Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Multi-channel Tree Algorithms for LTE RACH Reliable High Throughput Access

Alberto Martínez Alba
Murat Gürsu
Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Kommunikationsnetze

München, 27th September 2016

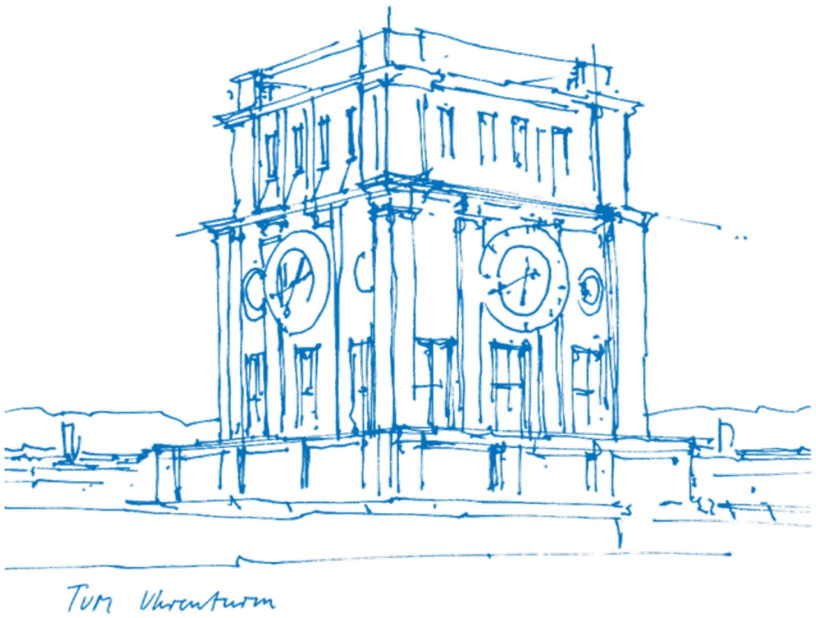


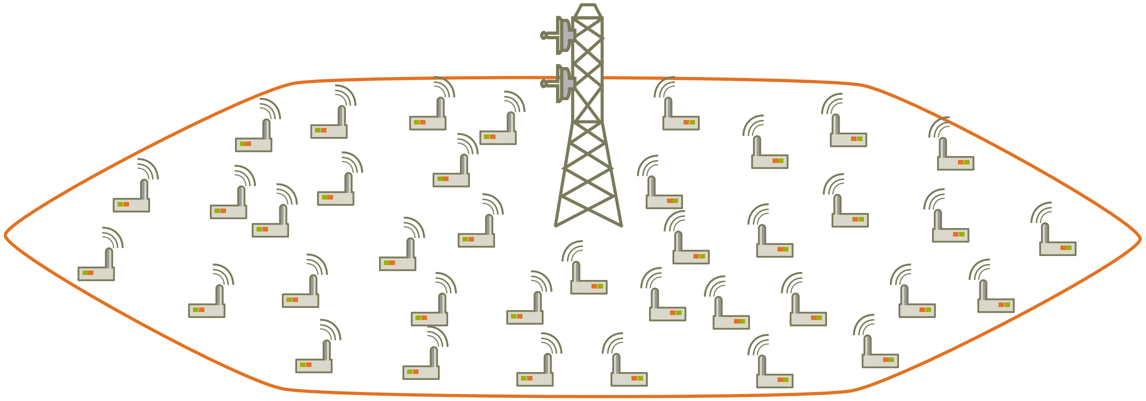
Table of Contents

Table of Contents

Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Motivation (I)

- LTE Radio Access Network (RAN) **overload control** [1]:
  - Large number of MTC devices
  - Network congestion
  - Protect the Uplink **Random Access Channel** (RACH) from overload



Congested network

Motivation (II)

- LTE RACH is based on (multi-channel) **Slotted ALOHA** access [2]
  - Slotted ALOHA cannot guarantee **reliability** [3]
  - **Tree Algorithms** can improve Slotted-ALOHA
- **Reliability:** ability of a network to **serve** some devices within the required time
- **Example:** 30,000 devices beta-distributed in 10 s [4]

	BO = 20 ms	BO = 240 ms	BO = 960 ms
Access success prob.	19.14%	23.84%	41.97%

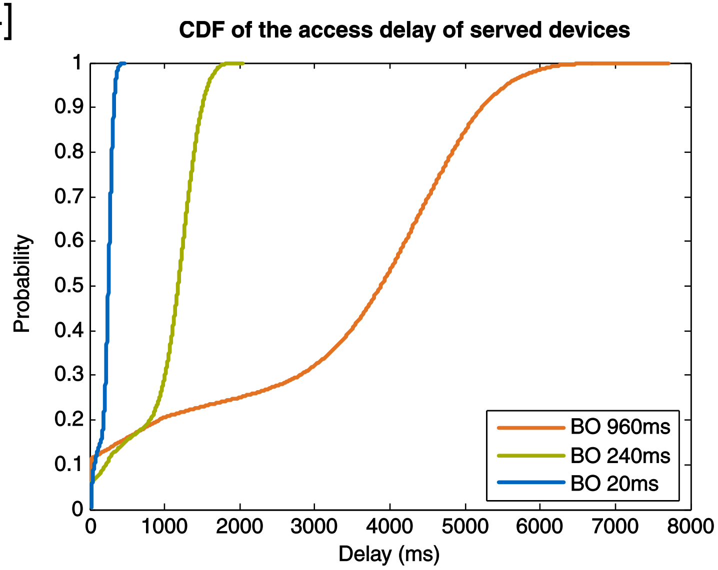


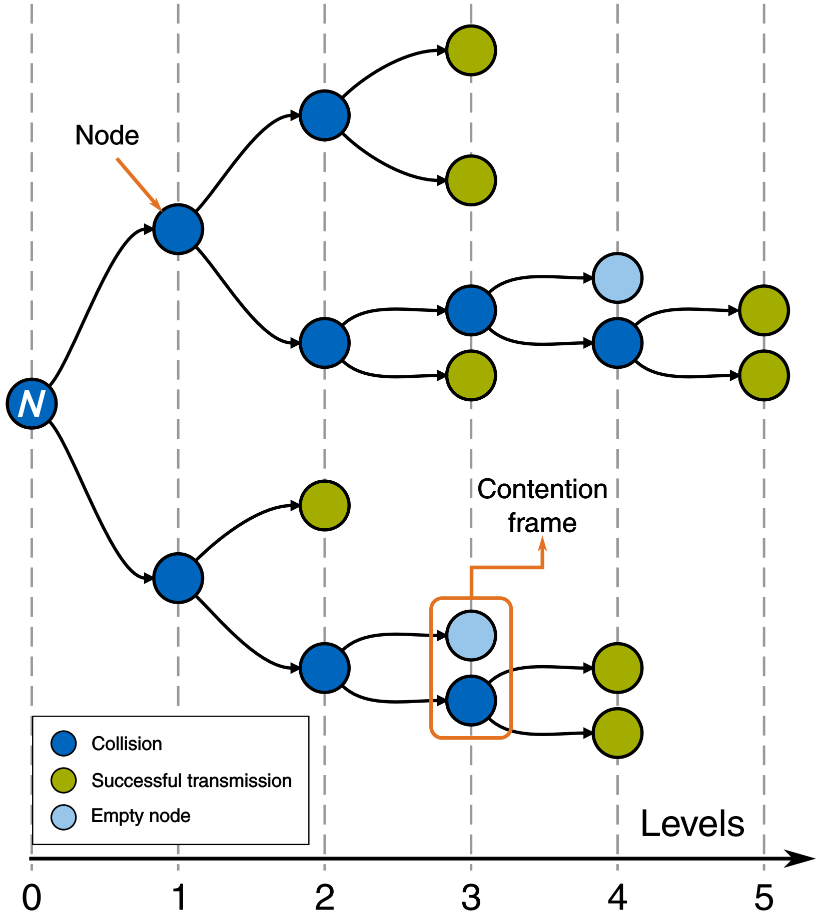
Table of Contents

LTE-RACH and Tree Algorithms

- Tree Algorithms are **Contention Resolution Algorithms** [5]
- Tree Algorithms are designed to guarantee **access success** [3][5]
- **Access delay** is still an issue
- LTE-RACH is **multi-channel** (orthogonal preambles) [6]
- Statistics of the access delay of **multi-channel Tree Algorithms** are needed

Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

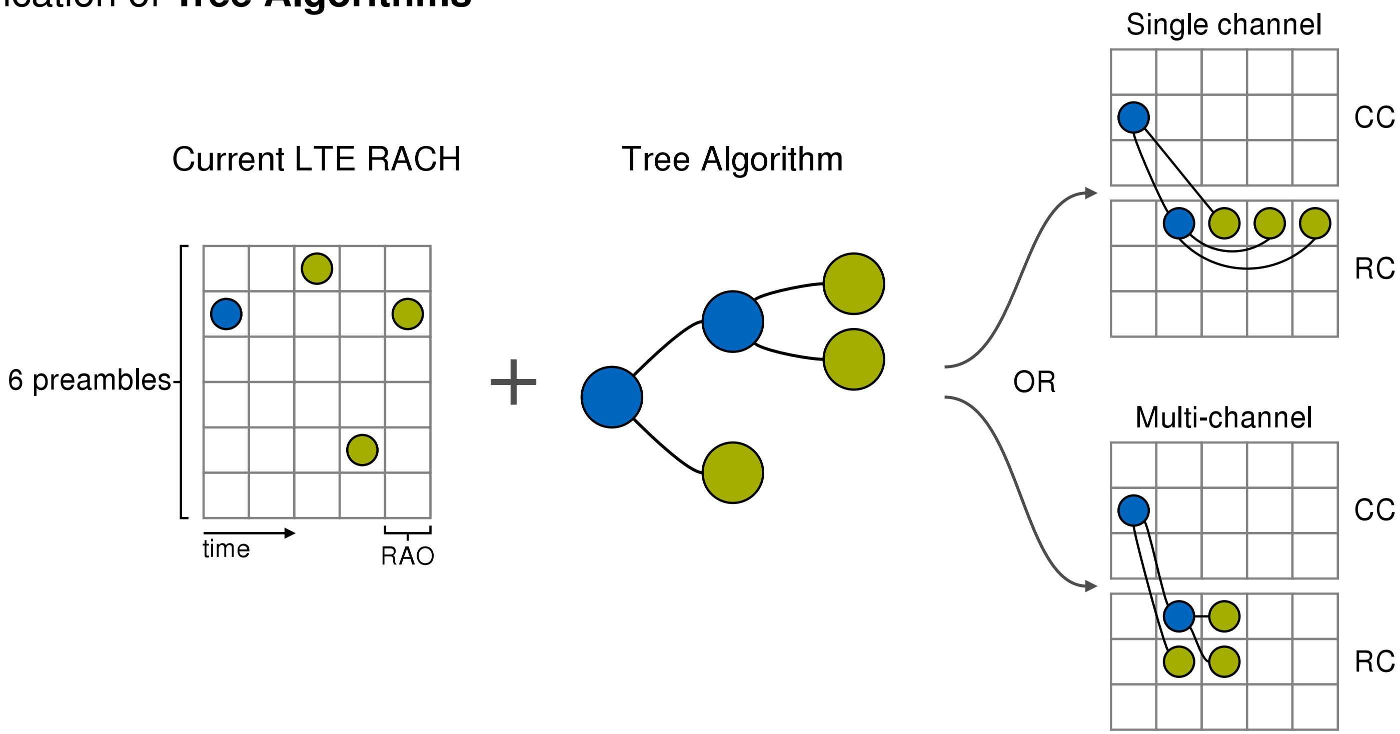
Example of a Tree Algorithm (I)



