A Quantum Self-Routing Packet Switch

Manish Kumar Shukla, Rahul Ratan and A. Yavuz Oruç
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Email: [manishk, rahulr, yavuz]@eng.umd.edu

***Abstract*— We present the design of a self-routing quantum packet switch that exploits quantum parallelism to route all the input packets of an internally blocking self-routing network to its outputs. The packets are represented using quantum bits (qubits) and the quantum self-routing switch creates a superposition of all the maximum size non-blocking subsets of the input packets at its outputs, which cannot be achieved by any classical self-routing switching network that is internally blocking.**

I. Introduction

Recently quantum computing has become a field of intense research and has generated tremendous interest. The main motivations for this have been the advantages of quantum parallelism and entanglement, which have been used to achieve breakthroughs like Shor's factorization algorithm [1] and Grover's quantum search algorithm [2]. These algorithms either solve problems that are not feasible (i.e., have exponential complexity) in the classical domain (Shor's Algorithm) or achieve very good results while having a much lower complexity than classical algorithms (Grover's Search). We use such quantum parallelism to our advantage in designing a self-routing interconnection network without internal blocking. In particular, we present the design of a self-routing Banyan network using quantum gates.

An interconnection network is non-blocking if it can route all possible one-to-one input-output mappings. For an $N \times N$ switch these mappings are the $N!$ permutations of the output addresses. A network which cannot route at least one of these $N!$ input-output mappings is said to be blocking. Many non-blocking interconnection networks exist in the literature [3], but either their crosspoint complexity is high [4] [5] [6] [7] or they do not have simple routing schemes [8]. Therefore, networks which are blocking but have the desirable properties of simple decentralized routing along with lower crosspoint complexity, e.g. the Banyan interconnection network [9], have been investigated.

An $N \times N$ (usually $N = 2^k$) Banyan network is composed of $\log_2 N$ stages, each having $N/2$ 2×2 switches. This network is self-routing, i.e., the routing decision for a packet at any stage in the network is made solely on the basis of the output address in that packet's header [3] [10] [11]. Thus, the routing is local, decentralized and easy to implement. Since there are $(N/2)\log_2 N$ 2×2 switches in the network, it can have $2^{(N/2)\log_2 N} = N^{N/2}$ states and because this network provides a unique path between an input-output pair, this is also equal to the number of unique input-output permutation maps it can route. As a result, a Banyan network *cannot* route $N! - N^{N/2}$ permutation maps.

Thus, in a self-routing network, a contention for an output can occur between the input packets at an internal switch. Using quantum superposition both the contending packets can be sent together to their desired output (without any increase in bandwidth or additional lines) when the packets are represented by qubits (quantum bits). In a classical network sending both the packets on the same line would require either multiple parallel outputs or higher bandwidth outputs, both of which have a higher cost. Consider packets P_1 and P_2 input to a 2×2 switch S with outputs O_1 and O_2. Both packets are addressed to a single output, say O_1. Classically, either we buffer or drop one randomly chosen packet and route the other. But using quantum superposition a state of the form $1/\sqrt{2}(|P_1, P_2\rangle + |P_2, P_1\rangle)$ can be created at the outputs of the switch. The two entries in each *ket* ($|\rangle$) correspond to packets at O_1 and O_2 respectively. We can see that both P_1 and P_2 are present at O_1 and a simple measurement in computational basis ($|0\rangle, |1\rangle$) on each qubit will give either P_1 or P_2 at O_1, each with probability $1/2$. Thus, blocking doesn't occur till the measurement is made. One implication of this is that packets can be routed to their destinations without incurring any time overhead to resolve the contentions between them. Other clever measurements can be devised that lead to different results by yielding information about both the packets. Also, the superposition created in this way can be used to process the packets in parallel.

In this paper we give the complete design of the Quantum Banyan Network using basic quantum gates and explain its functionality. The rest of the paper is organized as follows. In section II, we introduce some basic concepts of quantum computing and Banyan interconnection networks. In section III, we present the design of a 2×2 quantum switch that is capable of creating a superposition of its input packets in case of an output contention. In section IV, we present the design and properties of a self-routing Banyan interconnection network constructed using 2×2 quantum switches. Section V contains the concluding remarks.

II. Preliminaries

We give a short introduction to the concepts of quantum computing and self-routing interconnection networks and also give the notation used in the rest of the paper.

A. Qubits and Superposition

The indivisible unit of classical information is the bit, which can take either one of two values: 0 or 1. The corresponding unit of quantum information is the quantum bit or *qubit*. A qubit's state is a vector in a two dimensional complex Hilbert space. The elements of an orthonormal basis for this space are represented as $|0\rangle$ and $|1\rangle$. A qubit can also exist in a superposition of the '0' and '1' states. In general, a qubit's state can be written as $|x\rangle = a\,|0\rangle + b\,|1\rangle$ where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. $|x\rangle$ is also represented as $|x\rangle = [a\ b]^T$. On measurement (w.r.t. the above basis) the qubit is observed to be found either in state $|1\rangle$ or in state $|0\rangle$ with probability $|a|^2$ and $|b|^2$ respectively. The state of a system of multiple qubits can be written by taking the tensor product of individual state vectors [12] [13].

