SEP 1995

MODULAR EXPANSION AND RECONFIGURATION OF SHUFFLENETS IN MULTI-STAR IMPLEMENTATIONS

By

PHILIP PAK-TUNG TO

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF PHILOSOPHY DIVISION OF INFORMATION ENGINEERING THE CHINESE UNIVERSITY OF HONG HONG JUNE 1994 42

14 55

Mesis TK 5105.5 T6 1984

Acknowledgement

I would like to express my sincere gratitude towards my supervisor Dr. T.S. P. Yum for his advice, support and encouragement, which make this 2-year research period a fruitful and rewarding experience.

I would also like to thank Y.W. Leung for his useful discussions on the multistar implementation of ShuffleNet.

Abstract

ShuffleNet is one of the many architectures proposed for multihop optical networks. ShuffleNet is popular because of its relatively low mean-internodal distance and simplied routing schemes. ShuffleNet, however, is not without drawbacks. In this thesis, we investigate two problematic issues of ShuffleNet, namely modular expansion of the network and reconfiguration of the network in multistar implementation.

It is well-known that regular network architectures, such as ShuffleNet, cannot have arbitrary network size and hence modular expansion is generally difficult or impossible. In the first part of this thesis, we devise a way to expand a ShuffleNet modularly. The expansion procedure is based on the multi-star implementation of ShuffleNet. We show how a (p, k) ShuffleNet can be expanded to a (p, k+1) ShuffleNet in modular phases. In each phase, the number of nodes in the network is increased by only a small fraction and hardware and software reconfigurations are kept to a minimum.

In the second part of this thesis, we investigate the reconfigurability of ShuffleNet in multi-star implementation. Reconfigurability is a measure of the degree of freedom by which a network can be reconfigured to adapt to the changing traffic patterns. If reconfigurability is 100%, the logical identities of any pair of nodes can be exchanged by retuning their transceivers. In a multi-star network, however, we cannot always attain 100% reconfigurability. We show that the reconfigurability of a multi-star ShuffleNet is related to the number of star-couplers used and the number of wavelength channels available in each star-coupler. We also show that reconfigurability can be maximized a proper assignment of wavelength channels. Several channels assignment algorithms are presented and discussed.

Contents

1	Intr	oducti	on	1
2	Mo	dular I	Expansion of ShuffleNet	8
	2.1	Multi-	Star Implementation of ShuffleNet	10
	2.2	Modul	ar Expansion of ShuffleNet	21
		2.2.1	Expansion Phase 1	21
		2.2.2	Subsequent Expansion Phases	24
	2.3	Discus	ssions	26
3	Rec	configu	rability of ShuffleNet in Multi-Star Implementation	33
	3.1	Recon	figurability of ShuffleNet	34
		3.1.1	Definitions	34
		3.1.2	Rearrangable Conditions	35
		3.1.3	Formal Representation	38
	3.2	Maxir	nizing Network Reconfigurability	40
		3.2.1	Rules to maximize T_{sc} and R_{sc}	41
		3.2.2	Rules to Maximize Z	42
	3.3	Chan	nels Assignment Algorithms	43
		3.3.1	Channels Assignment Algorithm for $w = p \ldots \ldots$	45

		3.3.2	Channels Assignment Algorithm for $w = p$	$\cdot k$	•	•	•		•		46
		3.3.3	Channels Assignment Algorithm for $w = M$	p^k	•	•	•	•	•		49
	3.4	Discus	ssions \ldots \ldots \ldots \ldots \ldots \ldots	• •	•	•	÷	•	•	•	51
4	Cor	nclusio	ns								55

List of Tables

1.1	The number of nodes in a ShuffleNet for different values of p and k .	7
2.1	Mapping nodes in the $(2,3)$ ShuffleNet to the $(2,4,2)$ ShuffleNet.	24
2.2	Percentage growth in each phase of expanding a $(2,3)$ ShuffleNet.	28
3.1	Reconfigurability of the $(3,2)$ ShuffleNet in multi-star implemen-	
	tation for various values of w and C .	54

List of Figures

1,1	A 3-node network.	3
1.2	A (2,2) ShuffleNet	6
2.1	The $(2,2)$ ShuffleNet is a subgraph of the $(3,2)$ ShuffleNet	9
2.2	The broadcast-and-select structure	11
2.3	A multi-star ShuffleNet with a central hub	13
2.4	Nodes in a (p, k) ShuffleNet	14
2.5	Channels assignment for the (2,2) ShuffleNet	15
2.6	Multi-star implementation of the $(2,2)$ ShuffleNet	16
2.7	Connectivity and channels assignment of a $(2,3)$ ShuffleNet using	
	8 wavelength channels per fiber	22
2.8	A $(2,4,2)$ ShuffleNet using 8 wavelength channels per fiber	30
2.9	Multi-star implementation of the (2,3) ShuffleNet	31
2.10	Multi-star implementation of the (2,4,2) ShuffleNet	31
2.11	A $(2,4,3)$ ShuffleNet using 8 wavelength channels per fiber	32
3.1	Channels assignment for a (2,2) ShuffleNet	36
3.2	Rule 1	41
3.3	An example (2,2) ShuffleNet	44

3.4	Channels assignment for a $(2,2)$ ShuffleNet using 2 channels per	
	coupler	47
3.5	Channels assignment for a $(3,2)$ ShuffleNet using 6 channels per	
	coupler	49
3.6	The "Round Robin" assignment method	52

Chapter 1

Introduction

In the past decade, optical fibers have been extensively deployed in telecommunication networks because of the huge information transmission capacity they can potentially provide. The low-loss region of a single-mode optical fiber (ranging roughly from 1.2 to 1.6 μ m) provides around 30 terahertz of optical bandwidth. Capitalizing such transmission capacity, however, proves to be no trivial task because the speed of electronics cannot possibly catch up with the maximum bit-rate provided by an optical fiber [1].

Over the years researchers have been devising ways to tap the tremendous amount of bandwidth provide by the optical medium [2, 3, 4, 5]. A promising way is the Wavelength Division Multiplexing (WDM) technique. In a WDM system, the low-loss region of the optical fiber is divided into a number of independent wavelength channels. Each channel can be accessed at peak electronic processing speeds of, say a few Gbps. By performing multiple simultaneous transmission on different channels, a huge aggregate capacity can be achieved.

Early uses of the WDM technology are confined to upgrading the capacity of

point-to-point connections. Typically two, three or four additional wavelength channels are installed to boost the maximum bit-rate of existing optical links. These channels are usually separated by several tens to hundreds of nanometers in wavelength. With the advent of narrow linewidth lasers and narrowband optical filters, wavelength channels can be spaced more closely together and hence the number of channels available is increased. When more channels are available, it is becoming more feasible to use WDM in networking applications beyond the sole increase of link capacity [6]. One of them is multi-access optical networks that interconnect a large number of nodes and provide a huge network capacity such that video conferencing, High Definition Television (HDTV) broadcasting, and other state-of-the-art multimedia applications can be supported.

An optical network has its physical and logical topologies. The physical topology (e.g. star, bus or ring) describes how the network is implemented and the logical topology specifies how the nodes are connected and how wavelength channels are assigned. As an example, Figure 1.1(a) shows the logical topology of a 3-node network with node A transmitting to node B via channel λ_0 , node B to node C via channel λ_1 , and so on. The physical topology of this network can be a single-star network as shown in Figure 1.1(b) instead of a ring. This shows that a logical topology can be realized by different physical implementations. This relative independence of logical and physical topologies gives network designers the freedom to choose the most suitable logical topology for a particular application.

There are basically two types of logical topologies - single-hop and multihop. In a single-hop network, each node can communicate with any other nodes in the network using an all optical path without going through any intermediate





Figure 1.1: A 3-node network.

nodes. Usually every node in the network is equipped with tunable transmitters or tunable receivers (or both) which are capable of accessing the entire spectrum of wavelength channels in use. Dynamic coordinations among nodes are necessary to avoid contention and receiver collision [7]. If the network is packet-switched, the transceivers will need to be retuned in a per-packet basis and so they must be very agile. Unfortunately, rapidly tunable transceivers are not yet commercially feasible and their tuning range is usually limited, implying that the number of usable channels is small. Consequently, single-hop optical networks are inherently limited in size.

In a multihop network, each node is equipped with a small number of fixed (or slowly tunable) wavelength transmitters and receivers and is connected to a small number of nodes. Since a direct path exists only for a small set of node pairs, a message typically has to route through one or more intermediate nodes before reaching the destination. In each intermediate node, electro-optical conversion is required to store and forward the packets. This multihop approach avoids the need to retune the transmitters and/or receivers each time a packet is transmitted. In addition, no pre-transmission coordination is necessary. The price to pay is a reduced network throughput because each packet now takes a longer path to go from source to destination. Packets may also be dropped if an intermediate node becomes too congested. Nevertheless, the multihop approach is a viable solution when rapid tunable transceivers are not yet feasible.

