Research on Scheduling Algorithm of Three-Stage Clos Switching Networks

Dongtao Cao*, Yimin Ma
China Academy of Space Technology (Xi’an), Xian, China

*Corresponding author: 24760054@qq.com

Abstract. The three-stage Clos switching network is a kind of multi-stage switching network which is commonly used to build a high-speed and large-capacity switching network, and the scheduling algorithm of the switching network affects the performance of the switching network. The article introduces and analyzes the research ideas of the scheduling algorithm of the three-stage Clos network, and provides a reference for further research on high-performance scheduling algorithms.

Keywords: Clos, Scheduling Algorithm, Switching Network.

1. Introduction
With the continuous development of optical fiber communication systems, the rate of communication links continues to increase [1], and switching nodes have gradually become the bottleneck of the expansion of the switching system. Clos network is suitable to build large-capacity switching networks and is given a new life again. Clos network has the characteristics of good scalability, modularity, and multi-path. It is a commonly used multi-stage switching network for building high-speed and large-capacity switching networks. The topology of Clos network is widely used in data centers structure called "Fat Tree" [2]. Many communication manufacturers' routers and switch products also use the Clos network, such as H3C's data center core switch S12500, Huawei's data center network switch Cloud Engine 12800 series, Cisco's carrier-grade CRS series high-end routers and Juniper EX9208. And with the development of on-board switching technology and laser communication technology, the application of Clos network on the satellite to build a high fault-tolerant and large-capacity switching network has also achieved many research results [3-6].

2. Clos switching network model and scheduling algorithm

2.1. Clos switching network model
The first stage of the commonly used symmetrical three-stage Clos switching network is composed of $k \times m$ switching elements (SE), and the second stage is composed of $m \times k$ SEs, and the third stage is composed of $k \times n$ SEs. There is exactly one link between two switching modules at two adjacent switching stages. The total switching capacity of the entire network is $N = n \times k$, and symmetrical three-stage Clos switching network is often denoted by $C(m, n, k)$.

The three-stage Clos network can be divided into buffered Clos networks, such as memory-space-memory (MSM), MMM, SMM, SMS and bufferless $S^3$ Clos network. Each switching module in the
input and output stages of the MSM network uses a shared buffer to store the cell conflicting in the outputs and there is no buffer in the central stage and the problem of cell disorder is avoided. Therefore, rich research results about the MSM Clos switching network has achieved. Figure 1 shows the MSM Clos model.

2.2. Centralized scheduling algorithm of Clos switching network

The scheduling algorithm of the Clos network can be divided into two categories: centralized and distributed scheduling algorithms (whether all request information is obtained before the routing can also be divided into dynamic algorithms and static algorithms). The centralized scheduling algorithm requires unified selection of all requests and the algorithm complexity is higher, and the scalability is poor, which is not conducive to the distributed implementation of the network [7].

Centralized algorithms mainly include matrix decomposition algorithm [8-11], bipartite graph matching [12], and bipartite graph edge coloring algorithm [13, 14]. The distributed scheduling algorithm means that each switching module can independently perform its own input and output port matching without knowing the routing information of other modules, such as CRRD [15] and its improved algorithm [16-19]. This article mainly focuses on MSM and related scheduling algorithm of this type Clos network is introduced. Of course, many of those algorithms are not limited to the MSM Clos network.

The centralized algorithm needs to obtain the connection requests of all IM to construct a traffic matrix $H$. Each row of the matrix represents an IM, each column of the matrix represents an OM, and the elements in the matrix represent the number of requests from an IM to an OM. The matrix decomposition algorithm decomposes the traffic matrix into the sum of $m$ permutation matrices (each row and column of this matrix has only one element 1, and the remaining elements are zero). As shown in Equation1, each permutation matrix represents the connection mode of a CM.

$$H = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 2 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (1)$$

Another way of matrix decomposition is to generate the specification matrix $S$ shown [9]. Each row of the matrix represents an IM, and each column represents an OM. The elements in the matrix
indicate that \( IM(i) \) set up the link to the \( OM(j) \) by the \( CM(r) \), rewrite the traffic matrix \( H \) as shown in Equation 2. The symbol \( '*' \) represents the elements swapped each time.

\[
S = \begin{bmatrix}
0 & 0 & 1 & 2 \\
1 & 1 & 2 & 2 \\
0 & 0 & 1 & 2
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 2 \\
1 & 1 & 2 & 2 \\
0 & 0 & 1 & 2
\end{bmatrix}
\begin{bmatrix}
0 & 0' & 1 & 2' \\
1 & 1 & 2 & 2 \\
2 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 2 \\
1 & 1 & 2 & 2 \\
2 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 2 & 1 & 0 \\
1 & 1 & 2 & 2 \\
2 & 0 & 0 & 1
\end{bmatrix}.
\]  

(2)

It can be seen that in the first column of the matrix, \( IM(0) \) and \( IM(2) \) need to establish a path to \( OM(1) \) through \( CM(0) \), and contention will occur at the output of \( CM(0) \). At this time, the connection request of \( IM(2) \) can be transformed into the other \( CMs \), and the column positions of the elements of the same line can be swapped, until each element on each column of the matrix appears exactly once.

The bipartite graph matching algorithm uses graph theory to use each IM and OM as a set of vertices, and the connection request between the IM and OM as a set of edges. The traffic matrix \( H \) is represented by a bipartite graph as Figure 2(a) shown. Through the matching algorithm of the bipartite graph, \( m \) maximum matches are found consecutively, and each match represents the internal connection of a CM. The bipartite graph edge coloring algorithm uses a color to represent each CM, and put color on each edge of the bipartite graph, and each vertex cannot have two edges the same color. As shown in Figure 2(f), each kind of line represents a CM, colored edge between any two vertices indicates that the connection request between these two modules establishes a connection through this CM, and the edges of the same color can be routed using the same CM.