This work was supported in part by NSF under grant 9981187.

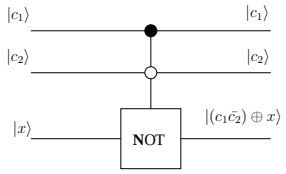


Fig. 1. A Controlled-Controlled NOT (CC-NOT) quantum gate

B. Quantum Gates

The state of qubits can be transformed via quantum gates and circuits made using such gates [14]. These gates are unitary transformations (hence are reversible) acting on a fixed number of qubits. Reversibility implies that given the output of a gate, the corresponding input is uniquely determined. A one qubit gate, which is extensively used is the Hadamard gate. The transformation matrix for this gate is

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1}$$

It transforms states $|0\rangle$ and $|1\rangle$ as: $|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Thus, a Hadamard gate puts a qubit in state $|0\rangle$ or $|1\rangle$ into an equal superposition of $|0\rangle$ and $|1\rangle$. Another type of gates that are extensively used for manipulating qubits are controlled quantum gates, e.g. a controlled-Hadamard or a controlled-NOT gate [13]. A controlled gate becomes active depending on the state of some control qubits. One such gate, controlled-controlled-NOT (CC-NOT) gate which has two control qubits (c_1 and c_2) is shown in figure 1. This gate does the following operation

$$|c_1\rangle \ \ |c_2\rangle \ \ |x\rangle \xrightarrow{\mathbf{CC-NOT}} |c_1\rangle \ \ |c_2\rangle \ \ |(c_1.\bar{c_2}) \oplus x\rangle \tag{2}$$

i.e., it inverts x when $c_1 = 1$ (indicated by solid circle) and $c_2 = 0$ (indicated by open circle). This can be extended to quantum gates with multiple control qubits. We will be using such NOT and Hadamard gates with multiple control qubits in our switch.

C. Self-Routing Interconnection Networks

A network has self-routing property if a packet can be routed by only knowing its input and output addresses, and nothing about other packets' output addresses e.g. the Omega network [10]. Usually, such an N-input network is made using 2×2 switches arranged in $\log_2 N$ stages each having $N/2$ switches ($N = 2^k$). The routing delay experienced by the packets is $O(\log N)$ which is much lower than the $O(N \log N)$ delay of centralized algorithms. Such a network (an 8×8 Banyan network) is shown in figure 2. Each 2×2 switch in this network can be either in *cross* state or in *through* state. In figure 2, switches B and C are in *through* state while A is in *cross* state. The self routing property of the network is illustrated by the example in figure 2. A packet is routed to the lower output of a switch at the i^{th} stage if the i^{th} most significant address bit is '1' and to the upper output if this bit is '0'. Thus the path taken by the packet at I_4 (with output address 010) goes via switches C, B and E. Figure 2 also shows that there can be a contention for an output at an internal switch. Such a situation occurs at switch D.

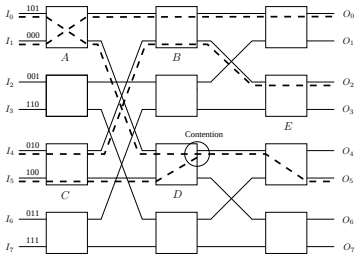


Fig. 2. An 8×8 Banyan network

III. Quantum Switch

In this section we first give the design of a basic quantum gate, which we call a switch gate. This gate is used to make a 2×2 switch that is capable of creating a superposition of its input packets at both outputs in case of a contention. We refine this design to a 2×2 switch that creates a superposition only on the output for which there is contention at the inputs. This refinement is critical for preventing erroneous routing of packets.

A. The Switch Gate

The basic building block of the Quantum switch is a quantum gate, which we call a switch gate. This gate is represented in figure 3. It is a $(2n + 1)$–qubit controlled-quantum gate having one control qubit and and two n-size sets of target qubits, each set representing a packet of size n. If the control qubit is set in state $|1\rangle$ then the gate interchanges the two sets of target qubits otherwise if the control qubit is in state $|0\rangle$, it leaves them unchanged. Representing the state of the control qubit by $|c\rangle$, and the states of the two sets of target qubits by strings $|x_1 x_2 \cdots x_n\rangle$ and $|y_1 y_2 \cdots y_n\rangle$ respectively, where $c, x_i, y_i \in \{0, 1\}$, $i = 1 \cdots n$, the function of this gate can be written as

$$|c\rangle|x_1 \cdots x_n\rangle|y_1 \cdots y_n\rangle \longrightarrow |c\rangle|u_1 \cdots u_n\rangle|v_1 \cdots v_n\rangle \tag{3}$$

