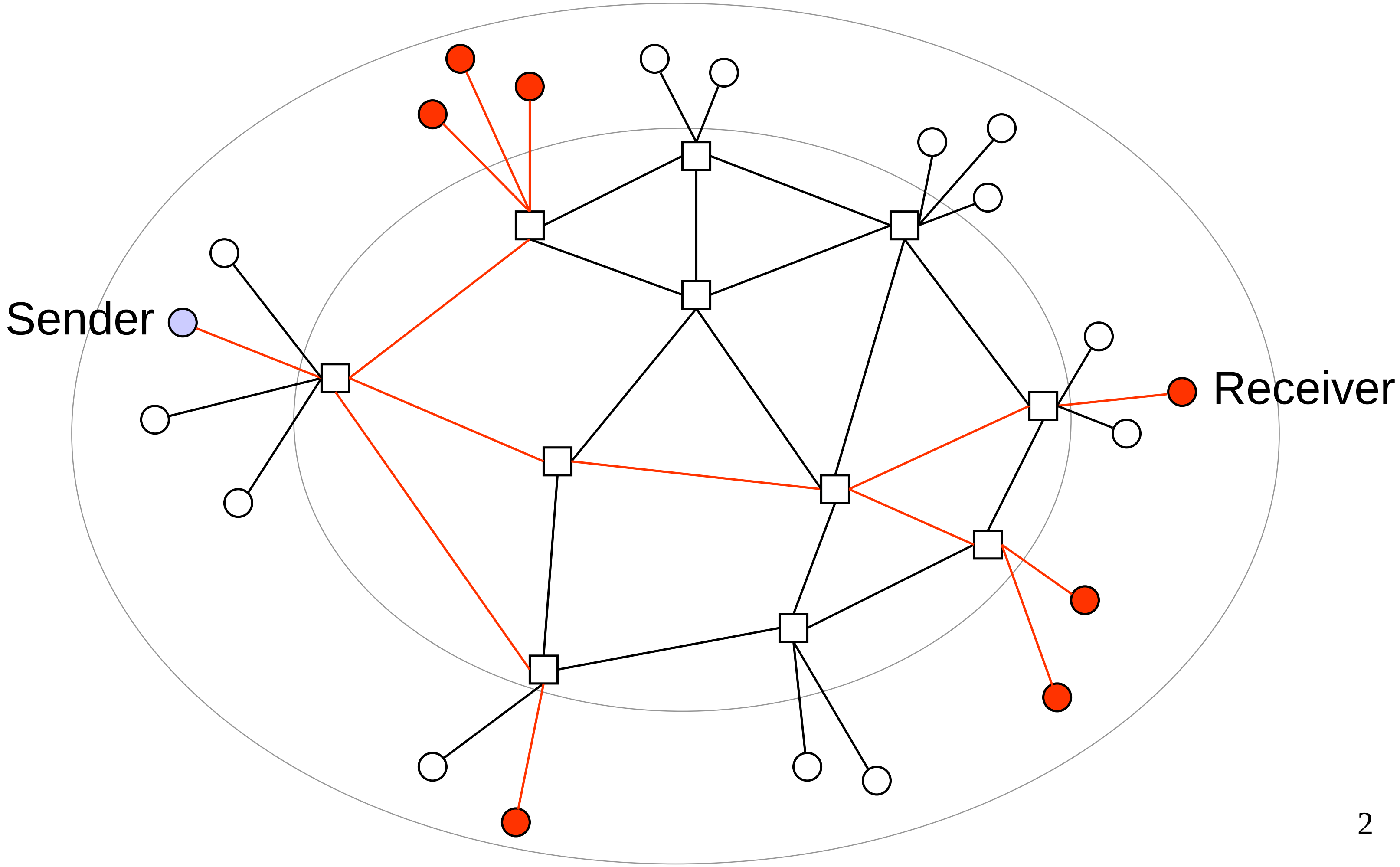
New Algorithms for Calculating Blocking Probabilities in Multicast Networks

