# Challenge #1: topology design

## Definitions

- A graph is a couple $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a finite, nonempty set of nodes and $\mathcal{A}$ is a subset of $\mathcal{N} \times \mathcal{N}$.
  - If $(i, j) \in \mathcal{A} \Rightarrow (j, i) \in \mathcal{A}$, we say the graph is undirected.
  - A very useful model for representing network connectivity as seen at a given architectural level.
  - The interpretation of nodes and arcs depends on the considered architectural level, e.g., nodes can be routers and arcs subnets connecting them at IP level or switches connected with physical links at layer 2.
- If a function $w : \mathcal{A} \mapsto \mathbb{R}$ is defined, we say the graph is weighted. The weight of arc $(i, j)$ is denoted with $w_{ij}$.
- We define:

  $n$ the number of nodes;

  $k$ the number of arcs.

## Connectivity

- A path between nodes $s$ and $t$ is an ordered sequence of nodes and arcs connecting the two nodes, i.e., $(s, k_1), (k_1, k_2), \dots, (k_{\ell-1}, t)$ is a path of length $\ell$.
- A graph is said to be connected if there exists a path connecting any two nodes.
- A graph is connected if one of the following statements is true
  - The adjacency matrix $\mathbf{A}$ is irreducible;
  - The second smallest eigenvalue of $\mathbf{L}$ is positive;
- An $n \times n$ matrix $\mathbf{A}$ is irreducible if

$$\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^{n-1} > \mathbf{0}$$

# Breadth-First-Search algorithm

Finds all paths form a given node.

Given a graph 'Graph' and a node 'root' in it:

```
for each node n in Graph:
  n.distance = INFINITY
  n.parent = NIL
endfor
create empty queue Q
root.distance = 0
Q.enqueue(root)
while Q is not empty:
  current = Q.dequeue()
  for each node n that is adjacent to current:
  if n.distance == INFINITY:
   n.distance = current.distance + 1
   n.parent = current
   Q.enqueue(n)
  endif
endwhile
```

# Laplacian

- The laplacian of a graph in an $n \times n$ matrix $\mathbf{L}$ defined as follows:
  - $L_{ii} = d_i$, where $d_i$ is the degree of node $i$;
  - $L_{ij} = -1$ if and only if $(i, j) \in \mathcal{A}$.
- We can write:
$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$
- $\mathbf{L}$ is a symmetric matrix, if the graph is undirected.
- The sum of each row and each column of $\mathbf{L}$ is 0. Therefore 0 is always an eigenvalue of $\mathbf{L}$

## Eigenvalues of the Laplacian

- The incidence matrix of a graph in an $a \times n$ matrix $\mathbf{M}$ defined as follows:
  - $M_{\ell i} = 1$, if $\ell = (i, j) \in \mathcal{A}$ (outgoing arc);
  - $M_{\ell j} = -1$, if $\ell = (i, j) \in \mathcal{A}$ (incoming arc);
  - $M_{\ell j} = 0$, otherwise.
- It can be verified that:

$$\mathbf{L} = \frac{1}{2}\mathbf{M}^T\mathbf{M}$$

- Hence $\mathbf{L}$ is a positive semi-definite matrix, i.e., $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$ for any vector $\mathbf{x}$.
- Positive semi-definite and symmetric $\rightarrow$ the eigenvalues of $\mathbf{L}$ are real and non-negative.

## Connectivity through Laplacian

**Theorem (From algebraic graph theory)**

*For an undirected graph the number of connected components is equal to the algebraic multiplicity of the smallest eigenvalue of the graph Laplacian.*

*Corollary. An undirected graph is connected if and only if the smallest eigenvalue of the graph Laplacian is simple, i.e., the second smallest eigenvalue of the Laplacian is positive.*

- The property stated in the Corollary can be used to assess the connectivity of a given graph.
- Let $\eta_1 = 0 \leq \eta_2 \leq \cdots \leq \eta_n$ denote the $n$ real, non-negative eigenvalues of $\mathbf{L}$.
- The graph is connected if and only if $\eta_2 > 0$.