- $N = 8$ initial contenders
- No **new devices** allowed
- **Feedback** is required
- Collision $= q$ **new nodes**
- $S = 19$ **nodes** to complete the tree
- $L = 5$ **levels**

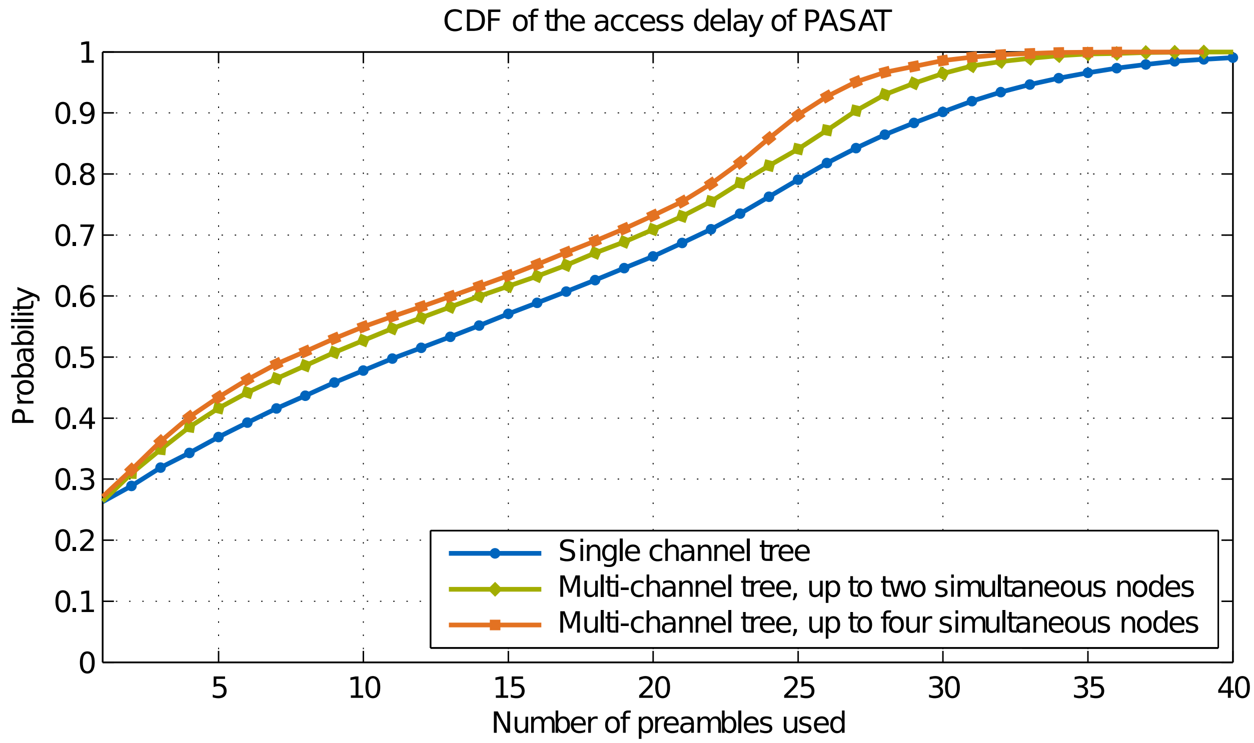
PASAT-Algorithm

- Key idea:
  - Separation of LTE RACH into **Contention Channel** (CC) and **Resolution Channel** (RC)
  - Application of **Tree Algorithms**



PASAT-Algorithm (II): simulation results

- For $N = 3000$ devices beta-distributed within 1 second (200 RAOs):



- **Conclusion**: in this case, multi-channel Tree Algorithms perform better.

Table of Contents

Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

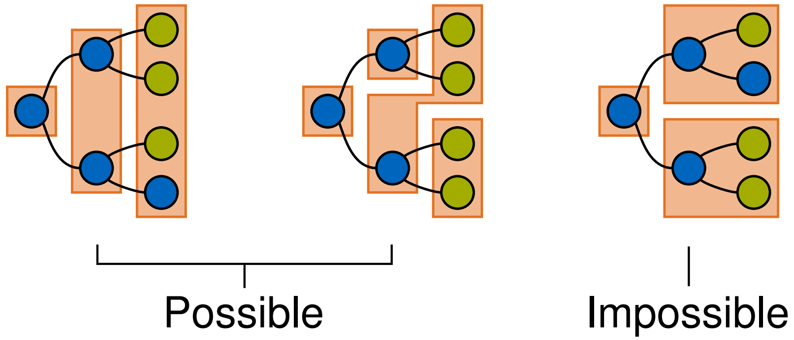
Multi-channel Tree Algorithms

Key idea:

Transmit multiple nodes at the same time $\longrightarrow$ Reduce access delay

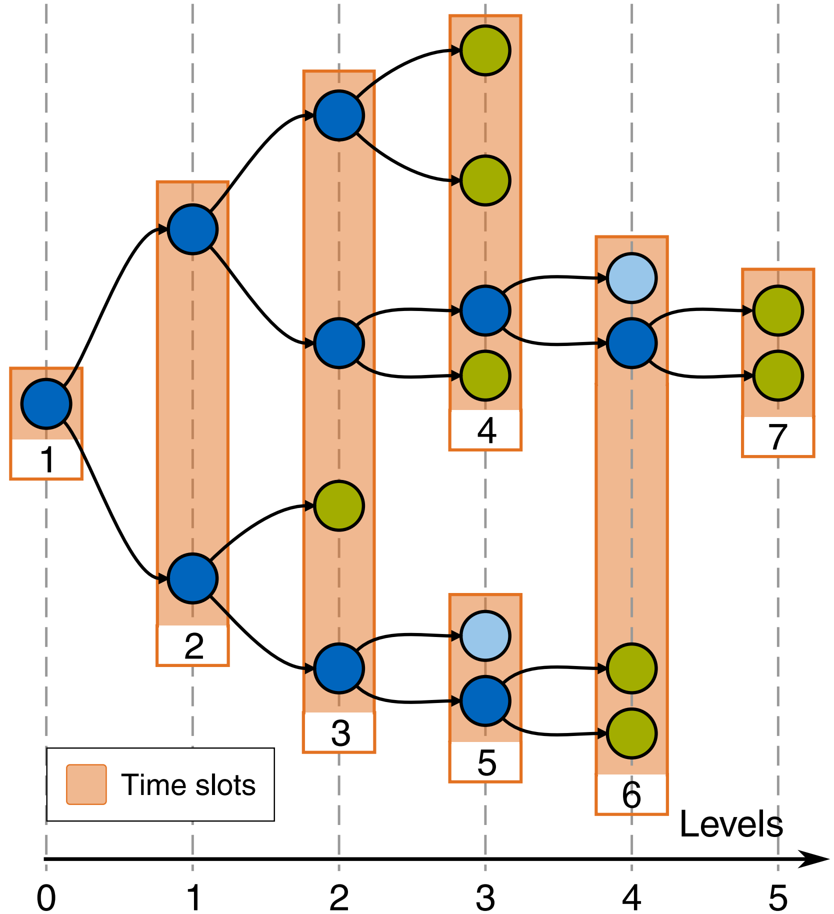
Limitation:

Only nodes that are not descendant from one another can be transmitted together



Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

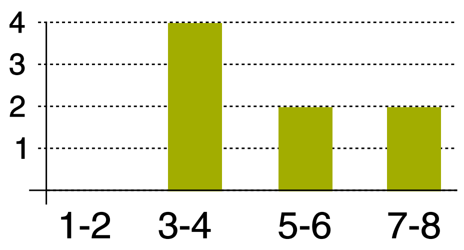
Example of a multi-channel Tree Algorithm



- $G = 2 \rightarrow 4$ nodes per slot
- No direct conversion from single channel
- Access delays:

$$\{3,4,4,4,6,6,7,7\}$$

- Average access delay: 5.125 slots
- Histogram of transmissions



State of the art of Tree Algorithms

- This statistics can be **found in the literature** about TAs:
  - Average, variance and probability-generating function (PGF) of the length of the tree [3][7][8][9]
  - Average and probability-generating function (PGF) of the access delay [9]
  - PMF of the number of levels to complete the tree [10]
  - PMF of the number of levels to successfully transmit [10]
- However, the following important statistics are **missing for multi-channel** trees:
  - CDF of the **length** of the tree
  - CDF of the **access delay**

