CMSC5733 Social Computing

Irwin King

The Chinese University of Hong Kong

king@cse.cuhk.edu.hk

Outline

- Graphs
  - Origins
  - Definition
  - Spectral Properties
  - Type of Graphs
  - Topological Structure
- Regular Networks
  - Diameter, Centrality, and Average Path Length
  - Type of Networks

GRAPHS

Origins

- Leonhard Euler--bridges of Konigsberg
- G. Yule—preferential attachment
- Kermack, McKendrick—epidemic model
- Paul Erdos--discrete mathematics, Erdos-Renyi algorithm
- Stanley Milgram—small-world network
- Duncan Watts—sparse networks in the physical world
- Steven Strogatz—network structure on complex adaptive systems
- Albert-Laszlo Barabasi—scale-free networks, nonrandom networks with hubs

Principles of Network Science

- Structure
- Emergence
- Dynamism
- Autonomy
- Bottom-Up Evolution
- Topology
- Power
- Stability

- Graph Definitions
- Graph Properties
- Matrix Representation
- Classes of Graphs

Set-theoretic Definition

- A *graph G* = [*N,L,f*] is a 3-tuple consisting of a set of nodes *N*, a set of links *L*, and a mapping function f: *L*→*N* × *N*, which maps links into pairs of node

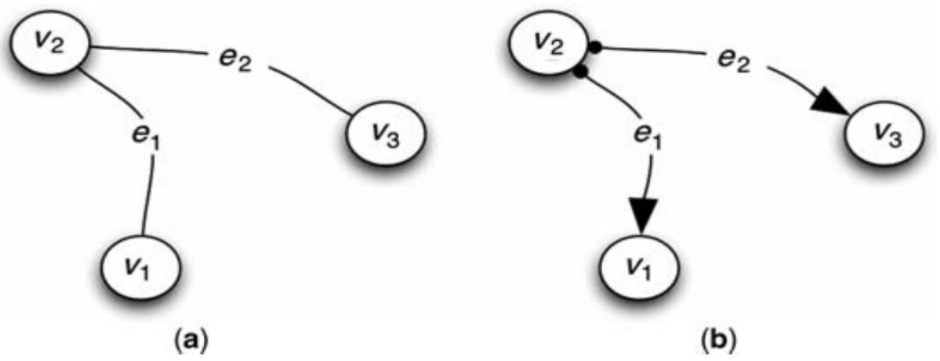
$G = [N, L, f]$ is a graph composed of three sets:

$N = [v_1, v_2, \ldots, v_n]$ are nodes; $n = |N|$ is the number of nodes in N.

$L = [e_1, e_2, \ldots, e_m]$ are links; $m = |L|$ is the number of links in L.

$f\colon L \longrightarrow N \times N$ maps links onto node pairs.

- Mapping function
  - Nondirectional link
  - Directional link



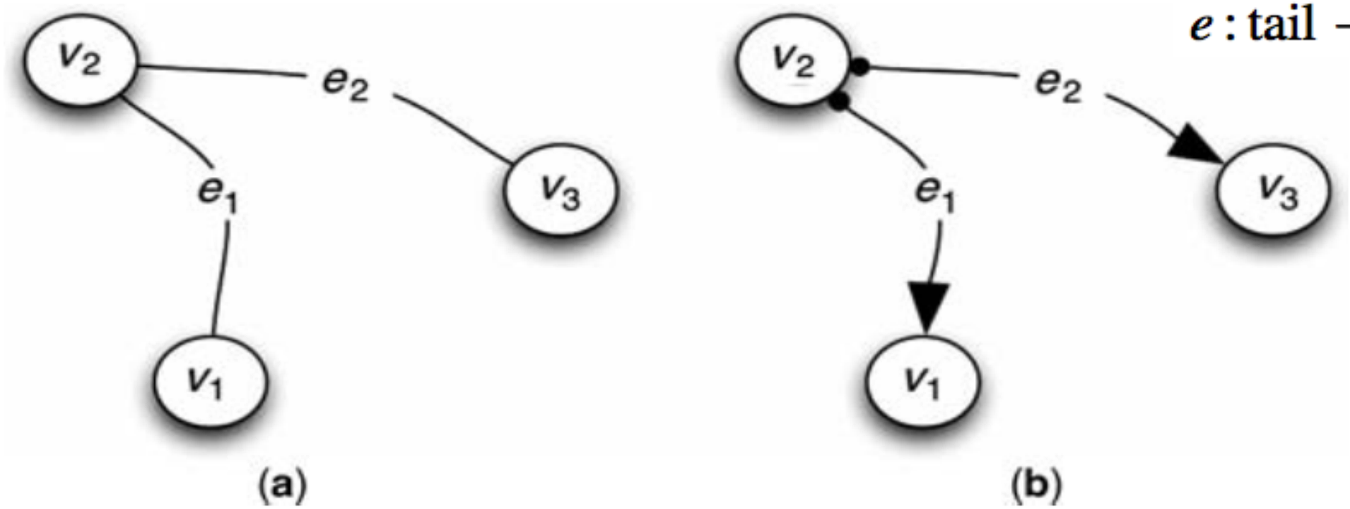
Node Degree and Hub

- Node degree
  - The number of links (directed or undirected) connecting a node *v* to the graph is called the *degree* of the node
- When the graph is directed
  - The *out-degree* of a node is equal to the number of outward-directed links
  - The *in-degree* is equal to the number of inward-directed links
- Hub
  - The *hub* of a graph is the node with the largest degree

