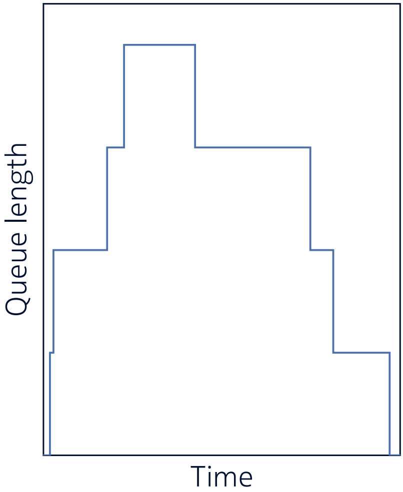
NOKIA Bell Labs TELECOM ParisTech

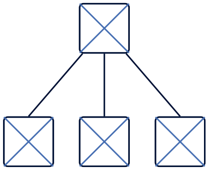
Building the tree



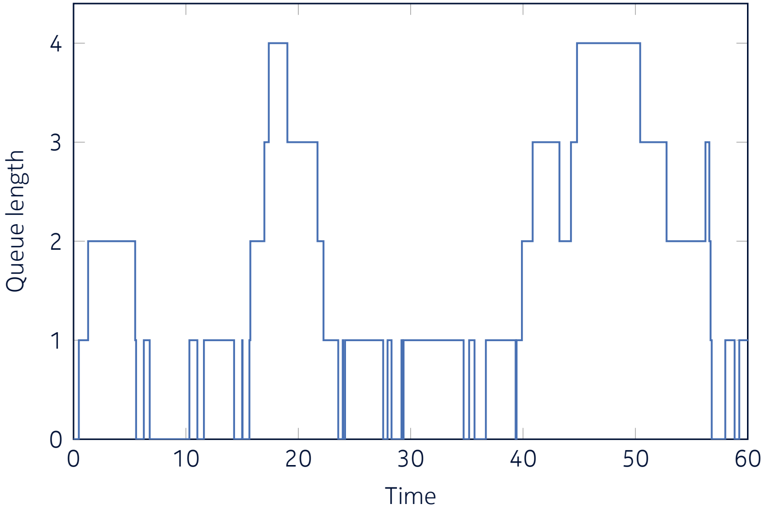


Building the tree

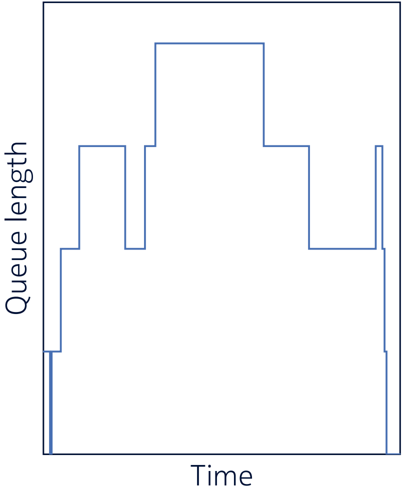




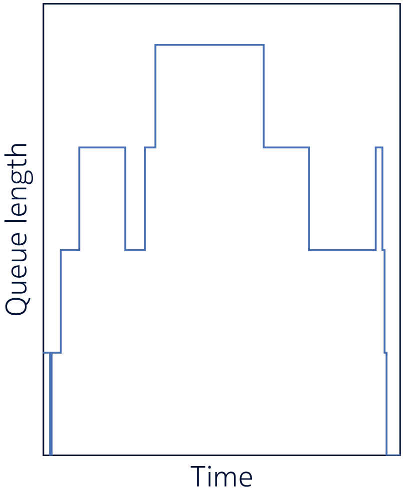
Realization



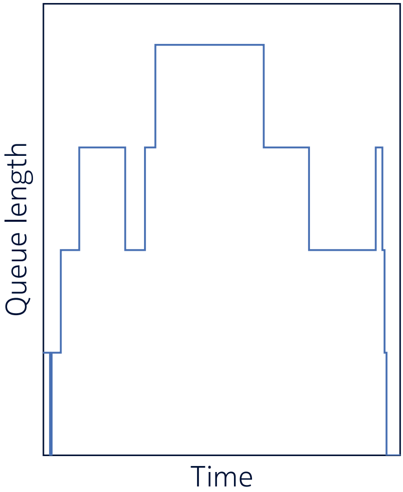
Building the tree

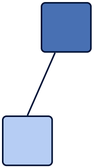


Building the tree



Building the tree

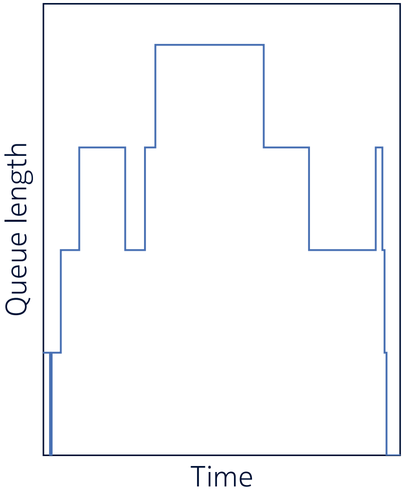


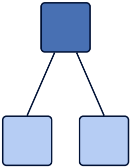


  Public

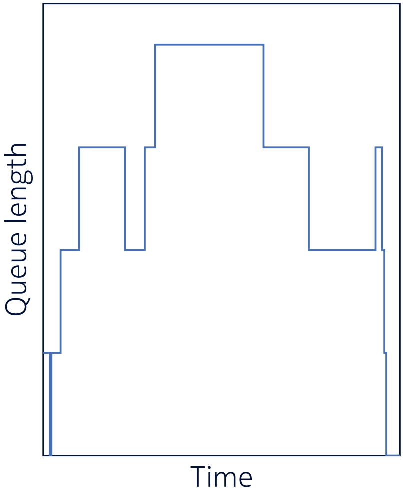
NOKIA Bell Labs TELECOM ParisTech

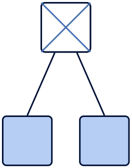
Building the tree



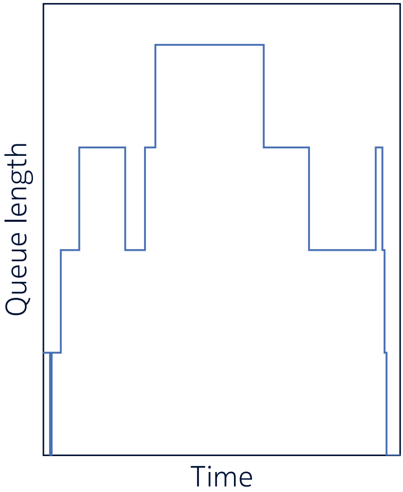


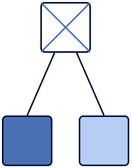
Building the tree



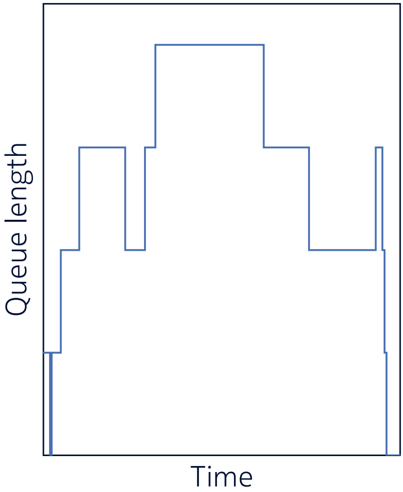


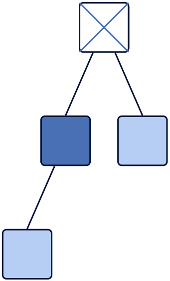
Building the tree





Building the tree

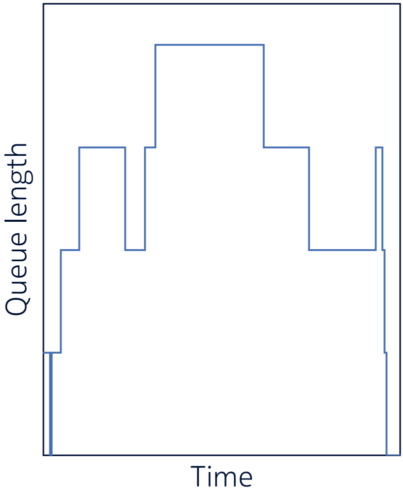


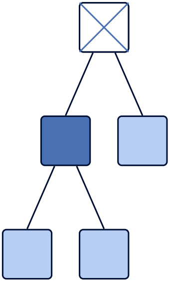


  Public

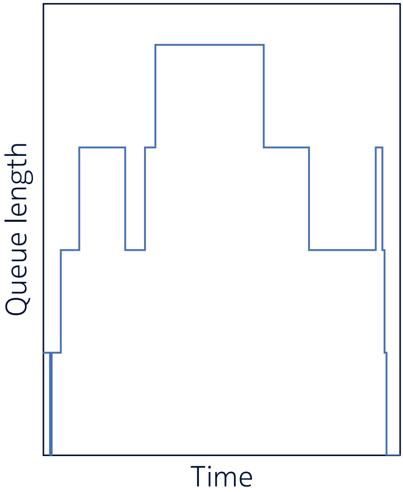
NOKIA Bell Labs TELECOM ParisTech

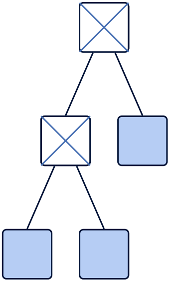
