3 Network Theory

In this section we will first give a basic introduction to graph theory and will then introduce some popular families of networks and investigate their structural properties.

3.1 Graph theory

A *graph* $G = (V, E)$ consists of a set of *nodes* (or *vertices*) V and a set of *edges* (or *arcs*) E. The nodes represent the processing units and the edges represent the communication links between the units. Often, we will set $n = |V|$ (the size of V) and $m = |E|$. The *size* of G is defined as the number of nodes it contains. For all $v, w \in V$, (v, w) denotes a *directed* edge from v to w, and $\{v, w\}$ denotes an *undirected* edge from v to w. G is called *undirected* if $E \subseteq \{\{v, w\} \mid v, w \in V\}$ and *directed* if $E \subseteq \{(v, w) \mid v, w \in V\}$. Unless explicitly mentioned, we assume for the rest of this lecture that G is undirected.

A sequence of contiguous edges in G is called a *path*. The *length* of the path is defined as the number of edges it contains. A path is called *node-simple* if it visits every node in G at most once. Similarly, it is called *edge-simple* (or *simple*) if it contains every edge in G at most once. G is called *connected* if, for any pair of nodes $v, w \in V$, there is a path in G from v to w. We call a simple path a *cycle* if it starts and ends at the same node. The *girth* of a graph G is defined as the length of the shortest cycle G contains. G is called a *tree* if it is connected and contains no cycle. A graph $T = (V', E')$ is called a *spanning tree* of G if $V' = V$, $E' \subseteq E$, and T is a tree. G is called *bipartite* if its node set can be partitioned into two node sets V_1 and V_2 such that $E \subseteq \{\{v, w\} \mid v \in V_1,\ w \in V_2\}$.

For any pair of nodes $v, w \in V$, let $\delta(v, w)$ denote the *distance* of v and w in G, that is, the length of a shortest path from v to w. The *diameter* D of G is defined as $\max\{\delta(v, w) \mid v, w \in V\}$. If $\{v, w\} \in E$ then v is called a *neighbor* of w. For any subset $U \subseteq V$, the *neighborhood* of U is defined as

$$\Gamma(U) = \{v \in V \setminus U \mid \exists u \in U : \ \{u, v\} \in E\} .$$

The number of neighbors of v is called the *degree* of v and denoted by d_v. The degree of G is defined as $d = \max\{d_v \mid v \in V\}$. If all nodes in G have the same degree, then G is called *regular*.

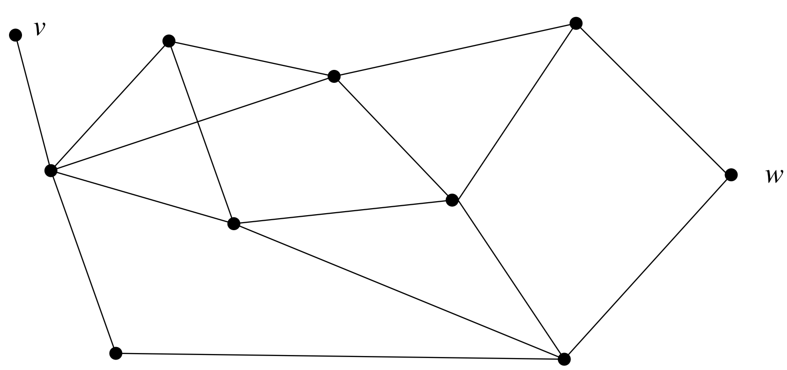

Figure 1: An example of an undirected graph with diameter 4.

A family of graphs $\mathcal{G} = \{G_n \mid n \in \mathbb{N}\}$ has degree $d(n)$ if for all $n \in \mathbb{N}$ the degree of G_n is $d(n)$. If it is clear to which family a graph belongs, we say that this graph has constant (or bounded) degree if and only if its family has constant degree.