$$\text{Hub} = \text{maximum}\{d(v_i)\}$$

$$f = [e_1 : v_2 \sim v_1,\ e_2 : v_3 \sim v_2]$$

$$e : \text{tail} \longrightarrow \text{head}$$

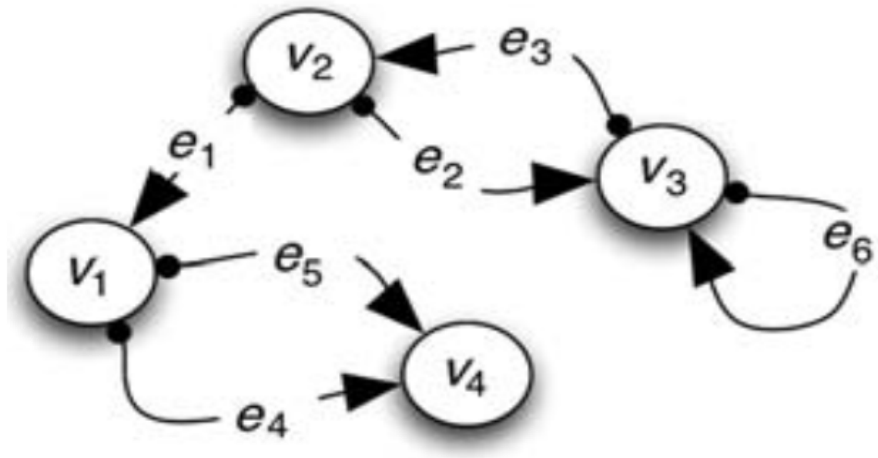


$$d(v_1) = d_1 = 1$$
$$d(v_2) = d_2 = 2$$
$$d(v_3) = d_3 = 3$$

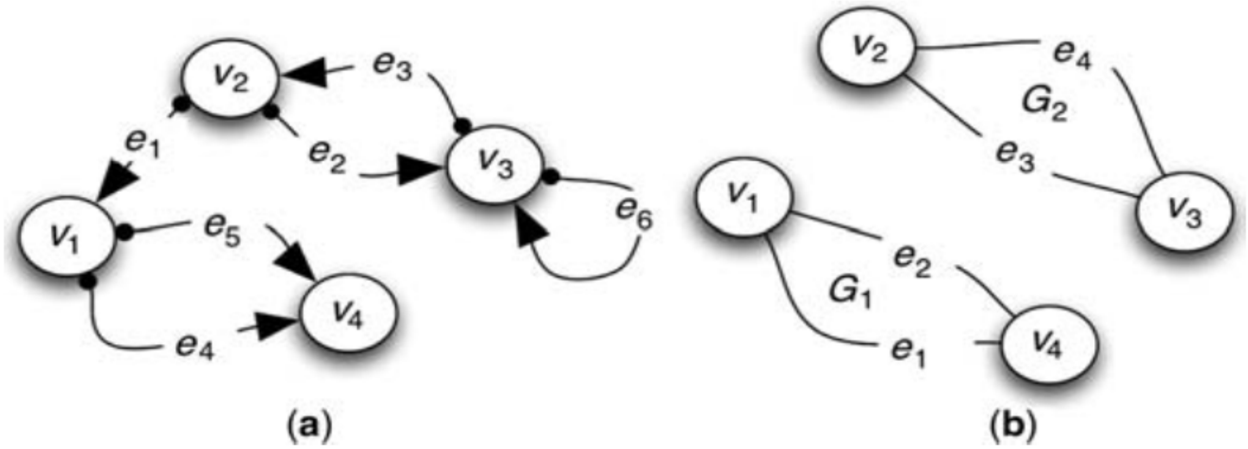
$$\text{in\_d}(v_1) = \text{in\_d}_1 = 1$$
$$\text{out\_d}(v_1) = \text{out\_d}_1 = 0$$
$$\text{in\_d}(v_2) = \text{in\_d}_2 = 0$$
$$\text{out\_d}(v_2) = \text{out\_d}_2 = 2$$
$$\text{in\_d}(v_3) = \text{in\_d}_3 = 1$$
$$\text{out\_d}(v_3) = \text{out\_d}_3 = 0$$

V_2 is the hub

- Path
  - A *path* is a sequence of nodes in G



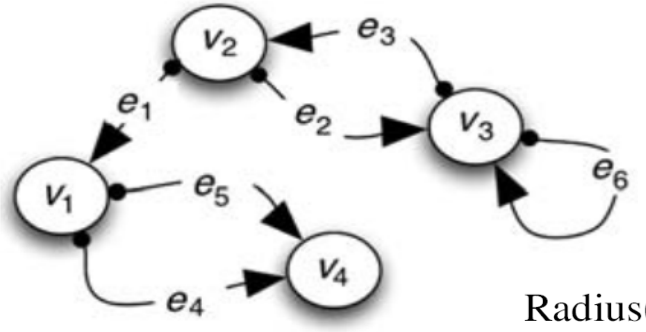
  - The *length of a path* is equal to the number of links (*hops*) between starting and ending nodes of the path
  - The *shortest path* is used as the path connecting nodes u and v. It is also called the *direct path* between two nodes.
  - The *average path length of G* is equal to the average over all shortest paths
- Circuit
  - A *path* that begins and ends with the same node is called a circuit
  - A *loop* is a circuit of length 1



- Conn
  - Undirected graph G is strongly connected if every node v_i is reachable along a path from every other node $v_i \neq v_j$, for j =1,2, . . . , i − 1, i + 1, . . . , n.
  - Weakly connected?
- Component
  - A graph G has *components* G_1 and G_2 if no (undirected) path exists from any node of G_1 to any node of G_2
  - A component is an isolated *subgraph*

Diameter and Radius

- Diameter
  - The longest path between any two nodes in a graph *G* is called the *diameter* of *G*
- Radius
  - The longest path from a node *u* to all other nodes of a connected graph be defined as the *radius* of node *u*
  - The largest radius over all nodes is the graph's diameter

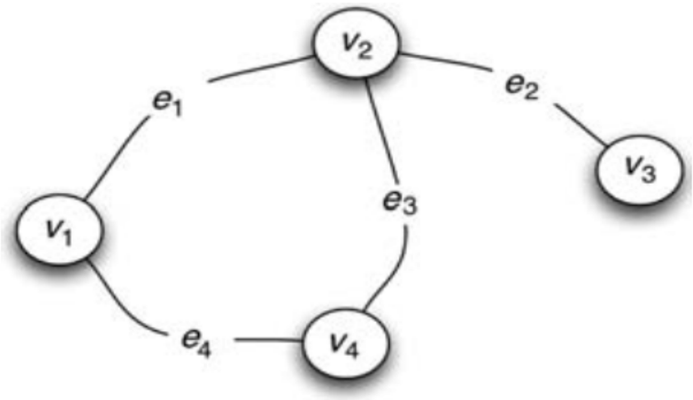


Node	Radius (Node)
1	2 hops
2	2 hops
3	3 hops

$$\text{Radius}(u) = \text{maximum}_i \left\{ \text{minimum}_j \{ \text{path}_j(u_i, v_i) \} \right\}$$

Centrality, Betweenness and Closesness

- Centrality
  - The *center* of the graph is the node with the smallest radius
- Betweenness
  - *Betweenness* of node *v* is the number of paths from all nodes (except *v*) to all other nodes that must pass through node *v*
- Closesness
  - Closeness of node *v* is the number of direct paths from all nodes to all other nodes that must pass through node *v*



Node	Betweenness	Closeness
1	6	0
2	6	4
3	0	0
4	2	0

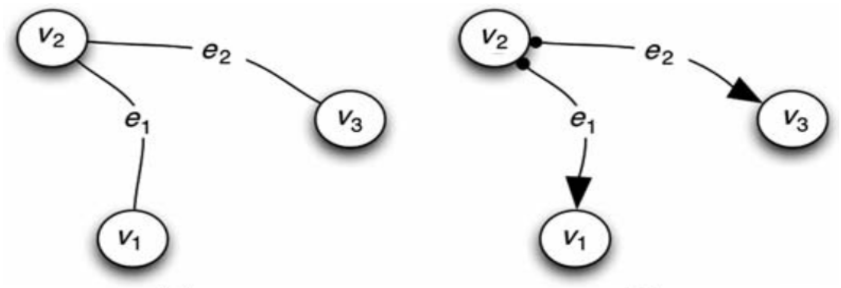
Measures of the power of an intermediary!

