# Combinatorial Algorithm for Calculating Blocking Probabilities in Multicast Networks

Samuli Aalto & Jorma Virtamo

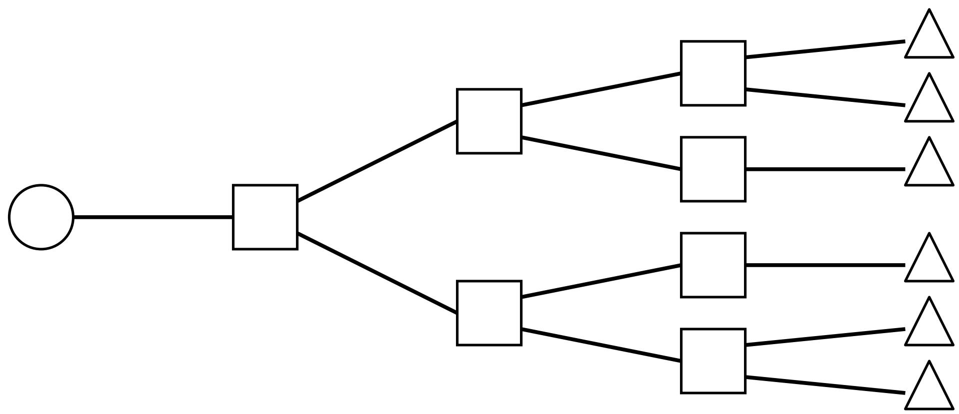
Laboratory of Telecommunications Technology