A *network* is specified by a graph $G = (V, E)$ with edge capacities given by a function $c : E \to \mathbb{R}^+$. Given a graph G with capacities c, let the capacity of a node $v \in V$ be defined as

$$c(v) = \sum_{w \in V} c(v, w)$$

and the capacity of any node set or edge set U be defined as $c(U) = \sum_{u \in U} c(u)$. Given a subset $U \subseteq V$, $(U, \bar{U})$ denotes the set of all edges $(u, v) \in E$ (or $\{u, v\} \in E$ if G is undirected) with $u \in U$ and $v \in \bar{U}$. So $c(U, \bar{U})$ is the sum of the capacities of all edges in $(U, \bar{U})$. The *expansion* α of a network G with capacities c is defined as

$$\alpha = \min_{U \subseteq V} \frac{c(U, \bar{U})}{\min\{c(U), c(\bar{U})\}} \ .$$

3.2 Basic network topologies

The most basic network topologies used in practice are trees, cycles, grids and tori. Many other suggested networks are simply combinations or derivatives of these. The advantage of trees is that the path selection problem is very easy: for every source-destination pair there is only one possible simple path. However, since the root of a tree is usually a severe bottleneck, so-called *fat trees* have been used. These trees have the property that higher-level edges have a (much) larger capacity than lower-level edges. See Figure 2 for an example.

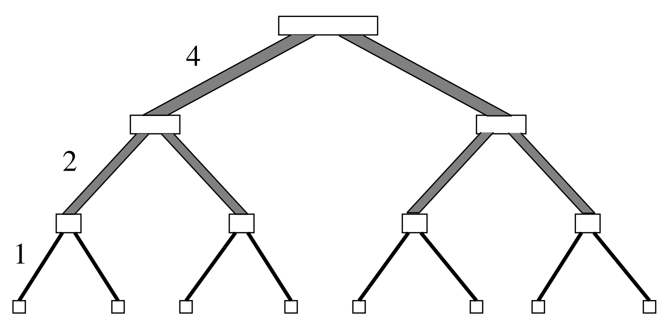

Figure 2: The structure of a fat tree.

Fat trees belong to a family of networks that require edges of non-uniform capacity to be efficient. Easier to build are networks with edges of uniform capacity. This is usually the case for grids and tori. Unless explicitly mentioned, we will treat all edges in the following to be of capacity 1. In the following, $[x]$ means the set $\{0, 1, \ldots, x-1\}$.

Definition 3.1 (Torus, Mesh) *Let* $m, d \in \mathbb{N}$. *The* (m, d)-mesh $M(m, d)$ *is a graph with node set* $V = [m]^d$ *and edge set*

$$E = \left\{ \{(a_{d-1} \ldots a_0), (b_{d-1} \ldots b_0)\} \mid a_i, b_i \in [m],\ \sum_{i=0}^{d-1} |a_i - b_i| = 1 \right\} \ .$$

The (m, d)-torus $T(m, d)$ *is a graph that consists of an* (m, d)*-mesh and additionally wrap-around edges from* $(a_{d-1} \dots a_{i+1}(m-1)\ a_{i-1} \dots a_0)$ *to* $(a_{d-1} \dots a_{i+1}\ 0\ a_{i-1} \dots a_0)$ *for all* $i \in [d]$ *and all* $a_j \in [m]$ *with* $j \neq i$*.* $M(m, 1)$ *is also called a* line*,* $T(m, 1)$ *a* cycle*, and* $M(2, d) = T(2, d)$ *a* d-dimensional hypercube*.*

Figure 3 presents a linear array, a torus, and a hypercube.

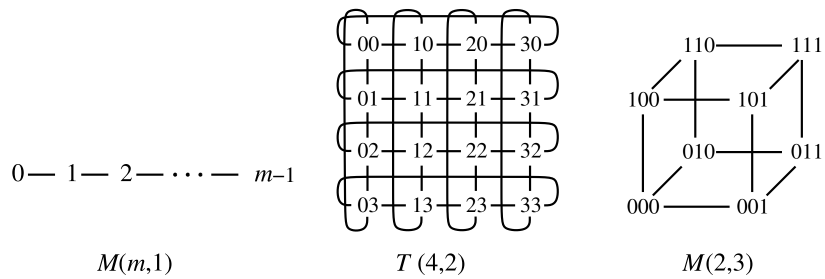

Figure 3: The structure of $M(m, 1)$, $T(4, 2)$, and $M(2, 3)$.

The hypercube is a very important class of networks, and many derivatives, the so-called *hypercubic networks*, have been suggested for it. Among these are the butterfly, cube-connected-cycles, shuffle-exchange, and de Bruijn graph. We start with the butterfly, which is basically a rolled out version of a hypercube.