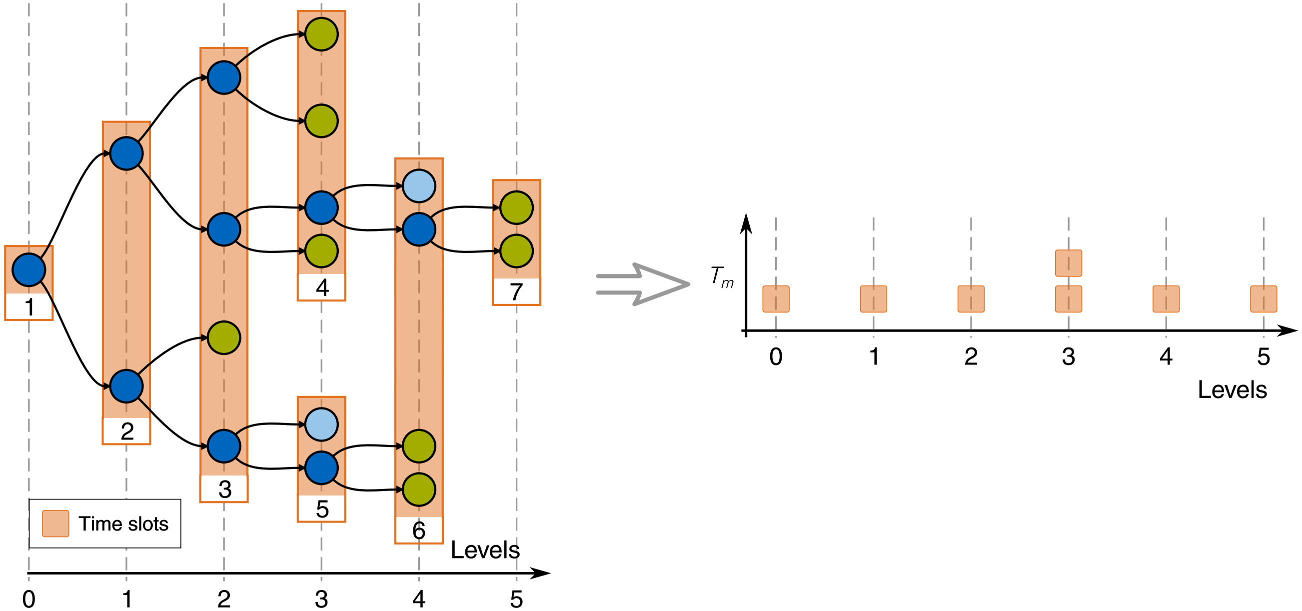
Table of Contents

Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

PMF of the length of a multi-channel Tree (I)

- Before facing the **CDF of the access delay**, the **CDF of the tree length** is needed
- The total number of **slots** T in a tree is given by:

$$T \approx 1 + \sum_{m=1}^{M} T_m$$



PMF of the length of a multi-channel Tree (II)

- We need to find out the **joint PMF** of all variables:

$$p_{T_1,\dots,T_M}(t_1,\dots,t_M) \quad \longrightarrow \quad p_T(t)$$

- Assuming a Markov process, the **joint distribution** will be [11]:

$$p_{T_1,\dots,T_M}(t_1,\dots,t_M) = p_{T_1}(t_1)p_{T_2|T_1}(t_2|t_1)\dots p_{T_M|T_{M_1}}(t_M|t_{M-1})$$

- There is a direct translation between **collisions** and slots:

$$X_m \rightarrow \text{Number of collisions in the level } m$$

$$p_{X_m|X_{m_1}}(x_m|x_{m-1}) \quad \longrightarrow \quad p_{T_m|T_{m_1}}(t_m|t_{m-1})$$

Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

PMF of the length of a multi-channel Tree (III)

Number of **collisions** is known

$\downarrow$

Prob. to have some **contenders**, given collisions

$\downarrow$

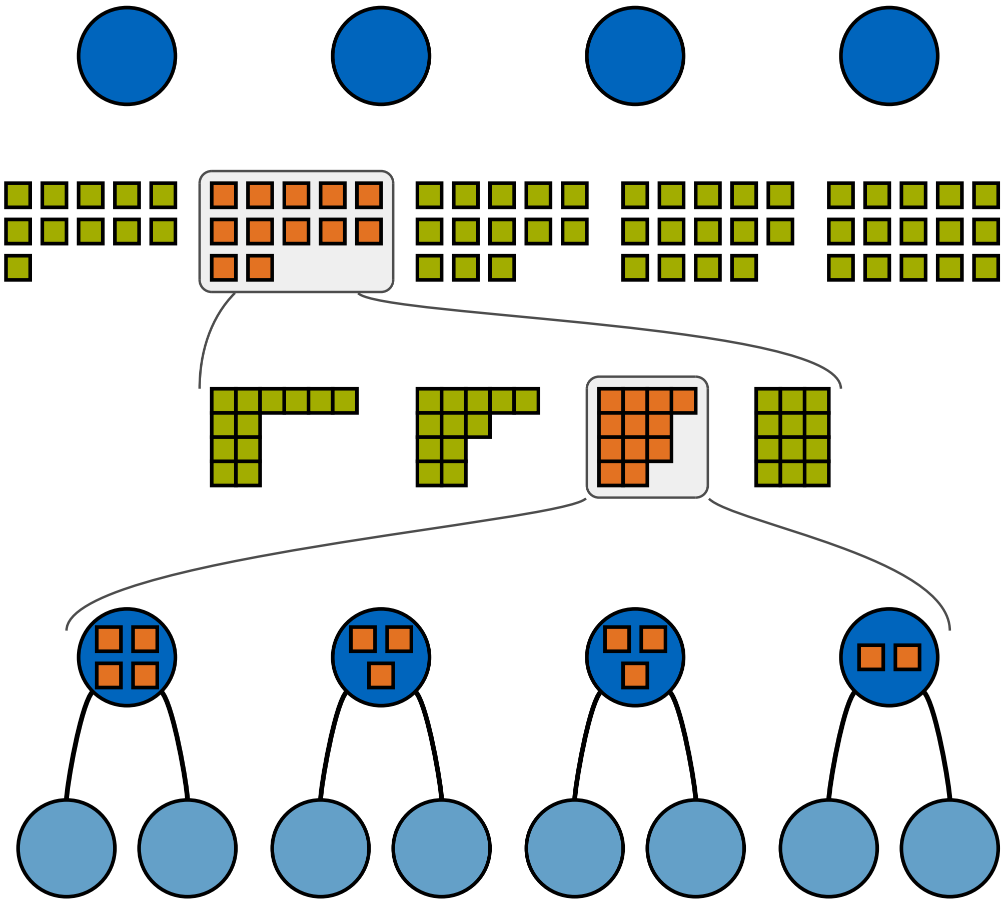
Prob. to have a certain **partition**, given collisions and contenders

$\downarrow$

Prob. to have generate **collisions** in each node, given collisions, contenders and a partition

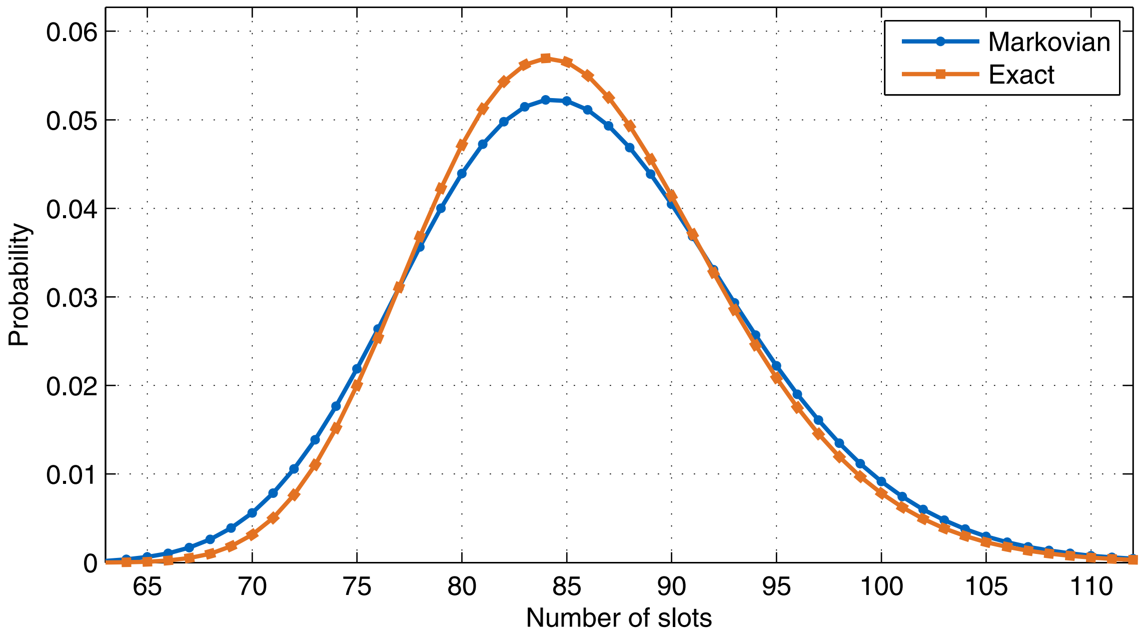
$\downarrow$

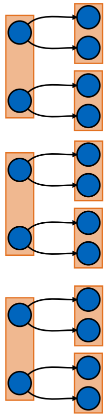
$p_{X_m|X_{m_1}}(x_m|x_{m-1})$



Results of the length of a multi-channel Tree (I)

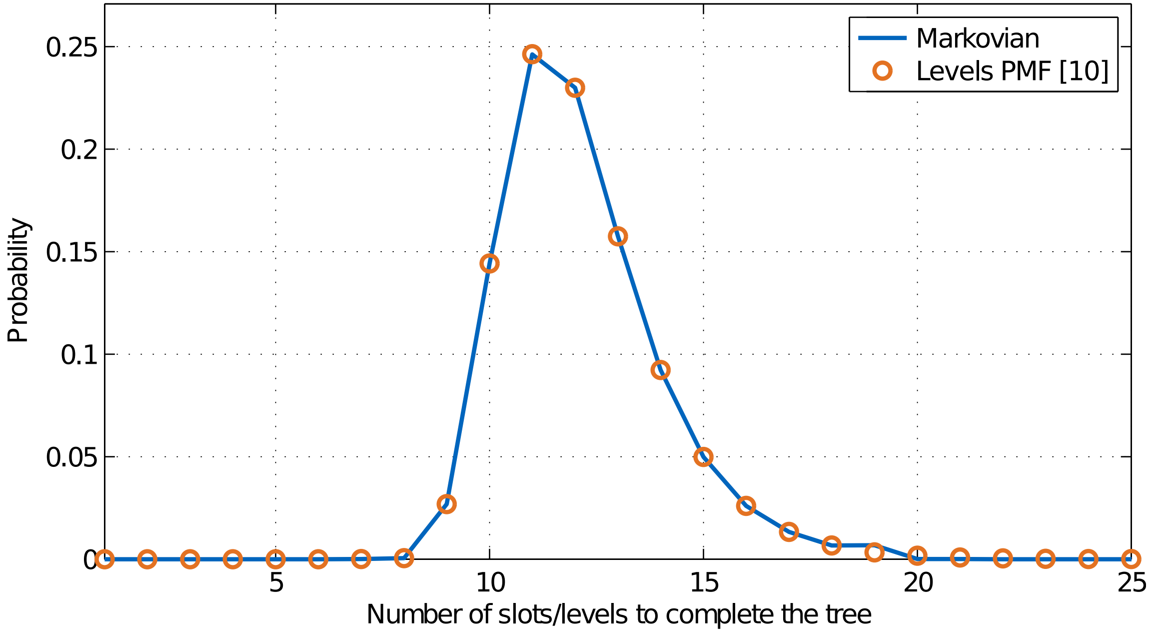
- If we set $G = 1 \longrightarrow$ **single channel** scenario
- We can use this value to **compare** our technique with the independent approach and the exact solution
- For $N = 60$:





Results of the length of a multi-channel Tree (II)

- Conversely, for $G \gg 1$, the number of used time slots should **converge** to the **number of levels**
- For $G = 25$ and $N = 60$ we obtain:



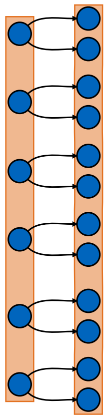
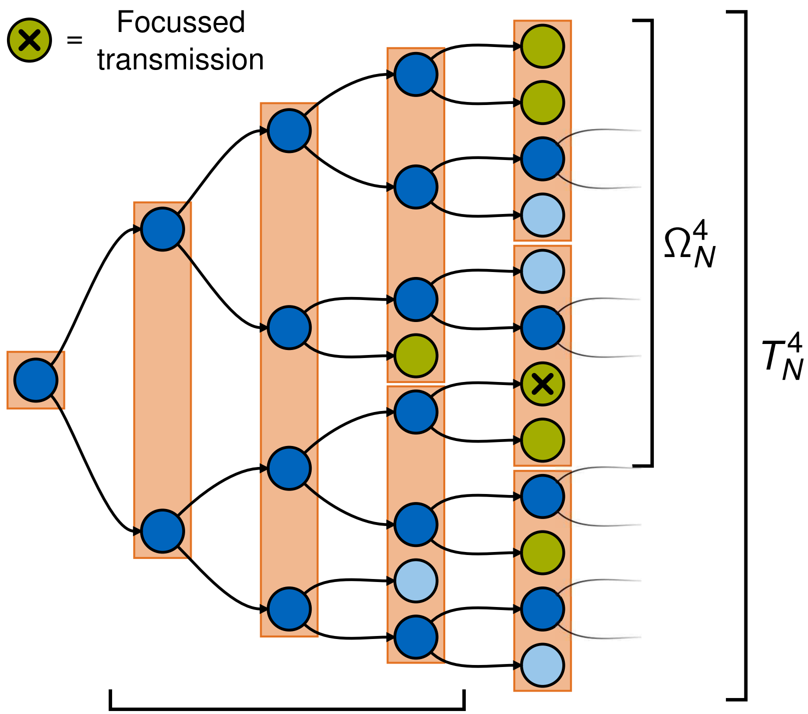


Table of Contents

Multi-channel access delay (I)

- Example of one device transmitting in the slot 2 in the level 4:



$$S_N^3 = \sum_{m=1}^{3} T_N^m$$

Total delay:

$$\Theta_N^4 = 1 + S_N^3 + \Omega_N^4$$

In general:

$$\Theta_N^m = 1 + S_N^{m-1} + \Omega_N^m$$

Multi-channel access delay (II)

- The joint PMF is needed:

$$p_{S_N^{m-1},\Omega_N^m}(s_{m-1},\omega_m) \quad \longrightarrow \quad p_{\Theta_N^m}(\theta_m)$$

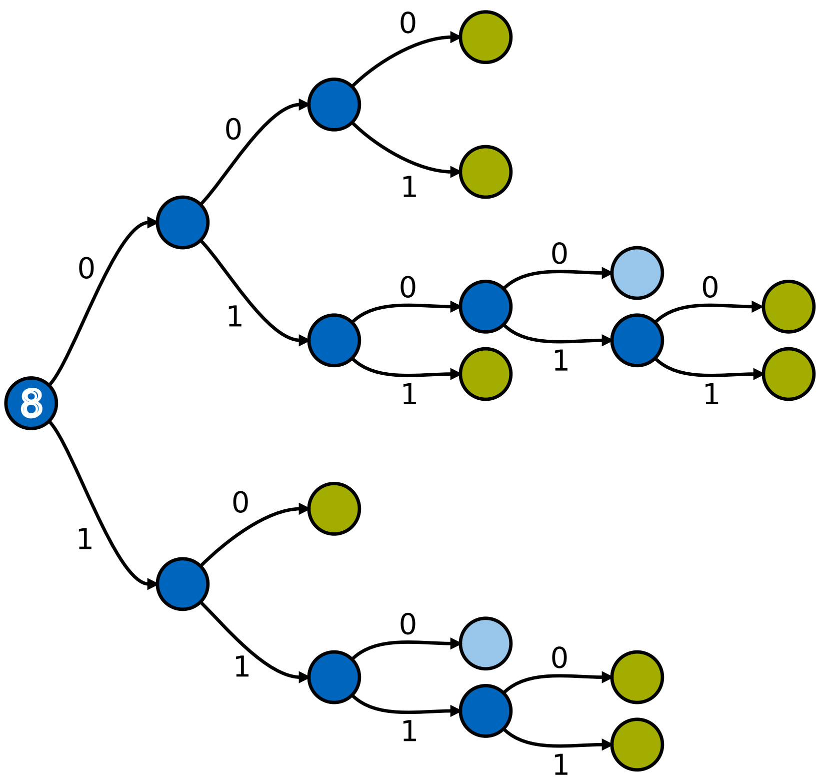
- This PMF can be obtained from the statistics of the length of the tree.
  - S_N^{m-1} is the length of the tree up to level $m-1$
  - Ω_N^m is (almost) an uniform random variable
- Finally, the probability $p_{M_N}(m)$ to transmit in the level m [10] is required:

$$p_{\Theta_N}(\theta) = \sum_{m=1}^{\infty} p_{\Theta_N^m}(\theta_m) \cdot p_{M_N}(m)$$

Table of Contents

Simulation design (I)

- Conversion of a tree into a **decisions matrix**.



$$\Longleftrightarrow \qquad \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & - & - \\ 0 & 0 & 1 & - & - \\ 0 & 1 & 0 & 0 & - \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & - & - & - \\ 1 & 1 & 1 & 0 & - \\ 1 & 1 & 1 & 1 & - \end{bmatrix}$$

Simulation design (II)

- Structure of **one run**:
  1. Randomly **generate** matrices **C**.
  2. **Process** matrices **C** to obtain length L and access delays θ.

- Calculation of the **number of runs**:
  1. Choose probability α to exceed a difference of ε between the empirical and the theoretical CDFs.
  2. Use the **Dvoretzky–Kiefer–Wolfowitz (DKW) inequality** to compute the number of iterations n:

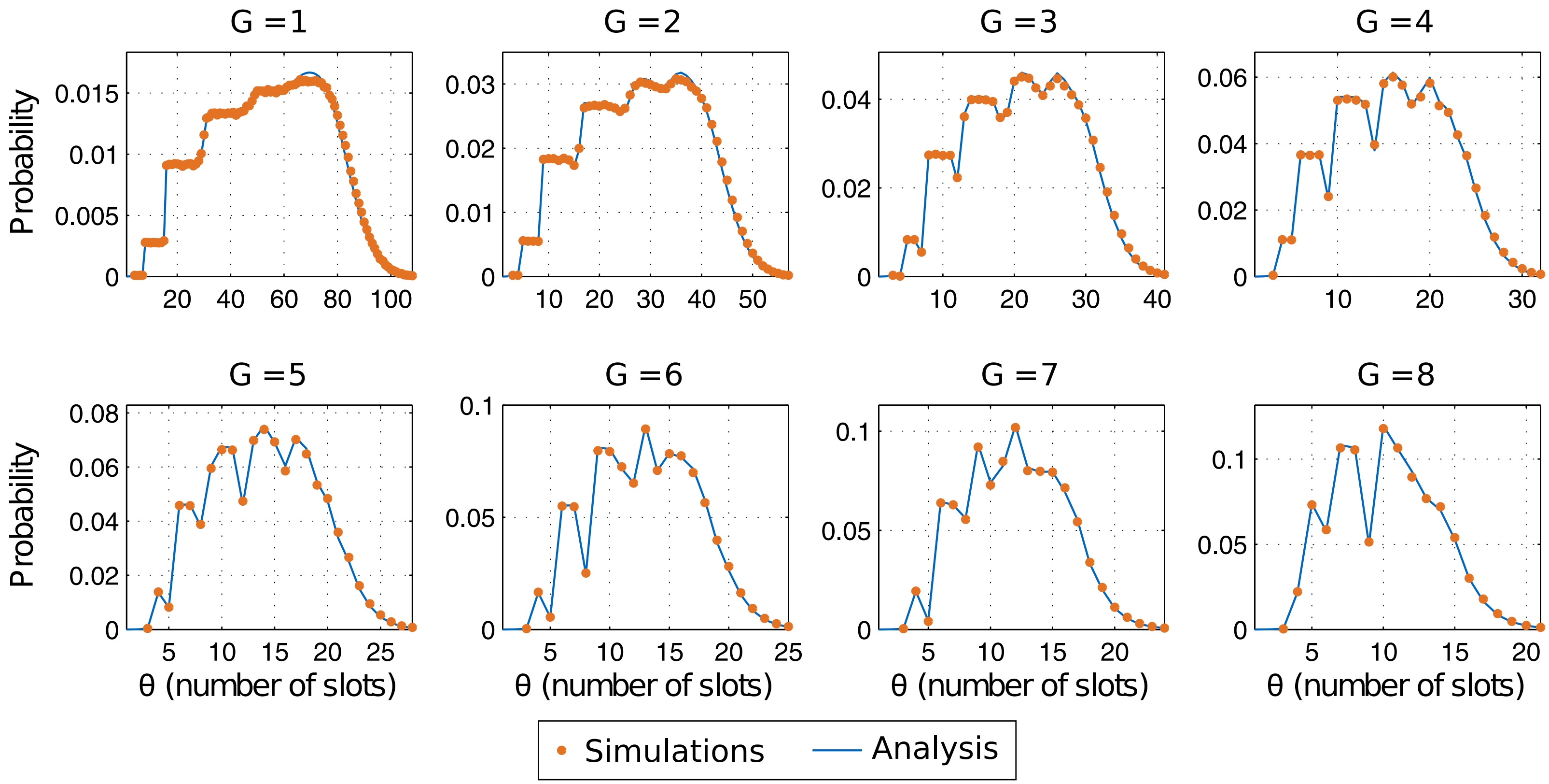
$$\alpha \leq 2e^{-2n\varepsilon^2}$$

For example, for $\alpha = 0.01$ and $\varepsilon = 0.01$:

$$n \approx 26500$$

Simulation results (I)