Multihop networks can have an irregular topology [8, 9, 10, 11]. Irregular multihop networks are usually constructed by optimizing for the network traffic pattern. Routing in irregular networks, however, are more complicated because they lack any structural connectivity pattern [11]. Regular structured multihop networks, on the other hand, have fixed node-connectivity pattern which permit simplified routing schemes to be used. However, due to their regular structure, the number of network nodes cannot be arbitrary. This also implies that modular growth of network size is difficult or even impossible. Some well-known regular logical topologies include the perfect shuffle, the de Bruijn graph, the toroid and the hypercube. The corresponding networks based on these topologies are the ShuffleNet, the de Bruijn graph network, the Manhattan Street Network and the Hypercube network [11].

In this thesis, we concentrate on the regular multihop network called ShuffleNet. ShuffleNet was first proposed in [3] and later extended in [4]. It is a multi-column network in which nodes in one column are connected to nodes in the next column in a perfect shuffle connection pattern. A ShuffleNet is characterized by two non-zero integer parameters p and k. In a (p, k) ShuffleNet, the total number of nodes N is equal to kp^k . They are numbered from 0 to $kp^k - 1$ and are arranged in k columns of p^k nodes each, with the kth column wrapped around to the first in a cylindrical fashion. The number of transmitters and the number of receivers per node are both equal to p. The total number of channels is kp^{k+1} . The location of a node in ShuffleNet can be represented by its (row, column) coordinates. For node n, it is readily seen that

$$row = rem(n/p^k) \quad 0 \le row \le p^k - 1 \tag{1.1}$$

$$column = int(n/p^k) \quad 0 \le column \le k-1$$
 (1.2)

where rem(x/y) denotes the remainder of x/y and int(x/y) denotes the integer part of x/y.

A (2,2) ShuffleNet is shown in Figure 1.2. Each directed link from one node to another represents a dedicated wavelength channel.

In this thesis, we investigate two problematic issues associated with ShuffleNet. The first one is the modular expansion of ShuffleNet. As mentioned above, the number of nodes N in a ShuffleNet cannot be arbitrary [11]. Table 1.1 shows the number of nodes N for various values of p and k. In implementing a ShuffleNet, we usually have to put in "dummy" nodes so that the total number of nodes in the network falls into this discrete set of integers. As a consequence, incremental growth of the network cannot be done easily. In Chapter 2 of this thesis, we devise a way to expand a ShuffleNet modularly. The expansion procedure is based on the multi-star realization of ShuffleNet, which is a feasible



Figure 1.2: A (2,2) ShuffleNet.

way to implement a ShuffleNet. We show how a (p, k) ShuffleNet can be expanded to a (p, k + 1) ShuffleNet in several discrete phases. In each phase, a "partial" ShuffleNet is constructed to enable fractional growth of the network size. Moreover, the hardware and software reconfigurations required are kept to a minimum. Specifically, the hardware part involves only rearranging certain fiber connections and the software part involves updating node addresses and routing tables. No transmitter or receiver has to be added to existing nodes and hence disturbance to the existing network is greatly reduced.

The second problem is less obvious and it concerns the reconfigurability of ShuffleNet in multi-star implementation. Reconfigurability is a measure of the degree of freedom by which a network can be reconfigured to suit the changing traffic pattern. In chapter 3, we prove that if a ShuffleNet is implemented out of multiple star-couplers, the network may not be able to attain the maximum reconfigurability of 100%. In fact, the reconfigurability of a multi-star ShuffleNet is directly related to the number of star-couplers used and the number

		p						
k	2	3	4	5	6	7		
2	8	18	32	50	72	98		
3	24	81	192	375	648	1,029		
4	64	324	1,024	2,500	5,184	9,604		
5	160	1,215	5,120	15,625	38,880	84,035		
6	384	4,374	24,576	93,750	279,936	705,894		
7	896	15,3909	114,688	546,875	1,959,552	5,764,801		

Table 1.1: The number of nodes in a ShuffleNet for different values of p and k.

of wavelength channels available in each star-coupler. We show that given a certain number of couplers is used, network reconfigurability can be maximized by an optimal assignment of wavelength channels. Several channels assignment algorithms for maximizing reconfigurability are presented and discussed.

Chapter 2

Modular Expansion of ShuffleNet

As discussed in the previous chapter, because of the regularity of the ShuffleNet structure, the number of nodes N in a ShuffleNet cannot be arbitrary. It is therefore difficult to expand the size of a ShuffleNet modularly.

One solution exists by observing that a (p, k) ShuffleNet is a subgraph of the (p + 1, k) ShuffleNet [13]. An example is shown in Figure 2.1 where a (2,2)ShuffleNet (the shaded nodes and the solid arrows) is shown to be imbedded in a (3,2) ShuffleNet. If a target (p + 1, k) ShuffleNet is to be built but not all the nodes are needed for the moment, we can first deploy those nodes corresponding to the imbedded (p, k) ShuffleNet, and switch to the target system when necessity calls for. Switching from a (p, k) ShuffleNet to a (p + 1, k) ShuffleNet in one step causes a huge jump in network size. Moreover, the size of the target system has

¹The materials presented in this chapter have been published in part in the Proceedings of IEEE GLOBECOM'93, Houston, USA, Nov. 1993; also to appear in IEEE Transactions on Networking, August 1994.

to be determined first. Further growth beyond the planned target system size would require a lot of hardware and software reconfigurations.



Figure 2.1: The (2,2) ShuffleNet is a subgraph of the (3,2) ShuffleNet.

In [14] Karol proposed a multi-connected ring implementation of ShuffleNet. By using a new representation of the ShuffleNet connectivity graph and a generalization of Gray code patterns, Karol showed that if a (p, k) ShuffleNet is to be built, we can start with k nodes connected in a ring and grow to the target system in several steps. In each step, k nodes connected in a ring, together with the necessary fiber connections, are added. Moreover, by using the fact that a (p, k) ShuffleNet is a subgraph of a (p + 1, k) ShuffleNet, a multi-connected ring (p, k) ShuffleNet can be expanded to a (p + 1, k) ShuffleNet in increments of k nodes at a time. This approach provides a way to grow a ShuffleNet gradually with p. Moreover, as shown in [15], when a (p, k) ShuffleNet is grown to a

Chapter 2 Modular Expansion of ShuffleNet

(p+1,k) ShuffleNet, the total network throughput and the per-node throughput are both increased. On the practical side, however, this expansion method requires a new transmitter and a new receiver to be added to all network nodes for each increase of p by one. New fibers also need to be laid for all these added transceivers. If the nodes are geographically dispersed, this expansion operation may be very involved and expensive.

In this chapter we consider expanding a ShuffleNet with k instead of with p. We show how a (p, k) ShuffleNet can be expanded to a (p, k + 1) ShuffleNet in several discrete phases. In each phase, a "partial" ShuffleNet is constructed to enable fractional growth of the network size. Moreover, the hardware and software reconfigurations required are kept to a minimum. Specifically, the hardware part involves only rearranging certain fiber connections and the software part involves updating node addresses and routing tables. No transmitter or receiver has to be added to existing nodes and hence disturbance to the existing network is greatly reduced.

This chapter is organized as follows. In Section 2.1, we describe the multi-star implementation of ShuffleNet and propose two channels assignment algorithms. In Section 2.2, we show how to expand a multi-star ShuffleNet with k in several phases. Section 2.3 discusses the implications of our expansion method.

2.1 Multi-Star Implementation of ShuffleNet

One simple way to implement a ShuffleNet is the broadcast-and-select structure using a single star-coupler, as shown in Figure 2.2. In such structure, each node is assigned one or more dedicated wavelength channels for transmission. The star-coupler combines all the wavelength channels and broadcasts them to all the nodes. Each node then receives from the appropriate channels according to its logical connectivity, as defined in the ShuffleNet topology. Since all the nodes are connected to the same star-coupler, they can potentially be connected to any other nodes in the network. Therefore this single-star approach gives the maximal nodal reconfiguration capability, which may not be attainable in the multi-star approach [22]. A single-star network, however, is limited in size by the available power budget and by the finite number of wavelength channels available [6]. The power splitting loss in the star-coupler becomes significant when the network size is large. Since each node must be assigned dedicated wavelength channels, the maximum number of nodes in a network is bounded by the number of wavelength channels available.



Figure 2.2: The broadcast-and-select structure.

One way to solve this problem is the shared-channel approach [16]. Group of p users in each column transmit on a common channel, with a separate group of p users in the next column receiving on each channel. The number of channels required to implement a (p, k) ShuffleNet is reduced from kp^{k+1} to kp^{k-1} . However, when multiple users are transmitting on a common channel, we must deal with the contention problem, which may lead to bandwidth wastage.

An alternate way is to observe that for a multihop network, each node needs only be connected to a subset of the nodes. Potential connectivity for each node to all the other nodes in the network is not absolutely necessary. By using this fact, we can implement a ShuffleNet as a multi-star network. The idea is that by using multiple small couplers, each interconnecting a subset of the nodes, the available wavelength channels can be spatially reused on each coupler, hence increasing the number of usable channels [17, 18, 19]. By adjusting the number of couplers to use, we can tradeoff between wavelength division multiplexing and space division multiplexing. When multiple couplers are used, the size and hence the power splitting loss of each coupler are reduced, resulting in a more relaxed power budget constraint. This, together with the fact that more channels are available, allows more network nodes to be attached. For a fixed required number of channels, we can space them farther apart. This can reduce the network cost as less expensive optical filters can be used.