Definition 3.2 (Butterfly) *Let* $d \in \mathbb{N}$*. The* d-dimensional butterfly $BF(d)$ *is a graph with node set* $V = [d+1] \times [2]^d$ *and an edge set* $E = E_1 \cup E_2$ *with*

$$E_1 = \{\{(i, \alpha), (i+1, \alpha)\} \mid i \in [d],\ \alpha \in [2]^d\}$$

and

$$E_2 = \{\{(i, \alpha), (i+1, \beta)\} \mid i \in [d],\ \alpha, \beta \in [2]^d,\ \alpha \textit{ and } \beta \textit{ differ only at the } i\textit{th position}\}\ .$$

The node set $\{(i, \alpha) \mid \alpha \in [2]^d\}$ *represents* level i *of the butterfly. The* d-dimensional wrap-around butterfly *W-BF*(d) *is defined by taking the* $BF(d)$ *and identifying level* d *with level 0.*

Figure 4 shows the 3-dimensional butterfly $BF(3)$. The $BF(d)$ has $(d+1)2^d$ nodes, $2d \cdot 2^d$ edges and degree 4. It is not difficult to check that combining the node sets $\{(i, \alpha) \mid i \in [d]\}$ into a single node results in the hypercube.

Next we define the cube-connected-cycles network. It only has a degree of 3 and it results from the hypercube by replacing the corners by cycles.

Definition 3.3 (Cube-Connected-Cycles) *Let* $d \in \mathbb{N}$*. The* cube-connected-cycles *network CCC(*d*) is a graph with node set* $V = \{(a, p) \mid a \in [2]^d, p \in [d]\}$ *and edge set*

$$\begin{aligned} E = &\Big\{\{(a,p),(a,(p+1) \bmod d)\} \mid a \in [2]^d, p \in [d]\Big\} \\ &\cup \Big\{\{(a,p),(b,p)\} \mid a, b \in [2]^d, p \in [d], a = b \text{ except for } a_p\Big\} \end{aligned}$$

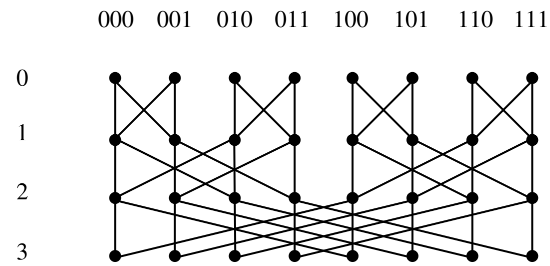

Figure 4: The structure of BF(3).