Helsinki University of Technology

samuli.aalto@hut.fi

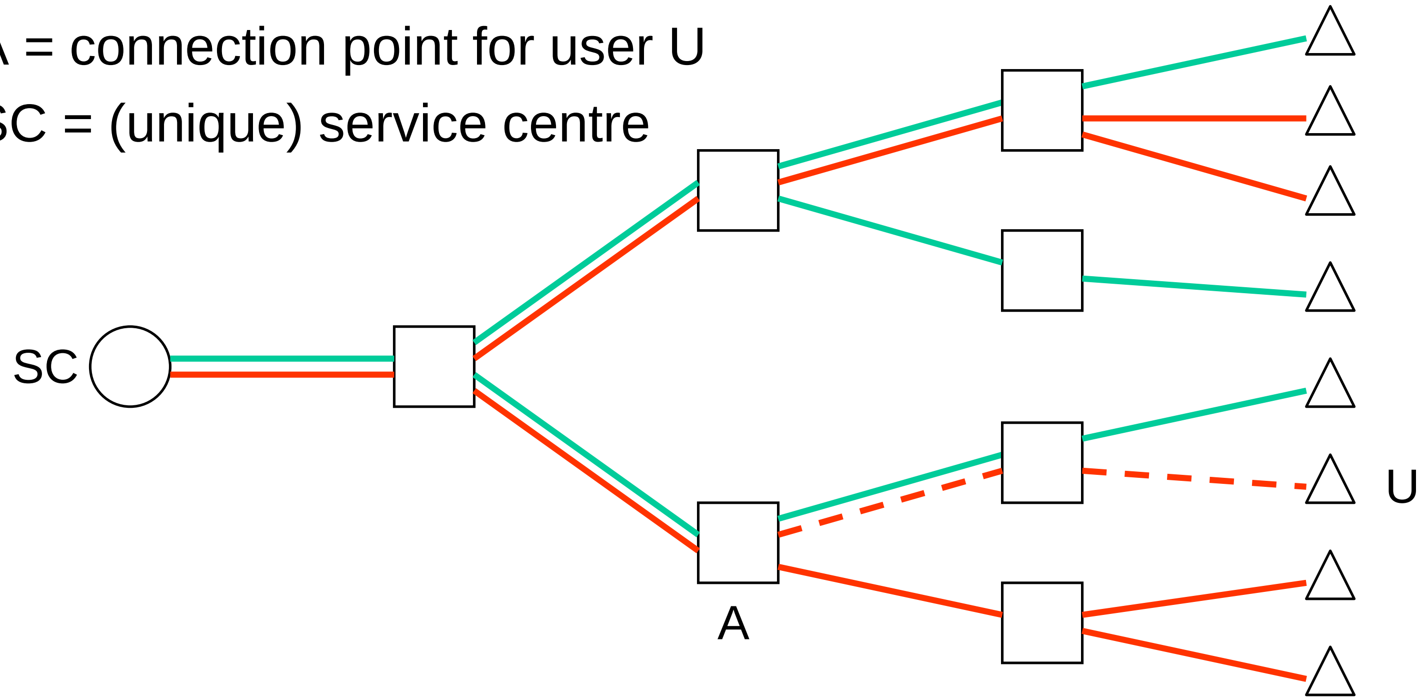
# Multicast network model

- **Setup** (consider e.g. distribution of TV or radio channels):
  - Unique **service center** offers a variety of **channels**
  - Each channel $i \in I$ is delivered by a **multicast connection** with **dynamic membership**
  - Each multicast connection uses the same **multicast tree** consisting of links $j \in J$ ($\Rightarrow$ **fixed routing)**
  - Service center located at the **root node** of the multicast tree
  - Users $u \in U$ located at the **leaf nodes** of the multicast tree



# Multicast connections with dynamic membership

- U = new user of channel 'red'
- A = connection point for user U
- SC = (unique) service centre



# Link states

- Consider first a network with infinite link capacities
- Let

$$Y_{ji} = 1\{\text{connection } i \text{ active on link } j\}$$

- Detailed link state (for any link $j \in J$)

$$\mathbf{Y}_j = (Y_{ji}; i \in I) \in S_Y := \{0,1\}^I$$

- Link state (for any link $j \in J$)

$$N_j = \sum_{i \in I} Y_{ji} \in S := \{0,1,\ldots,I\}$$

# Stationary state probabilities in a network with infinite link capacities

- Assume that the probabilities of the detailed **leaf link states** (which depend on the user population model adopted) are known, and denote them by

$$\pi_u(\mathbf{y}) := P\{\mathbf{Y}_u = \mathbf{y}\}$$

  – where $\mathbf{y} \in S_Y = \{0,1\}^I$

- Due to infinite link capacities and independent behaviour of the user populations, it follows that the probabilities of the detailed **network states** are also known:

$$\pi(\mathbf{x}) := P\{\mathbf{X} = \mathbf{x}\} = \prod_{u \in U} P\{\mathbf{Y}_u = \mathbf{y}_u\} = \prod_{u \in U} \pi_u(\mathbf{y}_u)$$

  – where $\mathbf{x} = (\mathbf{y}_u;\, u \in U) \in \Omega := \{0,1\}^{U \times I}$

# Stationary state probabilities in a network with finite link capacities

- If the **Truncation Principle** applies (which depends on the user population model adopted), then

$$\tilde{\pi}(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\sum_{\mathbf{x}\in\tilde{\Omega}} \pi(\mathbf{x})}$$

  – where $\mathbf{x} = (\mathbf{y}_u;\ u \in U) \in \tilde{\Omega}$ and

$$\tilde{\Omega} = \text{set of allowed network states}$$

# Blocking probability

- $B^t_{ui}$ = **time blocking** for user population $u$ and connection $i$
  = stationary probability of such network states in which
  a new request originating from user population $u$ to join
  connection $i$ would be rejected due to lack of link capacity

- How to calculate $B^t_{ui}$?