We can picture that in a real network, nodes will tend to cluster in groups and different groups may be geographically far apart from one another. A multi-star ShuffleNet is therefore likely to be implemented by co-locating the star-couplers in a central hub, with fiber trunks leading to each of the clusters. This is shown in Figure 2.3. Each fiber trunk serves one cluster and may contain more optical fibers than necessary for future network expansion. There are two reasons to co-locate the star-couplers. First, a node is connected to different star-couplers



Figure 2.3: A multi-star ShuffleNet with a central hub.

for transmission and reception. If the star-couplers are co-located, the cost of laying optical fibers can be reduced. Secondly, as we shall see, during network expansion, we need to add new star-couplers and rearrange the fiber connections. If all the star-couplers are co-located in a central hub, such procedure can be done entirely within the hub. The effort required and disturbance to the existing network can be greatly reduced.

One important objective of the implementation is to minimize the number of fiber connections for each node. It can be shown that a node needs only be connected to one star-coupler for transmission and one star-coupler for reception if the number of wavelength channels per fiber is no smaller than p^2 . To see this, consider an arbitrary node A in a (p, k) ShuffleNet as shown in Figure 2.4. If node A is receiving from star-coupler J, the p nodes transmitting to node A must also be connected to star-coupler J. Since each of the p nodes requires p distinct wavelength channels on star-coupler J, the minimum number of wavelength channels required is p^2 .



Figure 2.4: Nodes in a (p, k) ShuffleNet.

Let there be w wavelength channels available in a fiber and let these channels be labeled as channel 0 to channel w-1. Each column of nodes can be partitioned into groups of p nodes such that nodes in the same group are all connected to the same set of nodes in the next column [16]. In general, for any column in a (p, k)ShuffleNet, group *i* consists of nodes with the following set of row coordinates $\{i, i + p^{k-1}, i + 2 \cdot p^{k-1}, \ldots, i + (p-1) \cdot p^{k-1}\}$, where $0 \le i \le p^{k-1} - 1$. As an example consider the (2,2) ShuffleNet as shown in Figure 2.5. In the first column, group 0 consists of nodes $\{0, 2\}$ and group 1 consists of nodes $\{1, 3\}$. We can see that nodes $\{0, 2\}$ are both connected to nodes $\{4, 5\}$, and nodes $\{1, 3\}$ that nodes belonging to the same group must transmit to the same star-coupler in order that nodes to which they are connected in the next column can receive from a single star-coupler. Since there are p nodes in each group and each node requires p wavelength channels, a total of p^2 wavelength channels are needed for each group. This implies that the number of usable wavelength channels wmust be a multiple of p^2 . Here, we assume $w = Mp^2$, where M is an integer and $1 \leq M \leq p^{k-2}$.



Figure 2.5: Channels assignment for the (2,2) ShuffleNet.

In the following, we describe the Transmitter Channels Assignment Algorithm and the Receiver Channels Assignment Algorithm. We also construct two formulas which express the connectivity of node n given p, k, w and N.

In the Transmitter Channels Assignment Algorithm, each node is assigned p wavelength channels on a fiber that connects to a particular star-coupler. We divide the nodes in a column into groups and assign nodes of the same group to connect to the same star-coupler. We order the outgoing links of each node in a way such that the first link is the one that connects to the node in the



Figure 2.6: Multi-star implementation of the (2,2) ShuffleNet.

next column with the smallest row coordinate, the second link is the one that connects to the node with the second smallest row coordinate, etc. Within a group, wavelength channels are assigned to links in natural order. The first links are assigned first, followed by the second links, third links, etc. This is usually referred to as the row major order assignment in matrix theory. As an example, Figure 2.5 shows the connectivity and channels assignment for a (2,2)ShuffleNet. The integer pair associated with each link represents (*star-coupler*, *channel*). In this example, w is equal to $2^2 = 4$. We can see that in the first column, nodes 0 and 2 of group 0 are assigned connections to star-coupler 0. Within group 0, channels 0 to 3 of star-coupler 0 are assigned to the four outgoing links in a row major fashion. Similarly, nodes 1 and 3 of group 1 are assigned to connect to star-coupler 1 and the four channels of star-coupler 1 are assigned to the four outgoing links accordingly. Note that if there are more channels available, say w is equal to 8, nodes in group 1 can also be assigned to star-coupler 0 for transmission with the four outgoing links being assigned to channels 4 to 7. This procedure is repeated for all columns of the ShuffleNet.

In the Receiver Channels Assignment Algorithm, the corresponding set of receiver wavelength channels for each node is deduced directly from the connectivity graph. For example, in Figure 2.5, node 4 receives from node 0 and node 2. Since we have already assigned channels 0 and 1 of star-coupler 0 to nodes 0 and 2 respectively, node 4 will receive from channel 0 and channel 1 of star-coupler 0. With the Transmitter Channels Assignment done, the Receiver Channels Assignment is only a simple labeling algorithm.

For the formal description of the two algorithms we define the following notations: For node n, let T_n be the set of transmitter wavelength channels, R_n be the set of receiver wavelength channels, I_n be the star-coupler node n transmits packets to and J_n be the star-coupler node n receives packets from. Using the following algorithms, the multi-star implementation of the (2,2) ShuffleNet is obtained as shown in Figure 2.6.

Transmitter Channels Assignment Algorithm

Inputs: p, k, w and N.

Outputs: T_n and I_n for $0 \le n \le N - 1$.

begin

channel := 0; coupler := 0; for column := 0 to $(N/p^k) - 1$ do for row := 0 to $p^{k-1} - 1$ do

```
for link := 0 to p - 1 do

for shift := 0 to p - 1 do

begin

n := column \cdot p^k + row + shift \cdot p^{k-1}

T_n := T_n \cup \{channel\};

I_n := coupler;

if channel < w - 1 then

channel := channel + 1

else

begin

channel := 0;

coupler := coupler + 1;

end;
```

end;

end.

Receiver Channels Assignment Algorithm

Inputs: p, k, w and N.

Outputs: R_n and J_n for $0 \le n \le N - 1$.

begin

```
channel := 0; coupler := 0;
for node := p^k to N + p^k - 1 do
begin
n := rem(node/N);
```

```
J_n := coupler;

for link := 0 to p-1 do

begin

R_n := R_n \cup \{channel\};

if channel < w - 1 then

channel := channel + 1

else

begin

channel := 0;

coupler := coupler + 1;

end;

end;
```

end;

end.

A closed form solution for I_n and J_n can be put together as follows:

$$I_n = int \left[(rem(n/p^{k-1}) + p^{k-1}int(n/p^k))/(w/p^2) \right]$$
(2.1)

$$J_n = int \left[(n - p^k + N\delta \left[int(n/p^k) \right]) / (w/p) \right]$$
(2.2)

where $\delta(x) = 1$ if x = 0 and $\delta(x) = 0$ if $x \neq 0$. To see how (2.1) and (2.2) come about, we break them down and analyze them term by term. Basically (2.1) follows closely from our previous discussions on groups and the Transmitter Channels Assignment Algorithm. Since there are p^k nodes per column, and each group consists of p nodes, p^{k-1} represents the number of groups per column. The expression $rem(n/p^{k-1})$ gives the group to which node n belongs. If we number the groups in column 0 from 0 to $p^{k-1} - 1$, the groups in column 1 from p^{k-1} to $2p^{k-1}-1$ and so on, the group to which node *n* belongs becomes $rem(n/p^{k-1}) + p^{k-1}int(n/p^k)$ because $int(n/p^k)$ is the column coordinate of node *n*. Since a star-coupler can accommodate w/p^2 groups, the right hand side of (2.1) gives the star-coupler number to which a group connects to for transmission.

To interpret (2.2), refer to the (2,2) ShuffleNet in Figure 2.5. Starting from the second column, we see that both nodes 4 and 5 receive from star-coupler 0 while both nodes 6 and 7 receive from star-coupler 1. In general, due to the symmetry of ShuffleNet, if we start from the first node in the second column, i.e. from node p^k , the first set of w/p consecutive nodes receives from star-coupler 0, the second set receives from star-coupler 1, and so on. This is because w/prepresents the number of nodes receiving from a particular star-coupler. Special treatments are required for nodes in column 0 because the receiving side of the column 0 nodes is at the last column. The addresses for the nodes in column 0 must therefore be all increased by N, as indicated by the term $N\delta [int(n/p^k)]$.

Note that in the above algorithms and equations, the requirement on N is that it be divisible by p^k . Therefore, the algorithms and equations can also be used on partial ShuffleNets, or ShuffleNets having fewer columns than a corresponding full ShuffleNet, such as a (2,4) ShuffleNet with 3 columns instead of 4. A partial ShuffleNet is characterized by three parameters p, k and m. A (p, k, m) ShuffleNet is a (p, k) ShuffleNet having m columns, where $1 \le m \le k$. A full (p, k) ShuffleNet can be denoted as a (p, k, k) ShuffleNet. The total number of nodes N in a (p, k, m) ShuffleNet is equal to mp^k .