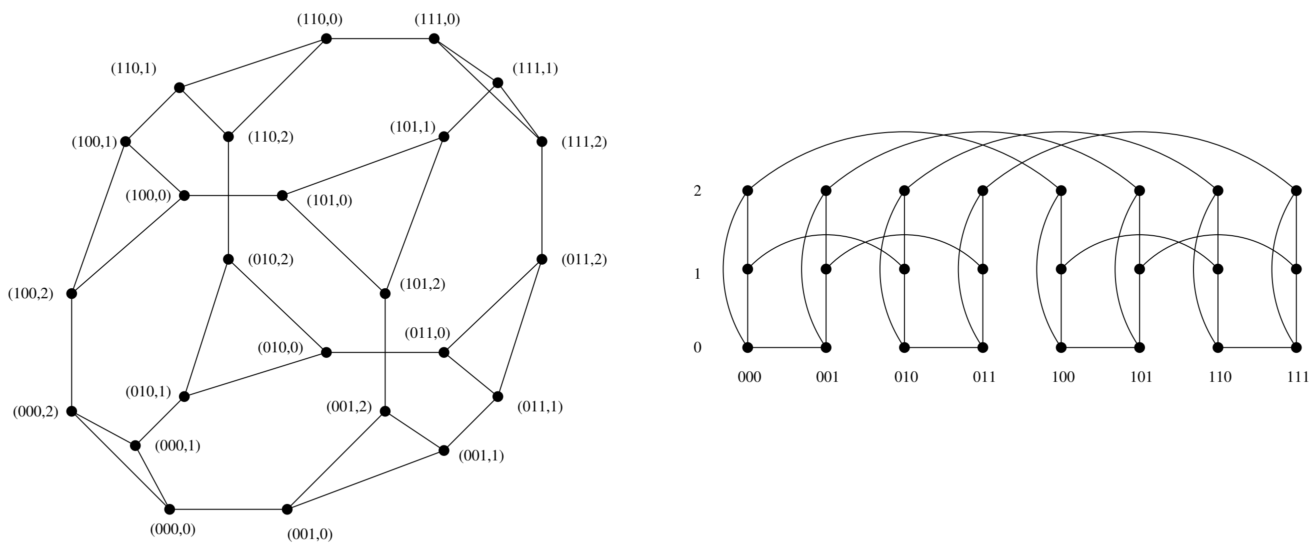

Figure 5: The structure of CCC(3).

Two possible representations of a CCC can be found in Figure 5.

The shuffle-exchange is yet another way of transforming the hypercubic interconnection structure into a constant degree network.

Definition 3.4 (Shuffle-Exchange) *Let* $d \in \mathbb{N}$. *The* d-dimensional shuffle-exchange $SE(d)$ *is defined as an undirected graph with node set* $V = [2]^d$ *and an edge set* $E = E_1 \cup E_2$ *with*

$$E_1 = \{\{(a_{d-1} \dots a_0), (a_{d-1} \dots \bar{a}_0)\} \mid (a_{d-1} \dots a_0) \in [2]^d,\ \bar{a}_0 = 1 - a_0\}$$

and

$$E_2 = \{\{(a_{d-1} \dots a_0), (a_0 a_{d-1} \dots a_1)\} \mid (a_{d-1} \dots a_0) \in [2]^d\} \ .$$

Figure 6 shows the 3- and 4-dimensional shuffle-exchange graph.

Definition 3.5 (DeBruijn) *The* b-ary DeBruijn graph of dimension d $DB(b,d)$ *is an undirected graph* $G = (V,E)$ *with node set* $V = \{v \in [b]^d\}$ *and edge set* E *that contains all edges* $\{v,w\}$ *with the property that* $w \in \{(x, v_{d-1}, \dots, v_1) :\ x \in [b]\}$, *where* $v = (v_{d-1}, \dots, v_0)$.

Two examples of a DeBruijn graph can be found in Figure 7.

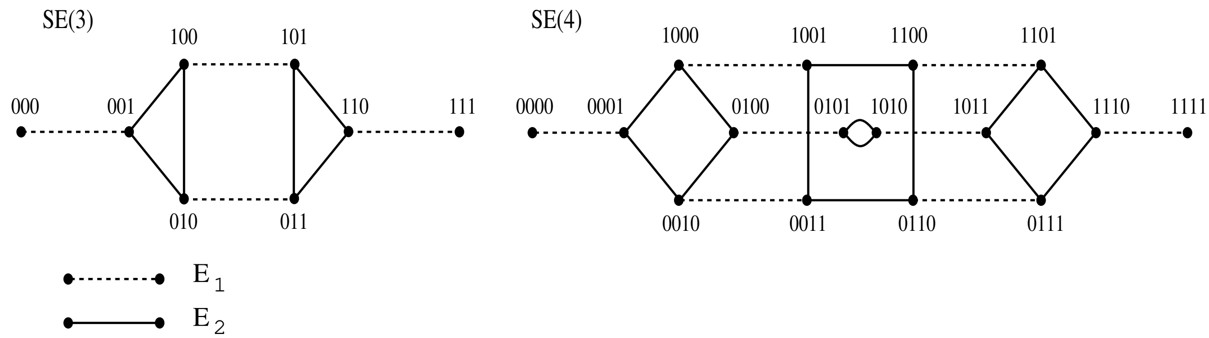

Figure 6: The structure of SE(3) and SE(4).

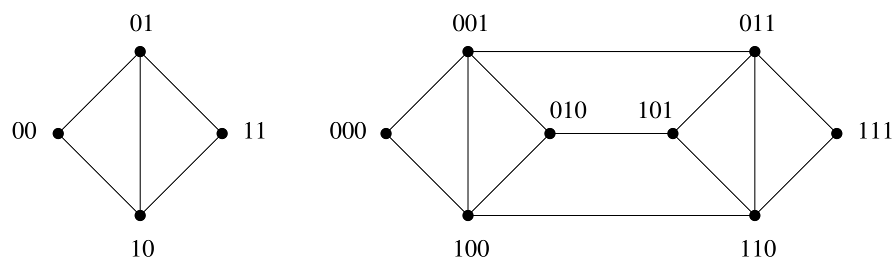

Figure 7: The structure of $DB(2,2)$ and $DB(2,3)$.