# Calculation of blocking probabilities (1)

- 1st possibility: closed form expression

$$B_{ui}^{t} := 1 - \sum_{\mathbf{x} \in \tilde{\Omega}_{ui}} \tilde{\pi}(\mathbf{x}) = 1 - \frac{\sum_{\mathbf{x} \in \tilde{\Omega}_{ui}} \pi(\mathbf{x})}{\sum_{\mathbf{x} \in \tilde{\Omega}} \pi(\mathbf{x})}$$

  - where

$$\tilde{\Omega}_{ui} = \text{set of nonblocking network states for } (u,i)$$

$$\tilde{\Omega} = \text{set of allowed network states}$$

- **Problem**: computationally extremely complex
  - exponential growth both in $U$ and $I$

# Calculation of blocking probabilities (2)

- 2nd possibility: recursive algorithm **exact** (see [4,5])

$$B_{ui}^{t} = 1 - \frac{\sum_{\mathbf{y} \in S_Y} Q_J^{ui}(\mathbf{y})}{\sum_{\mathbf{y} \in S_Y} Q_J(\mathbf{y})}$$

  – where probabilities $Q_j^{ui}(\mathbf{y})$ and $Q_j(\mathbf{y})$ can be calculated recursively (from the common link $J$ back to leaf links $u$)

- **Problem**: computationally complex
  – linear growth in $U$ but (still) exponential growth in $I$

# Calculation of blocking probabilities (3)

- 3rd possibility: **new** recursive algorithm **combi**

$$B_{ui}^{t} = 1 - \frac{\sum_{n \in S} Q_J^{ui}(n)}{\sum_{n \in S} Q_J(n)}$$

  – where probabilities $Q_j^{ui}(n)$ and $Q_j(n)$ can be calculated recursively (from the common link $J$ back to leaf links $u$)

- **Problem**: computationally resonable but …
  … restrictive assumptions have to be made!

# Restrictive assumptions

- (i) All receivers have a uniform preference distribution when making a choice (to join) between the multicast connections
- (ii) The mean holding time for any receiver to be joined to any connection is the same
- (iii) The capacity needed to carry any multicast connection in any link is the same
  - Make connections symmetric!
  - Users and network may still be “unsymmetrical”

# Basic results (1)

- Connections symmetric $\Rightarrow$
  - Whenever there are $n$ connections active on any leaf link $u \in U$, each possible index combination $\{i_1,\ldots,i_n\}$ is equally probable

- This and the independence of the user populations $\Rightarrow$
  - Whenever there are n connections active on any link $j \in J$, each possible index combination $\{i_1,\ldots,i_n\}$ is equally probable

- Consequence:
  - Combinatorics can be utilized

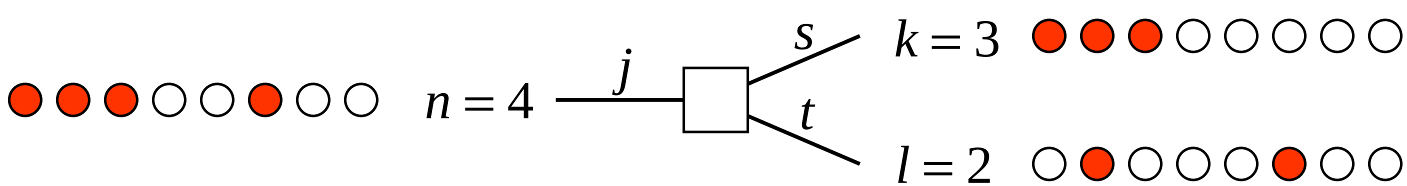
# Basic results (2)

- If link $j$ has two downstream neighbouring links $(s,t)$, then

$$\max\{N_s, N_t\} \le N_j \le \min\{N_s + N_t, I\}$$

- Assume (here) that $N_s = k \ge l = N_t$. Then

$$P\{N_j = n \mid N_s = k, N_t = l\} = \frac{\binom{k}{l-(n-k)}\binom{I-k}{n-k}}{\binom{I}{l}}$$



# Algorithm (1)

- Define (for all $j \in J$):

$$Q_j(n) = P\{N_j = n; N_k \le C_k, \forall k \in M_j\}$$

$$Q_j^{ui}(n) = P\{N_j^{(i)} = n; N_k^{(i)} \le C_k - 1, \forall k \in M_j \cap R_u; \quad N_k \le C_k, \quad \forall k \in M_j \setminus R_u\}$$

