Self-Routing Superconcentrators

Nicholas Pippenger*

Department of Computer Science
The University of British Columbia
Vancouver, British Columbia V6T 1Z4
CANADA

Abstract: Superconcentrators are switching systems that solve the generic problem of interconnecting clients and servers during sessions, in situations where either the clients or the servers are interchangeable (so that it does not matter which client is connected to which server). Previous constructions of superconcentrators have required an external agent to find the interconnections appropriate in each instance. We remedy this shortcoming by constructing superconcentrators that are "self-routing", in the sense that they compute for themselves the required interconnections.

Specifically, we show how to construct, for each n, a system S_n with the following properties. (1) The system S_n has n inputs, n outputs, and $O(n)$ components, each of which is of one of a fixed finite number of finite automata, and is connected to a fixed finite number of other components through cables, each of which carries signals from a fixed finite alphabet. (2) When some of the inputs, and an equal number of outputs, are "marked" (by the presentation of a certain signal), then after $O(\log n)$ steps (a time proportional to the "diameter" of the network) the system will establish a set of disjoint paths from the marked inputs to the marked outputs.

---

* This research was partially supported by an NSERC Operating Grant and an ASI Fellowship Award.

1. Introduction

Our main goal in this paper is to define the notion of a "self-routing superconcentrator" and show how to construct self-routing superconcentrators that are optimal (to within constant factors) in a number of respects. Superconcentrators (and the more specialized concentrators) are networks providing disjoint paths from inputs to outputs in situations wherein it does not matter which input is connected to which output. The most fundamental result concerning superconcentrators is that they can be built from a number of components (basic switching elements) proportional to the number of inputs and outputs. (In this respect they contrast with permuting networks, and other networks that provide paths between specific inputs and specific outputs, which require a non-linear number of components.) One obstacle to the application of these networks is that some external agent must find the paths provided by the network in each instance. This amounts to finding a maximum flow in a network with unit capacities or, equivalently, to finding a series of matchings in bipartite graphs. Unfortunately, all known algorithms for this problem require a decidedly non-linear number of operations, even for a serial algorithm running on a single central processor.

A self-routing superconcentrator overcomes this obstacle by solving its own routing problem, using a small amount of hardware associated with each switching element, together with protocols that allow this hardware to find the desired paths in a completely distributed way. These ideas are formulated in terms of a system of interconnected finite automata. This formulation ensures that the path-finding process, as well as the path-providing system, scales without any non-linearity.

Several previous results may reasonably be viewed as furnishing self-routing networks for certain problems. Firstly, we have the sorting, merging and classifying networks built from comparators (classical results are due to Batcher [7], and Ajtai, Komlós and Szemerédi [3, 4]). In these networks the finite automata are particularly simple, and the control information flows unidirectionally through the network. Secondly, Arora, Leighton and Maggs [5, 6] have described what may be called "self-routing non-blocking networks". Our formulation is based on theirs, and many of the techniques we use have been taken from their papers.

A subsidiary goal of this paper is to show how some of our ideas can be used to improve several previous results concerning circuit- and packet-switching networks. These improvements are somewhat technical, and deal with the amount of "expansion" needed for various constructions; this in turn affects whether elementary explicit constructions can be used, or whether known explicit constructions can be used at all, to supply this expansion. One of these previous result is that of Arora, Leighton and Maggs [5, 6]; another is a fault-tolerant packet-routing scheme due to Leighton and Maggs [11].

Section 2 of this paper discusses concentrators, superconcentrators and self-routing superconcentrators in turn, concluding with the statement of our main theorem. In Section 3 we present the basic combinatorial lemmas used in the proof of this theorem. In Section 4 we use these lemmas to develop some protocols that will serve as building blocks in our final routing protocol. In Section 5 we present the proof of the main theorem. In Section 6 we discuss the applicability of these ideas to the other problems mentioned above. A preliminary version of this paper appears as Pippenger [18′].

2. Self-Routing Networks

A *network* $N = (V, E, A, B)$ comprises (1) an acyclic directed graph $G = (V, E)$ with *vertices* V and *edges* E, (2) a set A of distinguished vertices having in-degree zero called *inputs*, and (3) a set B (disjoint from A) of distinguished vertices having out-degree zero called *outputs*. A network with m inputs and n outputs will be called an (m, n)-*network* or, if $m = n$, an n-*network*. A directed path joining an input to an output will be called a *route*, and a set of vertex-disjoint routes will be called a *state*.

We shall be concerned with three "complexity measures" for networks: the number of edges, which we shall call the *size*; the largest number of edges in any route, which we shall call the *depth*; and the largest total degree (in-degree plus out-degree) of any vertex, which we shall call the *valence*.

An (m, n)-*concentrator*, where $m > n$, is an (m, n)-network $N = (V, E, A, B)$ with the following property: given any set $X \subseteq A$ of inputs with $\#X \leq n$ (where $\#X$ denotes the cardinality of X), there exists a state of N containing routes originating at each input of X. (The routes of a state must of course terminate at distinct outputs.) A concentrator is schematic solution to the problem of interconnecting "clients" with "servers" during "sessions", in situations for which either all the clients, or all the servers, are equivalent, so that it does not matter which is connected to which.