3.3 Direct and indirect networks

Networks are usually separated into *direct* and *indirect* networks. Direct networks are networks in which every node represents a processing unit that can inject and absorb packets, whereas in indirect networks only certain nodes (the so-called *input nodes*) can inject packets and certain nodes (the so-called *output nodes*) can absorb packets. An important subclass of indirect networks are the so-called leveled graphs.

Definition 3.6 (Leveled Graph) *A graph $G = (V, E)$ is called* leveled with depth D *if the nodes of G can be partitioned into $D + 1$* levels *$L_0, \dots, L_D$ such that every edge in E connects nodes of consecutive levels. Nodes in level 0 are called* inputs*, and nodes in level D are called* outputs*. If, in addition, $|L_0| = |L_D|$ and L_0 is identified with L_D, then G is called a* wrapped leveled graph with depth D.

Examples of leveled graphs are the fat tree and the butterfly, and an example of a wrapped leveled graph is the wrap-around butterfly. In a butterfly it is usually assumed that the nodes in L_0 represent the input nodes and the nodes in level L_D represent the output nodes. In a fat tree the nodes in level L_D are usually both input and output nodes.

3.4 The diameter

Recall the definition of the diameter in Section 3.1. One important goal in choosing a topology for a network is that it has a small diameter. The following theorem presents a lower bound for this.

Theorem 3.7 *Every graph of maximum degree $d > 2$ and size n must have a diameter of at least $\lfloor(\log n)/(\log(d-1))\rfloor - 1$.*

Proof. Suppose we have a graph $G = (V, E)$ of maximum degree d and size n. Start from any node $v \in V$. In a first step at most d other nodes can be reached. In two steps at most $d \cdot (d-1)$ additional nodes can be reached. Thus, in general, in at most k steps at most

$$1 + \sum_{i=0}^{k-1} d \cdot (d-1)^i = 1 + d \cdot \frac{(d-1)^k - 1}{(d-1) - 1} \leq \frac{d \cdot (d-1)^k}{d-2}$$

nodes (including v) can be reached. This has to be at least n to ensure that v can reach all other nodes in V within k steps. Hence,

$$(d-1)^k \geq \frac{(d-2) \cdot n}{d} \quad \Leftrightarrow \quad k \geq \log_{d-1}((d-2) \cdot n/d) \; .$$

Since $\log_{d-1}((d-2)/d) > -2$ for all $d > 2$, this is true only if $k \geq \lfloor \log_{d-1} n \rfloor - 1$. □

Theorem 3.7 uses as a construction for the lower bound a complete $(d-1)$-ary tree with a root of degree d. However, it is easy to see that in this tree there are two nodes (see the leaves v and w in Figure 8) with a distance of approximately $2 \log_{d-1} n$, which is by a factor of 2 larger than the lower bound. Can networks with a better diameter be constructed? The next theorem gives an answer to this.

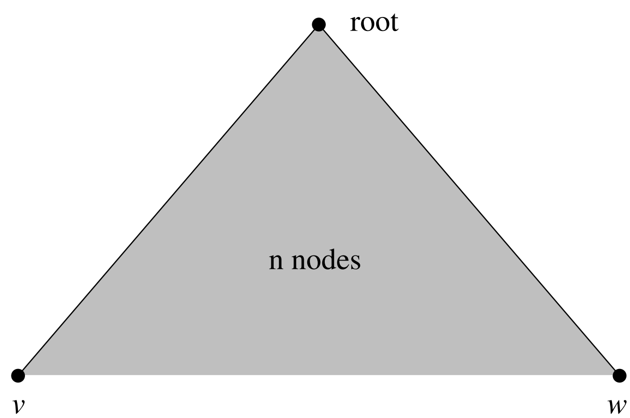

Figure 8: Nodes with highest distance in a tree.