Building the tree





Building the tree

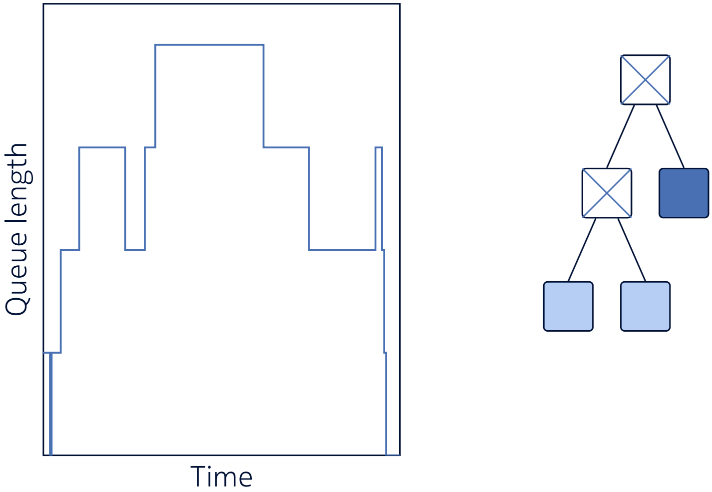




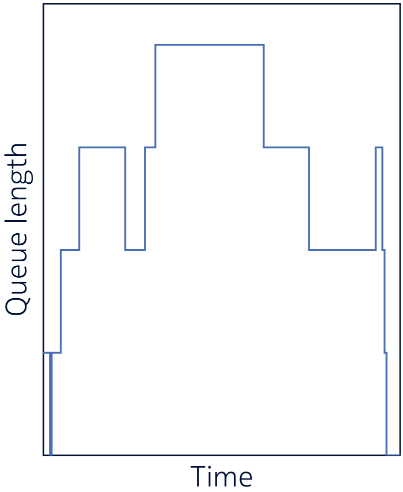
  Public

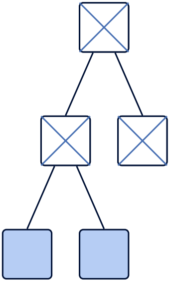
NOKIA Bell Labs TELECOM ParisTech

Building the tree

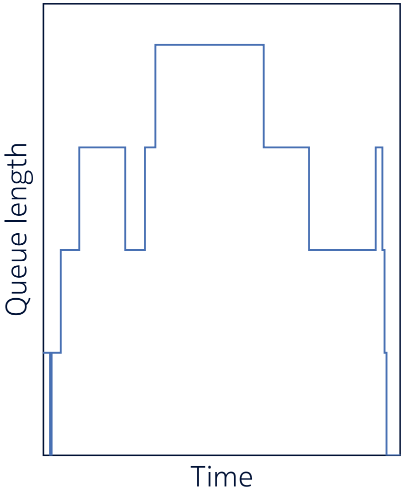


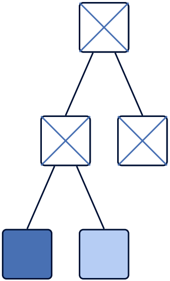
Building the tree





Building the tree

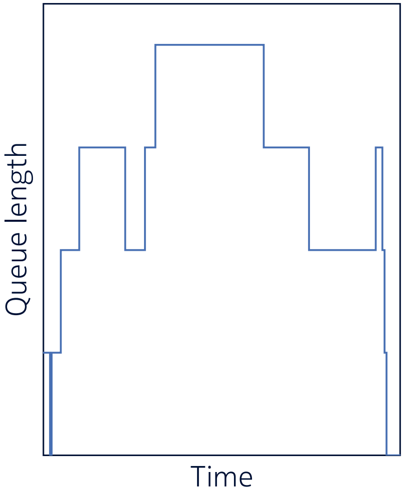


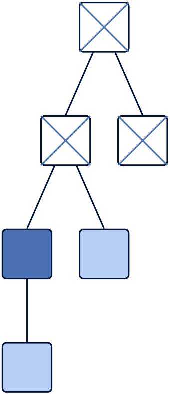


  Public

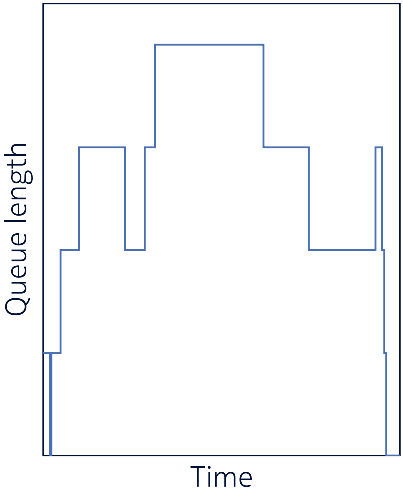
NOKIA Bell Labs TELECOM ParisTech

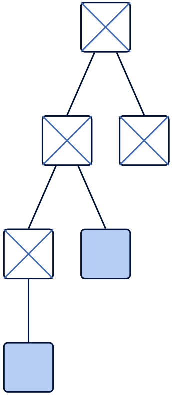
Building the tree



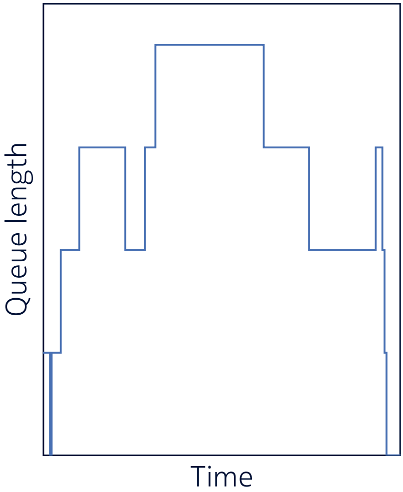


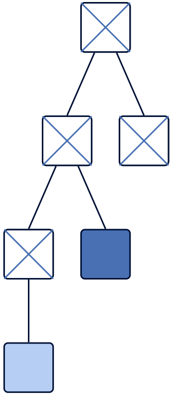
Building the tree



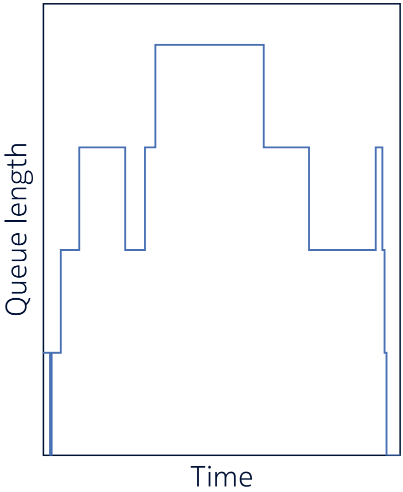


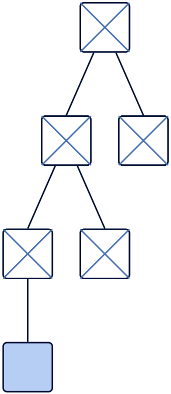
Building the tree



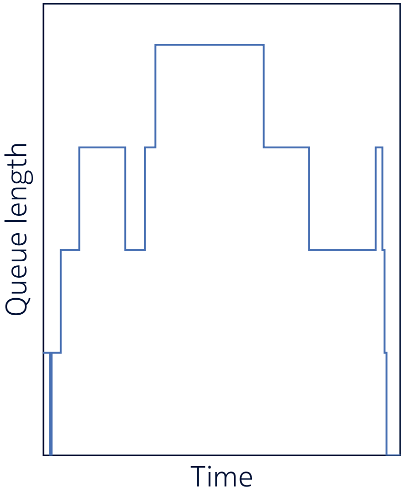


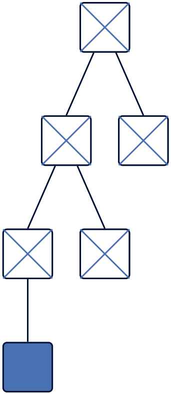
Building the tree



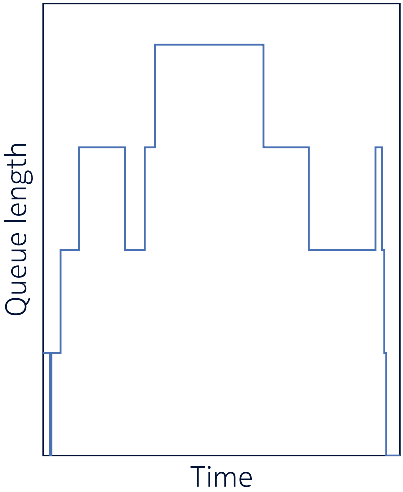


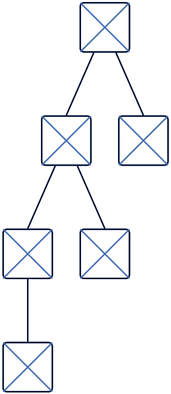
Building the tree





Building the tree

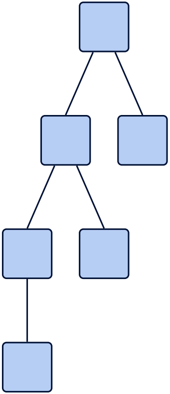




Why is it a branching process ?

Definition