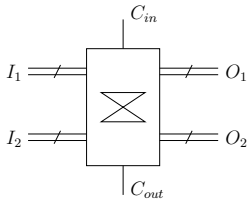
where $u_i = \bar{c}x_i + cy_i$ and $v_i = \bar{c}y_i + cx_i$. For the case when $n = 1$, i.e., when the packet size is one, it can be easily verified that the gate performs following functions depending on the state of the control qubit

$$|0\rangle|x\rangle|y\rangle \longrightarrow |0\rangle|x\rangle|y\rangle, \ |1\rangle|x\rangle|y\rangle \longrightarrow |1\rangle|y\rangle|x\rangle, \ x, y \in \{0, 1\} \tag{4}$$

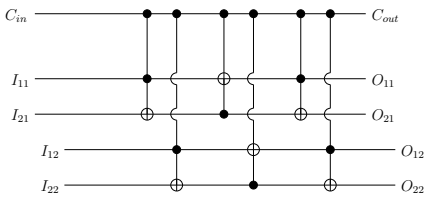
Thus for $n = 1$, matrix representation of this gate in the computational basis $|\text{control}, \text{target-1}, \text{target-2}\rangle$ is

$$\left[\begin{array}{c|cccc} \mathrm{I}_4 & \multicolumn{4}{c}{0_{4\times4}} \\ \hline & 1 & 0 & 0 & 0 \\ & 0 & 0 & 1 & 0 \\ 0_{4\times4} & 0 & 1 & 0 & 0 \\ & 0 & 0 & 0 & 1 \end{array}\right] \tag{5}$$

We use the switch gate to superpose the packets that contend for one output of a 2×2-switch and to route the superposition on that output. For example, when $n = 1$, if the control qubit of the gate is set in an equal superposition of states $|0\rangle$ and $|1\rangle$ then the action of the gate is $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|x\rangle|y\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle|x\rangle|y\rangle + |1\rangle|y\rangle|x\rangle)$. Thus an equal superposition (probability of observation = $1/2$) of packets x and y is created at both the outputs. Also, if we observe packet x at one of the outputs then packet y will be observed with certainty at the other and vice-versa.



(a) The Switch Gate



(b) Quantum circuit for the Switch Gate ($n = 2$)

Fig. 3. The switch gate

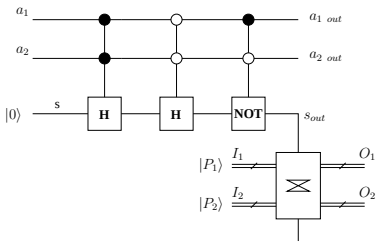


Fig. 4. A 2×2 Quantum Switch

B. A 2×2 *Quantum Switch*

As we have discussed earlier, in a self-routing switching network, the address bits present in the packet header are used to determine the path taken by the packet at the nodes of the network. For example, at the i^{th} stage of a Banyan network, a packet having i^{th} most significant address bit $a_i = 0$, is routed to the upper output of the 2×2 switch and a packet having $a_i = 1$ is routed to the lower output. The input packets of a 2×2 switch at the i^{th} stage are in contention for an output link if the i^{th} most significant bits of their addresses are same. The purpose of our quantum switch is to remove this blocking in the network so that the two contending packets can be routed in parallel on the same link using quantum superposition.

A simple design for such a 2×2 switch is depicted in figure 4. Two sets of qubits of size w each form the input packets of the switch. The address bits that determine the outputs are fed separately to the switch and are labeled a_1 and a_2 respectively. These bits function as the control inputs to the switch and are discarded after use. P_1 and P_2 (size $w-1$ each) are the input packets without a_1 and a_2 respectively. Another qubit s initialized to state $|0\rangle$ is used as a scratch qubit to generate the control input s_{out} to the switch gate.

The quantum circuit for this switch is given in figure 4. If $a_1 = a_2$, one of the Hadamard gates changes the state of s to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the switch gate creates an equal superposition of P_1 and P_2. If a_1 is $|1\rangle$ and a_2 is $|0\rangle$ then the NOT gate sets s to state $|1\rangle$, i.e., s_{out} is $|1\rangle$ and the switch gate is set in *cross* state. When a_1 is $|0\rangle$ and a_2 is $|1\rangle$, s remains in state $|0\rangle$ and the switch gate is set in *through* state. Thus, representing the state of the system by $|a_1, a_2, P_1, P_2\rangle$, the function of the switch is described as

$$\begin{aligned}
|0,0,P_1,P_2\rangle &\longrightarrow \frac{1}{\sqrt{2}}\left(|0,0,P_1,P_2\rangle + |0,0,P_2,P_1\rangle\right) \\
|0,1,P_1,P_2\rangle &\longrightarrow |0,1,P_1,P_2\rangle \\
|1,0,P_1,P_2\rangle &\longrightarrow |1,0,P_2,P_1\rangle \\
|1,1,P_1,P_2\rangle &\longrightarrow \frac{1}{\sqrt{2}}\left(|1,1,P_1,P_2\rangle + |1,1,P_2,P_1\rangle\right)
\end{aligned} \tag{6}$$