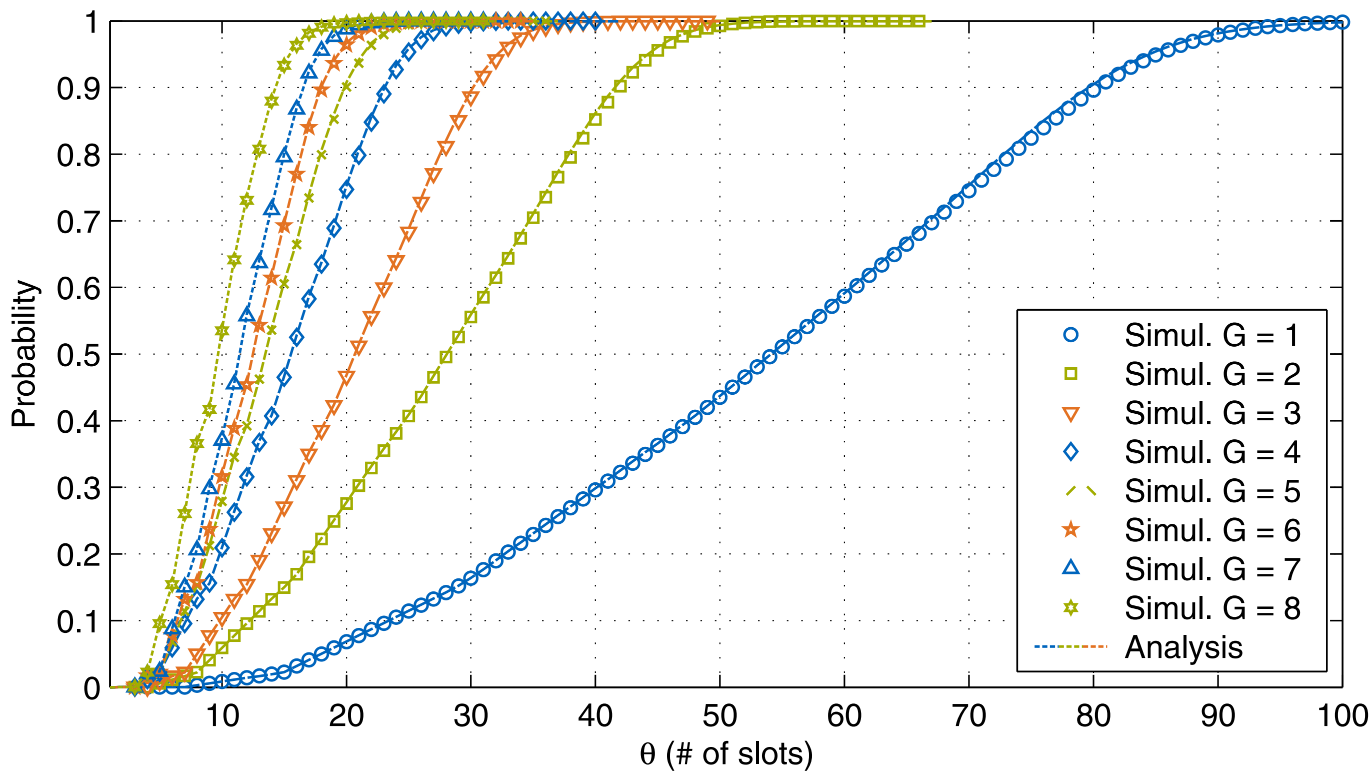
- After 26500 iterations, PMFs of the **access delay** for $N = 60$:



Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Simulation results (II)

- After 26500 iterations, CDFs of the **access delay** for $N = 60$:



Simulation results (III): goodness of fit

- Although simulations show **accurate results**, the model is still an **approximation**.

- Measure of the goodness of fit: **Kolmogorov-Smirnov statistic** $\mathfrak{D}$ (maximum difference between empirical and theoretical CDFs).

G	1	2	3	4	5	6	7	8
$\mathfrak{D}$	0.0108	0.0090	0.0096	0.0081	0.0082	0.0057	0.0042	0.0061

Table 1: Kolmogorov–Smirnov statistic for different values of G

- For $G > 1$, the **maximum error is below 1%.**

Table of Contents

Conclusions

- The problem of guaranteeing **reliability** of mobile networks under **massive arrivals** was addressed
- A **new approach** for improving the LTE RACH based on **Tree Algorithms** was presented
- A **Markovian approximation** was used to obtain an analytical model of multi-channel Tree Algorithms
- The **CDFs of the length of the tree and the access delay** were obtained as a result of the Markovian approach
- **Simulations** were performed in order to confirm the analytical results

Future work

- Implementation of multi-channel Tree Algorithms for different **traffic classes**
- Guaranteed reliability might be provided via **collision estimation**
- **Q-ary** multi-channel Tree Algorithms

References

[1] 3GPP. *MTC simulation results with specific solutions*. TR R2-104662. 3rd Generation Partnership Project (3GPP), Aug. 2010.

[2] C. H. Wei, G. Bianchi, and R. G. Cheng. "Modeling and Analysis of Random Access Channels With Bursty Arrivals in OFDMA Wireless Networks". In: *IEEE Transactions on Wireless Communications* 14.4 (2015), pp. 1940–1953. ISSN: 1536-1276. DOI: 10.1109/TWC.2014.2377121.

[3] James L. Massey. "Collision-Resolution Algorithms and Random-Access Communications". In: *Multi-User Communication Systems*. Ed. by G. Longo. CISM International Centre for Mechanical Sciences. Springer Vienna, 2014. ISBN: 9783709129005.

[4] 3GPP. *RACH Load Analysis for MTC in LTE*. TR R2-104617. 3rd Generation Partnership Project (3GPP), Aug. 2010.

[5] Jesus Alonso-Zárate et al. "Contention-based Collision-Resolution Medium Access Control Algorithms". In: *Medium Access Control in Wireless Networks*. Ed. by Hongyi Wu and Yi Pan. Wireless networks and mobile computing. Nova Science Publishers, 2008. ISBN: 9781600219443.

[6] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification*. TS 36.321. 3rd Generation Partnership Project (3GPP), Sept. 2008. URL: http://www.3gpp.org/ftp/Specs/html-info/36321.htm.

References (cont.)

[7] J. Capetanakis. "Tree algorithms for packet broadcast channels". In: *IEEE Transactions on Information Theory* 25.5 (1979), pp. 505–515. ISSN: 0018-9448. DOI: 10.1109/TIT.1979.1056093.

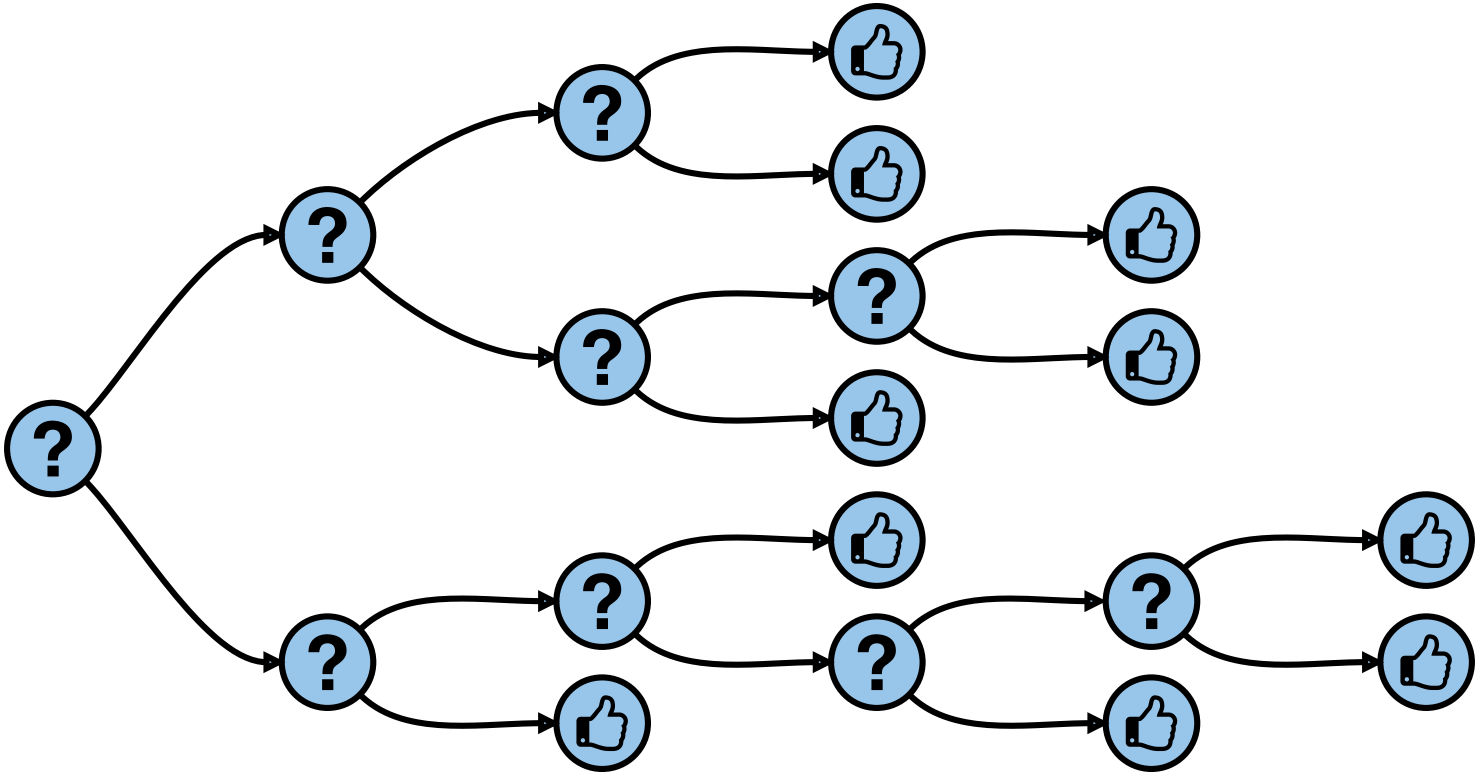
[8] M. Kaplan and E. Gulko. "Analytic properties of multiple-access trees". In: *IEEE Transactions on Information Theory* 31.2 (1985), pp. 255–263. ISSN: 0018-9448. DOI: 10.1109/TIT.1985.1057015.

[9] Mart L Molle and George C Polyzos. "Conflict resolution algorithms and their performance analysis". In: *Tec. Rep. Department of Computer Science and Engineering, University of California at San Diego, LaJolla* (1993).

[10] A. J. E. M. Janssen and M. J. de Jong. "Analysis of contention tree algorithms". In: *IEEE Transactions on Information Theory* 46.6 (2000), pp. 2163–2172. ISSN: 0018-9448. DOI: 10.1109/18.868486.