- Random tree
- One node at generation 0
- The children of the nodes of generation n belong to generation n + 1
- The number of children of a node is
  - random,
  - **with a probability distribution that is the same for all nodes**,
  - **independent of the number of children of other nodes**,
- Mean number of children per node $\rho < +\infty$



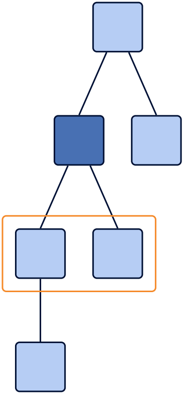
  Public

NOKIA Bell Labs TELECOM ParisTech

Why is it a branching process ?

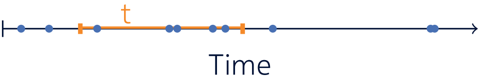
Why is it a branching process ?

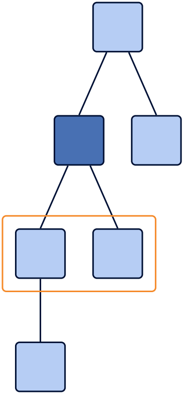
- p_k = probability that k customers arrive during a single service time
  = probability that a node has k children



Why is it a branching process ?

- p_k = probability that k customers arrive during a single service time
  = probability that a node has k children
- The number of arrivals during a service time of length t has a Poisson distribution with mean λt.

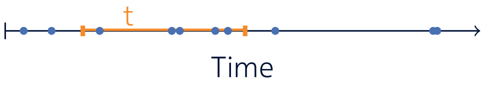


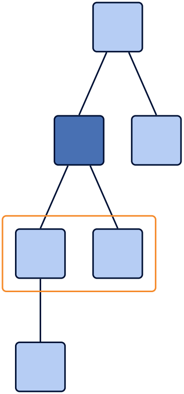


  Public NOKIA Bell Labs TELECOM ParisTech

Why is it a branching process ?