2.2 Modular Expansion of ShuffleNet

In this section we describe a procedure for expanding a (p, k) ShuffleNet to a (p, k + 1) ShuffleNet in several phases, each using a partial ShuffleNet. In each phase, the increase in network size is fractional. Since the first phase of expansion differs considerably from the subsequent phases, it will be described separately. Along with the descriptions, we will cite an example of expanding a (2,3) ShuffleNet to a (2,4) ShuffleNet and will show how this can be done in 3 phases. Figure 2.7 shows the logical connectivity and wavelength channels assignment for a (2,3) ShuffleNet using 8 wavelength channels per fiber.

2.2.1 Expansion Phase 1

There are three steps in the first phase of expansion. Denote the (p, k) ShuffleNet as Φ . Construct a partial $(p, k+1, m_0)$ ShuffleNet Φ_e . The total number of nodes in Φ_e is $m_0 p^{k+1}$. As the size of Φ_e must be large enough to accommodate all kp^k nodes in Φ , the smallest possible expansion must satisfy $m_0 p^{k+1} > kp^k$. In other words, $m_0 = int(k/p) + 1$. The number of new nodes added is therefore $m_0 p^{k+1} - kp^k = p^k(m_0 p - k)$ and the number of new star-couplers added is $p^{k+1}(m_0 p - k)/w$. The following steps are performed in the expansion.

1. Perform connectivity and channels assignment on Φ_e using the algorithms in Section II.

Remark: The connectivity and channels assignment for our example (2,4,2)ShuffleNet is shown in Figure 2.8.

2. Map each node in Φ to an equivalent node in Φ_e and update the node address accordingly.



Figure 2.7: Connectivity and channels assignment of a (2,3) ShuffleNet using 8 wavelength channels per fiber.

Remarks: Two nodes are said to be equivalent if their transmitter and receiver wavelengths are the same. They, however, can be connected to different star-couplers. For example, node 0 in the (2,3) ShuffleNet is equivalent to nodes 0, 4, 16 and 20 in the (2,4,2) ShuffleNet. Our goal is to map all the nodes in Φ to nodes in Φ_e so that these "old" nodes can continue to communicate with the others with a mere change of addresses and some connection rearrangements. In other words, the expansion should require no replacement or retuning of any transmitters and receivers in the "old" nodes. There may be more than one mapping available but the mapping we introduce here requires only connection rearrangements at the output side of the star-couplers. Specifically, a node α in Φ is mapped to a node β in Φ_e by the following formula

$$\beta = rem(a/p^{k-1}) + p^{k}int(a/p^{k-1}) + p^{k-1}rem(b/p) + p^{k+1}int(b/p)$$
(2.3)

where $a = rem(\alpha/p^k)$ and $b = int(\alpha/p^k)$.

To understand how we come up with such a mapping let us go back to our example. Figures 2.9 and 2.10 show the multi-star implementation of the (2,3) and (2,4,2) ShuffleNets respectively. Consider the input ports of each star-coupler. To reduce the number of reconnections, the nodes transmitting to a particular star-coupler in Figure 2.9 should be mapped onto the nodes transmitting to the same star-coupler in Figure 2.10. Thus nodes 0, 1, 4 and 5 in Figure 2.9 should be mapped onto nodes 0, 1, 8 and 9 in Figure 2.10 respectively. Note that nodes 0 and 1 do not even need to change addresses. If such a mapping is used, all the nodes can be "reused" with a mere change of addresses and the input ports of all star-couplers do not need any rearrangements. Unfortunately, this does not exempt us from rearranging the output ports. For instance, node 8 in Figure 2.9 is receiving from star-coupler 0. But when it is mapped onto node 4 in Figure 2.10, it has to receive from star-coupler 5. Therefore we must unplug the fiber connection of node 8 at star-coupler 0 and reconnect it to star-coupler 5. Table 2.1 shows the mapping used in our example. Equation (2.3) is simply a compact form of expressing this mapping.

Original address	New address	Original address	New address	Original address	New address
0	0	8	4	16	16
1	1	9	5	17	17
2	2	10	6	18	18
3	3	11	7	19	19
4	8	12	12	20	24
5	9	13	13	21	25
6	10	14	14	22	26
7	11	15	15	23	27

Table 2.1: Mapping nodes in the (2,3) ShuffleNet to the (2,4,2) ShuffleNet.

3. Disconnect some output ports of the star-couplers. Add new nodes and new star-couplers and connect all loose-end fibers according to the connectivity graph of Φ_e .

Remarks: In our example, eight new nodes numbered 24 to 31 are added. All the input ports of the six star-couplers do not need any rearrangement. The output ports of the six star-couplers except star-coupler 4 are all unplugged and new star-couplers 6 and 7 are added together with their attaching fibers. Finally, all loose-end fibers are connected according to the connectivity requirements of the (2,4,2) ShuffleNet. By setting p, k,w and N to 2, 4, 8 and 32 respectively in (2.2), which star-coupler a node is to receive from can readily be computed and connections can be made accordingly.

2.2.2 Subsequent Expansion Phases

In each subsequent phase, a column of nodes is added to the $(p, k+1, m_0)$ ShuffleNet until it becomes a full (p, k+1) ShuffleNet. Each subsequent phase is composed of three steps. In general, in phase i, the $(p, k+1, m_0+i-2)$ ShuffleNet is expanded to a $(p, k+1, m_0 + i - 1)$ ShuffleNet, where $2 \le i \le k - m_0 + 2$. Along with the descriptions we will expand our example (2,4,2) ShuffleNet to a (2,4,3) ShuffleNet.

1. Construct a (p, k+1, $m_0 + i - 1$) ShuffleNet and find the connectivity and wavelength channels assignment.

Remark: We illustrate this by the (2,4,3) ShuffleNet shown in Figure 2.11.

- Disconnect the output ports of the star-couplers that lead to node 0, node 1, ..., up to node (p^{k+1} 1) in the ShuffleNet of the previous phase.
 Remark: In our example, the output ports of star-couplers 4 to 7, originally connected to nodes 0 to 15, are now disconnected.
- 3. Connect a column of p^{k+1} new nodes, p^{k+2}/w new star-couplers and the loose-end fibers to the network according to the connectivity graph of the $(p, k+1, m_0 + i 1)$ ShuffleNet.

Remarks: Since p^{k+1} nodes are added and each node requires p channels, p^{k+2}/w new star-couplers are needed. In our example, nodes 32 to 47 are added together with four new star-couplers numbered from 8 to 11. Nodes 32 to 47 are connected to star-couplers 4 to 7 for reception and star-couplers 8 to 11 for transmission. Finally, the loose-end fibers of nodes 0 to 15 are connected to the newly added star-couplers 8 to 11. Similar to step 3 in expansion phase 1, (2.2) can be used to generate the list of connections.

Chapter 2 Modular Expansion of ShuffleNet

2.3 Discussions

The modular expansion of a (p, k) ShuffleNet to a (p, k + 1) ShuffleNet goes through the following list of partial ShuffleNets: $(p, k + 1, m_0)$, $(p, k + 1, m_0 + 1)$, ..., (p, k + 1, k + 1), where $m_0 = int(k/p) + 1$. The number of growing phases is k + 1 - int(k/p). In each phase of expansion, we need to rearrange fiber connections and update node addresses. Updating of node addresses involves only software changes and can be accomplished easily with the use of the mapping equation (2.3). Rearranging fiber connections is also very simple with the use of standard fiber connectors. In addition, if the star-couplers are centrally located, all plugging and unplugging of standard fiber connectors can be performed at the network hub. If done this way, rearranging fiber connections will not be particularly time-consuming.

For the multi-star implementation of a (p, k) ShuffleNet to be expandable, the number of wavelength channels used per fiber must be a multiple of p^2 and must not be larger than p^k . The former part was discussed in Section II. The latter part is due to that fact that if more than p^k channels are used, more than 1/p column of nodes will need to be connected to a star-coupler and a mapping for all the old nodes to be reused will not exist. This upper bound on w implies that if p is small, the number of usable channels would be small compared to the maximum number of channels that can potentially be supported. More couplers will be needed and the network cost is increased. Fortunately, if a small number of channels is used per fiber, these channels can be spaced farther apart so that less expensive transceivers can be used. On the other hand if we build a ShuffleNet with a large p such that more channels can be used per fiber, since p
represents the number of transceivers per node, the cost of transceivers will add to the overall network cost significantly. The best configuration will probably be determined by the cost of various optical devices.

When compared to Karol's approach of expanding with p [14], expanding a ShuffleNet with k results in a less gradual growth. In addition, when a (p, k)ShuffleNet is expanded to a (p, k + 1) ShuffleNet, the per-node throughput is decreased because each node now takes, on the average, a larger number of hops to reach the other nodes. But the advantages of expanding with k by the method outlined here are that no hardware change needs to be done on the network nodes and no new deployment of fibers are needed for existing nodes. Therefore whether it is more feasible to expand with p or with k depends on the specific application.