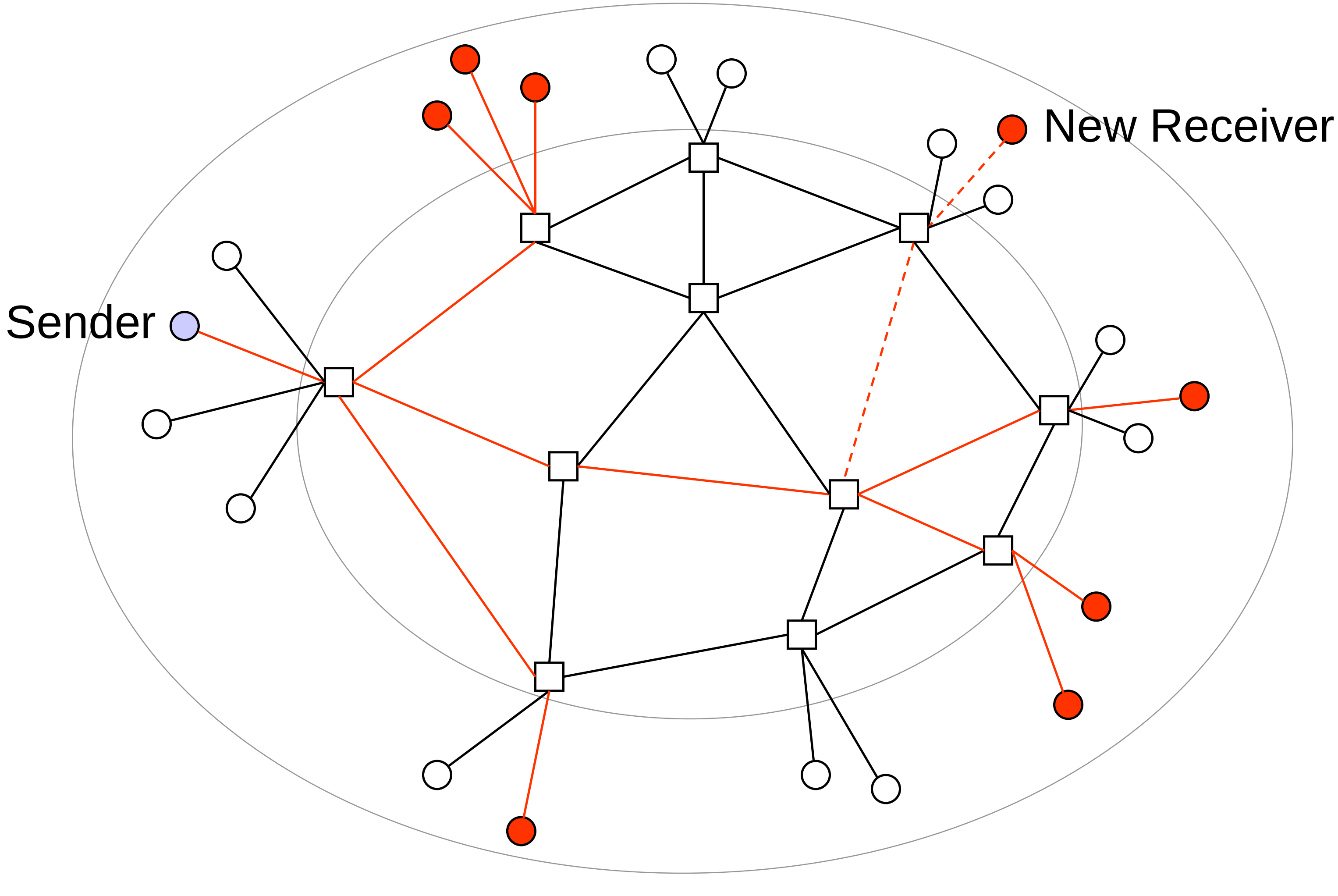
Samuli Aalto, Jouni Karvo & Jorma Virtamo
Networking Laboratory
Helsinki University of Technology

samuli.aalto@hut.fi

Multicast connection with dynamic membership (1)