Even though this design creates a superposition of the contending packets at the desired output, a complementary superposition is created on the other output also which is undesirable. Even if we go ahead and make a Banyan network using this switch, the outputs of the network might receive packets that are not addressed to them. Also, it will not be possible to verify whether the received packet was intended for that output or not because the output address bits are removed by the nodes (the 2×2 switches) of the network. We can overcome the problem of verifying the received packets by keeping a copy of the the output address in the data portion of the packets. Still, the problem of incorrect routing remains. Furthermore, the incorrectly routed packets will interfere with the routing of correctly routed packets, since their address bits will be used by the 2×2-switches in the later stages of the network, further increasing the erroneous routing.

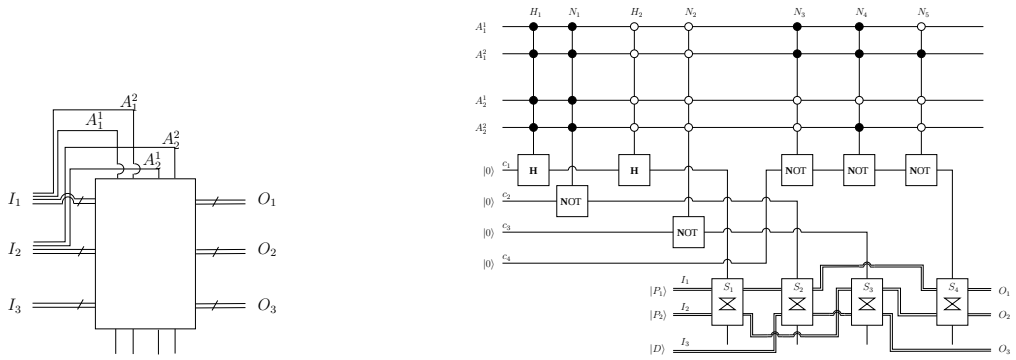
C. Quantum Switch with Dummy Input

To overcome the problems of the 2×2 quantum switch described in section III-B, we redesign the switch by introducing a dummy input-output pair. The undesirable superposition mentioned in section III-B is dumped on the dummy output. This switch is shown in figure 5(a). The inputs of the switch and the dummy input are labeled I_1, I_2, I_3 respectively. The two outputs and the dummy output are labeled O_1, O_2 and O_3 respectively. I_3 is *always* fed with dummy packets which are distinguishable from data packets. Note that the packets at I_1 and I_2 may also be dummy. To keep the dummy packets distinguishable from the data packets, the following scheme is used

a) The address bits of the data-packets are replicated, i.e., '0' and '1' bits of the address part are represented by '00' and '11' respectively.
b) The address part of the dummy packets is formed by repeating '01' $(\frac{p}{2} - 1)$-times, where p is the number of bits in the address part of packets at inputs I_1 and I_2, i.e., it is of the form '$0101\cdots01$'. Note that p is even.
c) The first bit of the data part is also used for distinguishing between data and dummy packets. This bit is set to '0' in the dummy packets and to '1' in data packets. It is used at the output of an interconnection network made using this switch. Since the address bits of the packets are removed by the network, this is the only bit available at the output of the network to determine whether the received packet is a data packet.

The quantum circuit for this switch is shown in figure 5(b). Two most significant bits from the address parts of packets at inputs I_1 and I_2, labeled A_1^1, A_1^2 and A_2^1, A_2^2 respectively, are used as control qubits. Note that the address part of the dummy packet on input I_3 is not used. Data packets at inputs I_1 and I_2, without the two most significant address bits, are denoted by P_1 and P_2 respectively. All the dummy packets are denoted by D. Four scratch qubits, c_1, c_2, c_3 and c_4 initialized in state $|0\rangle$ are used to generate the control inputs to the switch gates. The functionality of the circuit can be exhaustively described by following cases depending on the values of address qubits (where $A_1 = A_1^1A_1^2,\ A_2 = A_2^1A_2^2$)

a) When $A_1, A_2 = (00, 11)$ or $(00, 01)$ or $(01, 11)$ or $(01, 01)$, all the switch gates act in *through* state. Pack-