Theorem 3.8 *For every even $d > 2$ there is an infinite family of graphs G_n of maximum degree d and size n with a diameter of at most $(\log n)/(\log d - 1)$.*

Proof. The proof is part of the assignment. □

3.5 The expansion

Recall the definition of the expansion in Section 3.1. We start with an upper bound on the expansion that must hold for all networks.

Theorem 3.9 *For every network $G = (V, E)$ with non-negative edge capacities, the expansion can be at most 1.*

Proof. For every set $U \subseteq V$ let $E_U = \{\{v, w\} \in E \mid v \in U\}$, where an edge appears twice in E_U if both v and w are in U. Certainly, $(U, \bar{U}) \subseteq E_U$. Since $c(U) = c(E_U)$ it must therefore hold that $c(U, \bar{U}) \leq c(U)$. Equivalently, it must also hold that $c(U, \bar{U}) = c(\bar{U}, U) \leq c(\bar{U})$. Hence, $c(U, \bar{U}) \leq \min\{c(U), c(\bar{U})\}$ and therefore

$$\alpha(G) = \min_{U \subseteq V} \frac{c(U, \bar{U})}{\min\{c(U), c(\bar{U})\}} \leq 1 \ .$$

□

Interestingly, for any $d \geq 3$ there are graphs that can achieve a constant expansion. These are the so-called expanders. One explicit construction is known as the Gabber-Galil graph [1]:

Definition 3.10 *Gabber-Galil] Let $n \in \mathbb{N}$. The* Gabber-Galil graph *$GG(n)$ is a graph with node set $V = [n]^2$ and edge set E consisting of all edges $((x, y), (x', y'))$ with*

$$(x', y') \in \{(x, x+y), (x, x+y+1), (x+y, y), (x+y+1, y)\} \pmod{n}$$

Other explicit constructions of expanders can be found in [4, 5, 6]. Also random graphs are expanders, with high probability. For the classes of graphs we presented above the expansion is quite complicated to compute. Therefore, we just list some results here.

Theorem 3.11 *The d-dimensional hypercube, cube-connected-cycles, butterfly, shuffle-exchange, and DeBruijn graph with uniform edge capacities all have an expansion of $\Theta(1/d)$.*

Using the fact that for these networks $d = \Theta(\log n)$, where n is the number of nodes in the network, it follows that all of these networks have an expansion of $\Theta(1/\log n)$.

3.6 The flow number

We will look at one more parameter called the *flow number*. For this we first have to introduce the concept of multicommodity flows. Consider any network G with non-negative edge capacities given by c. A *multicommodity flow* instance on G is a set of ordered pairs of vertices $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. Each pair (s_i, t_i) represents a *commodity* with source s_i and target t_i. The objective is to maximize the amount of flow traveling from the sources to the corresponding destinations, subject to the capacity constraints. The problem comes in two flavors. In the first, called the *maximum multicommodity flow* problem, the total flow, summed over all commodities, is to be maximized. The second is called the *concurrent multicommodity flow* problem. Here, for each commodity (s_i, t_i) a non-negative demand d_i is specified. The objective is to maximize the *fraction* of the demand that can be shipped simultaneously for all commodities. In other words, we want to find the maximum f so that a flow of $f \cdot d_i$

can be shipped for every commodity i without exceeding the capacities of the edges. f is called *concurrent max-flow*. A *balanced multicommodity flow problem* (BMFP) is a concurrent multicommodity flow problem in which the sum of the demands of the commodities originating and the commodities terminating in a node v is equal to $c(v)$ for every $v \in V$. Both the maximum throughput problem and the maximum concurrent flow problem can be solved in polynomial time using linear programming.