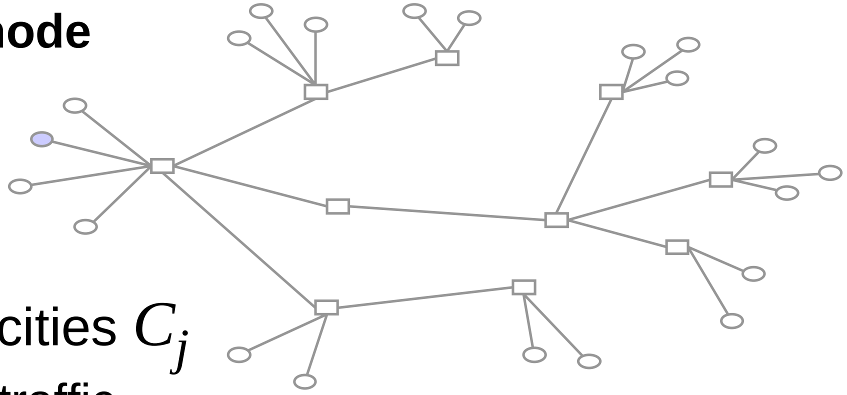


Multicast connection with dynamic membership (2)



Setup

- A (unique) **service center** offers a variety of **channels** $i \in I$
  - e.g. audio or video streams
  - required capacities d_i
- Each channel is delivered to users $u \in U$ by a **multicast connection** with **dynamic membership**
- Each multicast connection uses the same **routing tree**
  - service center located at the **root node**
  - users located at the **leaf nodes**



- Physical links $j \in J$ with finite capacities C_j
  - (possibly) shared with bg (unicast) traffic

Teletraffic study

- Calculation of call blocking (of a user requesting a channel)
  - call blocking = probability that the user fails to join the requested multicast connection

- **1st step**:
  Find out how to calculate time blocking (for that user)
  - time blocking = probability of such network states that do not allow the user to join the requested multicast connection

- **2nd step**:
  Express call blocking by means of time blocking
  (possibly in a slightly modified network)
  - exact expression depends always on the chosen user model

Time blocking (1)

- $\mathbf{X}$ = network state (**without** capacity constraints)
- $\tilde{\mathbf{X}}$ = network state (**with** capacity constraints)
- $\tilde{\Omega}$ = network state space (**with** capacity constraints)
- $\tilde{\Omega}_{ui}$ = nonblocking states for user u and channel i
- B^t_{ui} = time blocking probability for user u and channel i

$$B^t_{ui} := 1 - P\{\tilde{\mathbf{X}} \in \tilde{\Omega}_{ui}\} = 1 - \frac{P\{\mathbf{X} \in \tilde{\Omega}_{ui}\}}{P\{\mathbf{X} \in \tilde{\Omega}\}}$$

numerator

denominator

Due to the Truncation Principle!

Time blocking (2)

- In principle, there is a closed form analytical expression both for the numerator and the denominator

- **Problem**: computationally extremely complex
  - exponential in U

- **Solution**: use **recursive convolution-truncation algorithms** to calculate the numerator and the denominator
  - linear in U