(a) 2×2 quantum switch with dummy input

(b) Quantum circuit for the switch

Fig. 5. 2×2 Quantum switch with dummy input

ets P_1, P_2 and D are sent to outputs O_1, O_2 and O_3 respectively. Representing the state of the switch as $|A_1, A_2, P_1, P_2, D\rangle$, the function of the circuit is

$$\begin{aligned} |00\,11\,P_1\,P_2\,D\rangle &\longrightarrow |00\,11\,P_1\,P_2\,D\rangle \\ |00\,01\,P_1\,D_2\,D\rangle &\longrightarrow |00\,01\,P_1\,D_2\,D\rangle \\ |01\,11\,D_1\,P_2\,D\rangle &\longrightarrow |01\,11\,D_1\,P_2\,D\rangle \\ |01\,01\,D_1\,D_2\,D\rangle &\longrightarrow |01\,01\,D_1\,D_2\,D\rangle \end{aligned}$$

b) When $A_1, A_2 = (11, 00)$ or $(11, 01)$ or $(01, 00)$, exactly one of the gates N_3, N_4 and N_5 changes the state of c_4 to $|1\rangle$, setting S_4 in *cross* state. Since no other controlled-gate becomes active, c_1, c_2 and c_3 remain in state $|0\rangle$, thus S_1, S_2 and S_3 act in *through* state. Function of the circuit can be written as (D_i is a dummy packet on I_i, $i = 1, 2$)

$$\begin{aligned} |11\,00\,P_1\,P_2\,D\rangle &\longrightarrow |11\,00\,P_2\,P_1\,D\rangle \\ |11\,01\,P_1\,D_2\,D\rangle &\longrightarrow |11\,01\,D_2\,P_1\,D\rangle \\ |01\,00\,D_1\,P_2\,D\rangle &\longrightarrow |01\,00\,P_2\,D_1\,D\rangle \end{aligned}$$

c) When $A_1, A_2 = (00, 00)$, gate H_2 sets c_1 in state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Thus, gate S_1 creates a superposition of packets P_1 and P_2 at its outputs. Gate N_2 sets c_3 in state $|1\rangle$, setting gate S_3 in *cross* state. So, the superposition at the second output of gate S_1 is interchanged with the dummy input packet. Function of the circuit is

$$|00\,00\,P_1\,P_2\,D\rangle \longrightarrow \frac{1}{\sqrt{2}}(|00\,00\,P_1\,D\,P_2\rangle + |00\,00\,P_2\,D\,P_1\rangle)$$

d) When $A_1, A_2 = (11, 11)$, gate H_1 sets c_1 in state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ creating an equal superposition of packets P_1 and P_2 at the outputs of S_1. Gate N_1 sets c_2 to $|1\rangle$, making S_2 act in *cross* state to interchange the unwanted superposition at the first output of S_1 with the dummy packet. Function of the circuit is

$$|11\,11\,P_1\,P_2\,D\rangle \longrightarrow \frac{1}{\sqrt{2}}(|11\,11\,D\,P_1\,P_2\rangle + |11\,11\,D\,P_2\,P_1\rangle)$$

Thus, this switch behaves as a classical 2×2 switch when there is no contention between the input packets for an output. When contention occurs, a superposition of the input packets is sent to their intended output and a dummy packet is sent on the other output. Output O_3 is always discarded. Consequently, the problem of routing superposed packets to both the outputs is eliminated with the use of dummy inputs.

IV. Quantum Banyan Network

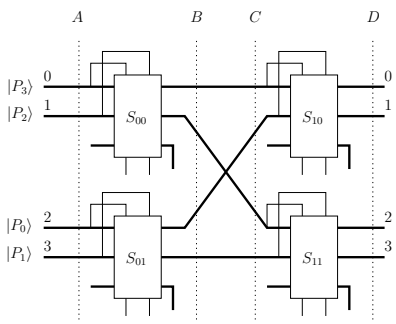
We first describe a 4×4 Banyan network made using the quantum switches designed in section III-C and then derive the properties of a general $n \times n$ quantum Banyan network.