## Random graphs

- **Erdős-Rényi random graphs**: two models are possible.
  - $G(n,m)$ model: the number $m$ of arcs is given; they are assigned uniformly at random to $m$ out of the $n(n-1)/2$ node couples.
  - $G(n,p)$ model: a graph is constructed by connecting nodes randomly. An arc is included in the graph with probability $p$ independently of any other arc.
- $r$**-regular random graph**: a graph selected from $\mathcal{G}_{n,r}$, which denotes the probability space of all $r$-regular graphs on $n$ vertices, where $3 \leq r < n$ and $nr$ is even.
  - An $r$ regular graph is a graph where each vertex has the same number $r$ of neighbors.
  - $r$ regular graphs for $r = 0, 1, 2$ are trivial.

## Traffic model and throughput

- The application-oblivious throughput bounds states that $TH \leq m/(\overline{h}\nu_f)$, where $m$ is the number of links of the network graph, $\overline{h}$ is the mean path length and $\nu_f$ is the number of flows.
- Assume an all-to-all traffic matrix: then $\nu_f = n(n-1)/2$.
- In case of $r$-regular random graph, the number of links is $m = nr/2$. In case of $p$-ER graph, the number of links is an outcome of the random generation, hence it is a random variable with mean $m = pn(n-1)/2$.
- The throughput bound can be calculated by estimating the mean path length $\overline{h}$, i.e., the mean of the lengths of the shortest paths connecting all node pairs.

# Assignment - Part 1

1. Use library scripts to generate $p$-ER random graphs and $r$-regular random graph. Let $K$ denote the number of nodes.
2. Write a script to check the connectivity of a given graph.
   - algebraic method 1 (irreducibility);
   - algebraic method 2 (eigenvalue of the Laplacian matrix);
   - breadth-first search algorithm.
3. Compare the complexity as a function of $K$ of the methods above by plotting curves of a complexity measure vs $n$.
4. Let $p_c(\mathcal{G})$ denote the probability that a graph $\mathcal{G}$ is connected. By running Monte Carlo simulations estimates $p_c(\mathcal{G})$ and produce two curve plots:
   - $p_c(\mathcal{G})$ vs. $p$ for Erdős-Rényi graphs with $K = 100$.
   - $p_c(\mathcal{G})$ vs. $K$, for $K \leq 100$, for $r$-regular random graphs with $r = 2$ and $r = 8$.

# Definitions

We aim to compare throughput offered by Fat-Tree and Jellyfish for the same amount of resources.

- $N = \#$ of servers.
- $S = \#$ of switches.
- $L = \#$ of links connecting switches.
- $n = \#$ number of ports of a switch.
- $\overline{h} =$ mean shortest path lengths for paths among switches.

# Assignment - Part 2

1. Find $r$ ($\#$ of switch ports to be connected to other switches in Jellyfish) as a function of $n$ so that $N$, $S$ and $L$ are the same for Jellyfish and Fat-tree.
2. Write the expression of the application-oblivious throughput bound $TH$ for an all-to-all (among switches) traffic matrix. Assume $\overline{h}$ as known). The expression of $TH$ must be a function of $\overline{h}$ and $n$ only.
3. Using the exact value of $\overline{h}$ as a function of $n$ for Fat-Tree (you must calculate it!) and the upper bound for $r$-regular random graphs (see slides, but be carful with notation) for Jellyfish, evaluate $TH$ for $n = 5 \cdot k$ for $k = 1, \ldots, 10$.

## Delivery of the assignment

The delivery of the assignment consists of a **2 pages written report** (Font size: 12 pt). Remember to put your given and family names, and enrollment number as a header on top of each page.

1. **PAGE 1** - Three curve plots: (i) Curve plot of complexity versus number of nodes $K$, for the three connectivity checking algorithms; (ii) Probability of a connected ER random graph as a function of $p$ for $K = 100$ nodes; (iii) Probability of a connected $r$-regular random graph as a function of the number of nodes $K$ for $r = 2$ and 8 (see slide Assignment - Part 1 for detailed specs). Accuracy and quality of the graphs will be fundamental in the evaluation. Also concise and significant comments will be appreciated.
2. **PAGE 2** - (i) Expression of $r$ as a function of $n$; (ii) Expression of $TH$ as a function of $n$ and $\overline{h}$; (iii) Table with six columns: $n$, $N$, $S$, $L$, $TH_{\text{Fat-Tree}}$, $TH_{\text{Jellyfish}}$