The algorithm

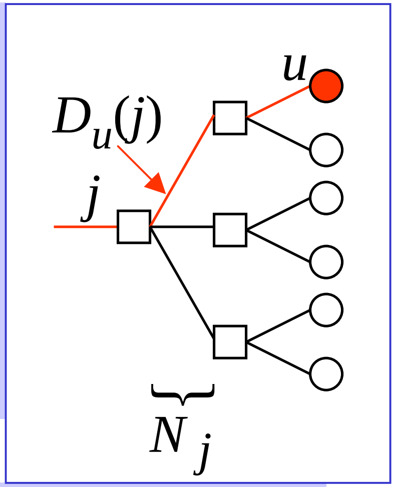
- Denominator:

$$P\{\mathbf{X}\in \tilde{\Omega}\} = \sum_{\mathbf{y}} Q_J(\mathbf{y})$$

link state; root link

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

truncation operator; "convolution" operator



- Numerator:

$$P\{\mathbf{X}\in \tilde{\Omega}_{ui}\} = \sum_{\mathbf{y}'} Q_J^{ui}(\mathbf{y}')$$

i-modified link state; root link

$$Q_j^{ui}(\mathbf{y}') = \begin{cases} T_j^{ui}[\pi_j](\mathbf{y}'), & j = u \\ T_j^{ui}[Q^{ui}_{D_u(j)} \odot \bigotimes_{k\in N_j\setminus R_u} Q_k](\mathbf{y}'), & j\in R_u\setminus\{u\} \end{cases}$$

ui-mod. trunc. oper.; modified "convolution" operator; route to user *u*

Cases studied

- Single link analysis (Karvo, Virtamo, Martikainen & Aalto, 1997-1998)
  - starting point
- Network wide analysis (Nyberg, Virtamo & Aalto, 1999)
  - all channels handled individually
  - first convolution-truncation algorithm

New algorithms!

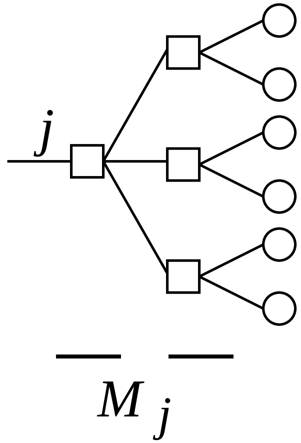
- Multi-class case (Aalto, Karvo & Virtamo, 2000)
  - class = group of statistically indistinguishable channels
  - combinatorial convolution needed
- Multi-layer case (Karvo, Aalto & Virtamo, 2000-2001)
  - layered coding of audio/video streams
  - individual channels / single class / multiple classes

Individually handled channels

- Denominator:
  - **Link state** = $\mathbf{y} = (y_1, \ldots, y_I)$
    - $y_i = 1$ (0) if channel i is active (idle)
  - Q**-function**:

total capacity req.

$$Q_j(\mathbf{y}) = P\{\mathbf{Y}_j = \mathbf{y}; r(\mathbf{Y}_{j'}) \le C_{j'}, j' \in M_j\}$$



  - **Truncation**:

$$T_j f(\mathbf{y}) = f(\mathbf{y}) 1\{r(\mathbf{y}) \le C_j\}$$

  - **OR-convolution**:

$$[f \otimes g](\mathbf{y}) = \sum_{\mathbf{u} \oplus \mathbf{v} = \mathbf{y}} f(\mathbf{u}) g(\mathbf{v})$$

componentwise OR

- Numerator: slightly different!

New algorithm

Single class

- Denominator:
  - **Link state** = n = number of active channels
  - Q**-function**:

$$Q_j(n) = P\{N_j = n; r(N_{j'}) \le C_{j'}, j' \in M_j\}$$

total capacity req.

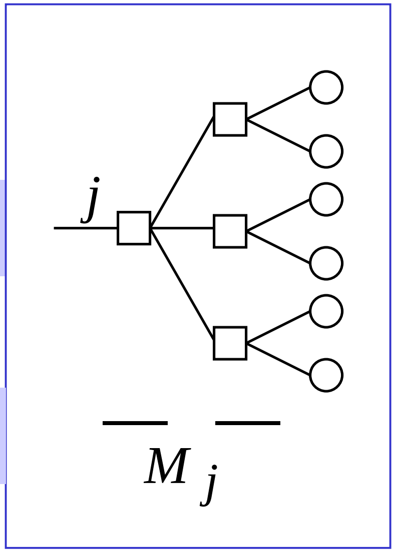
  - **Truncation**:

$$T_j f(n) = f(n) 1\{r(n) \le C_j\}$$

  - **Combinatorial convolution**:

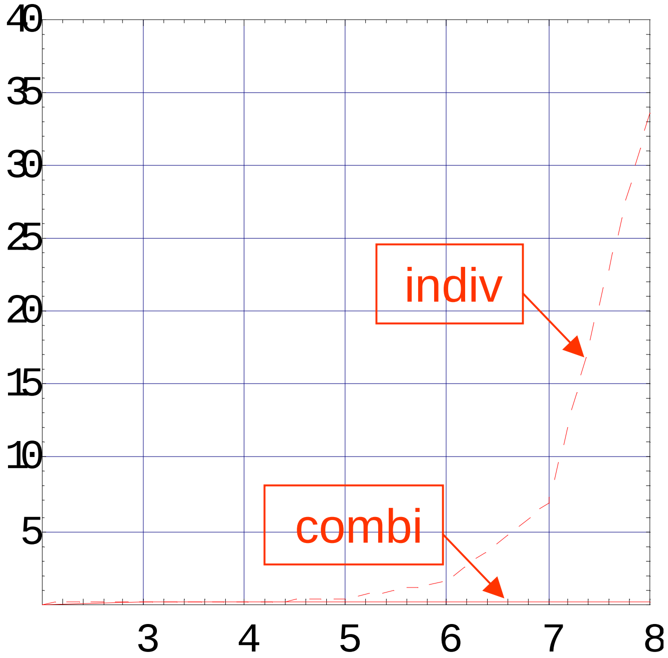
$$[f \otimes g](n) = \sum_{l,m} s(n \mid l, m, I) f(l) g(m)$$

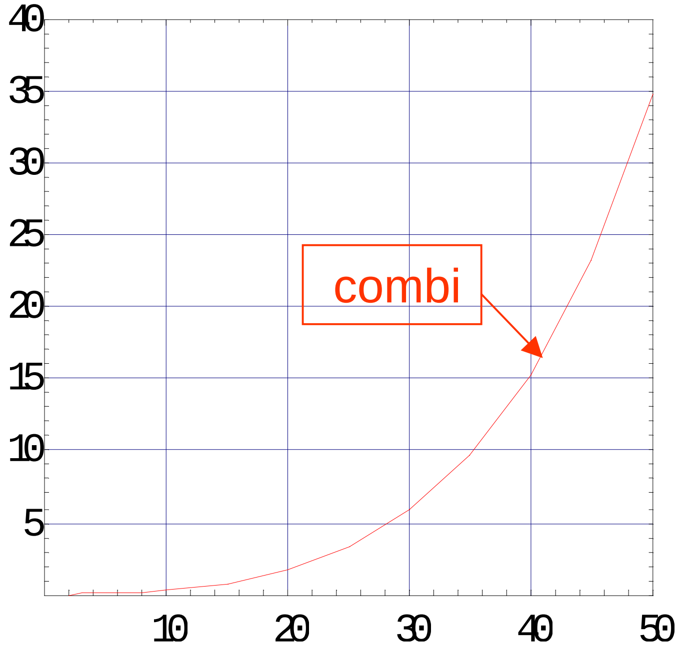
simple combinatorial factor by
a random sampling' -argument



- Numerator: slightly different!