Questions

- If u is connected to v, and v is connected to w, then is u connected to w?
- Is it possible for a graph to contain multiple paths connecting nodes?
- Is it true that there is no node farther away from all other nodes than the graph's diameter?
- Under what conditions would closeness not be a perfect measure of an intermediary's power over others?

Matrix Algebra Definition

- Connection matrix
  - The connection matrix of G, $C(G)$ is a mapping function f expressed as a square matrix, where
    - rows correspond to tail nodes
    - columns correspond to head nodes
    - $c_{i,j} = k$ if $v_i \sim v_j$ or $c_{i,j} = 0$ otherwise. $(i, j) = ([1,n], [1,n])$
    - k is the number of links that connect $v \sim v$



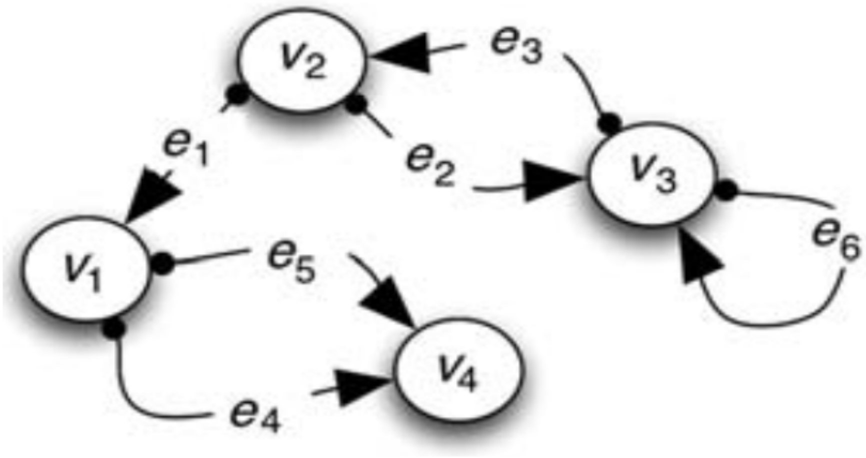
G $\quad$ G'

$$C(G) = \begin{array}{c|ccc} & v_1 & v_2 & v_3 \\ \hline v_1 & 0 & 1 & 0 \\ v_2 & 1 & 0 & 1 \\ v_3 & 0 & 1 & 0 \end{array} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$C(G') = \begin{array}{c|ccc} & v_1 & v_2 & v_3 \\ \hline v_1 & 0 & 0 & 0 \\ v_2 & 1 & 0 & 1 \\ v_3 & 0 & 0 & 0 \end{array} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Adjacency Matrix

- Adjacency matrix
  - The adjacency matrix *A* ignores duplicate links between node pairs
  - $a_{i,j} = 1$ if $v_i \sim v_j$ or $a_{i,j} = 0$ otherwise. $(i, j) = ([1,n], [1,n])$



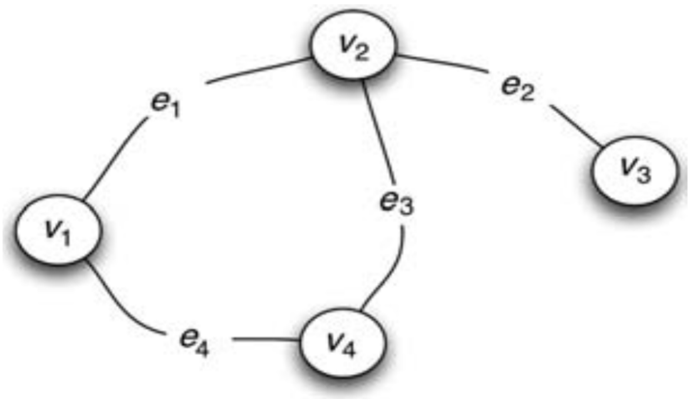
$$C(G) = \begin{array}{c|cccc} v_1 & 0 & 0 & 0 & 2 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 0 \end{array} = \begin{pmatrix} 0 & 0 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A(G) = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 0 & 0 & 1 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 0 \end{array} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Laplacian Matrix

- Laplacian matrix
  - The *Laplacian matrix* of graph *G*, namely, *L*(*G*), is a combination of the connection matrix and (diagonal) degree matrix: *L* = *C* – *D*, where *D* is a diagonal matrix and *C* is the connection matrix

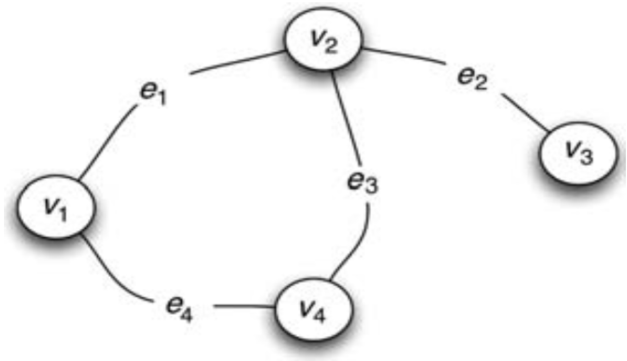
$$d_{i,j} = \begin{cases} \text{degree}(v_i) & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$



$$L(G) = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & -2 & 1 & 0 & 1 \\ v_2 & 1 & -3 & 1 & 1 \\ v_3 & 0 & 1 & -1 & 0 \\ v_4 & 1 & 1 & 0 & -2 \end{array} = \begin{pmatrix} -2 & 1 & 0 & 1 \\ 1 & -3 & 1 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & -2 \end{pmatrix}$$

Path Matrix

- Path matrix
  - Path matrix *P*(*G*) stores the number of hops along the direct path between all node pairs in a graph
  - *P*(*G*) enumerates the lengths of shortest paths among all node pairs