[11] J.G. Kemeny and J.L. Snell. *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"*. Undergraduate Texts in Mathematics. Springer New York, 1983. ISBN: 9780387901923. URL: https://books.google.de/books?id=0bTK5uWzbYwC.

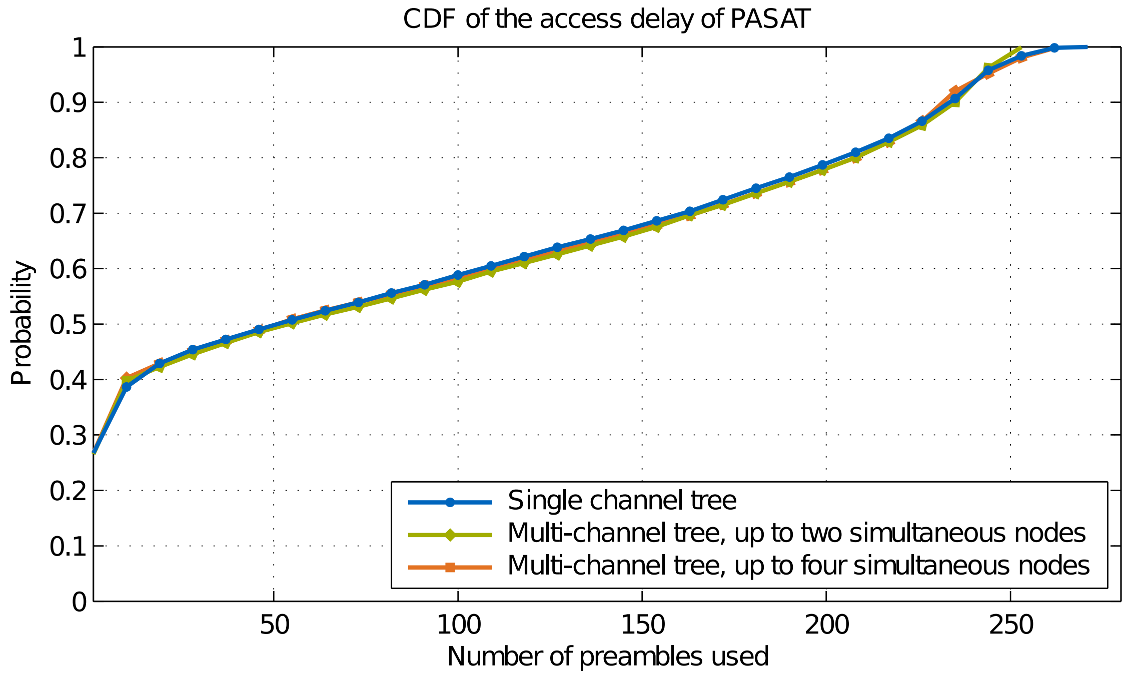
Questions?



1

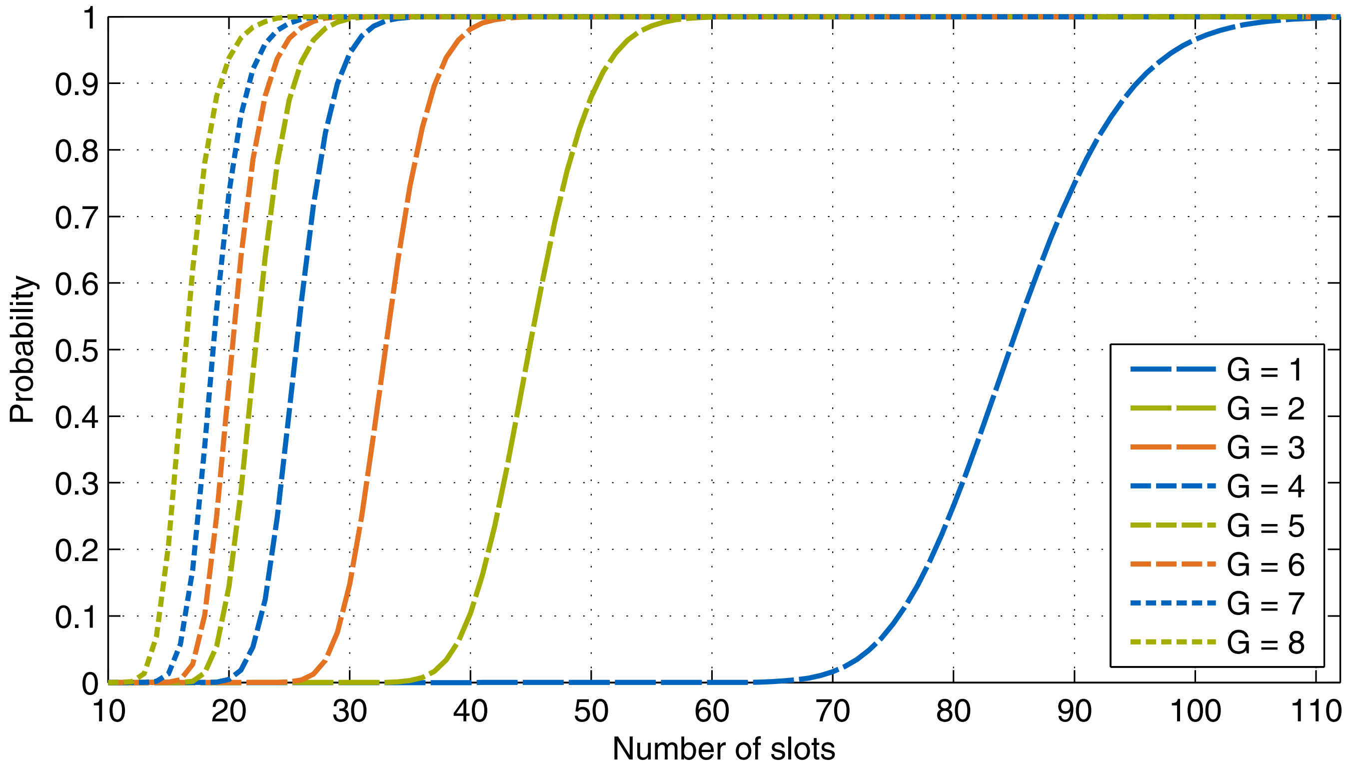
Appendix A: additional PASAT simulation results

- For $N = 30000$ devices beta-distributed within 10 second (200 RAOs):



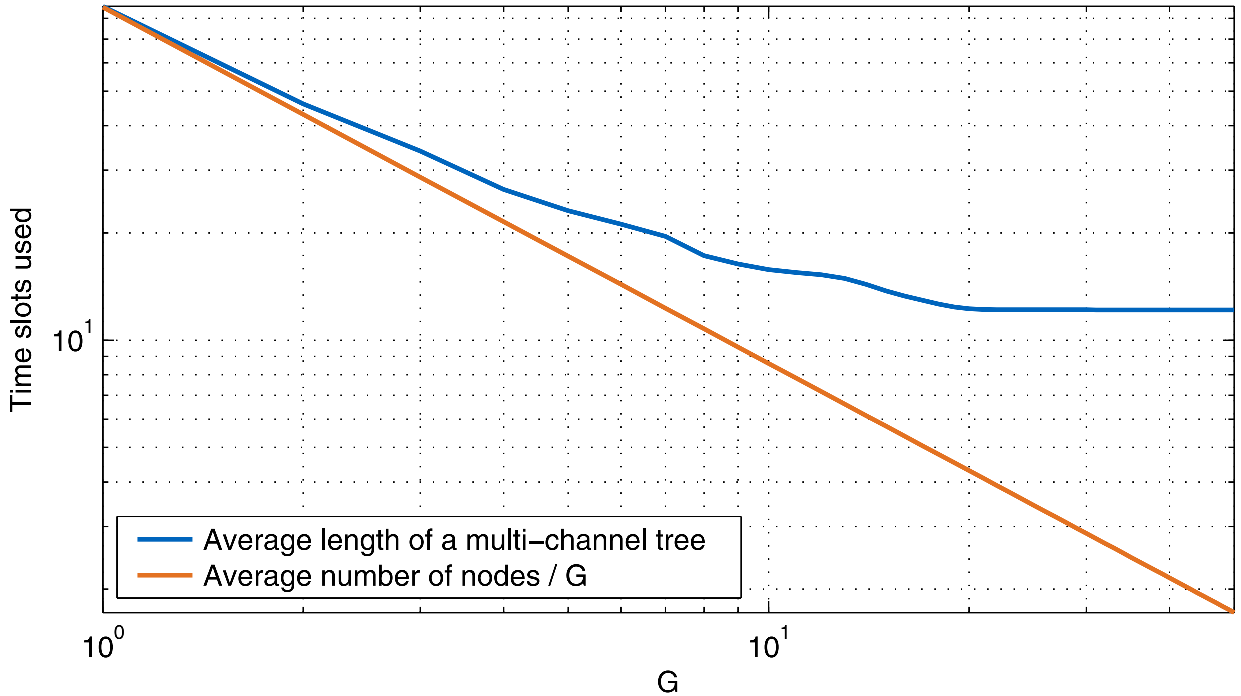
Appendix B: results of the length of a multi-channel Tree

- **CDFs of the multi-channel Tree length** for several values of G with $N = 60$