A. A 4×4 Quantum Banyan Network

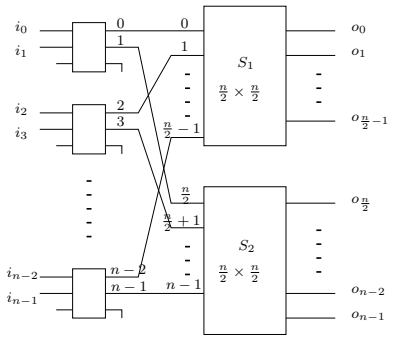
This network is depicted in figure 6(a). Here S_{ij} is the j^{th} 2×2 quantum switch (with dummy input) at stage i. Suppose the size of the input packets (including the address part) is w bits, then the sizes of the dummy packets used in the first and second stages of the network are $w-2$ and $w-4$ bits respectively in accordance with the scheme described in section III-C. The taps shown in the figure represent the fact that the address qubits are taken from the input packets. We give an example to explain the functioning of this switch. Suppose inputs 0, 1, 2 and 3 have packets for outputs 3, 2, 0 and 1 respectively. We represent the data part of these packets as P_3, P_2, P_0 and P_1 respectively. The packets are represented by the tuple (A, P), where A represents the address part and P is the data part. Also, we represent address bits '00', '11' and '01' by '0', '1' and 'd' respectively. Thus the packets at the four inputs are $(11, P_3)$, $(10, P_2)$, $(00, P_0)$ and $(01, P_1)$ respectively. We write the state of the quantum wires at locations marked in figure 6(a) by vertical dotted lines. The order of the components in ket-notation $|W_1, W_2, \cdots, W_n\rangle$, corresponds to the order in which we encounter the wires as we traverse the circuit from top to bottom. We ignore the third (dummy) input-output pair of all the 2×2 switches, when we write the state. Also, we do not write the state of the discarded address qubits.

The input state is $|(11, P_3), (10, P_2), (00, P_0), (01, P_1)\rangle$. The contentions for outputs at both the switches in the first stage cause them to create following state at location *B* in the figure

$$\frac{1}{2}\Big[\big|(d,D)(1,P_3)(0,P_0)(d,D)\big\rangle + \big|(d,D)(0,P_2)(0,P_0)(d,D)\big\rangle + \big|(d,D)(1,P_3)(1,P_1)(d,D)\big\rangle + \big|(d,D)(0,P_2)(1,P_1)(d,D)\big\rangle\Big]$$



(a) 4×4-Quantum Banyan Network



(b) Recursive construction of $n \times n$-Banyan Network ($n = 2^k, k \geq 1$)

Fig. 6. Quantum Banyan Network

After the shuffle, the state at *C* is given by interchanging the middle two packets in each of the four tuples (*kets*) above. There is no contention for any of the tuples (*kets*) at any of the switches in the second stages because one packet in each input (for every *ket*) is a dummy. Thus the output state of the switch is

$$\frac{1}{2}\big[|P_0\, D\, D\, P_3\rangle + |P_0\, D\, P_2\, D\rangle + |D\, P_1\, D\, P_3\rangle + |D\, P_1\, P_2\, D\rangle\big]$$

Note that all the address bits are consumed by the network. If we do a measurement at the outputs of the switch, we will observe one of the tuples (*kets*) in the above expression with probability $1/4$ each. The i^{th} component of a tuple is the packet at the i^{th} output. Thus, the probability of observing packet P_i, $i = 0, 1, 2, 3$, at output i is $1/2$. Also, we will observe data packets at two outputs, the remaining two outputs will have dummy packets.

B. Generalized Quantum Banyan Network

The generalized n-input Quantum Banyan network ($n = 2^k, k \geq 1$) is shown in figure 6(b). This network has $\log_2 n$ stages, each having $\frac{n}{2}$ 2×2 quantum switches. Output i of the first stage is connected to the input $\tilde{i}$ of the second stage, where

$$\tilde{i} = \begin{cases} \left(i + \left\lfloor \frac{i}{n/2} \right\rfloor\right) \bmod \frac{n}{2} & ;\ i \text{ even} \\ \frac{n}{2} + \left(i - 1 + \left\lfloor \frac{i}{n/2} \right\rfloor\right) \bmod \frac{n}{2} & ;\ i \text{ odd} \end{cases} \tag{7}$$

$i, \tilde{i} \in \{0, \cdots, n-1\}$, i.e., all the even numbered outputs are connected to the $\frac{n}{2} \times \frac{n}{2}$ switch S_1 and the odd numbered outputs are connected to the $\frac{n}{2} \times \frac{n}{2}$ switch S_2. S_1 and S_2 are constructed recursively in similar fashion.

We now analyze the routing properties of this network for permutation assignments. Unlike in a classical Banyan network, packets in the quantum Banyan network are superposed together whenever they contend for the same output at a switch. Therefore, the outputs will hold all the routable (non-conflicting) patterns of packets between them. In what follows we will establish that the output quantum state of the network is composed of all the *kets* that correspond to these patterns.