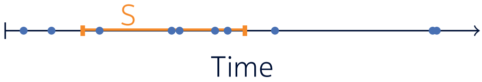
- p_k = probability that k customers arrive during a single service time
  = probability that a node has k children
- The number of arrivals during a service time of length t has a Poisson distribution with mean λt.
- Independence

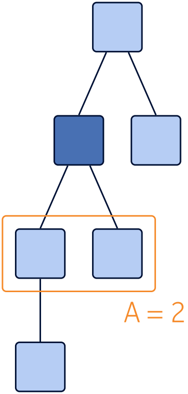




Mean number of children per node

- S = service time of a given customer.
  S is a random variable with mean $\frac{1}{\mu}$.
- A = number of customers arrived during the service of this customer
- Given S, A has a Poisson distribution with mean λS.



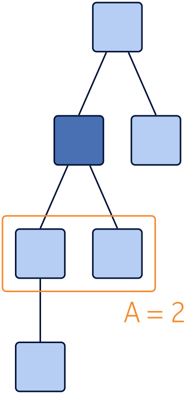


Mean number of children per node

- Mean number of children of a node

$$
\begin{aligned}
\mathbb{E}(A) &= \mathbb{E}(\mathbb{E}(A|S)) \\
&= \mathbb{E}(\lambda S) = \lambda \mathbb{E}(S) \\
&= \frac{\lambda}{\mu}.
\end{aligned}
$$

$\Rightarrow$ $\rho = \frac{\lambda}{\mu}$ = load of the queue,
= mean number of children of
a node in the tree.



M/G/1 queue

Branching process

Stability

Recurrence and transcience

Positive recurrence

Related work and references

M/G/1 queue

Branching process

Stability

Recurrence and transcience

Positive recurrence

Related work and references

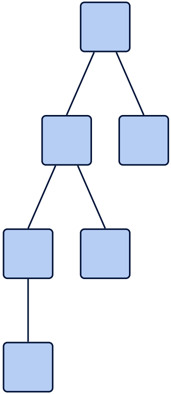
Recurrence and transcience

- $\rho = \frac{\lambda}{\mu}$ = load of the queue,
  = mean number of children per node.

- Branching process result :
  - $\rho \leq 1$: the tree dies out with probability 1.
    (assuming that $p_1 < 1$ if $\rho = 1$)
  - $\rho > 1$: the tree dies out with probability < 1.
- Queueing reformulation :
  - $\rho \leq 1$: $(Y_n)_{n \in \mathbb{N}}$ is recurrent.
  - $\rho > 1$: $(Y_n)_{n \in \mathbb{N}}$ is transcient.

Sketch of proof

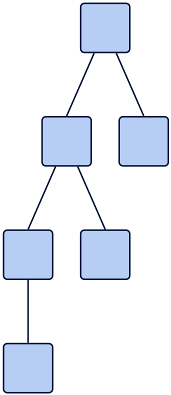
- p_k = probability that k customers arrive during a single service time
  = probability that a node has k children



Sketch of proof

- p_k = probability that k customers arrive during a single service time
  = probability that a node has k children

- P = probability that the queue empties
  = probability that the tree is finite



NOKIA Bell Labs
TELECOM ParisTech

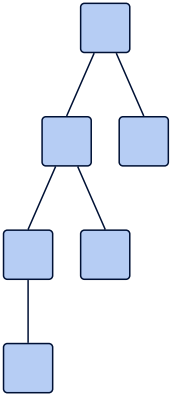
Sketch of proof

- p_k = probability that k customers arrive during a single service time
  = probability that a node has k children

- P = probability that the queue empties
  = probability that the tree is finite

- P satisfies the fixed-point equation

$$P = \sum_{k=0}^{+\infty} P^k p_k.$$



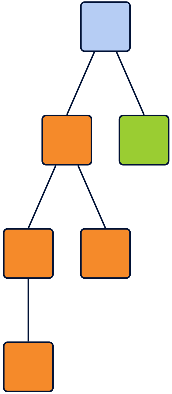
Sketch of proof

- p_k = probability that k customers arrive during a single service time
  = probability that a node has k children

- P = probability that the queue empties
  = probability that the tree is finite

- P satisfies the fixed-point equation

$$P = \sum_{k=0}^{+\infty} P^k p_k.$$



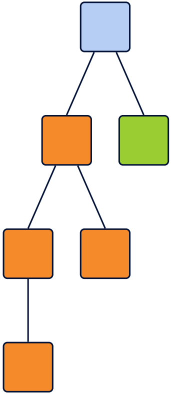
  Public

NOKIA Bell Labs TELECOM ParisTech

Sketch of proof

- P satisfies the fixed-point equation

$$P = \sum_{k=0}^{+\infty} P^k p_k$$



Sketch of proof

- P satisfies the fixed-point equation

$$P = \sum_{k=0}^{+\infty} P^k p_k$$

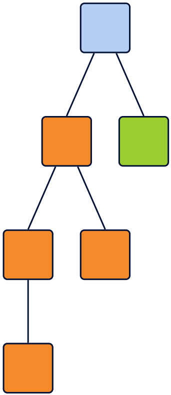
- P satisfies the fixed-point equation

$$P = \phi(P),$$

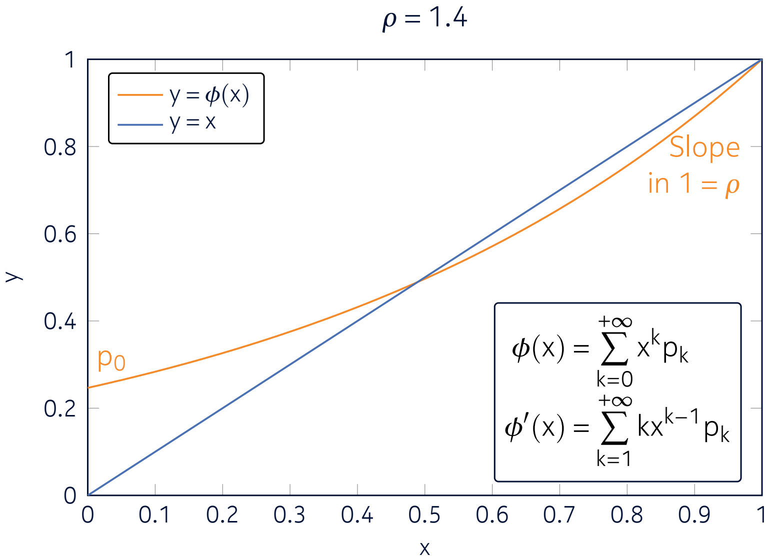
where

$$\phi(x) = \sum_{k=0}^{+\infty} x^k p_k$$

is the generating function of $(p_k)_{k \in \mathbb{N}}$



Sketch of proof



  Public NOKIA Bell Labs TELECOM ParisTech

Sketch of proof

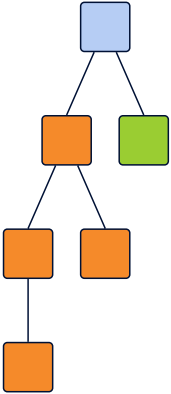
- P is **the smallest solution in** $[0,1]$ of the fixed-point equation