We assume that fixed wavelength transceivers are used in the network nodes. Recently there has been quite a lot of research on what can be achieved with frequency-selective devices. For example, Barry and Humblet [20] have shown how to build Latin Routers, which provide single-hop connectivity among Nnodes using only N wavelengths and can be constructed from small building blocks. Fixed wavelength devices are assumed because of its lower cost and better stability over wavelength agile devices. It is reasonable to anticipate that if wavelength agile devices are used, the hardware reconfigurations required in the expansion procedure may be reduced. However, such an issue involves designing a different set of expansion algorithms and is beyond the scope of this thesis.

With the use of partial ShuffleNets in our expansion phases, routing schemes such as the static self-routing scheme [16] and the dynamic routing scheme [21]

Chapter 2 Modular Expansion of ShuffleNet

cannot be used directly. These routing schemes, however, can also be used on partial ShuffleNets after some minor modifications. Since some columns are missing in a partial ShuffleNet, each node can no longer determine which link a packet should be routed by manipulating its own address with the *p*-ary representation of the destination address. Additional information is therefore required to assist routing. In the static routing scheme, for example, each node should keep a table mapping every destination address to a pre-computed path. The pre-computed path may represent the shortest path between two nodes, or a path that contains no faulty nodes and links. Based on the destination address, the corresponding pre-computed path is stored in the header of every packet to indicate which link to take in each intermediate node. In the dynamic routing scheme, whenever a packet is deflected, the pre-computed path in the packet header should be updated by the local node to reflect the new route the packet should take. With these simple modifications, these routing schemes can also be applied on partial ShuffleNets.

Table 2.2:	Percentage	growth	in each	phase of	expanding	a ((2,3)	ShuffleNet.
------------	------------	--------	---------	----------	-----------	-----	-------	-------------

Network size	Percentage growth			
24	00.00%			
32	33.33%			
48	50.00%			
64	33.33%			
96	50.00%			
128	33.33%			
160	25.00%			
192	20.00%			
256	33.33%			
320	25.00%			
384	20.00%			

As an example to show the modular growth property of our algorithm, Table 2.2 shows the percentage growth in network size when a (2,3) ShuffleNet is expanded to a (2,6) ShuffleNet in 10 expansion phases.

It can be shown that by skipping intermediate phases a (p, k) ShuffleNet can be grown directly to a (p, k+1) ShuffleNet without any hardware rearrangement. This implies a growth of more than 100% and would be too drastic for many applications.



Figure 2.8: A (2,4,2) ShuffleNet using 8 wavelength channels per fiber.



Figure 2.9: Multi-star implementation of the (2,3) ShuffleNet.



Figure 2.10: Multi-star implementation of the (2,4,2) ShuffleNet.



Figure 2.11: A (2,4,3) ShuffleNet using 8 wavelength channels per fiber.

Chapter 3

Reconfigurability of ShuffleNet in Multi-Star Implementation

If a ShuffleNet is implemented out of a single star-coupler, each node can potentially be connected to any other nodes in the network. Furthermore, if all the nodes are equipped with tunable transceivers capable of accessing the entire spectrum of wavelength channels in use, the logical topology can be reconfigured to adapt to the changing traffic pattern [10]. Such a network is considered to have maximum network reconfigurability. By "maximum reconfigurability", we mean any node can exchange its logical position in the network with another node and that can be done by simply retuning their transceivers. As reconfiguration is required only when there is a major shift in the traffic pattern, transceivers need only be slowly tunable. Note that during reconfiguration, we still have to conform to the ShuffleNet connectivity graph. In effect, we are rearranging the logical positions of the nodes in the ShuffleNet to obtain the best

¹The materials presented in this chapter will appear in the Proceedings of IEEE INFO-COM'94, Toronto, Canada, Jun. 1994.

nodes placement for a given traffic pattern. Several algorithms for finding the best nodes placement for a given traffic matrix are proposed in [23, 24].

On the practical side, however, the size of a single-star based system is limited by the available power budget and by the finite number of wavelength channels available [6]. To build a large local area network, the multi-star implementation is therefore more preferable [25]. In a multi-star network, however, we do not have the same degree of freedom for reconfiguration as in a single-star network. As we shall see, we are not free to exchange the logical positions of an arbitrary pair of nodes simply by retuning their transceivers because their physical connections may not allow them to do so. In other words, we may only be able to obtain sub-optimal nodes placement for a given traffic pattern with this limited freedom of reconfiguration.

In this chapter we investigate the reconfigurability issue in multi-star ShuffleNets. We show how reconfigurability is affected by the number of star-couplers used. Finally we propose a channels assignment algorithm to maximize reconfigurability for a given network configuration.

3.1 Reconfigurability of ShuffleNet

3.1.1 Definitions

In the single-star implementation, the logical positions of an arbitrary pair of nodes can be exchanged by retuning their transceivers. In the multi-star implementation, however, this is not always the case. Let us first explain the concept of reconfigurability. In a (p, k) ShuffleNet, each node transmits to p nodes in the next column and receives from p nodes in the previous column. Denote

the set of nodes which a node transmits to as the transmit set of that node. Similarly denote the set of nodes which a node receives from as the receive set of that node. For example, the transmit set and receive set of node 0 in a (2,2)ShuffleNet as shown in Figure 3.1 is $\{4, 5\}$ and $\{4, 6\}$ respectively. Exchanging the logical position of a pair of nodes is a two-part procedure. In the first part, the transmit sets of the two nodes are exchanged and in the second part, their receive sets are exchanged. Two nodes form a rearrangable pair if such exchange procedure can be done by retuning the transceivers. We define reconfigurability of a network as

$Reconfigurability = \frac{Total \ number \ of \ rearrangable \ pairs}{Total \ number \ of \ node \ pairs}$

The reconfigurability is a measure of the degree of freedom in node placements optimization. Its value lies between 0 and 1. The reconfigurability of a single-star network is 1. Note that the reconfigurability defined here is different from the definition as in [25]. In [25] reconfigurability is defined as "the number of nodes which can potentially be accessed by each node in a physical topology." However, in a multi-star ShuffleNet, even if each node can potentially access all the other nodes in the network, we still cannot guarantee that the logical identities of any pair of nodes can be exchanged by retuning.

3.1.2 Rearrangable Conditions

Here, we consider only symmetric multi-star ShuffleNets. In a symmetric multistar ShuffleNet, the star-couplers used are of identical size and each star has the same number of input and output ports. Moreover, the same number of



Figure 3.1: Channels assignment for a (2,2) ShuffleNet.

channels is used on each coupler. In such a network, at least one of the three conditions below must be satisfied before two nodes can become a rearrangable pair:

(a) Their transmit sets are the same and they receive from the same set of star-couplers; or

(b) Their receive sets are the same and they transmit to the same set of star-couplers; or

(c) They transmit to the same set of star-couplers and receive from the same set of star-couplers. The two sets may be different.

To explain the three conditions, we need to make use of 2 important observations:

(i) To exchange the transmit sets of two nodes just by retuning, the two nodes must transmit to the same set of star-couplers; and

(ii) To exchange the receive sets of two nodes just by retuning, the two nodes must receive from the same set of star-couplers.

Let us illustrate (i) by means of an example; (ii) follows by the same principle. Consider the (2,2) ShuffleNet as shown in Figure 3.1. The integer pair associated with each link represents (star-coupler, channel). In this example we assume that there are 4 star-couplers, each with 4 channels. We assume that there is only 1 transmitter or receiver associated with each fiber. In other words, we exclude the possibility that more than 1 transmitter or more than 1 receiver is connected to a fiber.

Consider nodes 0 and 3 in Figure 3.1. The transmit sets for nodes 0 and 3 are $\{4, 5\}$ and $\{6, 7\}$ respectively. Node 0 transmits to nodes 4 and 5 through star-couplers 0 and 1 respectively and node 3 transmits to nodes 6 and 7 through star-couplers 2 and 3 respectively. To exchange their transmit sets, node 0 must be able to transmit to nodes 6 and 7 through star-couplers 0 and 1. But such connections are not possible because the respective connections from star-couplers 0 and 1 to nodes 6 and 7 have already been occupied by node 1. Similarly node 3 cannot take up the transmit set of node 0 because the connections through star-couplers 2 and 3 to nodes 4 and 5 have already been occupied by node 2. There is no way for nodes 0 and 3 to exchange their transmit sets simply by retuning. For nodes 0 and 1, however, this is not the case. Both nodes 0 and 1 transmit through star-couplers 0 and 1 to their respective transmit sets, which are {4, 5} and {6, 7} respectively. Node 0 uses channel 0 while node 1 uses channel 1 in both stars. To exchange their transmit sets, we only need to tune the transmitters of node 0 to channel 1 and the transmitters of node 1 to channel 0. This exchange operation does not incur any conflict in fiber usage because the two nodes are just swapping their wavelength channels. As long as a pair of nodes are transmitting to or receiving from the same set of star-couplers, their

transmit sets or receive sets can be exchanged by swapping their wavelength channels.