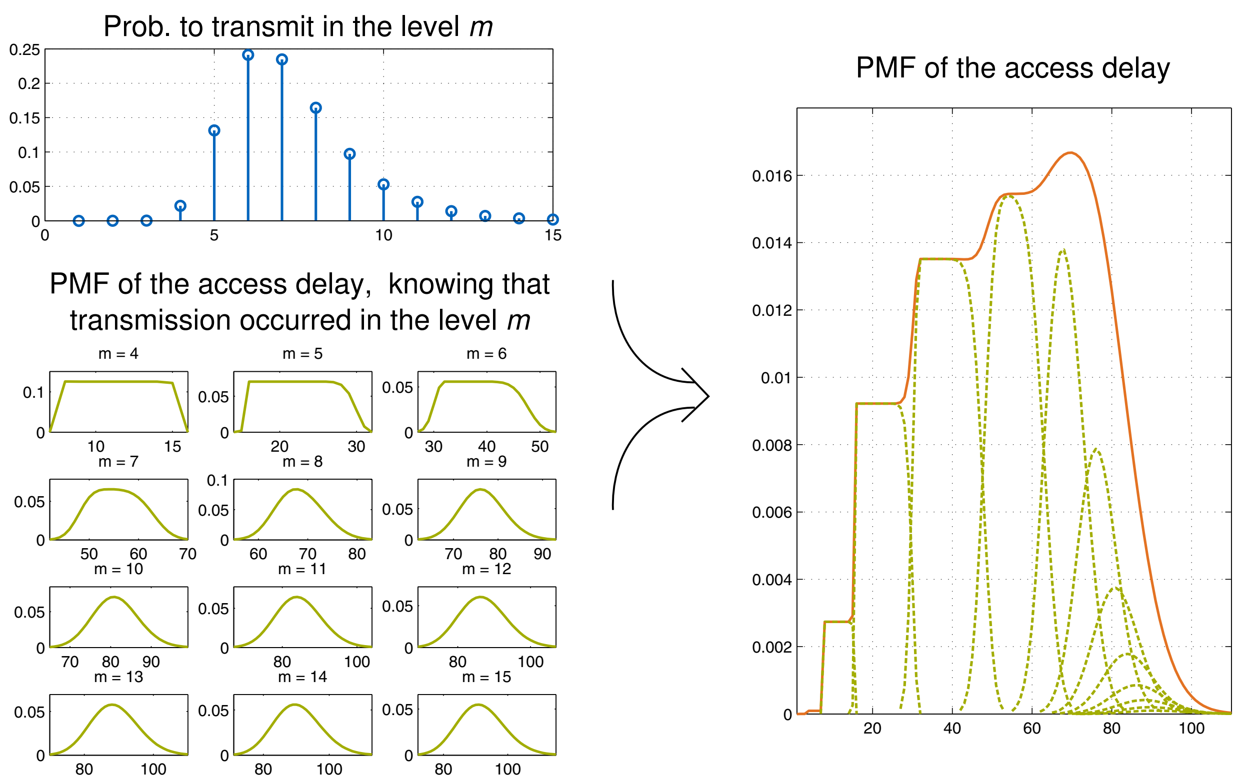
Appendix B: results of the length of a multi-channel Tree (IV)

- Evolution of **average of the multi-channel Tree length** for several values of G with $N = 60$



Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

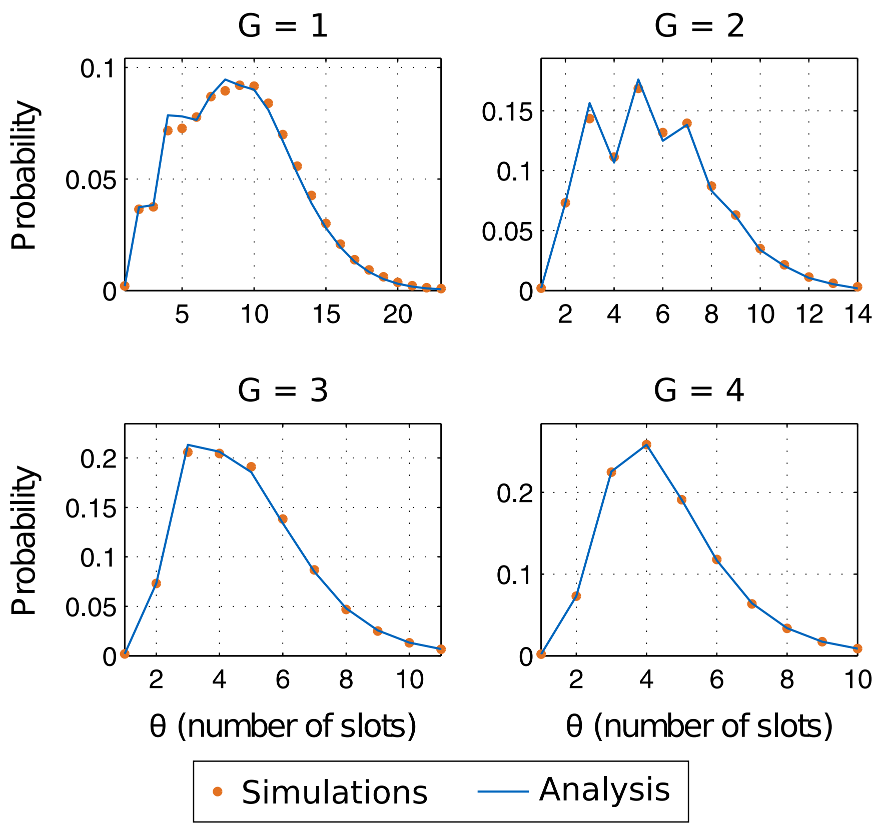
Appendix C: explanation of the PMF of the access delay



Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

Appendix D: additional simulation results

- After 26500 iterations, PMFs of the **access delay** for $N = 10$:



Appendix E: Modeling a Multichannel Tree

$$S_m \longrightarrow \text{Number of nodes at level } m$$
$$T_m \longrightarrow \text{Number of slots at level } m$$
$$G \longrightarrow \text{Number of contention frames in a slot}$$

The number of slots at level m can be expressed as:

$$T_m = \left\lceil \frac{S_m}{G \cdot q} \right\rceil = \left\lceil \frac{X_{m-1}}{G} \right\rceil$$

The total number of slots needed by the tree is then:

$$T = 1 + \sum_{m=1}^{\infty} T_m \approx 1 + \sum_{m=1}^{M} T_m$$

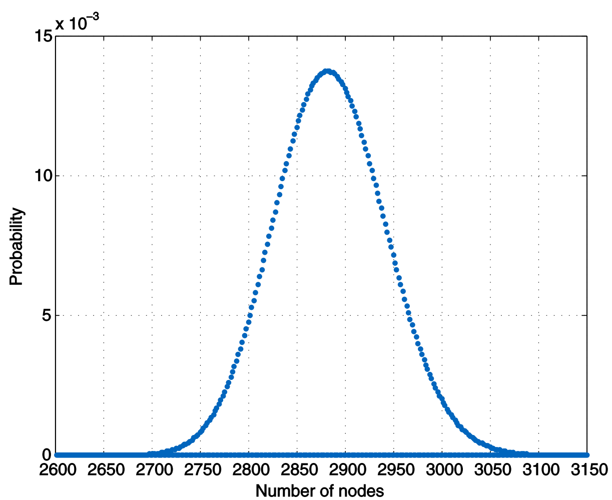
We can set M as $\Pr\{L > M\} \leq \varepsilon$ [10] for a small enough ε (e.g. $\varepsilon = 0.01$)

Lehrstuhl für Kommunikationsnetze
Fakultät für Elektrotechnik und Informationstechnik
Technische Universität München

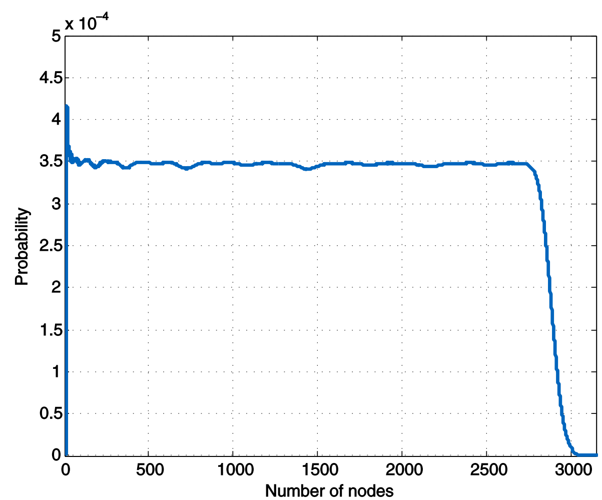
Appendix F: Proposed contributions to the analysis of tree algorithms (I)

- Extension of probability-generating function
  - We can enhance the utility of the PGF by using an FFT-computed expression instead of the classical Z-transform [10]
  - With this modification we can extend the analysis of the length of the tree and the access delay for thousands of devices with low computational cost

PMF of the total length of the tree



PMF of the access delay of the tree



Appendix F: Proposed contributions to the analysis of tree algorithms (II)

- Calculation of the marginal PMFs of the number of nodes in each level
  - It is possible to transform some tree-based problems to a bins-and-balls approach [8]
  - With this approach, we can obtain the marginal PMF of the number of nodes at level m as [2]:

$$f_{S_m}(s_m) = X_{2^{m-1}}(N, s_m)$$

$$X_R(N;k) = \sum_{x=k}^{\min(N-k,R)} \frac{\Psi^N_{x,k}\binom{R}{x}x!}{R^N}$$

    where $\Psi^N_{x,k}$ are the number of ways to arrange N balls into x bins, such that k of them have more than one ball. This numbers are defined throught the recursion [2]:

$$\Psi^N_{x,k} = k\Psi^{N-1}_{x,k} + (x-k+1)\,\Psi^{N-1}_{x,k-1} + \Psi^{N-1}_{x-1,k}$$

  - The expression for $X_R(N;k)$ is however hard to compute. We further propose:

$$\frac{\Psi^N_{x,k}\binom{R}{x}x!}{R^N} = \xi^{R,N}_{x,k} = \frac{1}{R}\left[k\xi^{R,N-1}_{x,k} + (x-k+1)\xi^{R,N-1}_{x,k-1} + (R-x+1)\xi^{R,N-1}_{x-1,k}\right]$$

Appendix G: Average length of a Multichannel Tree (I)

- One of the few parameters that whose analysis can be found in the literature [**7063635**]

- In [**7063635**], they stated:

$$\mathrm{E}\{T\} = 1 + \sum_{m=1}^{\infty} \left\lceil \frac{\mathrm{E}\{X_m\}}{G} \right\rceil$$

  where $\mathrm{E}\{X_m\}$ is the average number of collisions at level m

- However, this implies the folllowing relation

$$T_m = \left\lceil \frac{X_{m-1}}{G} \right\rceil \quad \longrightarrow \quad \mathrm{E}\{T_m\} = \left\lceil \frac{\mathrm{E}\{X_{m-1}\}}{G} \right\rceil$$

  which is in general **wrong**, and therefore it must be only taken as an approximation

Appendix G: Average length of a Multichannel Tree (II)

- However, an exact expression can be worked out
- First, we should reformulate the ceil function as:

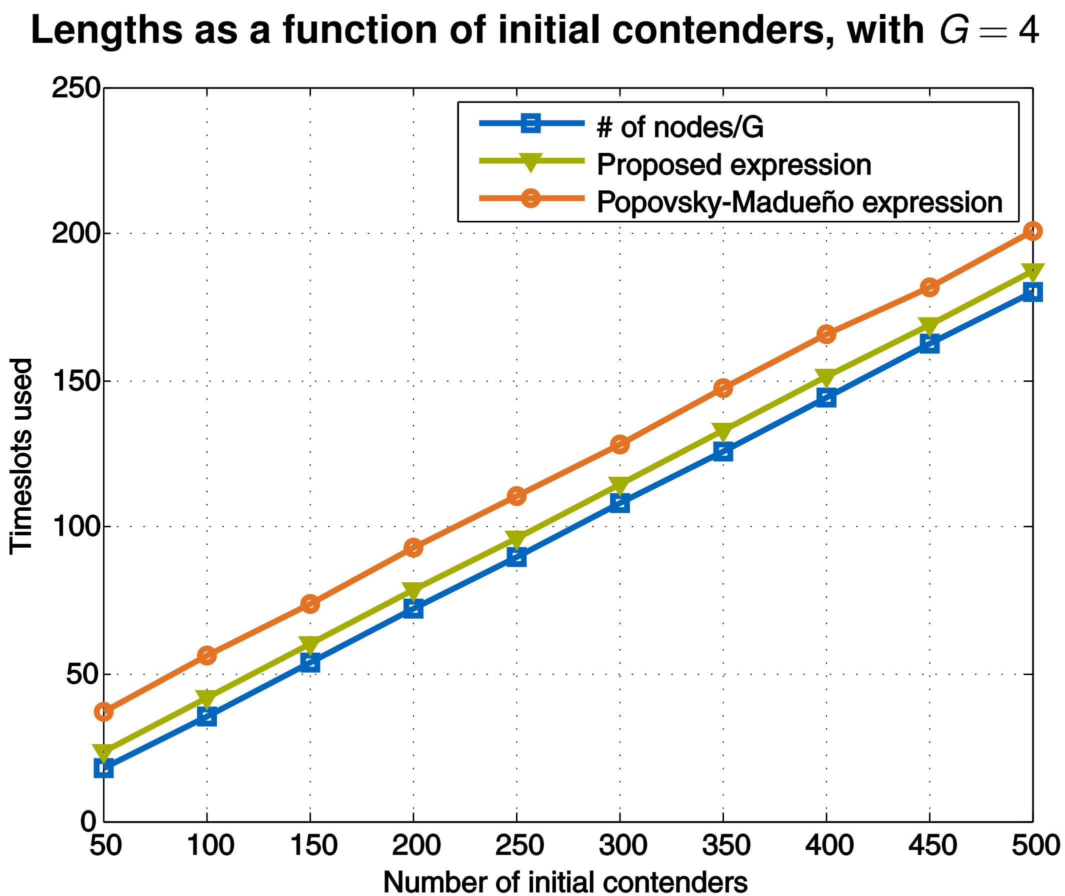
$$\left\lceil \frac{X_{m-1}}{G} \right\rceil = \begin{cases} \frac{X_{m-1}}{G} & \text{if } X_{m-1} \bmod G = 0 \\ \frac{X_{m-1}+G-1}{G} & \text{if } X_{m-1} \bmod G = 1 \\ \vdots & \vdots \\ \frac{X_{m-1}+1}{G} & \text{if } X_{m-1} \bmod G = G-1 \end{cases}$$

- Now we apply the law of total expectation:

$$\mathrm{E}\left\{\left\lceil \frac{X_{m-1}}{G} \right\rceil\right\} = \frac{1}{G}\mathrm{E}\{X_{m-1}\}\mathrm{Pr}\{X_{m-1} \bmod G = 0\} + \ldots + \frac{1}{G}\mathrm{E}\{X_{m-1}+1\}\mathrm{Pr}\{X_{m-1} \bmod G = G-1\}$$
$$= \frac{1}{G}\left(\mathrm{E}\{X_{m-1}\} + \sum_{k=1}^{G-1}(G-k)\cdot\sum_{j=0}^{\infty}\mathrm{Pr}\{X_{m-1} = G\cdot j + k\}\right)$$

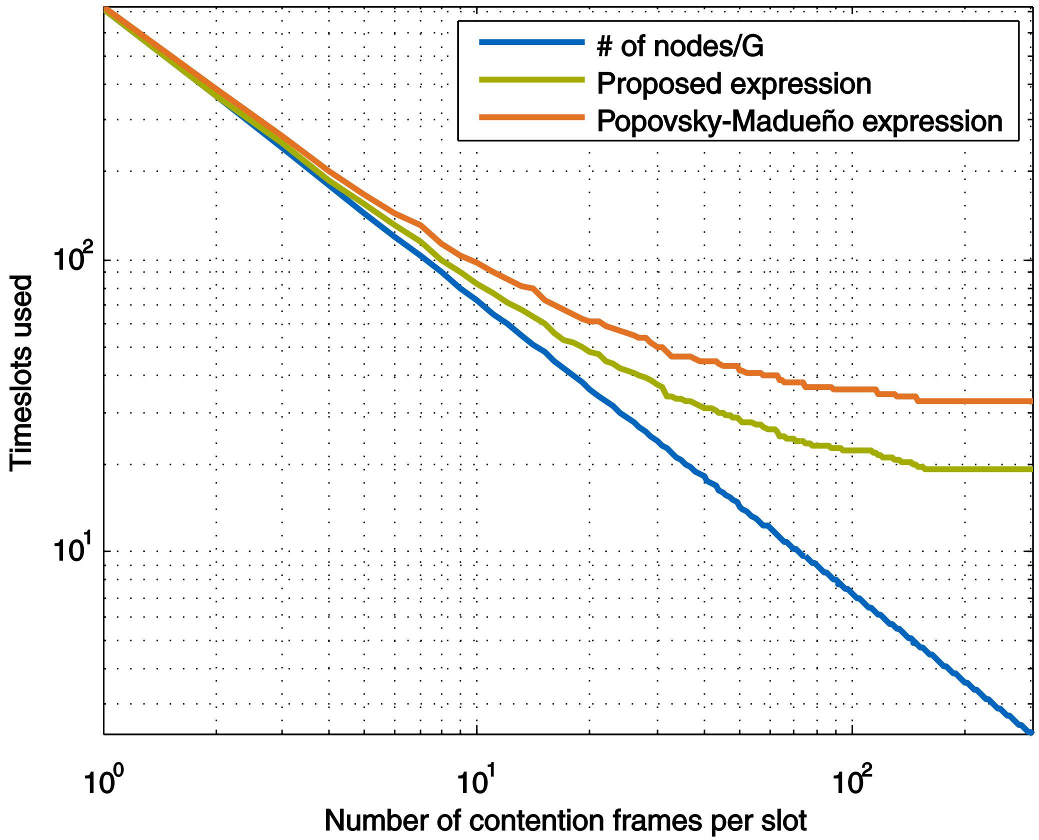
- In order to perform this calculation, we need to know the marginal PMFs of the collisions at each level

Appendix G: Average length of a Multichannel Tree (III)



Appendix G: Average length of a Multichannel Tree (IV)

Lengths as a function of the number of nodes per timeslots, with $N = 500$



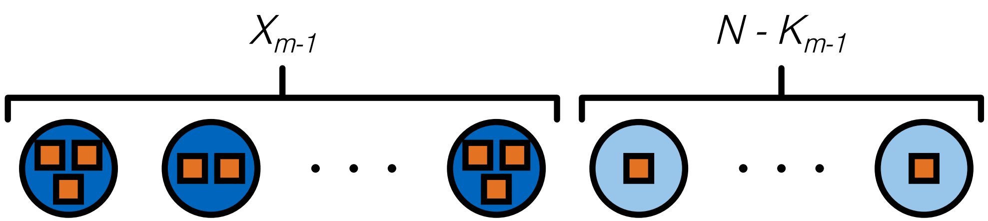
Appendix H: Markovian approach (I)

- The probability $f_{K_{m-1}|X_{m-1}}(k_{m-1}|x_{m-1})$ is computed as the following ratio:

$$f_{K_{m-1}|X_{m-1}}(k_{m-1}|x_{m-1}) = \frac{\Gamma^{X_{m-1}}_{K_{m-1}}}{\sum_{K_{m-1}} \Gamma^{X_{m-1}}_{K_{m-1}}} = \frac{\text{\# of ways to have } K_{m-1} \text{ contenders given } X_{m-1} \text{ collisions}}{\text{\# of ways to have } X_{m-1} \text{ collisions}}$$

- The term $\Gamma^{X_{m-1}}_{K_{m-1}}$ is obtained by approaching the problem as a **bins-and-balls problem** [2]:

$$\Gamma^{X_{m-1}}_{K_{m-1}} = \Psi^{N}_{N-K_{m-1}+X_{m-1},X_{m-1}} \binom{q^{m-1}}{N-K_{m-1}+X_{m-1}} (N-K_{m-1}+X_{m-1})!$$



Appendix H: Markovian approach (II)

- Now we are left with the computation of $f_{X_m|X_{m-1},K_{m-1}}(x_m|x_{m-1},k_{m-1})$

- We have to **decompose** K_{m-1} into partitions of X_{m-1} collisions. Let us call π_i the partition i of K_{m-1} into X_{m-1} parts, all of them greater than 1. Then:

$$f_{X_m|X_{m-1},K_{m-1}}(x_m|x_{m-1},k_{m-1}) = \sum_i f_{X_m|X_{m-1},K_{m-1},\Pi_i}(x_m|x_{m-1},k_{m-1},\pi_i) \cdot f_{\Pi_i}(\pi_i)$$

- Let us define $\pi_i = \{\kappa_1, \kappa_2, \ldots, \kappa_{X_{m-1}}\}$ and $\#_{\pi_i} = \{\#_1, \#_2, \ldots, \#_{X_{m-1}}\}$, where $\#_j$ is the number of parts equal to j of π_i. It can be shown that:

$$f_{\Pi_i}(\pi_i) = \frac{K_{m-1}!}{\Psi^{K_{m-1}}_{X_{m-1},X_{m-1}}} \cdot \prod_{j=1}^{X_{m-1}} \frac{1}{\kappa_j! \#_j!}$$

- Finally, given some partition π_i, the **total number of collisions** at level m will be the sum of the number of collisions in each independent subtree

Appendix I: From joint to sum distribution

- Let us define the auxiliary random variable:

$$\Sigma_m = \sum_{j=1}^{m} X_j$$

- We can apply the associative property to express our desired PMF as a function of a bivariate joint PMF:

$$f_{X=X_1+\ldots+X_m}(x = x_1 + \ldots + x_m) = f_{X=\Sigma_{m-1}+X_m}(x = \sigma_{m-1} + x_m) = \sum_{j=0}^{x} f_{\Sigma_{m-1},X_m}(j, \sigma_{m-1} - j)$$

- After some basic manipulation, we can express that bivariate function as:

$$f_{\Sigma_{m-1},X_m}(x_m, \sigma_{m-1}) = \sum_{x_{m-1}} f_{X_m|X_{m-1}}(x_m|x_{m-1}) f_{X_{m-1},X_{m-1}+S_{m-2}}(x_{m-1}, \sigma_{m-2})$$

which exhibits a clear recursive structure, much more suitable to compute