$$P = \phi(P),$$

where

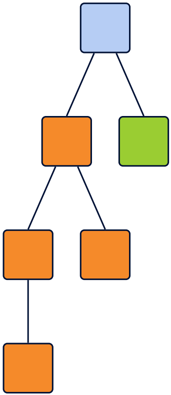
$$\phi(x) = \sum_{k=0}^{+\infty} x^k p_k$$

is the generating function of $(p_k)_{k \in \mathbb{N}}$



Sketch of proof

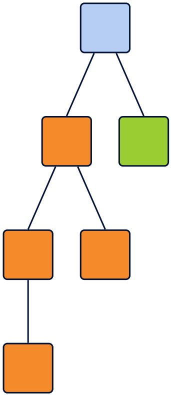
- P_n = probability that the population is extincted at generation n



Sketch of proof

- P_n = probability that the population is extincted at generation n

- $P = \lim_{n \to +\infty} P_n$



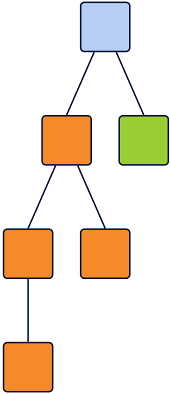
  Public NOKIA Bell Labs TELECOM ParisTech

Sketch of proof

- P_n = probability that the population
  is extincted at generation n

- $P = \lim_{n \to +\infty} P_n$

  $\mathbb{P}$(the population is finite)

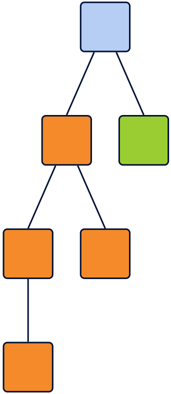


Sketch of proof

- P_n = probability that the population is extincted at generation n

- $P = \lim_{n \to +\infty} P_n$

$$\mathbb{P}(\text{the population is finite}) = \mathbb{P}\left(\bigcup_{n=0}^{+\infty} \left\{\begin{matrix}\text{the population is extincted} \\ \text{at generation } n\end{matrix}\right\}\right)$$

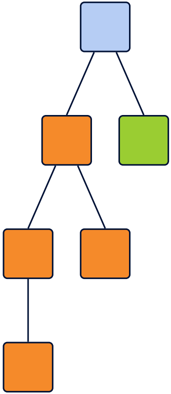


Sketch of proof

- P_n = probability that the population is extincted at generation n

- $P = \lim_{n \to +\infty} P_n$

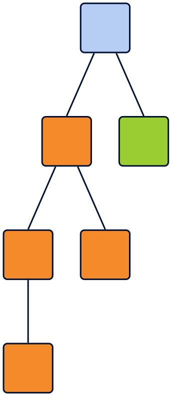
$$\mathbb{P}(\text{the population is finite})$$
$$= \mathbb{P}\left(\bigcup_{n=0}^{+\infty} \left\{ \begin{array}{c} \text{the population is extincted} \\ \text{at generation n} \end{array} \right\} \right)$$
$$= \lim_{n \to +\infty} \mathbb{P}\left(\begin{array}{c} \text{population is extincted} \\ \text{at generation n} \end{array} \right)$$



Sketch of proof

- P_n = probability that the population is extincted at generation n

- $P = \lim_{n \to +\infty} P_n$



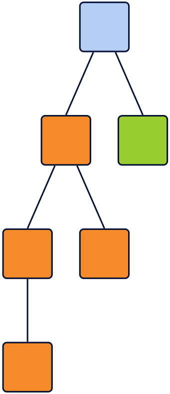
Sketch of proof

- P_n = probability that the population is extincted at generation n

- $P = \lim_{n \to +\infty} P_n$

- $(P_n)_{n \in \mathbb{N}}$ satisfies

$$P_{n+1} = \sum_{k=0}^{+\infty} P_n{}^k p_k, \quad \text{that is,} \quad P_{n+1} = \phi(P_n)$$



  Public NOKIA Bell Labs TELECOM ParisTech

Sketch of proof

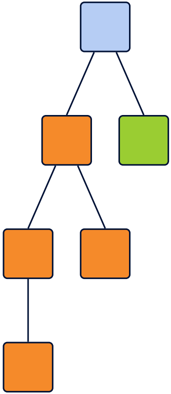
- P_n = probability that the population is extincted at generation n

- $P = \lim_{n \to +\infty} P_n$

- $(P_n)_{n \in \mathbb{N}}$ satisfies

$$P_{n+1} = \sum_{k=0}^{+\infty} P_n{}^k p_k, \quad \text{that is,} \quad P_{n+1} = \phi(P_n)$$

- $P_0 = 0$



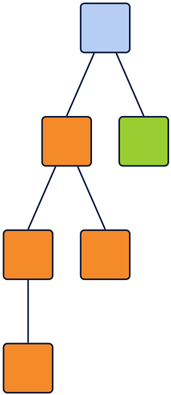
Sketch of proof

- P is a solution of the fixed-point equation

$$P = \phi(P)$$

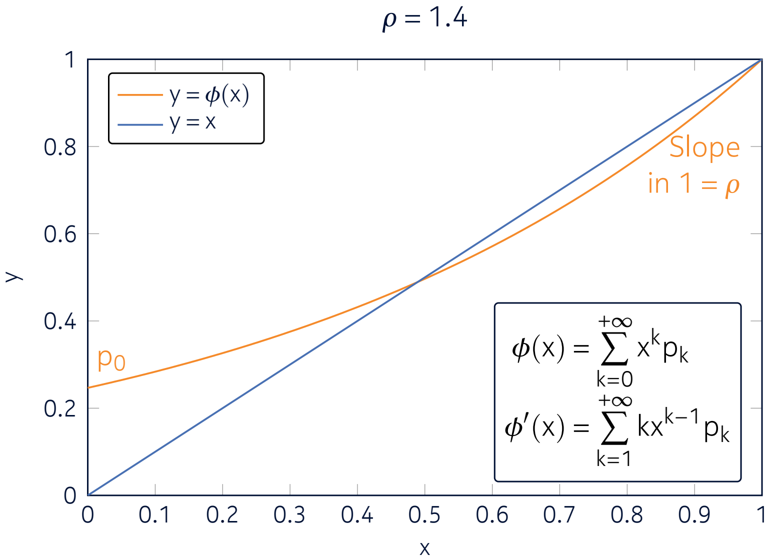
- It is also the limit of the sequence $(P_n)_{n\in\mathbb{N}}$ defined recursively by $P_0 = 0$ and