Using observations (i) and (ii), we can derive conditions (a), (b) and (c) easily. As stated, to exchange the identities of 2 nodes, we need to swap their transmit sets and receive sets. If two nodes have identical transmit sets and they receive from the same set of couplers, we only need to swap their receive sets and that can be done by retuning. Hence condition (a) is sufficient. Similarly if two nodes have identical receive sets and they transmit to the same set of couplers, we only need to swap their transmit sets and that can be done by retuning. Therefore condition (b) is also sufficient. Lastly if the two nodes do not have identical transmit sets or receive sets, but they transmit to the same set of couplers and receive from the same set of couplers, their transmit sets and receive sets can still be swapped by retuning. This is condition (c). Note that due to the structure of ShuffleNet, a pair of nodes cannot have the same transmit and receive sets.

3.1.3 Formal Representation

To represent the three conditions formally, let Ω be the set of nodes in the ShuffleNet and define:

$$N_{ts} = \{(x, y) \in \Omega \times \Omega : x < y, \text{ the transmit sets of } x \text{ and } y \text{ are equal}\} (3.1)$$

In other words, N_{ts} is a set of node pairs in which the nodes in each pair have identical transmit sets. The condition x < y is used to remove duplicated node pairs because we consider the pair (x, y) to be equivalent to the pair (y, x).

Similarly define:

$$N_{rs} = \{(x,y) \in \Omega imes \Omega : x < y, the receive sets of \}$$

$$x and y are equal$$
 (3.2)

 $T_{sc} = \{(x,y) \in \Omega \times \Omega : x < y, x \text{ and } y \text{ transmit to } \}$

the same set of couplers
$$\}$$
 (3.3)

 $R_{sc} = \{(x, y) \in \Omega \times \Omega : x < y, x \text{ and } y \text{ receive from } \}$

the same set of couplers} (3.4)

 $Z = \{(x, y) \in \Omega \times \Omega : x < y, x \text{ and } y \text{ form a rearrangable pair}\}(3.5)$

Using the above notations, we can combine conditions (a), (b) and (c) and rewrite Z in (3.5) as

$$Z = (N_{ts} \cap R_{sc}) \cup (N_{rs} \cap T_{sc}) \cup (T_{sc} \cap R_{sc})$$

$$(3.6)$$

The number of rearrangable pairs is therefore equal to |Z|, where the $|\cdot|$ operator denotes the number of elements in set Z. As an example, for the (2,2) ShuffleNet with the channels assignment as shown in Figure 3.1, $N_{ts} = \{(0,2), (1,3), (4,6), (5,7)\}, N_{rs} = \{(0,1), (2,3), (4,5), (6,7)\}, T_{sc} = \{(0,1), (0,4), (0,5), (1,4), (1,5), (4,5), (2,3), (2,6), (2,7), (3,6), (3,7), (6,7)\}, and <math>R_{sc} = \{(0,2), (0,4), (0,6), (2,4), (2,6), (4,6), (1,3), (1,5), (1,7), (3,5), (3,7), (5,7)\}$. Therefore $Z = \{(0,1), (0,2), (0,4), (1,3), (1,5), (2,3), (2,6), (3,7), (4,5), (4,6), (5,7), (6,7)\}$. The number of rearrangable pairs, |Z|, is 12. Reconfigurability of the network is therefore equal to

$$\begin{aligned} Reconfigurability &= \frac{Total \ number \ of \ rearrangable \ pairs}{Total \ number \ of \ node \ pairs} \\ &= \frac{12}{C_2^8} = \frac{12}{28} = 0.4286 \end{aligned}$$

3.2 Maximizing Network Reconfigurability

The reconfigurability of a ShuffleNet depends on its physical implementation. For a given configuration, however, network reconfigurability can be maximized by proper channels assignment.

In theory the best channels assignment which gives the maximum reconfigurability can be found by exhaustive search. In practice, however, such exhaustive search will be very costly even for the smallest ShuffleNet. For example, suppose we want to find the best channels assignment for the 4-star implementation of a (2,2) ShuffleNet. There are a total of 16 links and we need to assign each link to a channel on one of the four star-couplers. Since we are only interested in which star-coupler a link is assigned to, not the specific wavelength channel used, the number of combinations is given by the multinomial coefficient $\begin{pmatrix} 16\\ 4 \ 4 \ 4 \ 4 \end{pmatrix}$. In addition, the 4 star-couplers are indistinguishable. The total number of trials needed in an exhaustive search is equal to $\begin{pmatrix} 16\\ 4 \ 4 \ 4 \ 4 \ 4 \end{pmatrix} \times \frac{1}{4!} = 2,627,625$. As the size of the ShuffleNet increases, this figure increases exponentially and the method of exhaustive search becomes impractical.

We can maximize network reconfigurability by maximizing Z. From (3.6), we can see that maximizing Z is equivalent to maximizing $N_{ts} \cap R_{sc}$, $N_{rs} \cap T_{sc}$ and $T_{sc} \cap R_{sc}$ simultaneously. For a given ShuffleNet, N_{ts} and N_{rs} are fixed, but T_{sc} and R_{sc} depend on the channels assignment. We therefore want to find a "good" channels assignment so that Z is maximized. In the following, we show how we can construct such a "good" channels assignment algorithm step-by-step.

3.2.1 Rules to maximize T_{sc} and R_{sc}

Here we first derive two rules to maximize T_{sc} and R_{sc} . The two rules will be used to construct a channels assignment algorithm which maximizes reconfigurability. The two rules are:

Rule 1. To maximize T_{sc} , connect the p outgoing links of each node to p different star-couplers.

Rule 2. To maximize R_{sc} , connect the p incoming links of each node to p different star-couplers.



Figure 3.2: Rule 1.

The idea behind rule 1 and rule 2 is very simple. Let us explain rule 1 using a simple example. Suppose we want to assign 8 outgoing links of 4 nodes to 2 couplers. To maximize T_{sc} , we should maximize the number of nodes transmitting to the same set of couplers. Among other possible alternatives, we can either assign the 2 links of each node to different couplers, as shown in Figure 3.2a, or we can assign them to the same coupler, as shown in Figure 3.2b. We can easily see that the channels assignment in Figure 3.2a is the best in term of maximizing T_{sc} . Rule rule 2 can be explained similarly.

3.2.2 Rules to Maximize Z

In the following, we derive 3 additional rules which are used to construct the channels assignment algorithm. The 3 rules below are derived from conditions (a), (b) and (c) respectively and are targeted to maximize $N_{rs} \cap T_{sc}$, $N_{ts} \cap R_{sc}$ and $T_{sc} \cap R_{sc}$.

Rule 3. To maximize $N_{rs} \cap T_{sc}$: Apply rule 1 to maximize T_{sc} and assign the node pairs in N_{rs} to transmit to the same set of couplers.

Rule 4. To maximize $N_{ts} \cap R_{sc}$: Apply rule 2 to maximize R_{sc} and assign the node pairs in N_{ts} to receive from the same set of couplers.

Rule 5. To maximize $T_{sc} \cap R_{sc}$: Apply rule 1 and rule 2 to maximize T_{sc} and R_{sc} respectively and take advantage of the structural properties of ShuffleNet to maximize $T_{sc} \cap R_{sc}$.

Basically rule 3 aims at constructing a T_{sc} with maximal overlapping with N_{rs} . In the extreme case, T_{sc} should be made identical to N_{rs} . However, this may not be possible because of the constraints imposed by the system parameters such as the number of wavelength channels available per coupler. The best we

can do is to maximize T_{sc} using rule 1 and enlarge the overlapping of N_{rs} and T_{sc} as much as possible. By assigning the node pairs in N_{rs} to transmit to the same set of couplers, we are making as many node pairs as possible in N_{rs} to appear in T_{sc} , thus maximizing $N_{rs} \cap T_{sc}$. The explanation for rule 4 is very similar to that of rule 3. For rule 5, the explanation is not that strict forward. Maximizing $T_{sc} \cap R_{sc}$ is more tricky because merely maximizing T_{sc} and R_{sc} independently do not guarantee a maximal $T_{sc} \cap R_{sc}$. As we will soon see, we need to maximize $T_{sc} \cap R_{sc}$ based on the structural properties of ShuffleNet and on the number of channels available per star-coupler.

3.3 Channels Assignment Algorithms

Let there be w channels available in each coupler and let C be the number of couplers used. Since kp^{k+1} channels are needed to implement a (p, k) ShuffleNet, $C = kp^{k+1}/w$. We assume that kp^{k+1} is divisible by w.

The parameters w and C determines the reconfigurability of a multi-star ShuffleNet. In general, w being a factor of kp^{k+1} is the sole requirement for a multi-star Implementation. However, certain values of w can give a relatively higher reconfigurability. A plot of reconfigurability versus w will show that local maximums appear at points where: (i) w = p; (ii) $w = p \cdot k$; and (iii) $w = Mp^k$, where M is an integer. The reason is that these values of w match the regular structure of ShuffleNet and give a large $T_{sc} \cap R_{sc}$. In this paper, we will construct channels assignment algorithms for this three cases of w.