The notion of a concentrator was defined in 1973 by Pinsker [16], who proved the existence of (m, n)-concentrators with size at most $29m$. Pinsker's proof was noteworthy as being the first published "randomized construction" of a switching network, thus introducing what has become one of the central tools of this theory.

An n-*superconcentrator* is an n-network $N = (V, E, A, B)$ with the following property: given any set $X \subseteq A$ of inputs and any set $Y \subseteq B$ of outputs with $\#X = \#Y$, there exists a state containing routes originating at each of the inputs of X and terminating at each of the outputs of Y. The notion of a superconcentrator was introduced by Aho, Hopcroft and Ullman [2], who attributed it to conversations with R. W. Floyd. Their intended use was as a tool for establishing non-linear lower bounds for the complexity of circuits computing Boolean functions. This hope was dashed by Valiant [20], who used Pinsker's result in a recursive construction to show that there exist n-superconcentrators of size $O(n)$.

Though a failure for their original purpose, superconcentrators have proved useful for constructing other types of switching networks, including concentrators: they satisfy a condition that is similar to that of concentrators, but that is symmetrical between inputs and outputs; and this symmetry facilitates recursive constructions. Pippenger [17] showed (by a direct randomized construction, without using concentrators as a building block) the existence of n-superconcentrators with size $O(n)$, depth $O(\log n)$ and valence $O(1)$. (These depth and valence bounds will be important for us later; Valiant's original argument established only depth $O\big((\log n)^2\big)$ and valence $O(\log n / \log\log n)$.)

All of the interconnection properties discussed in the previous section are defined in terms of the *existence* of certain paths or sets of paths. In all cases, practical exploitation of these networks requires the use of algorithms that actually find these paths in each instance. This raises the question of the computational complexity of these path-finding or *routing* problems. For linear-sized superconcentrators, all known constructions require the solution of matching problems in bipartite graphs; for this problem the best algorithm known (due to Hopcroft and Karp [9]) yields a routing algorithm running in serial time $O(n^{3/2})$. A significant advance in algorithms for bipartite matching would be required to bring even this serial algorithm within logarithmic factors of the linear input and output size.

Since parallel and distributed computation constitutes one of the main areas of application for switching networks, it is natural to seek switching networks having parallel and distributed routing algorithms, ideally those in which the routing can be performed by simple hardware associated with each switching element. Sorting networks, and other networks based on comparators, were one of the earliest embodiments of this idea.

With appropriate input-output conventions, two sorting networks can be used as a self-routing superconcentrator. Imagine that the k superconcentrator inputs seeking connections present a "1" signal, while the remaining $n-k$ superconcentrator inputs present a "0" signal. If these signals are presented at the inputs of a sorting network, the 1's will appear at the k highest-number outputs of this sorting network. If the k superconcentrator outputs seeking connections present "1" signals to the inputs of another sorting network, these 1's will appear at the k highest-number outputs of this second sorting network. Thus by connecting the two sorting networks together, outputs to outputs, the signals from the superconcentrator inputs and outputs seeking connections will rendezvous at the common outputs of the sorting networks. If the best constructions known (see Ajtai, Komlós and Szemerédi [3, 4]) are used for the sorting networks, we obtain a construction for self-routing superconcentrators of size $O(n \log n)$, depth $O(\log n)$ and valence $O(1)$.

To make further progress, we abandon the requirement that the network be composed of comparators, and allow the vertices to be copies of an arbitrary finite automaton. The edges then carry signals from an arbitrary finite alphabet, and signals may pass in both direction over the edges. This model was introduced by Arora, Leighton and Maggs [5, 6]. In the purely graph-theoretic model, it is customary to require routes to be vertex-disjoint. When discussing protocols, however, it is often more convenient to require only that they be edge-disjoint, or even to allow an arbitrary fixed number of routes to pass through each vertex and edge (in this situation, one speaks of "congestion" $O(1)$). Such extended networks can be reduced to standard form by replacing each link (that is, each vertex that is not an input or output) by a complete bipartite graph of appropriate fixed size, and each edge by a bundle of edges.

For self-routing networks, it will be convenient to introduce two additional complexity measures to account for the resources they use to solve their routing problem. The total number of steps taken, between the presentation of the input and output signals and the arrival at a stable state incorporating the established routes, will be called the *latency* of the network. The sum over all components of the number of actual steps taken by that component (that is, the number of times that the component actually changes its state) will be called the *action* of the network. (In reckoning either the latency or the action, we consider the input and output signals that maximize the quantity in question, so that we are considering worst-case performance.) If a connected n-network has valence $O(1)$, it will have depth $\Omega(\log n)$, and this implies that the latency is also $\Omega(\log n)$. On the other hand, the action may be less that the product of the size and the latency: in the example of sorting networks, the size and latency are $O(n \log n)$ and $O(\log n)$, respectively, but the action is $O(n \log n)$, since each comparator changes state just once.