Let S_{ij} be the j^{th} switch in the i^{th} stage where $i = 1, \cdots, \log_2 n$ and $j = 0, \cdots, \frac{n}{2} - 1$. Assume input i of the first stage has a packet destined to output o_i of the last stage of the network, where $o_0, \cdots, o_{n-1}$ is a permutation of $0, \cdots, n-1$. Representing packets by their output addresses, the input quantum state of the network is $|o_0, \cdots, o_n\rangle$. Suppose that in the first stage, there is a contention for outputs at m switches $S_{(1,v_i)}$, $i = 0, \cdots, m-1$, $m \leq n/2$. We denote the outputs for which there is contention as u_i, $i = 0, \cdots, m-1$ and their complimentary outputs (output of $S_{(1,v_i)}$ to which no packet is addressed) by $\tilde{u_i}$ respectively. The indices of the even numbered inputs and outputs of the remaining $\frac{n}{2} - m$ switches are denoted by w_j, $j = 0, \cdots, \frac{n}{2} - m - 1$. Then the output state of the first stage is

$$\left(\frac{1}{\sqrt{2}}\right)^m \sum_{\substack{a_{u_i} \in \{o_{u_i}, o_{\tilde{u_i}}\} \\ i\ =\ 0,1,\cdots,m-1}} |a_0, a_1, \cdots, a_{(n-1)}\rangle \tag{8}$$

where $a_{\tilde{u_i}} = d$ (d denotes a dummy packet) and

$$(a_{w_j}, a_{w_j+1}) = \begin{cases} (o_{w_j}, o_{w_j+1}) & ;\ S_{1,\frac{w_j}{2}} \text{ is in } \textit{through} \text{ state} \\ (o_{w_j+1}, o_{w_j}) & ;\ S_{1,\frac{w_j}{2}} \text{ is in } \textit{cross} \text{ state} \end{cases}$$

Note that there are 2^m terms (*kets*) in the above summation. Each *ket* corresponds to a sequence of packets, including dummy packets, at the output of a stage (in this case, the first stage) of the network. We will use the terms 'packet sequence' and '*ket*' interchangeably in our discussion. The packets in the *kets* in (8) are shuffled according to the interconnection pattern between first and second stages. These shuffled *kets* form the input quantum state to stage two. They are processed at stage two in a fashion similar to that in stage one. The *kets* having contending packets are broken into more *kets* and the packets in all the *kets* (including those in which no output contentions occur) are routed according to their addresses. This process is repeated for all the stages. Thus, the quantum state at the output of the final stage is of the form $\sum_{i=1}^{M} b_i |p_0^i, p_1^i, \cdots, p_{(n-1)}^i\rangle$, where p's are the packets and b_i, M are defined as follows. Suppose that the i^{th} *ket*, $|p_0^i, p_1^i, \cdots, p_{(n-1)}^i\rangle$, which we denote as $|P^i_{(\log_2 n+1)}\rangle$, ($\log_2 n$ indicating the last stage), evolves via the following *ket* sequence, originating from the initial state $|P_1^i\rangle = |o_0, o_1, \cdots, o_n\rangle$ (m_j : number of contentions at j^{th}

stage, arrow superscripts are the stage numbers)

$$\underbrace{|P_1^i\rangle}_{m_1} \xrightarrow{1} \underbrace{|P_2^i\rangle}_{m_2} \xrightarrow{2} \cdots \xrightarrow{\log_2 n-1} \underbrace{|P_{\log_2 n}^i\rangle}_{m_{(\log_2 n)}} \xrightarrow{\log_2 n} |P_{(\log_2 n+1)}^i\rangle$$

then $b_i = \left(1/\sqrt{2}\right)^{\sum_{j=1}^{\log_2 n} m_j}$ and M is equal to the number of all *ket* sequences of the form shown above, which is also equal to the total number of kets at the output of the network. Note that all these sequences originate from state $|o_0, \cdots, o_n\rangle$ and together form a tree. The i^{th} level of this tree corresponds to the i^{th} stage of the network, and a node at i^{th} level corresponds to a *ket* in the expression for the output state of the i^{th} stage. Number of branches from this node is equal to 2^m, where m is the number of contentions in this *ket*.

Since each contention introduces one dummy packet in the ket sequence above, $d_i = \sum_{j=1}^{\log_2 n} m_j$ packets out of the n packets in $|p_0^i, \cdots, p_{(n-1)}^i\rangle$ are dummy packets. The remaining $n - d_i$ packets are correctly routed to their destinations. This gives the coefficient of $|p_0^i, \cdots, p_{(n-1)}^i\rangle$ in the expression of final state as $1/2^{d_i/2}$.

Remark: The probability of observing packets p_0^i, $\cdots$, $p_{(n-1)}^i$ at the output of the network is $1/2^{d_i}$. So, the *kets* with less number of dummy packets have a higher probability of being observed and this is desirable. Obviously a *ket* can not have more than $n-2$ dummy packets. Thus in worst case, it is possible to observe only two data packets and $n-2$ dummy packets with probability $1/2^{n-2}$.

The foregoing discussion characterized the form of the output quantum state of the network. We now establish that the *kets* in the output quantum state are maximal with respect to the resolution of contentions.

Definition 1: We call $\pi^{(\sigma)} = |a_0, \cdots a_{n-1}\rangle$ a sub-permutation of permutation $\sigma = |b_0, \cdots, b_{n-1}\rangle$ iff $a_i \in \{b_i, d\}$ $\forall i \in \{0, \cdots, n-1\}$. We express this as $\pi^{(\sigma)} \subseteq \sigma$.