- Then time blocking probability for class $(u,i)$ is

$$B_{ui}^t = 1 - \frac{P\{\mathbf{X} \in \tilde{\Omega}_{ui}\}}{P\{\mathbf{X} \in \tilde{\Omega}\}} = 1 - \frac{\sum_{n=0}^{C_J - 1} Q_J^{ui}(n)}{\sum_{n=0}^{C_J} Q_J(n)}$$

## Algorithm (2)

- Recursion 1 to calculate the denominator $Q_j(n)$:

$$Q_j(n) = \begin{cases} T_j[\pi_j](n), & j \in U \\ T_j[\bigotimes_{k \in N_j} Q_k](n), & j \notin U \end{cases}$$

  - where probabilities $\pi_j(n) = P\{N_j = n\}$ depend on the chosen user population model
- Truncation operator 1:
  - Let $f$ be any real-valued function defined on $S = \{0,1,\ldots,I\}$.
  - Then define

$$T_j[f](n) = f(n) \cdot 1\{n \le C_j\}$$

# Algorithm (3)

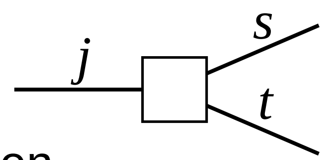
- Definition of operator $\otimes$:
  - Let $f$ and $g$ be any real-valued function defined on $S = \{0,1,\ldots,I\}$.
  - Then define

$$[f \otimes g](n) = \sum_{k=0}^{n} \sum_{l=n-k}^{n} s(n \mid k,l) f(k) g(l)$$

- where

$$s(n \mid k,l) = \frac{\binom{\max\{k,l\}}{k+l-n}\binom{I-\max\{k,l\}}{n-\max\{k,l\}}}{\binom{I}{\min\{k,l\}}}$$

# Algorithm (4)




- Key result:
  - If link $j$ has two downstream neighbouring links ($s$,$t$), then

$$P\{N_j = n\} = \sum_{k=0}^{n} \sum_{l=n-k}^{n} s(n \mid k,l) P\{N_s = k\} P\{N_t = l\}$$

  - In other words,

$$\pi_j(n) = [\pi_s \otimes \pi_t](n)$$

  - Proved by a “sampling without replacement” argument!

# Algorithm (5)

- Recursion 2 to calculate the numerator $Q_J^{ui}(n)$ :

$$Q_j^{ui}(n) = \begin{cases} T_u^{\circ}[\pi_u^{(i)}](n), & j = u \\ T_j^{\circ}[Q_{D_u(j)}^{ui} \odot \bigotimes_{k \in N_j \setminus R_u} Q_k](n), & j \in R_u \setminus \{u\} \end{cases}$$

  - where probabilities $\pi_u^{(i)}(n) = P\{N_u^{(i)} = n\}$ depend on the chosen user population model
- Truncation operator 2:
  - Let $f$ be any real-valued defined on $S = \{0,1,\ldots,I\}$.
  - Then define

$$T_j^{\circ}[f](n) = f(n) \cdot 1\{n \le C_j - 1\}$$

# Algorithm (6)

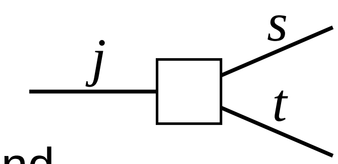
- Definition of operator $\odot$:
  - Let $f$ and $g$ be any real-valued function defined on $S = \{0,1,\ldots,I\}$.
  - Then define

$$[f \odot g](n) = \sum_{k=0}^{n} \sum_{l=n-k}^{n} s^{\circ}(n \mid k,l)\, f(k)\left[\left(1-\frac{l}{I}\right)g(l) + \frac{l+1}{I}\, g(l+1)\right]$$

  - where

$$s^{\circ}(n \mid k,l) = \frac{\binom{\max\{k,l\}}{k+l-n}\binom{I-1-\max\{k,l\}}{n-\max\{k,l\}}}{\binom{I-1}{\min\{k,l\}}}$$