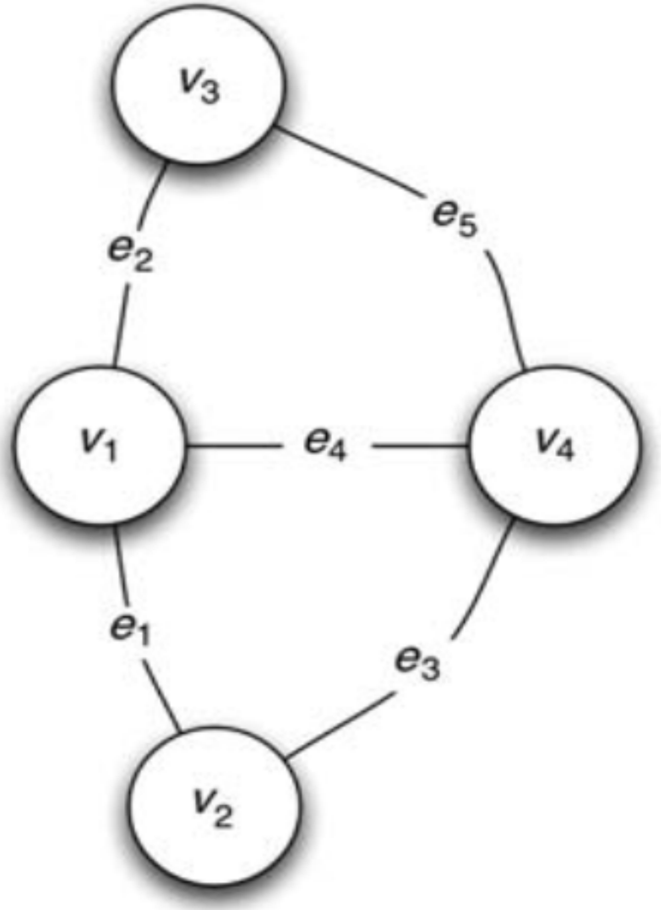
$$P = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 2 & 1 \\ v_2 & 1 & 0 & 1 & 1 \\ v_3 & 2 & 1 & 0 & 2 \\ v_4 & 1 & 1 & 2 & 0 \end{array} = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix}$$

  - Let D be the size of the longest path – the diameter of G

$$P = \min_{k=1}^{D}\{kA^k\}$$

Euler Path & Euler Circuit

- A path that returns to its starting point is called a *circuit*
- A path that traverses all links of a graph is called an *Euler path*
- A *Euler circuit* is a *Euler path* that begins and ends with the same node



Spectral Properties of Graphs

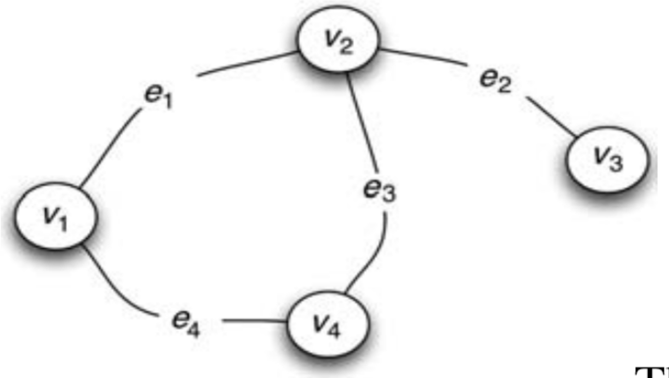
- Adjacency matrix
- Spectral decomposition
  - If λ is a diagonal matrix, then *A* may be decomposed as *A* = λ ***I*** where ***I*** is the identity matrix and λ is a matrix containing eigenvalues

$$\lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

  - det [*A* - λ ***I***] = 0
  - Eigenvalues are the diagonals $\lambda_1, \lambda_2, \ldots, \lambda_n$

Spectral Radius

- Spectral radius
  - The *spectral radius* ρ (G) is the largest nontrivial eigenvalue of det [$A(G) - \lambda \boldsymbol{I}$] =0
  - $\boldsymbol{A}$ is the adjacency matrix and $\boldsymbol{I}$ is the identity matrix
- How to compute?



$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \Longrightarrow \det \begin{bmatrix} -\lambda & 1 & 0 & 1 \\ 1 & -\lambda & 1 & 1 \\ 0 & 1 & -\lambda & 0 \\ 1 & 1 & 0 & -\lambda \end{bmatrix} = 0$$

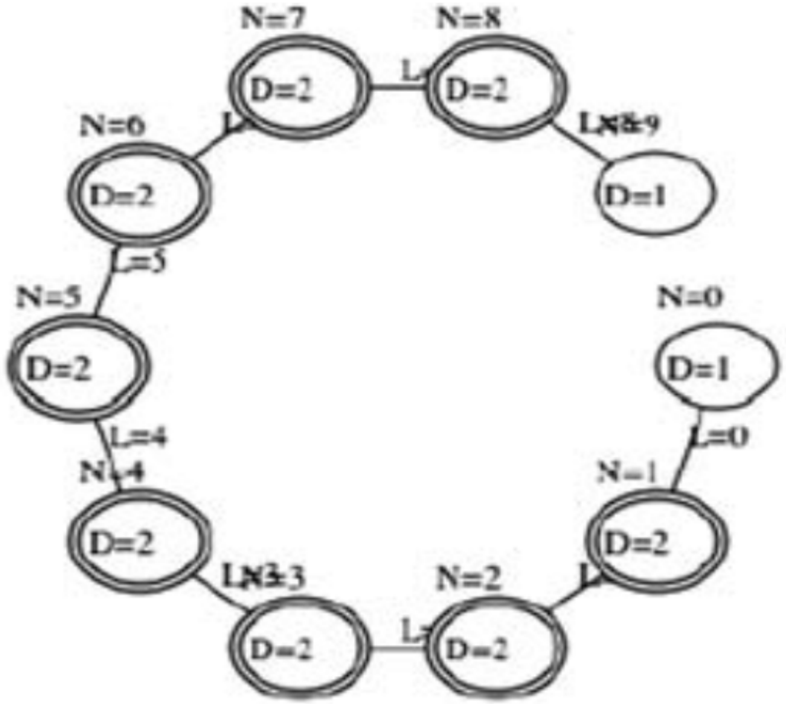
Expanding the determinant along column 3 using Laplace's expansion formula:

$$\lambda^4 - 4\lambda^2 - 2\lambda + 1 = 0$$

The roots are $\{-1.48, -1.0, 0.311, 2.17\}$, so $\rho = 2.17$.

Type of Graphs

- Line
  - The mapping function of a *line graph* defines a linear sequence of nodes, each connected to a *successor* node
  - The first and last nodes have degree 1
  - All intermediate nodes have degree



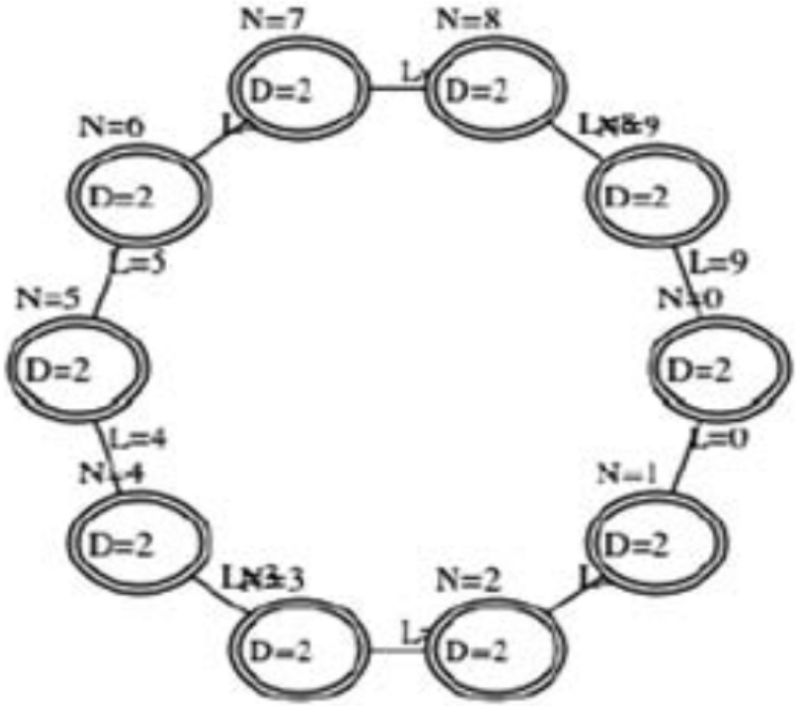
$$f_{\text{line}} = [e_i : v_i \sim v_{i+1}]; \quad i = 1,2,\ldots,n-1$$