Theorem 1: There is a finite set of finite automata from which, for every n, a self-routing superconcentrator with size $O(n)$ and depth $O(\log n)$, latency $O(\log n)$ and action $O(n)$, can be explicitly constructed in space $O(\log n)$.

3. Combinatorial Lemmas

In this section we present some graph-theoretic lemmas that will be needed for the construction of self-routing superconcentrators. The first two (Lemmas 2 and 3, which support Proposition 4) are based on the notion of a "compressor", as introduced by Pippenger [18]. The third (Lemma 5) is a straightforward instance of an "expander".

Let $G = (A, B, E)$ be a regular bipartite multigraph of *size* n and *degree* d (that is, in which $\#A = \#B = n$, and in which every vertex in $A \cup B$ has degree d, counting edges according to their multiplicities). Let M denote the $n \times n$ adjacency matrix of G (rows are indexed by A, columns are indexed by B, and the (i, j)-th entry is the multiplicity of $\{i, j\}$ in E). Let M^T denote the transpose of M. Then $M^T M$ and $M M^T$ are real symmetric matrices with non-negative entries. Their row and column sums are all equal to d^2; thus they each have an eigenvalue of d^2 (corresponding to the constant eigenvector), and all other eigenvalues are at most d^2 in absolute value. We shall say that G has *eigenvalue separation* ε if $M^T M$ and $M M^T$ both have d^2 as a simple eigenvalue, and all other eigenvalues are at most $\varepsilon^2 d^2$ is absolute value.

We shall assume that we have available an explicit construction (say, one that can be carried out in logarithmic space) for regular bipartite multigraphs of various sizes n, but fixed degree d and fixed eigenvalue separation $\varepsilon < 1$. (For example, Jimbo and Maruoka [10] give a construction for n any perfect square, $d = 8$ and $\varepsilon = \sqrt{3}/2$. In fact, a result of Alon [1] ensures that any explicit construction for expanders with fixed degree and expansion also yields a fixed eigenvalue separation.) For $k \geq 0$, let us write G^k for the regular bipartite multigraph whose adjacency matrix is M^k. Then the degree of every vertex in G^k is d^k, and if G has eigenvalue separation ε, then G^k has eigenvalue separation ε^k. Thus we can obtain graphs with fixed (though perhaps very large) degree and eigenvalue separation as small as we please.

We may also assume that we can obtain such graphs with size n any integral power of 2. To see this, we note that the construction of Jimbo and Maruoka [10] (like those of Margulis [13] and of Gabber and Galil [8] before it) works for n any integral power of the perfect square 4; this gives us graphs for the even integral powers of 2. If we have a graph G with size n, degree d and eigenvalue separation $\varepsilon < 1$, we can obtain from 4 copies of G a graph G' (the product of G with the complete bipartite graph on two sets of 2 vertices) having size $2n$, degree $2d$ and eigenvalue separation at most ε. Thus by doubling the degree, we obtain graphs for the odd integral powers of 2.

Lemma 2: Suppose that $G = (A, B, E)$ has size n, degree d and eigenvalue separation $\varepsilon \leq 1/8$. If $X \subseteq A$ satisfies $\#X/n = \alpha \leq 1/64$, and if $Y \subseteq B$ denotes the set of vertices adjacent to more than $d/4$ vertices in X, then $\#Y/n = \beta \leq \alpha/2$.

Proof: Suppose X contains k elements, and let f be the characteristic vector of X, so that $(f, f) = k$. Let e denote the constant vector whose entries are all 1. Let $g = e(e, f)/n$ denote the projection of f onto the subspace spanned by e, and let $h = f - g$ denote the projection of f onto the complementary subspace. Then we have

$$\begin{aligned}(f, M^T M f) &= (g, M^T M g) + (h, M^T M h) \\ &\leq d^2\,(g, g) + \varepsilon^2 d^2\,(h, h) \\ &\leq \frac{d^2 k^2 + \varepsilon^2 d^2 k(n-k)}{n} \\ &\leq n d^2\big(\alpha^2 + \varepsilon^2 \alpha(1-\alpha)\big),\end{aligned}$$

since $\alpha = k/n$. On the other hand,

$$(f, M^T M f) = (Mf, Mf) = \sum_{j \in B} d_X(j)^2,$$

where $d_X(j)$ denotes the number of edges from vertices in the set X to j. Since each vertex in

$$Y = \{j \in B : d_X(j) \geq d/4\}$$

contributes at least $d^2/16$ to this sum of non-negative terms, we have

$$\sum_{j \in B} d_X(j)^2 \geq l d^2/16 = n d^2 \beta/16,$$

where l is the cardinality of Y and $\beta = l/n$. Combining these estimates yields

$$\beta \leq 16\alpha^2 + 16\varepsilon^2 \alpha(1-\alpha).$$

Since $\alpha \leq 1/64$ and $\varepsilon^2 \leq 1/64$, this implies $\beta \leq \alpha/2$. $\triangle$

Lemma 3: Suppose that $G = (A, B, E)$ has size n, degree d and eigenvalue separation $\varepsilon \leq 1/8$. If $V \subseteq B$ satisfies $\#V/n = \gamma \leq 1/64$, and if $W \subseteq A$ denotes the set of vertices adjacent to more than $d/2$ vertices in V, then $\#W/n = \delta \leq \gamma/8$.