# Algorithm (7)




- Key result:
  - If link $j$ has two downstream neighbouring links ($s$,$t$), and link $s$ belongs to the interesting route, i.e. $s = D_u(j)$, then

$$P\{N_j^{(i)} = n\} = \sum_{k=0}^{n} \sum_{l=n-k}^{n} s^{\circ}(n \mid k,l) P\{N_s^{(i)} = k\}$$
$$\times [(1-\tfrac{l}{I}) P\{N_t = l\} + p(\tfrac{l+1}{I}) P\{N_t = l+1\}]$$

  - In other words,

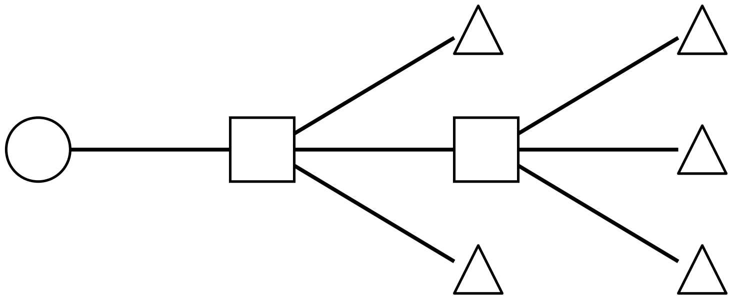
$$\pi_j^{(i)}(n) = [\pi_s^{(i)} \odot \pi_t](n)$$

  - Proved by another “sampling without replacement” argument!

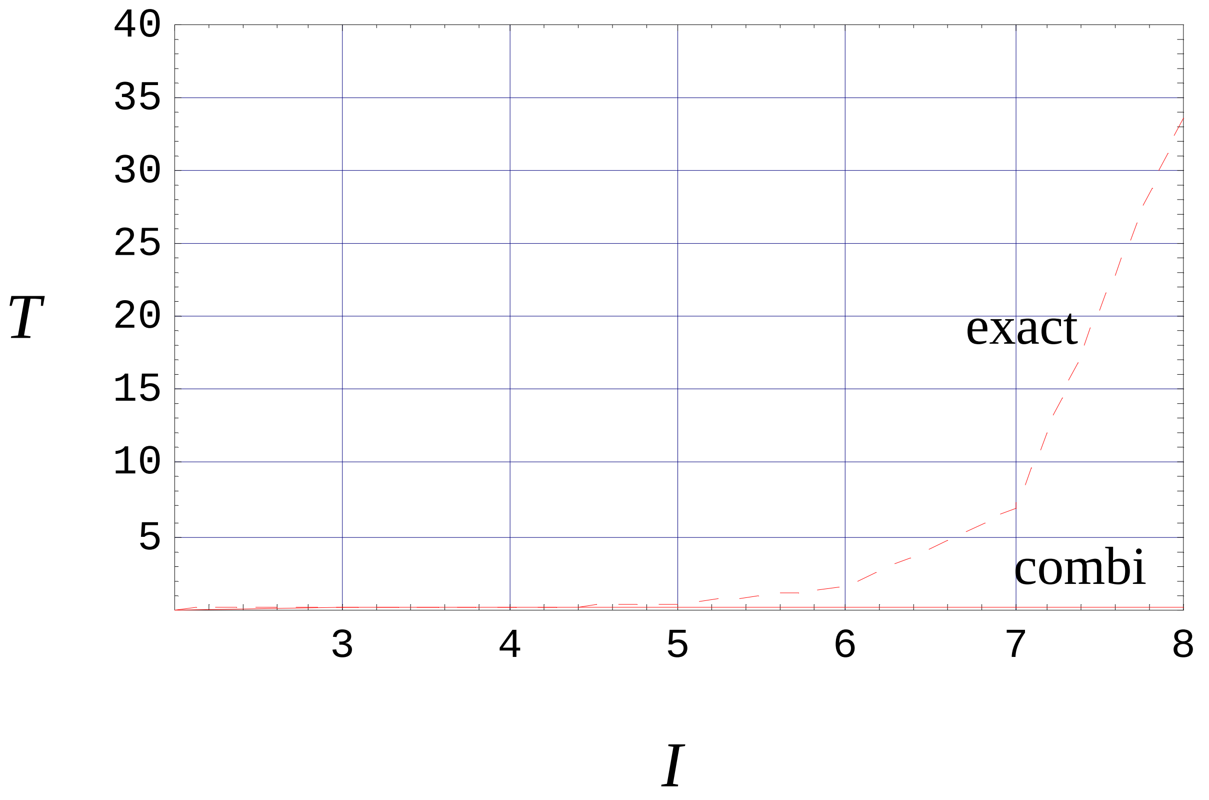
# Calculation of call blocking probabilities