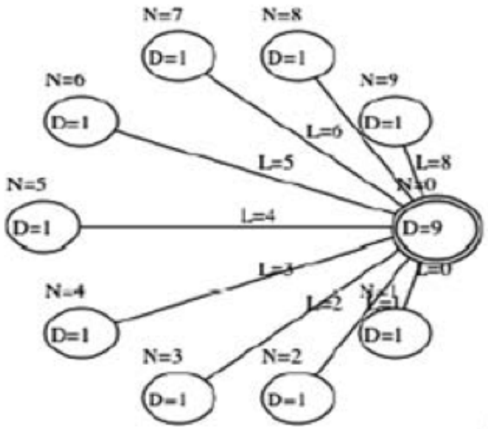
- Barbell

Type of Graphs

- Ring
  - Similar to *line graph,* but the ending node in the chain or sequence connects to the starting node

$$f_{\text{ring}} = \left[e_i : v_i \sim v_{i(\bmod n)+1}\right]; \quad i = 1,2,\ldots,n$$





(c)

Average Path Length

- The APL for a line and ring graph of *n* nodes
  - Analyze the path matrix for two cases: even and odd *n*.
  - Sum all nonzero elements of the path matrix (denoted as T).
    - In the case of a line graph, *T* is the sum of off-diagonal elements.
    - In the case of the ring network, *T* is the sum of the rows of the path matrix.
  - The number of nonzero elements of the symmetric path matrix is equal to $n(n-1)$
  - Average path length is $T/\, n(n-1)$.

$$\text{avg\_path\_length(line)} \sim O\left(\frac{n}{3}\right)$$

$$\text{avg\_path\_length(ring)} \sim O\left(\frac{n}{4}\right)$$

Average Path Length

- Line graph

$$\text{Matrix total} = T = 2\sum_{i=1}^{n-1}\{i(n-i)\} = 2\left[n\sum_{i=1}^{n-1} i - \sum_{i=1}^{n-1} i^2\right]$$

$$\text{Given } \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \quad \text{and} \quad \sum_{i=1}^{n-1} i^2 = \frac{(n(n-1)(2n-1))}{6}$$

$$T = n^2(n-1) - \frac{n(n-1)(2n-1)}{3} = n(n-1)\left[\frac{n-(2n-1)}{3}\right] = \frac{n(n-1)[n+1]}{3}$$

$$\text{avg\_path\_length} = \frac{T}{(n(n-1))} = \frac{(n+1)}{3} \quad \text{or} \quad O\left(\frac{n}{3}\right);\ n \gg 1 \text{ [Line]}$$

Average Path Length

- Ring graph

$$\text{row\_total} = 2\sum_{i=1}^{(n/2)-1} i + \left(\frac{n}{2}\right);\ \text{even } n$$

$$= 2\sum_{i=1}^{(n-1)/2} i;\ \text{odd } n$$

There are n rows, so $T = n(\text{row\_total})$. But the average path length is $T/(n(n-1))$, so

$$\text{avg\_path\_length} = \frac{(n(\text{row\_total}))}{(n(n-1))} = \frac{\text{row\_total}}{(n-1)}$$

$$= \frac{(n/2)^2}{(n-1)} = \frac{n^2}{(4(n-1))};\ \text{even } n$$

$$= \frac{(n+1)}{4};\ \text{odd } n$$

Assuming $n \gg 1$, so that $n/(n-1) \sim 1$, the average path length of a ring is

$$\text{avg\_path\_length} \sim O\left(\frac{n}{4}\right);\ \ n \gg 1\ \text{[Ring]}$$

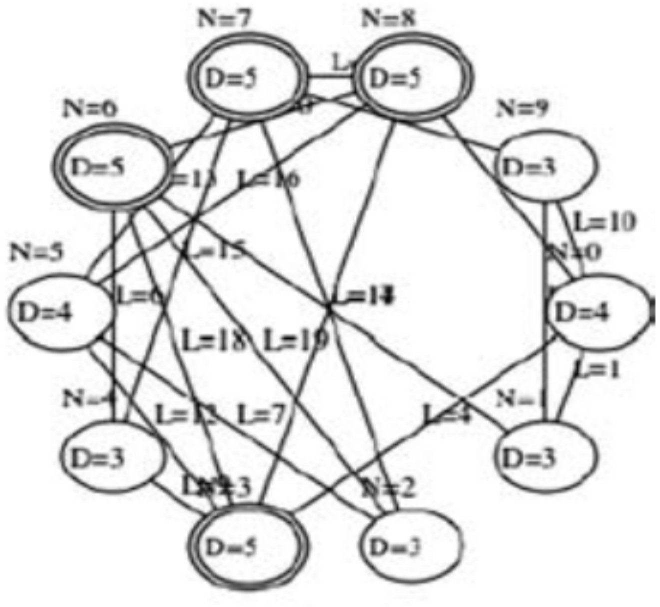
Random Graph

- A random graph is constructed by randomly selecting a tail, then randomly selecting a head node, and then connecting them with a link