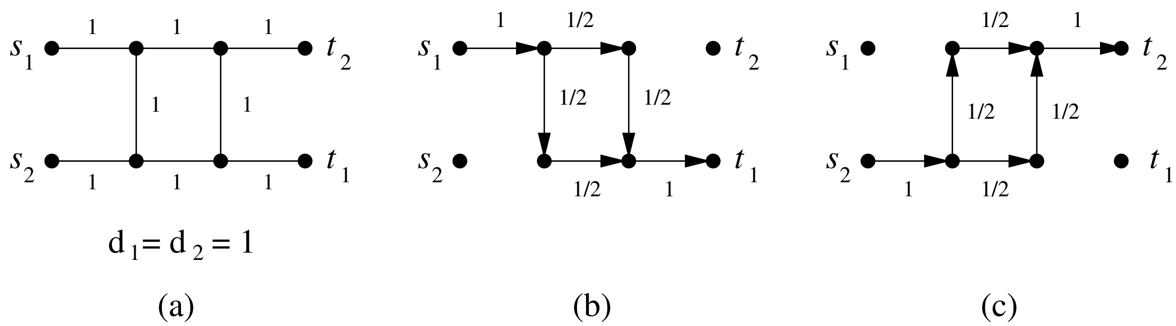

Figure 9: Solution to a 2-commodity flow problem (a). The routing of the first commodity is shown in (b) and the second commodity is shown in (c).

Given a concurrent multicommodity flow problem with feasible solution S, let the *dilation* $D(S)$ of S be defined as the length of the longest flow path in S and the *congestion* $C(S)$ of $\mathcal{S}$ be defined as the inverse of its concurrent flow value (i.e., the congestion says how many times the edge capacities would have to be increased in order to satisfy the demands of all commodities when using the same set of flow paths). Let $\mathcal{B}$ be the special BMFP in which each pair of nodes (v, w) has a commodity of demand $c(v) \cdot c(w)/c(V)$. The *flow number* $F(G)$ of a network G is defined as the minimum over all feasible solutions S of $\mathcal{B}$ of $\max\{C(S), D(S)\}$ [2]. In the case that there is no risk of confusion, we will simply write F instead of $F(G)$. Note that the flow number of a network is invariant to a scaling of the capacities.

The flow number of a network G can be computed in polynomial time. Another advantage of the flow number is that, as shown by the next theorem, it can be applied to much more general multicommodity flow problems than just the one that defines it.

Theorem 3.12 *For any network G with flow number F and any instance I of the BMFP for G, there is a feasible solution for I with congestion and dilation at most $2F$.*

Proof. The idea is to decompose I into two multicommodity flow problems: for every commodity i with source s_i and destination t_i, the first problem I_1 has commodities i_u from s_i to u for all $u \in V$ with demands $d_{i_u} = d_i \cdot c(u)/c(V)$, and the second problem I_2 has commodities i'_u from u to t_i for all $u \in V$ with demands $d_{i'_u} = d_i \cdot c(u)/c(V)$. For every commodity i from the original problem, the total demand of corresponding commodities in I_1 is d_i and is d_i in I_2 as well. Moreover, for every node $u \in V$ the amount of commodity i shipped to u in I_1 is equal to the amount of commodity i shipped from u in I_2.

Interestingly, both of the flow problems I_1 and I_2 are equal to the special flow problem $\mathcal{B}$ because for any pair $v, w \in V$, the total demand of the commodities with source v and destination w in I_1 is equal to

$$\sum_{i:\, s_i = v} \frac{d_i \cdot c(w)}{c(V)} = \frac{c(v) \cdot c(w)}{c(V)} \ ,$$

and in I_2 it is also equal to

$$\sum_{i:\, t_i=w} \frac{d_i \cdot c(v)}{c(V)} = \frac{c(v) \cdot c(w)}{c(V)} \ .$$

Thus, according to the definition of the flow number, both I_1 and I_2 have a feasible solution with congestion and dilation at most F. Hence, the original problem I has a feasible solution with congestion and dilation at most $2F$, which proves the claim. $\square$

With techniques similar to those used in the proof of Theorem 5.0.3 in [7] one can also prove the following result.

Theorem 3.13 *On average over all BMFPs I, the minimum $\max\{C(\mathcal{S}), D(\mathcal{S})\}$ over all feasible solutions $\mathcal{S}$ of I is $\Omega(F)$.*

Hence, the flow number truthfully captures the problem of routing BMFPs in networks. Next, we explore the relationship of the flow number with the diameter and the expansion of a network. The first result immediately follows from the definition of F.

Fact 3.14 *For every network with diameter D and flow number F, it holds that $F \geq D$.*

The next result reveals a very close relationship between the expansion and the flow number of a network.

Theorem 3.15 *For any network G with expansion α and flow number F it holds that*

$$\alpha^{-1} \leq F \leq c \cdot \alpha^{-1} \log n$$

for some constant c.