The channels assignment algorithms we are going to present make use of two unique features of ShuffleNet:

F1. If we divide the nodes in a column sequentially into p groups, the member nodes of each group have identical receive sets.

F2. The outgoing links in the first group of a column determines the first incoming links of every node in the next column, while the outgoing links in the second group determines the second incoming links of every node in the next column, and so on.



Figure 3.3: An example (2,2) ShuffleNet.

We can explain F1 and F2 easily by an example. A (2,2) ShuffleNet is shown in Figure 3.3. Each link is marked with a number in circle. We can divide the nodes in the each column into p=2 groups. For example, in the second column, nodes 4 and 5 belong to one group while nodes 6 and 7 belong to the other. We can easily see that the receive sets of nodes 4 and 5 are the same and the receive sets of nodes 6 and 7 are also the same. This explains F1. For F2, consider the outgoing links of nodes 4 and 5, which are links 8, 9, 10 and 11. The 4 links constitute the first incoming links of the 4 nodes in the next column. Similarly

the 4 outgoing links of nodes 6 and 7 constitute the second incoming links of the 4 nodes in the next column.

In the following we construct a channels assignment algorithm for each of the three cases of w. The channels assignment algorithms assign the star-couplers and wavelength channels to the outgoing links only. Because of the regular structure of ShuffleNet, the wavelength channels of the incoming links of each node can be found by a simple labeling algorithm [26].

3.3.1 Channels Assignment Algorithm for w = p

This is the simplest case. By rule 3 and F1, we assign nodes of a group to transmit to the same set of couplers. By rule 1, the p outgoing links of each node are assigned to different couplers. It can be shown that due to the shuffle connection pattern and the fact that the same couplers assignment pattern is applied to every node within a group, rule 4 is automatically satisfied once rule 3 is satisfied. Since there are only p channels available in each coupler, we cannot further increase the reconfigurability using rule 5.

An example channels assignment is shown in Figure 3.4. We can see from the figure that once we have assigned nodes of a group (e.g. nodes 0 and 1) to transmit to the same set of couplers, nodes which have the same transmit sets (e.g. nodes 4 and 6) are automatically assigned to receive from the same set of couplers. The formal description of the algorithm is given below. The algorithm outputs a 2-dimensional array network[] which contains the channels assignment.

Channels Assignment Algorithm for w = p

Inputs: p, k and w.

Outputs: channels assignment array network[].

```
begin

channel := 0; coupler := 0;

for node := 0 to kp^{k-1} do

begin

for link := 0 to p - 1 do

network[node, link] := (coupler + link, channel);

if channel = w - 1

begin

channel := 0;

coupler := coupler + p;

end

else

channel := channel + 1;

end;

end.
```

3.3.2 Channels Assignment Algorithm for $w = p \cdot k$

The case where $w = p \cdot k$ is similar to the case of w = p except that with additional channels available on each coupler, we can apply rule 5 to further increase the size of $T_{sc} \cap R_{sc}$ and hence increasing Z. To apply rule 5, the idea is to make the assignment of couplers to the outgoing links in a column be identical for all columns, such as the channels assignment for a (3,2) ShuffleNet as shown in Figure 3.5. In Figure 3.5, we can see that the couplers assignment to the links in the first column is identical to that in the second column. Since $w = p \cdot k$



Figure 3.4: Channels assignment for a (2,2) ShuffleNet using 2 channels per coupler.

and there are k columns in the network, each coupler can provide p channels for each column. Therefore, similar to the case of w = p, rule 3 and rule 4 can be satisfied. In addition, since the couplers assignment for every column are the same, nodes along the same row in the ShuffleNet connectivity graph will have equal transmit sets and receive sets, which implies that they are rearrangable. For example, in Figure 3.5, nodes 0 and 9 are rearrangable because they both transmit to the same set of couplers (i.e. 0, 1 and 2) and receive from the same set of couplers (i.e. 0, 3 and 6). It can be shown that if we do not have identical couplers assignment for each column, the reconfigurability will be reduced. The formal description of the channels assignment algorithm is given as follows:

Channels Assignment Algorithm for $w = p \cdot k$

Inputs: p, k and w.

Outputs: channels assignment array network[].

begin

```
channel := 0; coupler := 0; current_channel := -p;
node := 0;
while (node < kp^k) do
   begin
      if rem(node, p^k) = 0
         begin
            current\_channel := current\_channel + p;
            channel := current_channel;
            coupler := 0;
         end;
      for link := 0 to p - 1 do
         network[node, link] := (coupler + link, channel);
      ifchannel = current\_channel + p - 1;
         begin
            channel := current_channel;
            coupler := coupler + p;
         end
      else
         channel := channel + 1;
      node := node + 1;
```

end;

end.



Figure 3.5: Channels assignment for a (3,2) ShuffleNet using 6 channels per coupler.

3.3.3 Channels Assignment Algorithm for $w = Mp^k$

With additional channels available from each coupler, we can further increase Z by applying rule 5. The idea is to assign all the nodes in a column to transmit to the same set of couplers so that the nodes in the next column will all receive from the same set of couplers. To achieve that, a total of p couplers are used in each column, each providing p^k channels. In addition, a "round robin" assignment

method is introduced. In such method, a particular couplers assignment pattern is applied to consecutive links of the nodes within a group. From one group to the next, the pattern is shifted to the left by 1 position. By using this method, we can guarantee that the nodes in the next column all receive from that same set of couplers. Since they are also made to transmit to the same set of couplers (although that may be a different set), any pair of nodes within a column will become rearrangable. The same procedure is applied to all columns in the network. Let us illustrated this by using an example ShuffleNet as shown in Figure 3.6. In the first column, the 2 outgoing links of nodes 0 and 1 are assigned in the order of couplers 0 and 1. As a result, the first incoming links of nodes 4 to 7 are assigned to couplers 0, 1, 0, and 1 respectively. If we assign the 2 outgoing links of nodes 2 and 3 in the order of couplers 1 and 0, which is a shifted version, the second incoming links of nodes 4 to 7 will be assigned to couplers 1, 0, 1, and 0 respectively. Nodes 4 to 7 will then receive from the same set of couplers. This method is in fact based on feature F2 described previously. If a total of p couplers are used in the whole network, each coupler will provide kp^k channels. In that case, the couplers assignment of every column will be identical, implying that every node will transmit to and receive from exactly the same set of couplers. A 100% reconfigurability will be attained.

The formal description of the algorithm is given below. The function rem(x,y) denotes the remainder of x/y.

The "Round Robin" Channels Assignment Algorithm for $w = Mp^k$ Inputs: p, k and w.

Outputs: channels assignment array network[].

begin

```
for column := 0 to k - 1 do
   for group := 0 to p - 1 do
      for shift := 0 to p^{k-1} - 1 do
          begin
             offset := rem(column, k/M) \times p;
             current\_coupler := rem(column, k/M) \times p + group;
             if (M \ll 1) then
                current\_channel := (column/M) \times p^k + group \times p^{k-1} + shift
             else
                current\_channel := rem((group \times p^{k-1} + shift), p^k);
             coupler := current_coupler;
             channel := current_channel;
             node = column \times p^k + group \times p^{k-1} + shift;
             for link := 0 to p - 1 do
                 network[node, link] := (rem(coupler + link) + p, channel);
          end;
```

end.

3.4 Discussions

The use of multiple star-couplers is inevitable during implementation if the network size is large. The multi-star implementation enables a reduced complexity of each individual coupler and relaxes the power budget constraint, thereby increasing the number of nodes that can be accommodated. The price to pay is a reduced network reconfigurability.



Figure 3.6: The "Round Robin" assignment method.

The reconfigurability of a network is a measure of the degree of freedom one can reconfigure the network in respond to the change of network traffic patterns. If reconfigurability is equal to 1, we can freely exchange the logical positions of an arbitrary pair of nodes just by retuning their transceivers. There are different choices of node pairs to be exchanged. One possibility is that we may exchange the logical positions of a pair of nodes so that those "busy" nodes are logically closer to each other and so packets can take fewer hops to reach their destinations. This can reduce the weighted mean hop count of the network [23]. If we do not have a reconfigurability of 1, we can only settle for a sub-optimal network configuration. It is obvious that we need to trade off reconfigurability against the advantages of the multi-star implementation.

We found that to implement a (p, k) ShuffleNet, if we use p star-couplers (i.e. $w = kp^k$), we can still attain the maximum reconfigurability while enjoying the advantages of a multi-star network. The reason is that if p couplers are used,

the number of channels available on each coupler must be $kp^{k+1}/p = kp^k$, which is a multiple of p^k . In each column, p^k channels are needed from each coupler. We can therefore assign the whole column of nodes to transmit to the set of p couplers. Moreover, all the nodes in the next column will also receive from exactly the same set of couplers. Since there are k columns, and the number of channels available per coupler is kp^k , we can assign the nodes in every column to transmit to the same set of couplers. In other words, all the nodes in the network will transmit to and receive from the same set of couplers. Any arbitrary pair of nodes can be rearrangable, which leads to maximal reconfigurability.