- The mapping function uses random numbers r_t and r_h to select nodes:

$$f_{\text{random}} = [e_i : v_{1+r_t n} \sim v_{1+r_h n}];$$

$$i = 1,2,\ldots,m, \text{ where } m = \text{number of links}$$

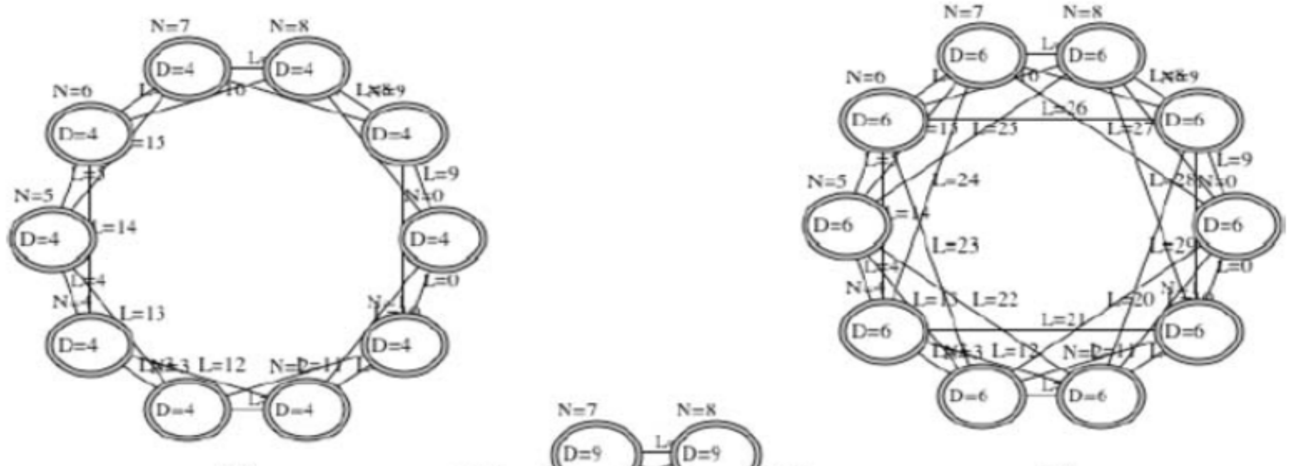


- Sample r from a uniform distribution

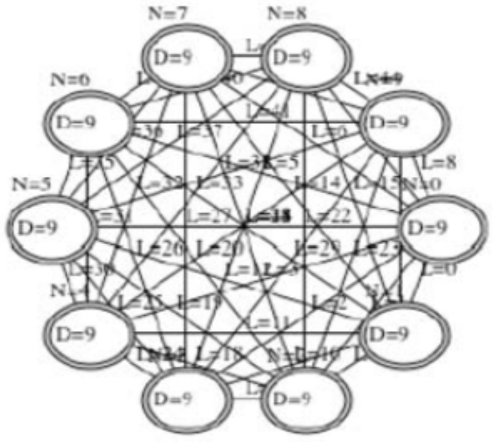
Structured Versus Random

- Structured graphs (regular graphs)
  - The mapping function establishes some kind of pattern (visually or in the adjacency matrix)
    - Ring graph, line graph, complete graph…
- Unstructured graphs (random graphs)
  - No discernible pattern appears
- Between structured graph and unstructured graph
  - *k*-Regular Graphs
    - Each node has *k* degree exactly

k-Regular Graphs



(a) (b)



(c)

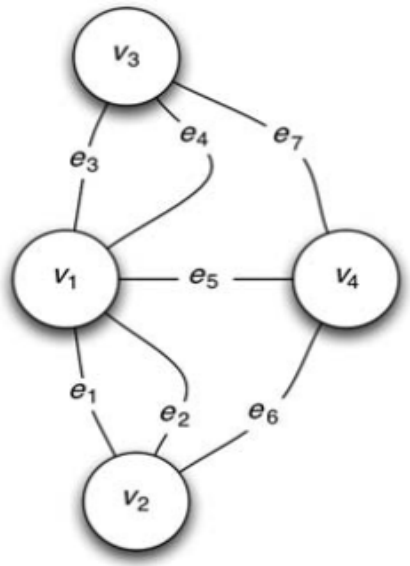
$$\text{Density}(G) = \frac{\#\text{ links}}{\left(n\frac{(n-1)}{2}\right)} = \frac{2m}{(n(n-1))}$$

Figure k-Regular graphs: (a) 2-regular graph nodes connect to two sequential successors; (b) 3-regular graph nodes connect to three sequential successors; (c) a complete graph links every node to every other node.

Topological Structure

- Degree sequence
  - g = [d_1, d_2, …, d_n] define a *degree sequence* containing the degree values of all n nodes in *G*
- Degree sequence distribution
  - g' = [h_1, h_2, …, h_{max_d}] where

$$h_1 = \text{fraction of nodes with degree 1}$$
$$h_2 = \text{fraction of nodes with degree 2}$$
$$\vdots$$
$$h_{\max\_d} = \text{fraction of nodes with } \max\_d = \text{maximum degree (hub) of G}$$



$$g = [5,3,3,3] \qquad g' = \left[0,0,\frac{3}{4},0,\frac{1}{4}\right] = [0,0,0.75,0.25]$$

Topological Structure

- Scale-free topology
  - Poisson process: the probability of obtaining exactly *k* successes in *m* trials is given by the binomial distribution

$$B(k,m) = C\binom{m}{k} p^k (1-p)^{m-k}$$

$B(k,m)$ is approximated by the Poisson distribution by replacing p with (λ/m), in $B(k,m)$, and letting m grow without bound:

$$H(k) = \lambda^k \frac{\exp(-\lambda)}{k!}$$

where $\lambda =$ mean node degree; $k =$ node degree.

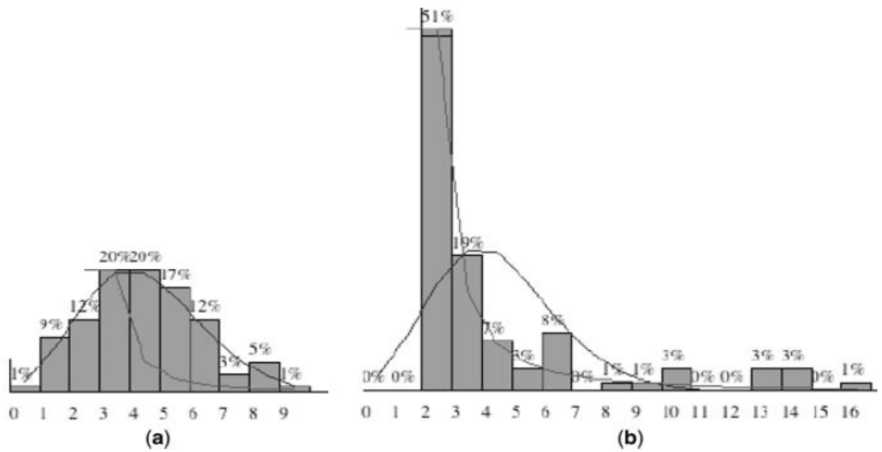
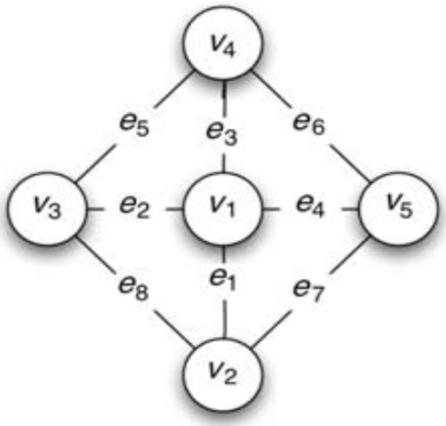


Figure Degree histogram for (a) a random graph and (b) a scale-free graph. One line graph shows a Poisson distribution, and the other line graph shows a power-law fit to the histogram data.