Processing time (sec) vs. number of channels





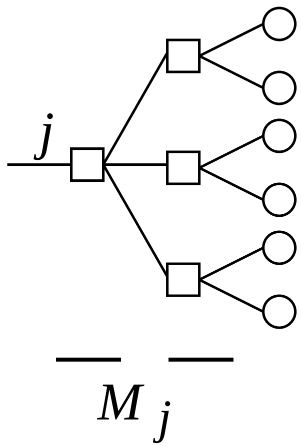
New algorithm

Multi-class case

- Denominator:
  - **Link state = n** = $(n_1, \ldots, n_K)$
    - n_k = number of active channels of class k
  - Q**-function**:

total capacity req.

$$Q_j(\mathbf{n}) = P\{\mathbf{N}_j = \mathbf{n}; r(\mathbf{N}_{j'}) \le C_{j'}, j' \in M_j\}$$



  - **Truncation**:

$$T_j f(\mathbf{n}) = f(\mathbf{n}) 1\{r(\mathbf{n}) \le C_j\}$$

  - **Combinatorial convolution**:

$$[f \otimes g](\mathbf{n}) = \sum_{\mathbf{l},\mathbf{m}} \prod_k s(n_k \mid l_k, m_k, I_k) f(\mathbf{l}) g(\mathbf{m})$$

the same combinatorial factor as before

- Numerator: slightly different!

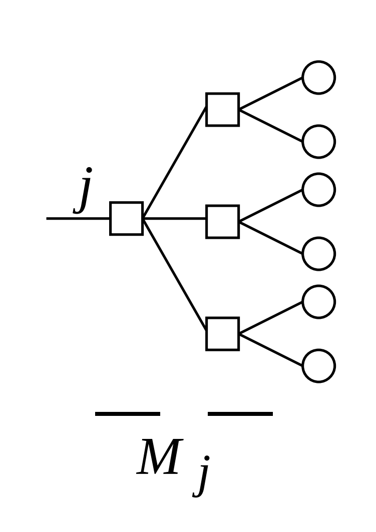
New algorithm

Multi-layer case (individually handled channels)

- Denominator:
  - **Link state** = $\mathbf{y} = (y_1, \ldots, y_I)$
    - y_i = layer on which channel i is active (0 when idle)
  - Q**-function**:

total capacity req.

$$Q_j(\mathbf{y}) = P\{\mathbf{Y}_j = \mathbf{y}; r(\mathbf{Y}_{j'}) \le C_{j'}, j' \in M_j\}$$



  - **Truncation**:

$$T_j f(\mathbf{y}) = f(\mathbf{y}) 1\{r(\mathbf{y}) \le C_j\}$$

  - **Max-convolution**:

$$[f \otimes g](\mathbf{y}) = \sum_{\max\{\mathbf{u},\mathbf{v}\}=\mathbf{y}} f(\mathbf{u}) g(\mathbf{v})$$

componentwise max

- Numerator: slightly different!

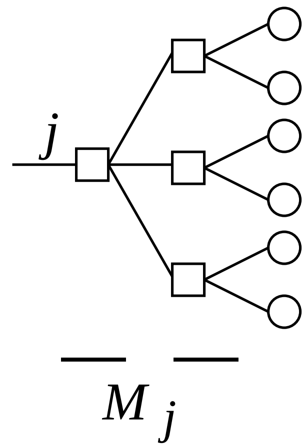
New algorithm

Multi-layer case (single class)

- Denominator:
  - **Link state** = $\mathbf{n} = (n_1, \ldots, n_L)$
    - n_l = number of active channels on layer l
  - Q**-function**:

total capacity req.

$$Q_j(\mathbf{n}) = P\{\mathbf{N}_j = \mathbf{n}; r(\mathbf{N}_{j'}) \le C_{j'}, j' \in M_j\}$$



  - **Truncation**:

$$T_j f(\mathbf{n}) = f(\mathbf{n}) 1\{r(\mathbf{n}) \le C_j\}$$

  - **Combinatorial convolution**:

$$[f \otimes g](\mathbf{n}) = \sum_{\mathbf{l},\mathbf{m}} s(\mathbf{n} \mid \mathbf{l}, \mathbf{m}) f(\mathbf{l}) g(\mathbf{m})$$

much more complicated combinatorial factor

- Numerator: slightly different!

Multi-layer case (multiple classes)

To be done!

THE END