Proof: Let f be the characteristic vector of V. Then

$$(f, M^T M f) \leq nd^2\big(\gamma^2 + \varepsilon^2\gamma(1-\gamma)\big),$$

as in the proof of Lemma 2. On the other hand,

$$(f, M^T M f) = (Mf, Mf) = \sum_{i \in A} d_V(i)^2,$$

where $d_V(i)$ denotes the number of edges from i to vertices in the set V. Since each vertex in

$$W = \{i \in A : d_V(i) \geq d/2\}$$

contributes at least $d^2/4$ to this sum of non-negative terms, we have

$$\sum_{i \in A} d_V(i)^2 \geq kd^2/4 = nd^2\delta/16,$$

where k is the cardinality of W and $\delta = k/n$. Combining these estimates yields

$$\delta \leq 4\gamma^2 + 4\varepsilon^2\gamma(1-\gamma).$$

Since $\gamma \leq 1/64$ and $\varepsilon^2 \leq 1/64$, this implies $\delta \leq \gamma/8$. $\triangle$

Let $G' = (A_1 \cup A_2, B, E_1 \cup E_2)$ be the bipartite graph obtained by taking two disjoint n-vertex sets A_1 and A_2, and two disjoint sets of edges E_1 and E_2, such that (A_1, B, E_1) and (A_2, B, E_2) are each isomorphic to (A, B, E).

Proposition 4: If $R \subseteq A_1 \cup A_2$ satisfies $\#R/n = \varrho \leq 1/64$, if $S \subseteq B$ denotes the set of vertices adjacent to more than $d/2$ vertices in R, and if $T \subset A$ denotes the set of all vertices adjacent to more than $d/2$ vertices in S, then $\#T/n \leq \varrho/4$.

Proof: For $i \in \{1, 2\}$, let S_i denote the set of vertices adjacent to more than $d/4$ vertices in $R \cap A_i$. Applying Lemma 2 with $X = R \cap A_i$, so that $\alpha \leq \varrho$, and $Y = S_i$ yields $\#S_i/n \leq \varrho/2$. Since $S \subseteq S_1 \cup S_2$, we obtain $\#S/n \leq \varrho$. For $i \in \{1, 2\}$, let T_i denote the set of vertices in A_i adjacent to more than $d/2$ vertices in S. Applying Lemma 3 with $V = S$, so that $\gamma \leq \varrho$, and $W = T_i$ yields $\#T_i/n \leq \varrho/8$. Since $T = T_1 \cup T_2$, we obtain $\#T/n \leq \varrho/4$. $\triangle$

We shall now choose a degree d_0 such that, whenever n is an integral power of 2, we can construct a regular bipartite multigraph of size n, degree d_0 and eigenvalue separation $\varepsilon_0 = 1/8$ (to which we can therefore apply Lemmas 2 and 3, and Proposition 4). For example, since $(\sqrt{3}/2)^5 \leq 1/2$, the discussion preceding Lemma 2 implies that we can choose $d_0 = 2 \cdot 8^{15} = 2^{46}$.

We now set $\varepsilon_1 = 1/256\sqrt{d_0}$, and choose a degree d_1 such that, whenever n is an integral power of 2, we can construct a regular bipartite multigraph of size n, degree d_1 and eigenvalue separation ε_1. For example, if we choose $d_0 = 2^{46}$, we can then choose $d_1 = 2 \cdot 8^{155} = 2^{466}$.

Lemma 5: Suppose that $K = (A, B, L)$ is a regular bipartite graph with size n, degree d_1, and eigenvalue separation ε_1. If $X \subseteq A$ and $Y \subseteq B$ satisfy $\#X/n \geq 1/256$ and $\#Y/n \geq 1/256d_0$ (or $\#Y/n \geq 1/256$ and $\#X/n \geq 1/256d_0$), then some vertex in X is adjacent to some vertex in Y.

Proof: Suppose that $X \subseteq A$ with $\#X = k$ and $Y \subseteq B$ with $\#Y = l$ are not joined by an edge. Suppose further that

$$\alpha\beta \geq \varepsilon_1^2, \tag{3.1}$$

where $\alpha = k/n$ and $\beta = l/n$. We shall show that these suppositions lead to a contradiction. Since either of the hypotheses of the lemma implies (3.1), this will complete the proof of the lemma.