One point worth mentioning is that under the condition of limited reconfigurability, even if we have exchanged the logical identities of two nodes, their physical identities remain the same because their physical identities are already fixed during implementation.

It is observed that the node placement optimization algorithms proposed for single-star ShuffleNet in [23] may work for the multi-star case as well. However, it is not possible to obtain a sub-optimal node placement strategy when the reconfigurability is less than 1.

To implement a (p, k) ShuffleNet, the best configuration is to use p starcouplers, each with kp^k wavelength channels. In this way, we can take advantage of the multi-star implementation while maintaining a 100% reconfigurability. As an example, Table 3.1 shows the reconfigurability for different values of w and C in the multi-star implementation of a (3,2) ShuffleNet. We can see from the table that if we use 3 star-couplers in the implementation, we can attain the maximum reconfigurability of 1.

Table 3.1: Reconfigurability of the (3,2) ShuffleNet in multi-star implementation for various values of w and C.

w	C	Reconfigurability
3	18	0.2353
6	9	0.2941
9	6	0.4706
18	3	1.0000

Chapter 4

Conclusions

In this thesis we have investigated two important issues of ShuffleNet, namely modular expansion of network size and reconfigurability in multi-star implementation.

We discuss in Chapter 2 that modular expansion of ShuffleNet is difficult because of the regularity of the ShuffleNet structure. Since it is not possible to separate an expansion procedure from its physical implementation, we first develop a multi-star implementation method for ShuffleNet. The multi-star implementation method is especially suitable for large networks because the limited available wavelength channels are spatially reused across different star-couplers. Based on this multi-star implementation, an expansion algorithm is proposed which enables modular growth of ShuffleNets. The expansion algorithm expands a (p, k) ShuffleNet to a (p, k + 1) ShuffleNet in several discrete phases. In each expansion phase, the increase in network size is only fractional. Hardware and software reconfigurations are kept to a minimum and they involve rearranging fiber connections and updating node addresses and routing tables. Since the star-couplers are likely to be centrally located, the expansion procedure is not particularly time-consuming and will not cause much disturbance to existing nodes except for the network down-time during reconfiguration.

In Chapter 3, we have investigated the reconfigurability of ShuffleNet in the multi-star implementation. We show how reconfigurability is related with w, the number of channels available per coupler, and C, the number of couplers used. Based on a set of observations and rules, we propose three Channels Assignment Algorithms for different values of w to maximize reconfigurability under a particular implementation. We found that to implement a (p, k) ShuffleNet, the best configuration is to use p star-couplers, each with kp^k wavelength channels. In this way, we can take advantage of the multi-star implementation while maintaining a 100% reconfigurability.

Finally, some related directions for further research are outlined in the following. The expansion procedure developed in this thesis expands a ShuffleNet with the parameter k while the one proposed by Karol in [14] expands with the parameter p. Both methods have their pros and cons, as discussed in the previous chapters. It would be worthwhile to derive an expansion algorithm by combining the two algorithms. The combined algorithm should allow expansion with either parameter and possess the advantages of the original ones. Secondly, although the problems studied in this thesis are targeted for ShuffleNets only, similar problems exist in other regular structure multihop networks. Results and insights obtained in this thesis can possibly be extended to other regular multihop networks.

Bibliography

- [1] P.E. Green, Fiber Optic Networks, Prentice-Hall, 1993.
- [2] M.C. Brain, "Coherent Transmission Field Demonstrations," Conf. Record, Optical Fiber Commun. Conf., P.29, 1991.
- [3] M.J. Creaner, R.C. Steele, G.R. Walker, N.G. Walker, J. Mellis, S. Al-Chalabi, I. Sturgess, M. Rutherford, J. Davidson, and M. Brain, "Field Demonstration of 565 Mbit/s DPSK Coherent Transmission System over 176 km of Installed Fiber," *Elect. Ltrs.*, Vol.24, No.22, P.1354–1356, 1988.
- [4] T. Imai et. al, "Field Demonstration of 2.5 Gbit/s Coherent Optical Transmission Through Installed Submarine Fiber Cables," Elect. Ltrs., Vol.26, No.17, P.1407-1409, 1990.
- [5] N.A. Olsson et. al, "68.3 km Transmission With 1.37 Tbitkm/s Capacity using Wavelength Division Multiplexing of Ten Single-Frequency Lasers at 1.5 micrometers," Elect. Ltrs., Vol.21, P.105–106, 1985.
- [6] C.A. Brackett, "Dense Wavelength Division Multiplexing Network: Principles and Applications," *IEEE Journal on Selected Areas in Communications*, Vol.8, No.6, Aug. 1990, P.948-964.

- B. Mukherjee, "WDM-Based Local Lightwave Networks Part I: Single-Hop Systems," *IEEE Network Magazine*, May 1992, P.12-27.
- [8] J.A. Bannister, L. Fratta and M. Gerla, "Topological Design of the Wavelength-Division Optical network," *Proceedings of IEEE GLOBE-COM*'89, Nov. 1989, P.1022-1029.
- [9] J.A. Bannister and M. Gerla, "Design of the Wavelength-Division Optical Network," Proceedings of IEEE ICC'90, Apr. 1990, P.962-967.
- [10] J.-F.P. Labourdette and A.S. Acampora, "Logically Rearrangeable Multihop Lightwave Networks," *IEEE Transaction on Communications*, Vol.39, Aug. 1991, P.1223-1230.
- [11] B. Mukherjee, "WDM-Based Local Lightwave Networks Part II: Multihop Systems," IEEE Network Magazine, Jul. 1992, P.20-32.
- [12] C. Rose, "Mean Internodal Distance in Regular and Random Multihop Networks," *IEEE Transactions on Communications*, Vol.40, No.8, Aug. 1992, P.1310-1318.
- [13] A.S. Acampora and M.J. Karol, "An Overview of Lightwave Packet Networks," *IEEE Network Magazine*, Jan. 1989, P.29-41.
- [14] M.J. Karol, "Optical Interconnection Using ShuffleNet Multihop Networks in Multi-Connected Ring Topologies," Proceedings of ACM SIGCOMM'88 Symposium, 1988, P.25-34.

- [15] A.S. Acampora, M.J. Karol and M.G. Hluchyj, "Terabit Lightwave Networks: The Multihop Approach," AT&T Technical Journal, Nov./Dec. 1987.
- [16] M.G. Hluchyj and M.J. Karol, "ShuffleNet: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks," *Proceedings of IEEE INFOCOM'88*, 1988, P.379-390.
- [17] Y. Birk, M.E. Marhic and F.A. Tobagi, "Selective Broadcast Interconnections (SBI) for Wideband Fiber-Optic Local Area Networks," Proc. 2nd Intl. Technical Symp. on Optical and Electro-Optical Applied Science and Engineering, Conf. on Broadband Networks (SPIE), Cannes, France, 28-29 Nov. 1985.
- [18] Y. Birk, "Fiber-Optic Bus-Oriented Single-Hop Interconnection among Multi-Transceiver Stations," *IEEE Journal of Lightwave Technology*, Vol.9, No.12, Dec. 1991, P.1657–1664.
- [19] Y. Birk, "Power Efficient Layout of a Fiber-Optic Multistar that permits log₂ N Concurrent Baseband Transmissions among N Stations," to appear in *IEEE Journal of Ligthwave Technology*, special issue on broadband optical networks, 1993.
- [20] R.A. Barry and P.A. Humblet, "Latin Routers, Design and Implementation," *IEEE Journal of Lightwave Technology*, Vol.11, No.5/6, May/June 1993, P.891-899.

- [21] M.J. Karol and S. Shaikh, "A Simple Adaptive Routing Scheme for Congestion Control in ShuffleNet Multihop Lightwave Networks," *IEEE Journal* on Selected Areas in Communications, Vol.9, Sept. 1991, P.1040-1051.
- [22] P.P. To, "Reconfigurability of ShuffleNets in Multi-Star Implementation," to appear in *Proceedings of IEEE INFOCOM'94*, Toronto, Canada, June 1994.
- [23] S. Banerjee and B. Mukherjee, "Algorithms for Optimized Node Arrangements in ShuffleNet Based Multihop Lightwave Networks," Proceedings of IEEE INFOCOM'93, 1993, P.557-564.
- [24] T.K. Chan and T.S. Yum, "Action Node Placement in ShuffleNets," to appear in Proceedings of IEEE INFOCOM'94, Toronto, Canada, 1994.
- [25] A. Ganz, B. Li and L. Zenou, "Reconfigurability of Multi-Star Based Lightwave LANs," Proceedings of IEEE GLOBECOM'92, 1992, P.1907-1910.
- [26] P.P. To, T.S. Yum and Y.W. Leung, "Multi-Star Implementation of Expandable ShuffleNets," Proceedings of IEEE GLOBECOM'93, 1993, P.468-473.