Topological Structure

- Small-world topology
  - A *small-world* graph, *G*, is a graph with relatively small average path length, and a relatively high *cluster coefficient*, CC(G).
  - For a node u, suppose that the neighbors share c links, then the cluster coefficient of node u, Cc(u), is

$$Cc(u) = \frac{2c}{\text{degree}(u)(\text{degree}(u) - 1)}$$



$$Cc(v_1) = \frac{2(4)}{12} = \frac{8}{12} = \frac{2}{3} \qquad CC(G) = \sum_{i=1}^{n} \frac{Cc(v_i)}{n} = \frac{2}{3}$$

TABLE 1 Some Common Examples of Small-World Networks

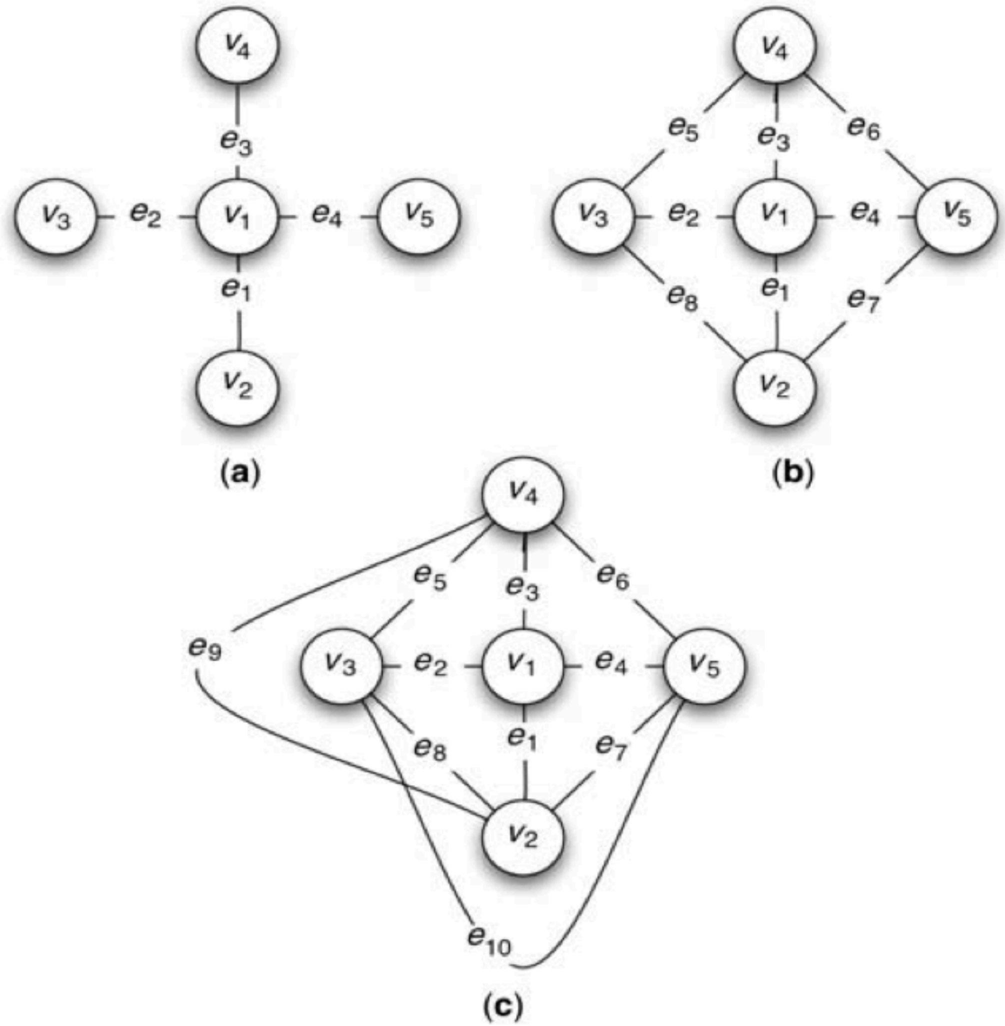
Graph	Size, n	Small-World Cluster Coefficient	Random Cluster Coefficient
World Wide Web	153,127	0.11	0.00023
Internet	6,209	0.30	0.00100
Actors in same movie	225,226	0.79	0.00027
Coauthor scientific papers	52,909	0.43	0.00018
Western US power grid	4,941	0.08	0.00200
C. elegans neural network	282	0.28	0.05000
Foodweb (ecological chain)	134	0.22	0.06000

TABLE 2 Comparison of Some Properties of Graphs (n = 100, m = 200), and p = 5% for the Small-World Graph

Property	Random	Scale-Free	Small-World	2-Regular
Hub degree	10	21	10	4
Average degree	4.0	3.94	4.0	4
Distribution	Poisson	Power	Poisson-like	Delta(4)
Average path length	3.38	3.08	4.0	12.88
Diameter	7	5	9	25
Cluster coefficient	0.045	0.156	0.544	0.500
Entropy	2.9	2.3	0.9	0.0

(*p* = 5% means only 5% of the Small-World Graph links are random, while 95% are regular)

Calculate Cluster Coefficients



REGULAR NETWORKS

Link Efficiency

- Link Efficiency
  - The tradeoff between number of links and number of hops in the average path length of a network:

$$\mathrm{E}(G) = \frac{m - avg\_path\_length(G)}{m}$$

    where *m* is the number of links in *G*

  - Let *t* be the total number of paths and $r_{i,j}$ the length of the direct path between node v_i and v_j:

$$\mathrm{avg\_path\_length} = \sum_i \sum_j \frac{r_{i,j}}{t}$$

  - A network is *scalable* if link efficiency approaches 100% as network size *n* approaches infinity

TABLE 1 **Link Efficiency of Several Network Classes, $n \gg 1$**

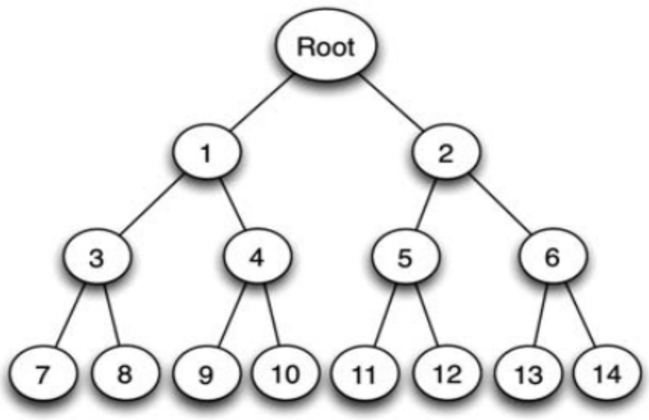
Network Class	Efficiency	Example
Line	$\frac{2n-4}{3(n-1)}$	Asymptotic to $\frac{2}{3}$
Ring	$\frac{3n-1}{4n}$	Asymptotic to $\frac{3}{4}$
Binary tree	$1-\frac{2\log_2(n+1)-6}{n-1}$	$n=127$, $m=126$, $E=93.4\%$
Toroid	$1-\frac{1}{4\sqrt{n}}$	$n=100$, $m=200$, $E=97.5\%$
Random	98.31%	$n=100$, $m=200$, avg_path_length $=3.38$
Hypercube	$1-\frac{1}{n-1}$	$n=128$, $m=448$, $E=99.2\%$
Complete	~ 1.0	$m=n\frac{n-1}{2}$, avg_path_length $=1$