Let N denote the adjacency matrix of K, and let f denote the characteristic vector of X. Then we have

$$(f, N^T N f) \leq \frac{d_1^2 k^2 + \varepsilon_1^2 d_1^2 k(n-k)}{n},$$

as in the proof of Lemma 2. Now let u be the characteristic vector of the complement $B \setminus Y$ of Y, so that $(u, u) = n - l$. Then we have

$$\begin{aligned}(f, N^T N f) &= (Nf, Nf) \\ &\geq \frac{(Nf, u)^2}{(u, u)},\end{aligned}$$

by Cauchy's inequality. Since no edge joins X and Y, we have

$$\frac{(Nf, u)^2}{(u, u)} = \frac{(Nf, e)^2}{(u, u)} = \frac{(f, N^T e)^2}{(u, u)} = \frac{d_1^2 k^2}{(n-l)}.$$

Combining these estimates yields

$$\varepsilon_1^2 \geq \frac{\alpha}{1-\alpha}\frac{\beta}{1-\beta},$$

which contradicts (3.1). $\triangle$

4. Basic Protocols

In this section we shall describe two protocols that will be basic building blocks in the proof of Theorem 1. Each of these protocols is based on a bipartite graph, and each can be executed by a system obtained from this graph by replacing each vertex by a finite automaton and replacing each edge by a communication channel with a finite signalling alphabet. Our first protocol is based on the graph $G' = (A', B, E')$ constructed for Proposition 4; the resulting system, operating according to this protocol, will be called a "compactor". The second protocol will based on the graph $K = (A, B, L)$ constructed for Lemma 5; the resulting system will be called a "broker".

Let $G' = (A', B, E')$ be the graph constructed for Proposition 4. Suppose that each of the $2n$ vertices in A' represents a "boy", each of the n vertices in B represents a "girl", and that each of the $2nd_0$ edges in E' represents a "boy-knows-girl" relationship. (Since G' is a multigraph, we allow boys to know girls with multiplicities greater than 1.) Suppose further that each boy in some set of at most $n/64$ boys wants to have some number not exceeding $d_0/2$ of "dates" with girls that he knows (where a given boy may date a given girl as many times as the multiplicity with which he knows her). Suppose still further that each girl is willing to have any number of dates not exceeding $d_0/2$. Then the boys will have as many dates as they want, and the girls will have no more than they are willing, if they all execute the following protocol. The protocol comprises a sequence of "rounds", where each round comprises the following three steps.

(1) Each boy that wants one or more dates sends as many "invitations" to each girl he knows as the multiplicity with which he knows her.

(2) Each girl that receives more than $d_0/2$ invitations sends back "rejections", one for each invitation received, to the boys who sent them.

(3) Each boy that receives at most $d_0/2$ rejections proceeds to have as many dates as he wants, choosing from among girls to whom he sent invitations that were not rejected.

Suppose that at the outset of the first round $\varrho n \le n/64$ boys each want one or more dates. By Proposition 4, at most $\varrho n/4$ boys receive more than $d_0/2$ rejections, and thus all but $\varrho n/4$ of the boys who want one or more dates will have as many dates as they want in step (3). Thus at the conclusion of the first round, all but one-quarter of the boys who wanted dates will be "satisfied".

During the second round, only those boys who were not satisfied during the first round will send out invitations; by the same reasoning, all but at most one-quarter of these will be satisfied during the second round. We can continue in this way until all boys have been satisfied.

It remains to verify that the girls are also "satisfied", that is, that no girl has more than $d_0/2$ dates during all the rounds of the protocol. Say that the "debut" of a girl (who has one or more dates) is the round in which she has her first date. Then she sends out no rejections during her debut (since no boy would date her if she did), and thus she receives at most $d_0/2$ invitations during her debut. Any invitations she receives in rounds after her debut will be duplicates of invitations she received during her debut, and each of these invitations results in at most one date, no matter how many times it is duplicated (since each boy has all his dates during a single round). Thus each girl has at most $d_0/2$ dates during all the rounds of the protocol.

A network whose underlying graph is G' and which operates according to the protocol just described will be called a "compactor". We shall also need "mirror-image compactors", which are obtained from compactors by exchanging the inputs and the outputs (so that there are n inputs and $2n$ outputs), and reversing the directions of the edges. We turn now from compactors to brokers.

Let $K = (A, B, L)$ be the graph constructed for Lemma 5. Suppose that each of the n vertices in A represents a boy, that each of the n vertices in B represents a girl, and that each of the nd_1 edges in L represents a boy-knows-girl relationship. Suppose further that each boy and each girl want some number (not exceeding d_0) of dates with members of the opposite sex whom they know, allowing now for multiple dates between the same boy and girl, irrespective of the multiplicity with which they know each other. Let us also assume for simplicity at this point that the total number of dates wanted by boys equals the total wanted by girls.

The nd_1 edges of the regular bipartite multigraph K can be partitioned into d_1 matchings, in such a way that each vertex is adjacent to exactly one edge of each matching. (For all currently known explicit constructions for expanders, such a partition is a manifest byproduct of the construction.)