$$P_{n+1} = \phi(P_n), \quad \forall n \in \mathbb{N}.$$

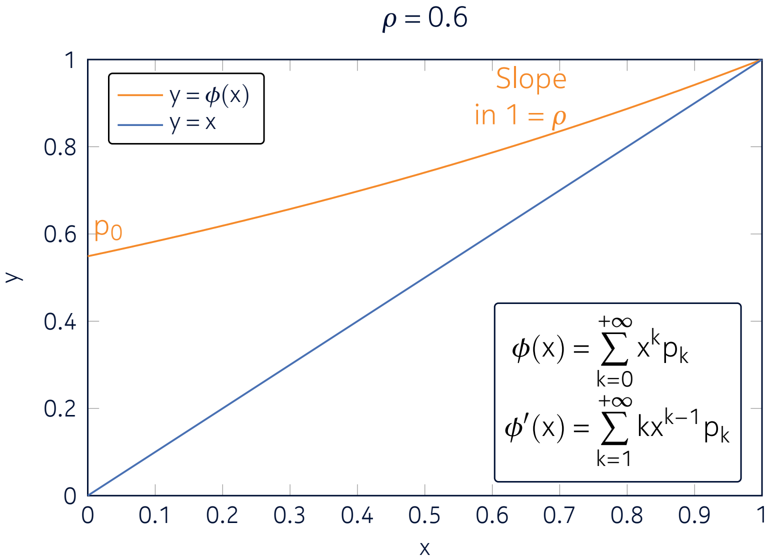


  Public NOKIA Bell Labs TELECOM ParisTech

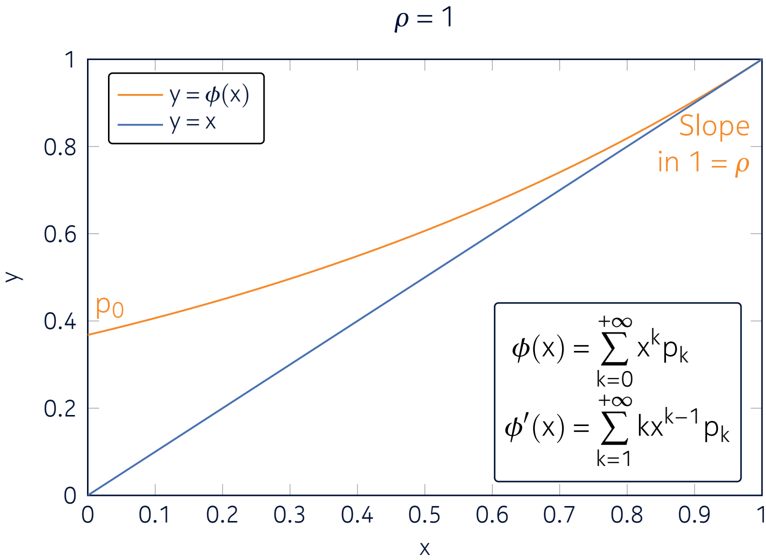
Sketch of proof



Sketch of proof



Sketch of proof



Recurrence and transcience

- $\rho = \frac{\lambda}{\mu}$ = load of the queue,
  = mean number of children per node.

- Branching process result :
  - $\rho \leq 1$: the tree dies out with probability 1.
    (assuming that $p_1 < 1$ if $\rho = 1$)
  - $\rho > 1$: the tree dies out with probability < 1.
- Queueing reformulation :
  - $\rho \leq 1$: $(Y_n)_{n \in \mathbb{N}}$ is recurrent.
  - $\rho > 1$: $(Y_n)_{n \in \mathbb{N}}$ is transcient.

M/G/1 queue

Branching process

Stability

Recurrence and transcience

Positive recurrence

Related work and references

Positive recurrence

- What we have shown so far :
  - $\rho \leq 1$: $(Y_n)_{n \in \mathbb{N}}$ is recurrent.
  - $\rho > 1$: $(Y_n)_{n \in \mathbb{N}}$ is transcient.

Positive recurrence

- What we have shown so far :
  - $\rho \leq 1$: $(Y_n)_{n \in \mathbb{N}}$ is recurrent.
  - $\rho > 1$: $(Y_n)_{n \in \mathbb{N}}$ is transcient.

- Branching process result :
  - $\rho < 1$: the mean population size is finite,
  - $\rho = 1$: the mean population size is infinite.

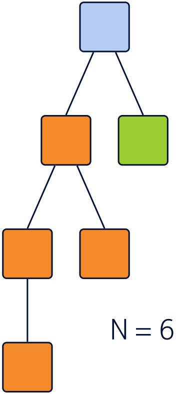
Positive recurrence

- What we have shown so far :
  - $\rho \leq 1$: $(Y_n)_{n \in \mathbb{N}}$ is recurrent.
  - $\rho > 1$: $(Y_n)_{n \in \mathbb{N}}$ is transcient.

- Branching process result :
  - $\rho < 1$: the mean population size is finite,
  - $\rho = 1$: the mean population size is infinite.
- Queueing reformulation :
  - $\rho < 1$: $(Y_n)_{n \in \mathbb{N}}$ is **positive** recurrent,
  - $\rho = 1$: $(Y_n)_{n \in \mathbb{N}}$ is **null** recurrent.

Mean population size

- N = total population size

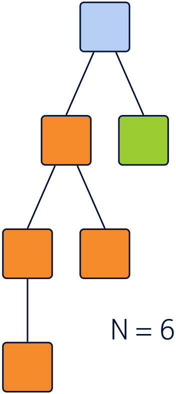


Mean population size

- N = total population size
- Mean population size

$$\mathbb{E}(N) = 1 + \sum_{k=0}^{+\infty} k\mathbb{E}(N) \times p_k$$

$$= 1 + \left(\sum_{k=0}^{+\infty} kp_k\right) \times \mathbb{E}(N)$$



N = 6

  Public

NOKIA Bell Labs TELECOM ParisTech

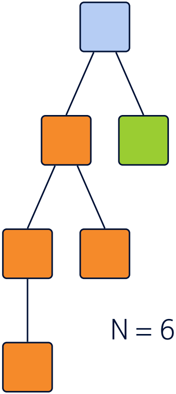
Mean population size

- N = total population size
- Mean population size

$$\mathbb{E}(N) = 1 + \sum_{k=0}^{+\infty} k\mathbb{E}(N) \times p_k$$
$$= 1 + \left(\sum_{k=0}^{+\infty} kp_k \right) \times \mathbb{E}(N)$$