Binary Tree Network

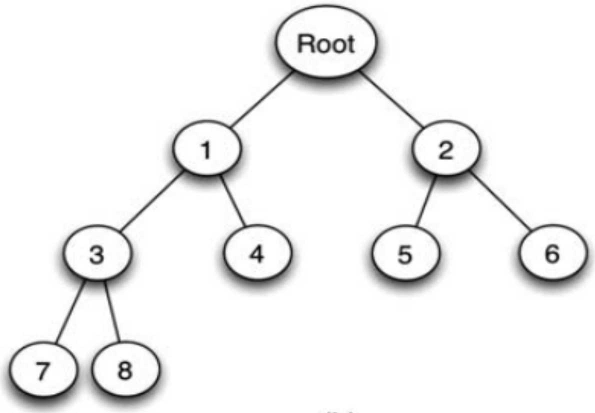
- A line graph is not link-efficient
  - The number of links grows as fast as the number of hops in its average path length
- The *binary tree* is more link-efficient
- A binary tree is defined recursively
  - The root node, has degree 2 and connects two subtrees, which in turn connect to two more subtrees, and so forth
  - This recursion ends with a set of nodes called the leaf nodes, which have degree 1
  - As it grows, its average path length grows much slower than its number of links

Binary Tree Network

- Balanced binary tree
  - A balanced binary tree contains *k* levels and exactly $2^k - 1$ nodes, m = (n – 1) links, for *k* =1,2, ...
- Unbalanced binary tree
  - An unbalanced binary tree contains less than 2k 2 1 nodes



(a)



(b)

Properties

- Center
  - The root node with radius r = k – 1
  - the leaf nodes lie at the extreme diameter, which is D = 2(k – 1) hops
- Diameter
  - Grows logarithmic with size n because $k = O(\log_2(n))$
- Average path length
  - Also grows logarithmically, is proportional to its diameter

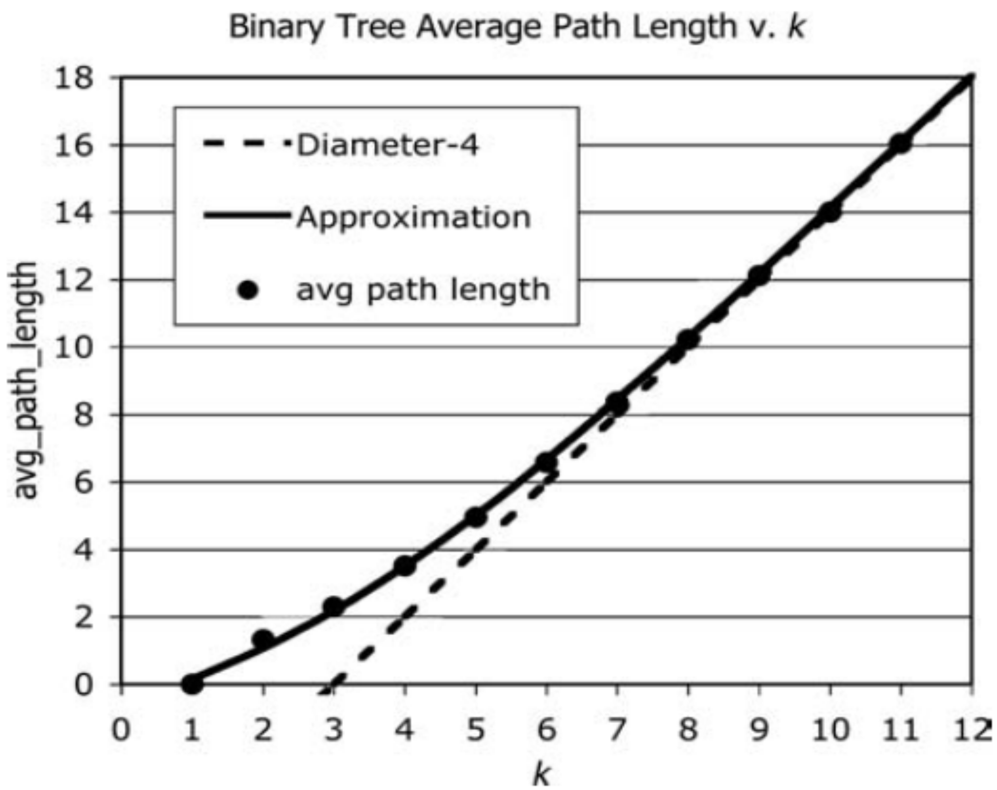


Figure Path length and $(D - 4)$ versus level k for a balanced binary tree with $n = 2^k - 1$ nodes, $m = n - 1$ links, and diameter $= D = 2(k - 1)$.

Average path length and $(D - 4)$ merge for high values of k. Thus, average path length is asymptotic to $(D - 4)$:

$$\text{avg\_path\_length(balanced binary tree)} = (D - 4);\ k \gg 1$$

$$D = 2(k - 1), \text{ so avg\_path\_length} = 2k - 6 = 2\log_2(n + 1) - 6$$

- For smaller values of *k, say, k < 9,* the approximation breaks down
  - The nonlinear portion of the approximation diminishes exponentially as *k* increases — reaching zero as (D – 4) dominates:

$$\text{avg\_path\_length} = (D-4) + \frac{A}{1+\exp(B\,\mathrm{k})}$$

where A =10.67, B = 0.45 gives the best fit.

  - Substituting $D = 2(k - 1)$ and $k = \log_2(n+1)$

$$\text{avg\_path\_length} = 2\log_2(n+1) - 6 + \frac{10.67}{1+\exp(0.45\log_2(n+1))}$$

Link Efficiency

- A balanced binary tree has m = n – 1 links
- Link efficiency of a "large" balanced binary tree is:

$$E(\text{balanced binary tree}) = 1 - \frac{D-4}{m} = 1 - \frac{(2k-1)-4}{n-1}; \quad k > 9$$

$$E = 1 - \frac{2\log_2(n+1) - 6}{n-1}, \text{ because } k = \log_2(n+1)$$

- Assuming *k* >> 1

$$E(\text{balanced binary tree}) = 1 - \frac{2\log_2(n)}{n}; \quad k > 9$$

- Binary tree link efficiency approaches 100%, as n grows without bound