We consider a protocol which comprises d_1 "steps", with each step corresponding to one of the matchings in K, and with the order of the steps being arbitrary.

During the step corresponding to the matching M, whenever a boy who wants one or more dates is matched in M to a girl who wants one or more dates, they proceed to have as many dates as the smaller of these numbers, thereby reducing the number of dates wanted by each of them by at least one, and reducing one of these numbers to zero.

Consider now the number of boys who still want one or more dates after all of the steps. We claim that this number is less than $n/256$. For if there were $n/256$ boys wanting one or more dates, there would be at least $n/256d_0$ girls wanting one or more dates (since each girl wants at most d_0 dates and the totals for girls and boys always remain equal). By Lemma 5, K would contain an edge between one of these boys and one of these girls. This would contradict the fact that either the boy or the girl wants no more dates after the step corresponding to the matching containing this edge. A symmetrical argument shows that fewer than $n/256$ girls still want one or more dates after consideration of all the matchings. Furthemore, even if the total numbers of dates wanted by boys and girls are not equal, we may still conclude that the number of boys (and also the number of girls) who still want dates after all the steps is at most $n/256$ plus the imbalance (the absolute value of the difference) between the numbers wanted. This can be seen by setting aside a number of unmatched boys or girls equal to the imbalance, then applying the argument given above for the balanced case.

A network whose underlying graph is K and which operates according to the protocol just decribed will be called a "broker".

5. Conclusion of the Proof

We shall now construct the self-routing superconcentrators promised by Theorem 1. The approach that we shall take is to reduce the problem to a case in which the simple construction based on sorting networks (described in Section 2) can be applied. This approach has the merit of requiring only local arguments, in that the protocols used by the various components can be considered in isolation from one another.

We begin by describing the network that will be used. Let $b = \lfloor \log_2 \log_2 n \rfloor$. Let I_0 denote the set of inputs of the network, and let J_0 denote the set of outputs. For $k = 0, 1, \ldots, b-1$, when I_k and J_k have been defined, install a broker from I_k to J_k, install a compactor whose inputs are I_k and whose outputs form a new set I_{k+1}, and install a mirror-image compactor whose outputs are J_k and whose inputs form a new set J_{k+1}. These three networks together will be called "level k". Finally, we shall install a self-routing superconcentrator Φ between I_b and J_b. This superconcentrator will have d_0 inputs for each vertex in I_b and d_0 outputs for each vertex in J_b. Thus it will have $l = nd_0/2^b = O(n/\log n)$ inputs and outputs. If it is constructed from two sorting networks as described in Section 2, and if these sorting networks are themselves constructed as described by Ajtai, Komlós and Szemerédi [3, 4], the network Φ will have size $O(l \log l) = O(n)$ and depth $O(\log l) = O(\log n)$. Thus the entire network just constructed has size $O(n)$ and depth $O(\log n)$.

We now describe the protocol by which the network finds routes satisfying given sets of requests at its inputs and outputs. There are two types of routes that can be used to satisfy requests. The first type proceeds through the broker at level k (for some $0 \le k \le b-1$), passing through the compactors at levels smaller than k. The second type proceeds through the network Φ, passing through all the compactors.

The protocol for routing will be divided into two parts, Part 1 and Part 2, where Part 1 will be be responsible for either satisfying each request or advancing it through the compactors to an input or output of Φ, and where Part 2 will be responsible for the routing in Φ. We shall focus attention on Part 1, since Part 2 is carried out by the sorting networks.

Part 1 of the protocol will itself be divided into two parts, Part 1.1 and Part 1.2. Part 1.1 will fulfill the responsibility of Part 1 for all but at most $m = nd_0/(2d_0)^b$ pairs of requests; these m or fewer pairs of requests will be "abandoned" at various times during Part 1.1. Part 1.2 will be responsible for advancing the abandoned requests to the inputs and outputs of Φ, fulfilling the responsibility of Part 1 to them.

Part 1.1 is divided into b phases, which take place at levels $k = 0, 1, \ldots, b-1$ in turn. In phase k, the networks at level k will each begin operation with at most $d_0/2$ requests at each of the $n/2^k$ vertices in each of I_k and J_k. First the broker will operate according to its protocol (using $O(1)$ steps). We shall see later that this will reduce the number of vertices in each of I_k and J_k that still have requests to at most $n/128 \cdot 2^k$.

Then the compactor will operate according to its protocol for a rounds (using $O(a)$ steps), where $a = 4 + \lceil (b+1)\log_4(2d_0)\rceil$. This will advance most of the requests at vertices in I_k to vertices in I_{k+1}, with each vertex in I_{k+1} receiving at most $d_0/2$ requests. Concurrently the mirror-image compactor will operate according to its protocol for a rounds (using $O(a)$ steps) to advance most of the requests at vertices in J_k to vertices in J_{k+1}, with each vertex in J_{k+1} receiving at most $d_0/2$ requests. At the end of these a rounds at most $d_0/2$ requests at each of at most $n/4^a 2^k$ vertices in each of I_k and J_k will remain. These requests that do not advance will be abandoned. Part 1.1 has b phases, each using $O(a)$ steps, and thus uses a total of $O(ab) = O((\log\log n)^2)$ steps in all.