Definition 2: Let $\Pi_\sigma = \{\pi : \pi \subseteq \sigma\}$, where $\sigma = |o_0, o_1, \cdots, o_{n-1}\rangle$ is an input permutation. Then, $\gamma_i^{(\sigma)}$ is an output sub-permutation of σ at stage i iff $\exists\ \pi \in \Pi_\sigma$ which can be routed through the Banyan network without any contention from stage 1 through stage i to give $\gamma_i^{(\sigma)}$.

Definition 3: An input sub-permutation $\pi \subseteq \sigma$ is maximal iff it can be routed through the Banyan network without any contentions, and for any other input sub-permutation $\psi \subseteq \sigma$ which can be routed through the network without any contention, π is not a sub-permutation of ψ.

Definition 4: An output sub-permutation $\gamma_i^{(\sigma)}$ (at stage i) of the input permutation $\sigma = |o_0, \cdots, o_{n-1}\rangle$ is maximal iff $\exists$ a maximal input sub-permutation π such that routing π through the network till stage i gives $\gamma_i^{(\sigma)}$.

Theorem 1: The output quantum state of the Banyan network is a superposition of all the maximal output sub-permutations (at stage $\log_2 n$) of the input permutation $|o_0, \cdots, o_{n-1}\rangle$.

The proof of this theorem will be deferred to a fuller version of this paper. This theorem also implies that the output quantum state contains all the possible packet combinations obtained when in a corresponding classical network contentions at the 2×2 internal switches are resolved by dropping packets based on a fair coin toss. It can also be shown that the probability of observing each such packet combination is the same as it would be for the classical network with the aforementioned random packet drop scheme.

V. Concluding Remarks and Future Work

In this paper we outlined the design of a quantum self-routing Banyan switch that creates a superposition of all the maximal output sub-permutations of an input permutation assignment. This was accomplished by introducing dummy packets wherever contentions occur within the network. Thus, all packets are routed through the network without undergoing any blocking. A simple measurement (measuring each qubit in computational basis) destroys this superposition and gives only one such output sub-permutation, which is equivalent to classical routing through a Banyan network with random packet drops in case of contentions. More sophisticated measurements can be done that use this superposition in a better way and obtain more information about the packets. For example a measurement using a superposed basis can give information about correlations between the packets. Another potential advantage of the quantum Banyan network over the conventional Banyan network is that there is no need to take local decisions to resolve contentions.

The results established in this paper show that quantum packet switching can be an effective approach to mitigate contentions in packet switches. Much work remains to be done to make quantum switching networks a reality. Another direction would be to extend our results to more powerful switching structures such as the Beneš and Clos networks.

Acknowledgments

We would like to thank Professor Tekin Dereli for sharing his knowledge and ideas on quantum computing with us and Professor Erdal Arikan for suggesting to the third author to explore the use of quantum gates in the design of interconnection networks.

References

[1] P. W. Shor, "Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, May 1996, pp. 212–219.

[3] F. K. Hwang, *The Mathematical Theory of Nonblocking Switching Networks*, ser. Series on applied mathematics, F. K. Hwang, Ed. Singapore: World Scientific Publishing Co. Pte. Ltd, 1998, vol. 11.

[4] C. Clos, "A study of non-blocking switching networks," *Bell Syst. Tech. J.*, vol. 32, no. 2, pp. 406–424, Mar. 1953.

[5] D. G. Cantor, "On nonblocking switching networks," *Networks*, vol. 1, pp. 367–377, 1971.

[6] M. Koppelman and A. Y. Oruç, "A self-routing permutation network," *Journal of Parallel and Distributed Computing*, pp. 140–151, Oct. 1990.

[7] C. Y. Lee and A. Y. Oruç, "Design of efficient and easily routable generalized connectors," *IEEE Trans. Commun.*, pp. 646–650, Feb. 1995.

[8] V. E. Beneš, *Mathematical Theory of Connecting Networks and Telephone Traffic*. Academic Press, 1965.

[9] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, R. Gallager, Ed. Kluwer Academic Publishers, 1990.

[10] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, vol. 25, pp. 1145–1155, 1976.

[11] J. H. Patel, "Performance of processor memory interconnections for multiprocessors," *IEEE Trans. Comput.*, vol. 30, no. 10, pp. 771–780, Oct. 1981.

[12] J. Preskill. (1998) Physics 229: Advanced mathematical methods of physics – quantum computation and information. [Online]. Available: http://www.theory.caltech.edu/people/preskill/ph229

[13] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, Sept. 2000.

[14] D. Deutsch, "Quantum computational networks," *Proc. R. Soc. Lond. Series A, Mathematical and Physical Sciences*, vol. 425, no. 1868, pp. 73–90, Sept. 1989.