Proof. We only prove here that $F \geq \alpha^{-1}$. (The entire proof can be found in [2].) For this we need some notation. Given a concurrent multicommodity flow problem, the *cut ratio* of a cut $(U, \bar{U})$ is defined as

$$R_U = \frac{c(U,\bar{U})}{d(U,\bar{U})} \quad \text{where} \quad d(U,\bar{U}) = \sum_{(s_i,t_i) \in (U \times \bar{U}) \cup (\bar{U} \times U)} d_i \ .$$

Now, let f be the concurrent max-flow of the problem $\mathcal{B}$ used for the definition of F. Consider any cut $(U, \bar{U})$ and let $i_1, i_2, \ldots, i_r$ denote the commodities whose source and target are separated by this cut. Since all flows for these commodities must cross $(U, \bar{U})$, we know that

$$\sum_{j=1}^{r} f \cdot d_{i_j} \leq c(U, \bar{U}) \ .$$

Since $\sum_{j=1}^{r} d_{i_j} = d(U, \bar{U})$, this means that

$$f \leq \frac{c(U,\bar{U})}{d(U,\bar{U})} \ .$$

For $\mathcal{B}$ it holds that

$$d(U,\bar{U}) = \sum_{(u,v)\in(U\times\bar{U})\cup(\bar{U}\times U)} \frac{c(u)\cdot c(v)}{c(V)} = \frac{2c(U)\cdot c(\bar{U})}{c(V)} .$$

We distinguish between two cases. If $c(U) \geq c(V)/2$, then $c(U)\cdot c(\bar{U})/c(V) \geq c(\bar{U})/2$. Thus,

$$f \leq \frac{c(U,\bar{U})}{2\cdot c(\bar{U})/2} = \frac{c(U,\bar{U})}{\min\{c(U),c(\bar{U})\}} .$$

If $c(\bar{U}) \geq c(V)/2$, then $c(U)\cdot c(\bar{U})/c(V) \geq c(U)/2$ and therefore

$$f \leq \frac{c(U,\bar{U})}{2\cdot c(U)/2} = \frac{c(U,\bar{U})}{\min\{c(U),c(\bar{U})\}} .$$

Hence, in both cases,

$$f \leq \frac{c(U,\bar{U})}{\min\{c(U),c(\bar{U})\}}$$

and therefore $f \leq \alpha$ or $1/f \geq \alpha^{-1}$. Since according to the definition of F, $F \geq 1/f$, it follows that $F \geq \alpha^{-1}$. □

Since the flow number of a network is an upper bound on its diameter, it follows from Theorem 3.15:

Corollary 3.16 *For every network with expansion α the diameter is at most $O(\alpha^{-1}\log n)$.*

From Theorem 3.7 it follows that this bound is exact for constant degree expanders. Do there exist networks where the flow number is in $O(\max\{D,\alpha^{-1}\})$? The next theorem lists some.

Theorem 3.17 *The d-dimensional hypercube, cube-connected-cycles, butterfly, shuffle-exchange, and DeBruijn graph with uniform edge capacities all have a flow number of $\Theta(d)$.*

For proofs see, for example, [3] or [7]. Thus, for these networks it actually holds that $F = \Theta(\alpha^{-1})$, i.e. the expansion describes very well the routing ability of the network. It also follows from the bound that all networks must have a diameter of $O(\log n)$.

References

[1] O. Gabber and Z. Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

[2] P. Kolman and C. Scheideler. Improved bounds for the unsplittable flow problem. In *ACM/SIAM Symp. on Discrete Algorithms (SODA)*, 2002.

[3] F. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays $\cdot$ Trees $\cdot$ $Hypercubes$.* Morgan Kaufmann Publishers (San Mateo, CA), 1992.

[4] A. Lubotzky, R. Phillips, and R. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[5] G. Margulis. Explicit group theoretical constructions of combinatorial schemes and their application to the design of expanders and superconcentrators. *Problems Inform. Transmission*, 11:39–46, 1988.

[6] M. Morgenstern. Existence and explicit constructions of $q + 1$ regular Ramanujan graphs for every prime power q. *Journal of Combinatorial Theory, Series B*, 62:44–62, 1994.

[7] C. Scheideler. *Universal Routing Strategies for Interconnection Networks*, volume 1390 of *Lecture Notes in Computer Science*. Springer, 1998.