The total number of requests abandoned in all of $I_0 \cup \cdots \cup I_{b-1}$ is at most

$$\sum_{0 \le k \le b-1} nd_0/4^a 2^k 2 \le nd_0/4^a = m.$$

The same bound applies to the requests abandoned in all of $J_0 \cup \cdots \cup J_{b-1}$. This gives the desired bound on the number of pairs of abandoned requests.

It remains to verify that the broker protocol in phase k reduces the number of vertices in each of I_k and J_k that still have requests to at most $n/128 \cdot 2^k$. If the numbers of requests in I_k and J_k were equal, the broker protocol would reduce the numbers of vertices that still have requests to at most $n/256 \cdot 2^k$. The numbers of requests may not be equal, because the numbers of requests abandoned on the input and output sides may not be equal, but the imbalance can be at most $m \le n/256 \cdot 2^k$, and thus can contribute at most another $n/256 \cdot 2^k$ vertices, for a total of $n/128 \cdot 2^k$.

Part 1.2 of the protocol will itself be divided into three parts, Part 1.2.1, Part 1.2.2 and Part 1.2.3. During Part 1.2.1, some vertices will be marked as "active". At the outset, the vertices in I_0 and J_0 at which requests were abandoned during Part 1.1 will be marked as active. For $k = 0, 1, \ldots, b-1$, after the active vertices in I_k and J_k have been marked, all vertices in I_{k+1} and J_{k+1} at which requests were abandoned during part 1.1, or which are adjacent to an active vertex in I_k or J_k are marked as active. Since there are at most m abandoned requests, and since each vertex in I_k or J_k has just d_0 neighbors in I_{k+1} or J_{k+1}, there will be at most $md_0^k \le n/128 \cdot 2^k$ marked vertices in each of I_k and J_k.

During Part 1.2.2, each of the compactors will operate according to its protocol (using $O(\log(md_0^k)) = O(\log n)$ steps) to assign to each marked vertex at level k a set of $d_0/2$ adjacent vertices at level $k+1$ so that each marked vertex at level $k+1$ is the assignee of at most $d_0/2$ vertices on level k.

During Part 1.2.3, edge-disjoint routes will be traced from each vertex at which a request was abandoned during Part 1.1 to a vertex in I_b or J_b. This is accomplished by considering the levels $k = 0, 1, \ldots, b-1$ in turn, using the assignments found in Part 1.2.2. In this way all requests that are have not been satisfied through brokers are advanced to I_b and J_b, with at most d_0 requests per vertex ($d_0/2$ from each of Parts 1.1 and 1.2.3), so that these request can be satisfied through Φ in Part 2.

Adding the contributions from the various parts, we find that the entire routing protocol has latency $O(\log n)$. It remains to verify that the action is $O(n)$. The components in the brokers and in the network Φ each act only $O(1)$ times during the protocol, so these networks contribute to the action in proportion to their sizes, which sum to $O(n)$. In the compactors, the number of components that act decreases geometrically at each round; thus these subnetworks also contribute in proportion to their sizes, which also sum to $O(n)$. It follows that the entire protocol has action $O(n)$, which complete the proof of Theorem 1.

6. Other Applications

The techniques used in this paper are applicable to several other problems concerning circuit- and packet-switching. To describe these applications, it will be helpful to review the ways in which expanders are used for such problems. For the purposes of this review, we may identify four "grades" of expansion that are used in various applications.

Grade 0: Sufficiently small sets must be expanded by some fixed factor exceeding 1. An example is provided by ordinary superconcentrators (see Gabber and Galil [8]).

Grade 1: Sufficiently small sets must be expanded by some fixed factor exceeding $c > 1$, where c is a threshold that depends on the application. An example is provided by ordinary packet-routers (see Upfal [19]).

Grade 2: Sufficiently small sets must be expanded by some factor exceeding $c\sqrt{d}$, where c is a constant that depends on the application, and d is the degree of the graph. Examples are the original token distribution networks (see Peleg and Upfal [15]) and the current self-routing non-blocking networks (see Arora, Leighton and Maggs [6]).

Grade 3: Sufficiently small sets must be expanded by some factor exceeding cd, where $c > 1/2$ is a constant that depends on the application, and d is the degree of the graph. Examples are fault-tolerant packet-routing networks (see Leighton and Maggs [11]) and the original self-routing non-blocking networks (see Arora, Leighton and Maggs [5]).

The differences among these grades is best appreciated by considering the sources of the expanders used. Once upon a time, random graphs were the only source of expanders, and they were used for all grades. The earliest line of work on explicit constructions for expanders, starting with Margulis [13], progressing through Gabber and Galil [8] and culminating with Jimbo and Maruoka [10], yields expanders of grades 0 and 1. These constructions are now based on elementary algebraic arguments.