- In the paper, a similar algorithm is derived for calculating call blocking probabilities
- Dependence on the user population model has to be taken carefully into account
  - Infinite user population model:
    - call blocking $B^c_{ui}$ equals time blocking $B^t_{ui}$
  - Single user model:
    - call blocking $B^c_{ui}$ equals time blocking in a modified network where user $u$ is removed

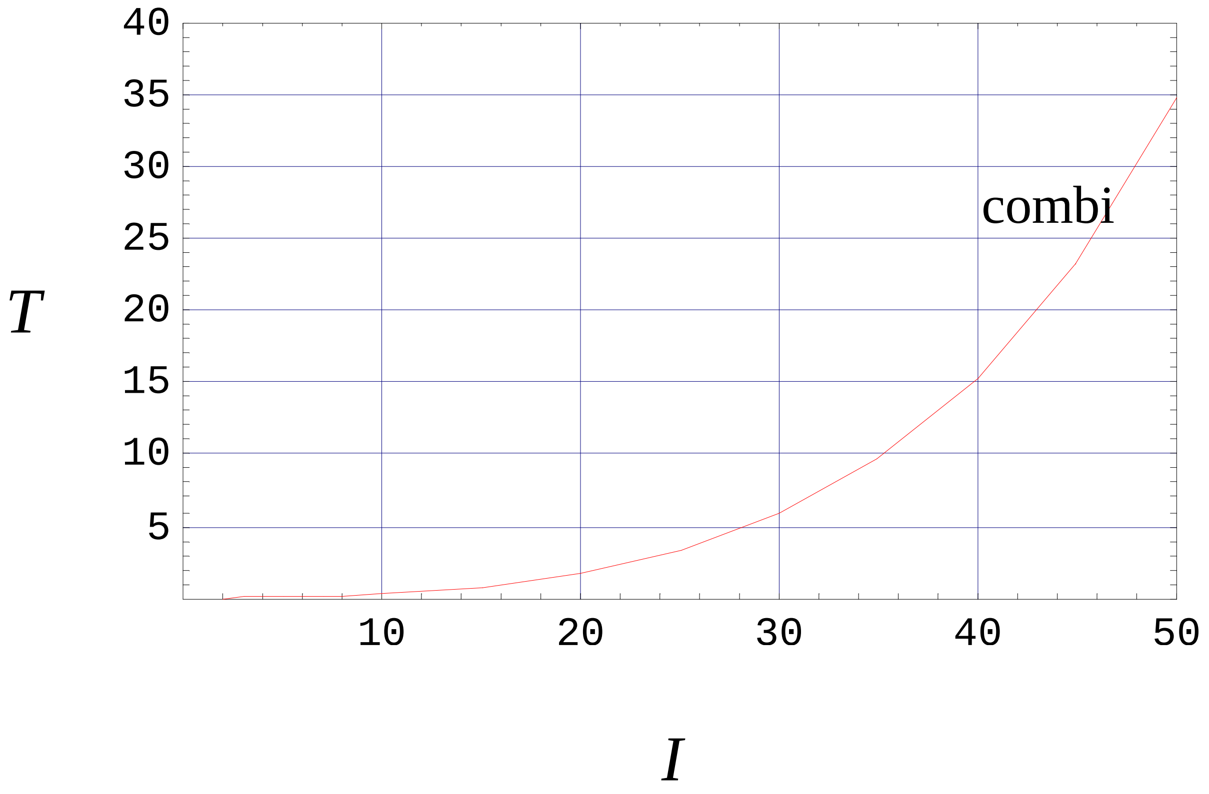
# Example network 1 (figure 2 in [4])