- Mean number of children per node

$$\sum_{k=0}^{+\infty} kp_k = \rho$$



N = 6

Mean population size

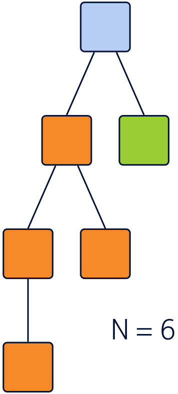
- N = total population size
- Mean population size

$$\mathbb{E}(N) = 1 + \sum_{k=0}^{+\infty} k\mathbb{E}(N) \times p_k$$
$$= 1 + \left(\sum_{k=0}^{+\infty} kp_k\right) \times \mathbb{E}(N)$$

- Mean number of children per node

$$\sum_{k=0}^{+\infty} kp_k = \rho$$

- We obtain $\mathbb{E}(N) = 1 + \rho\mathbb{E}(N)$

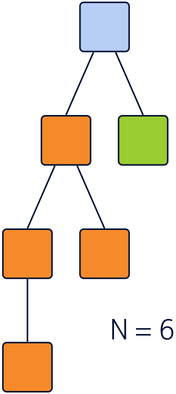


  Public

NOKIA Bell Labs TELECOM ParisTech

Mean population size

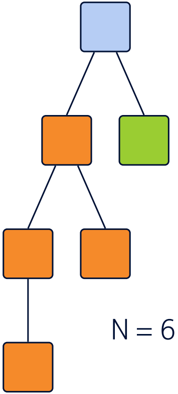
- We obtain $\mathbb{E}(N) = 1 + \rho\mathbb{E}(N)$



Mean population size

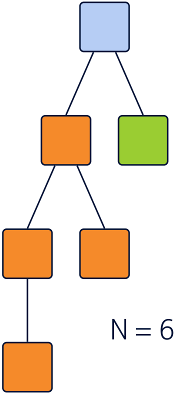
- We obtain $\mathbb{E}(N) = 1 + \rho\mathbb{E}(N)$

- If $\rho > 1$, then $\mathbb{E}(N) > 1 + \mathbb{E}(N) > \mathbb{E}(N)$
  $\rightarrow$ $\mathbb{E}(N) = +\infty$



Mean population size

- We obtain $\mathbb{E}(N) = 1 + \rho\mathbb{E}(N)$
- If $\rho > 1$, then $\mathbb{E}(N) > 1 + \mathbb{E}(N) > \mathbb{E}(N)$
  $\rightarrow$ $\mathbb{E}(N) = +\infty$
- If $\rho = 1$, then $\mathbb{E}(N) = 1 + \mathbb{E}(N)$
  $\rightarrow$ $\mathbb{E}(N) = +\infty$



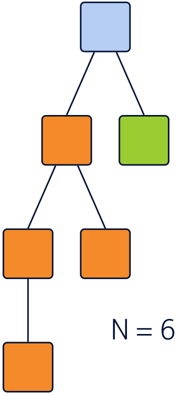
Mean population size

- We obtain $\mathbb{E}(N) = 1 + \rho\mathbb{E}(N)$

- If $\rho > 1$, then $\mathbb{E}(N) > 1 + \mathbb{E}(N) > \mathbb{E}(N)$
  → $\mathbb{E}(N) = +\infty$

- If $\rho = 1$, then $\mathbb{E}(N) = 1 + \mathbb{E}(N)$
  → $\mathbb{E}(N) = +\infty$

- What if $\rho < 1$? We can't conclude !
  → **If** $\mathbb{E}(N) < +\infty$**, then** $\mathbb{E}(N) = \frac{1}{1-\rho}$
  → Study the population size generation by generation

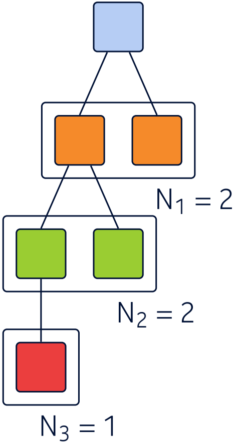


  Public

NOKIA Bell Labs TELECOM ParisTech

Mean number of nodes per generation

- N_k = number of nodes at generation k

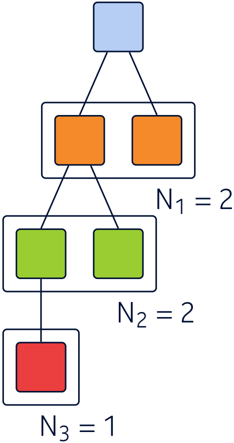


  Public

Mean number of nodes per generation

- N_k = number of nodes at generation k
- For each $k \geq 1$,

$$\begin{aligned}\mathbb{E}(N_k) &= \mathbb{E}(\mathbb{E}(N_k | N_{k-1})) \\ &= \mathbb{E}(\rho N_{k-1}) \\ &= \rho \mathbb{E}(N_{k-1})\end{aligned}$$

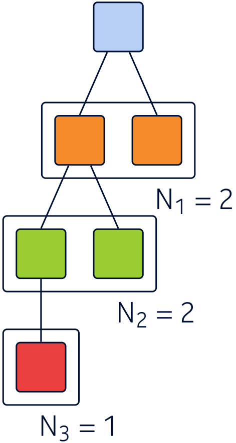


Mean number of nodes per generation

- N_k = number of nodes at generation k
- For each $k \geq 1$,

$$\begin{aligned}\mathbb{E}(N_k) &= \mathbb{E}(\mathbb{E}(N_k | N_{k-1})) \\ &= \mathbb{E}(\rho N_{k-1}) \\ &= \rho \mathbb{E}(N_{k-1})\end{aligned}$$

- $\mathbb{E}(N_0) = 1$



Mean number of nodes per generation

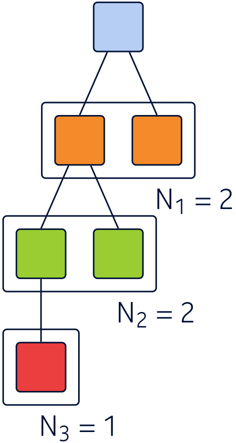
- N_k = number of nodes at generation k

- For each $k \geq 1$,

$$\begin{aligned}\mathbb{E}(N_k) &= \mathbb{E}(\mathbb{E}(N_k | N_{k-1})) \\ &= \mathbb{E}(\rho N_{k-1}) \\ &= \rho \mathbb{E}(N_{k-1})\end{aligned}$$