The introduction of Ramanujan graphs, in the works of Lubotzky, Phillips and Sarnak [12] and Margulis [14], brought explicit constructions for grade 2 expanders, but the mathematics required to establish the properties of these graphs lies much deeper. No explicit constructions have yet been found for expanders of grade 3.

Our original construction for self-routing superconcentrators required grade 3 expanders, and thus was not explicit. This requirement was lightened, over the course of research, through grade 2 expanders to the present requirement for grade 0 expanders, and can thus be met with elementary constructions. The techniques we have used to achieve this lightening are applicable to at least three other problems that currently require grade

2 or 3 expanders, lightening their requirements to grade 0. Specifically, we can adapt the constructions of Leighton and Maggs [11] for fault-tolerant packet-routing networks and of Arora, Leighton and Maggs [6] for self-routing non-blocking networks to use any construction for regular expanders, and we can adapt the token-distribution algorithm of Peleg and Upfal to work on any regular expander. (Broder, Frieze, Shamir and Peleg [7′] have also given token-distribution algorithms that work on any expander, using techniques related to, but slightly different from, compression.) These additional applications give us confidence that the techniques introduced in this paper will have broad use for circuit- and packet-switching problems.

Acknowledgment

The author is indebted Tom Leighton and Bruce Maggs for many discussions concerning their work in [5], [6] and [11].

References

[1] N. Alon, "Eigenvalues and Expanders", *Combinatorica*, 6 (1986) 83–96.

[2] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[3] M. Ajtai, J. Komlós and E. Szemerédi, "Sorting in $c \log n$ Parallel Steps", *Combinatorica*, 3 (1983) 1–19.

[4] M. Ajtai, J. Komlós and E. Szemerédi, "An $O(n \log n)$ Sorting Network", *ACM Symp. on Theory of Computing*, 15 (1983) 1–9.

[5] S. Arora, T. Leighton and B. Maggs, "On-Line Algorithms for Path Selection in a Nonblocking Network", *ACM Symp. on Theory of Computing*, 22 (1990) 149–158.

[6] S. Arora, T. Leighton and B. Maggs "On-Line Algorithms for Path Selection in a Nonblocking Network", preprint, 1992.

[7] K. E. Batcher, "Sorting Networks and Their Applications", *Proc. AFIPS Spring Joint Computer Conf.*, 32 (1968) 307–314.

[7′] A. Z. Broder, A. M. Frieze, E. Shamir and E. Upfal, "Near-Perfect Tolen Distribution", *Internat. Colloq. on Automata, Languages and Programming*, 19 (1992) 308–317.

[8] O. Gabber and Z. Galil, "Explicit Constructions of Linear-Sized Superconcentrators", *J. Comp. and Sys. Sci.*, 22 (1981) 407–420.

[9] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ Algorithm for Maximal Matchngs in Bipartitte Graphs", *SIAM J. Computing*, 2 (1975) 225–231.

[10] S. Jimbo and A. Maruoka, "Expanders Obtained from Affine Transformations", *Combinatorica*, 7 (1987) 343–355.

[11] T. Leighton and B. Maggs, "Expanders Might Be Practical: Fast Algorithms for Routing around Faults on Multibutterflies", *IEEE Symp. on Foundations of Computer Science*, 30 (1989) 384–389.

[12] A. Lubotzky, R. Phillips and P. Sarnak, "Ramanujan Graphs", *Combinatorica*, 8 (1988) 261–277.

[13] G. A.Margulis, "Explicit Construction of Concentrators", *Problems of Inform. Transm.*, 9 (1974) 71–80.

[14] G. A. Margulis, "Explicit Group-Theoretical Constructions of Combinatorial Schemes and Their Application to the Design of Expanders and Concentrators", *Problems of Inform. Transm.*, 24 (1988) 39–46.

[15] D. Peleg and E. Upfal, "The Token Distribution Problem", *SIAM J. Computing*, 18 (1989)229–243.

[16] M. S. Pinsker, "On the Complexity of a Concentrator", *Internat. Teletraffic Congr.*, 7 (1973) 318/1–318/4.

[17] N. Pippenger, "Superconcentrators", *SIAM J. Computing*, 6 (1977) 298–304.

[18] N. Pippenger, "On Networks of Noisy Gates", *IEEE Symp. on Foundations of Computer Science*, 26 (1985) 30–38.

[18′] N. Pippenger, "Self-Routing Superconcentrators", *ACM Symp. on Theory of Computing*, 25 (1993) 355–361.

[19] E. Upfal, "An $O(\log N)$ Deterministic Packet-Routing Scheme", *ACM Symp. on Theory of Computing*, 21 (1989) 241–250.

[20] L. G. Valiant, "Graph-Theoretic Properties in Computational Complexity", *J. Comp. and Sys. Sci.*, 13 (1976) 278–285.