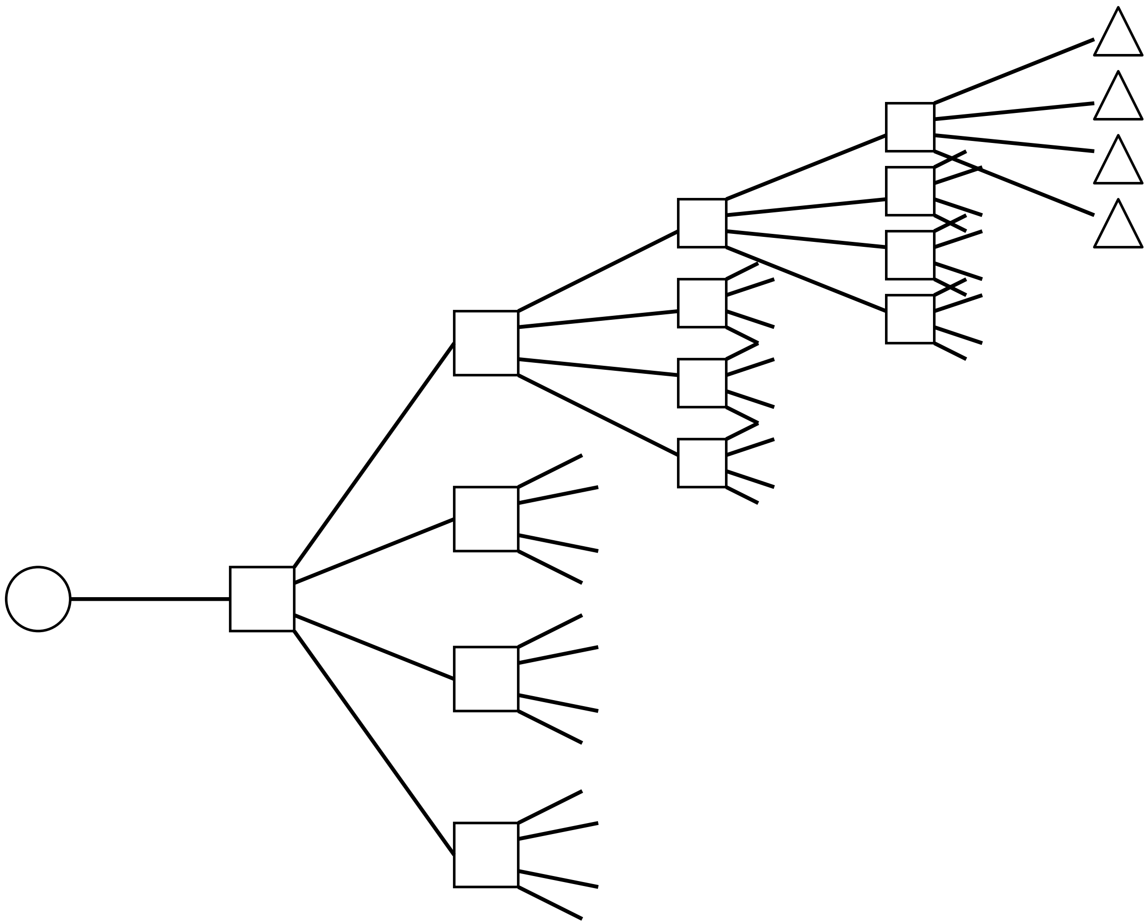
# Processing time T vs. nr of multicast connections I (normal scale)



# Processing time T vs. nr of multicast connections I (normal scale)



# Example network 2 (figure 5 in [5])



# THE END