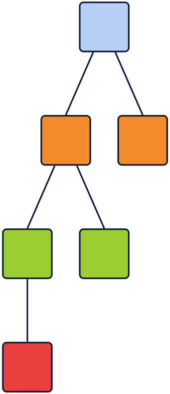
- $\mathbb{E}(N_0) = 1$

- By induction, $\mathbb{E}(N_k) = \rho^k$



Mean population size

- By induction, $\mathbb{E}(N_k) = \rho^k$

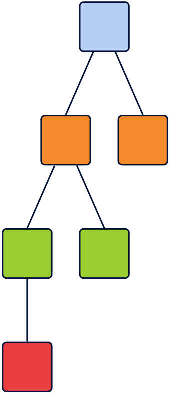


  Public

NOKIA Bell Labs TELECOM ParisTech

Mean population size

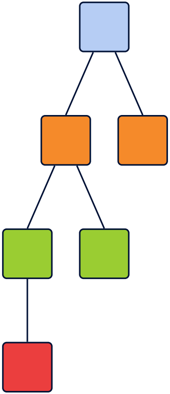
- By induction, $\mathbb{E}(N_k) = \rho^k$
- Population size : $N = \sum_{k=0}^{+\infty} N_k$



Mean population size

- By induction, $\mathbb{E}(N_k) = \rho^k$
- Population size : $N = \sum_{k=0}^{+\infty} N_k$
- Mean population size

$$\mathbb{E}(N) = \mathbb{E}\left(\sum_{k=0}^{+\infty} N_k\right) = \sum_{k=0}^{+\infty} \mathbb{E}(N_k) = \sum_{k=0}^{+\infty} \rho^k$$



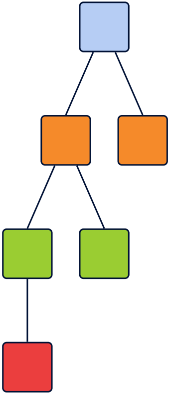
Mean population size

- By induction, $\mathbb{E}(N_k) = \rho^k$
- Population size : $N = \sum_{k=0}^{+\infty} N_k$
- Mean population size

$$\mathbb{E}(N) = \mathbb{E}\left(\sum_{k=0}^{+\infty} N_k\right) = \sum_{k=0}^{+\infty} \mathbb{E}(N_k) = \sum_{k=0}^{+\infty} \rho^k$$

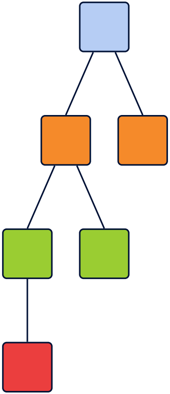
Hence,

$$\mathbb{E}(N) = \begin{cases} +\infty & \text{if } \rho \geq 1 \\ \frac{1}{1-\rho} & \text{if } \rho < 1 \end{cases}$$



Length of a busy period

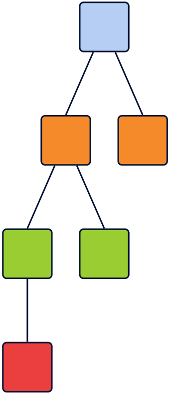
- B = time length of the busy period



Length of a busy period

- B = time length of the busy period
- Mean length of the busy period

$$
\begin{aligned}
\mathbb{E}(B) &= \frac{1}{\mu} + \sum_{k=0}^{+\infty} k\mathbb{E}(B) \times p_k \\
&= \frac{1}{\mu} + \left(\sum_{k=0}^{+\infty} k p_k\right) \times \mathbb{E}(B) \\
&= \frac{1}{\mu} + \rho \mathbb{E}(B)
\end{aligned}
$$



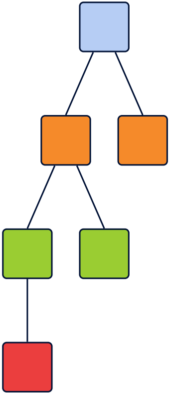
Length of a busy period

- B = time length of the busy period
- Mean length of the busy period

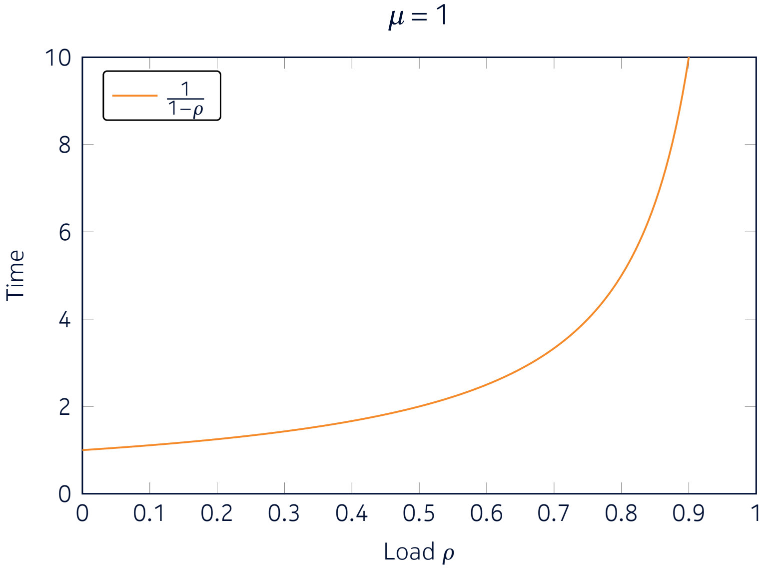
$$\begin{aligned}\mathbb{E}(B) &= \frac{1}{\mu} + \sum_{k=0}^{+\infty} k\mathbb{E}(B) \times p_k \\ &= \frac{1}{\mu} + \left(\sum_{k=0}^{+\infty} kp_k\right) \times \mathbb{E}(B) \\ &= \frac{1}{\mu} + \rho\mathbb{E}(B)\end{aligned}$$

By the same reasoning, we obtain

$$\mathbb{E}(B) = \begin{cases} +\infty & \text{if } \rho \geq 1 \\ \frac{1}{\mu}\frac{1}{1-\rho} & \text{if } \rho < 1 \end{cases}$$



Length of a busy period



M/G/1 queue

Branching process

Stability

Recurrence and transcience

Positive recurrence

Related work and references

Related work

- M/G/1 queue with LCFS service discipline
- GI/M/1 queue
- Pooling systems

Bibliography

David G. Kendall (1951). “Some Problems in the Theory of Queues”. In : Journal of the Royal Statistical Society. Series B (Methodological) 13.2, p. 151–185.

David G. Kendall (1953). “Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain”. In : The Annals of Mathematical Statistics 24.3, p. 338–354.

Pierre Brémaud (2013). Markov Chains : Gibbs Fields, Monte Carlo Simulation, and Queues. Texts in Applied Mathematics. Springer New York.

Berestycki Nathanaël (2014). Applied Probability - Part II. Course. University of Cambridge.

Yoram Clapper (2017). Branching Processes in Queuing Theory. Bachelor’s thesis mathematics. University of Amsterdam.

NOKIA