![Figure 2. Bipartite graph matching and edge coloring](image)

Similar to finding the maximum matching of the bipartite graph multiple times, the system distinct representative(SDR) algorithm based on Hall's theorem [20] can use the SDR solved each time as the connection of an CM. For example, the connection of \( C(4,4,3) \) in Equation 1 shows that the matrix can be divided into three subsets \( A_1(1,0,0,2), A_2(1,2,2,1), A_3(2,0,0,1) \), four groups of different representative systems can be found \((1,2,0), (0,2,1), (2,1,0), (0,1,2) \) which are corresponding to path allocation of the CMs.

In addition, there is also a blocking and rearrangement algorithm based on Paull's theorem [21]. This kind of algorithm first searches for the availability of CMs according to the selection rules of intermediate modules, such as random selection, or sequential selection and so on, to find a path for a new valid request. If there is no available CM that meets the conditions, the partially established connection needs to be rearranged into a new connection request to release an available CM.

Centralized scheduling algorithms are often relatively high, and the complexity of the algorithm is related to the number of ports in the network, and it is difficult to apply to actual situations, so easy to expand, and distributed scheduling algorithms are proposed.

2.3. Distributed scheduling algorithm of Clos switching network

Distributed scheduling algorithms do not require centralized scheduler to obtain the overall request information and link connection status of IM. The typical representative of this type of algorithm is the concurrent round-robin-based dispatching (CRRD) algorithm [15]. The CRRD algorithm is a heuristic
algorithm, extended from the single-stage Crossbar’s iSLIP algorithm to the three-stage Clos network. In order to convenient description later, some terminologies used in this paper as shown in Table.1.

**Table.1.** Terminology for Clos network

| Notation | Description |
|----------|-------------|
| IP(i, h) | (h+1)th input port at IM(i), where 0 ≤ i ≤ k – 1, 0 ≤ h ≤ n – 1 |
| OP(j, h) | (h+1)th output port at OM(j), where 0 ≤ j ≤ k – 1, 0 ≤ h ≤ n – 1 |
| LI(i, r) | Output link at IM(i) that is connected to CM(r), where 0 ≤ r ≤ m – 1, |
| LC(r, j) | Output link at CM(r) that is connected to OM(j) |
| VOMQ(i, j) | Virtual output-module queue at IM(i) that stores cells destined for OM(j) |
| G(i, j) | Virtual output queue VOQ(i, j, h) at IM(i) that destined for OM(j) |

The CRRD algorithm is divided into two phase, there are three types of pointers, virtual output queue (VOQ) pointer PV(i, v), IM output port pointer PL(i, r) and CM output port pointer PC(r, j).

The phase 1 is matching within each IM, the maximum match is obtained within IM by multiple iterations. In each time iteration, each unmatched non-empty VOQ sends a request to all unmatched output port, and the output port selects one of the multiple requests it has received according to the pointer PL(i, r) to accept and send the grant to the corresponding VOQ, and VOQ selects one of the received grant according to the pointer PV(i, v) and sends the accept to the output port. Repeat the above process until the maximum number of matches is obtained, as shown in Figure 3.

Phase 2 is the internal matching within the CM. The IM output link LI(i, r) that have matched the VOQ sends a request to the CM output link LC(r, j). LC(r, j) selects one request according to the pointer PC(r, j) to accept and return grant to the LI(i, r) and VOQ. The VOQ obtained grant can send a cell in the next time slot.

In order to reduce the collision probability in the CM of phase 2, the queue number of VOQ(i, v) in CRRD algorithm is set to ν = hk + j (0 ≤ ν ≤ nk – 1), and the pointers PV(i, v), PL(i, r), PC(r, j) are updated only if the matching within the IM is achieved at the first iteration in phase 1 and the request is also granted by the CM in phase 2.

The improved concurrent master-slave round-robin dispatching (CMSD)[15] scheme and static round-robin dispatching (SRRD) scheme[16] based on the CRRD algorithm have been proposed.

CMSD have improved IM queue organization of CRRD and SRRD improved the CMSD’s pointer initialization. The CMSD algorithm introduces the concepts of VOQ group G(i, j) and master-slave arbiter ML(i, r) and SL(i, j, r) in IM, and VOQ from IM(i) destined to the same OM(j) is called G(i, j), each IM(i) has k G(i, j), the pointer PML(i, r) of the master arbiter ML(i, r) is the same as CRRD PL(i, r) that represent m CMs, and the slave arbiter SL(i, j, r) is equivalent to the projection of the master

**Figure 3.** The match within IM of CRRD  
**Figure 4.** The match within IM of CMSD

The improved concurrent master-sale round-robin dispatching (CMSD)[15] scheme and static round-robin dispatching (SRRD) scheme[16] based on the CRRD algorithm have been proposed. CMSD have improved IM queue organization of CRRD and SRRD improved the CMSD’s pointer initialization. The CMSD algorithm introduces the concepts of VOQ group G(i, j) and master-slave arbiter ML(i, r) and SL(i, j, r) in IM, and VOQ from IM(i) destined to the same OM(j) is called G(i, j), each IM(i) has k G(i, j), the pointer PML(i, r) of the master arbiter ML(i, r) is the same as CRRD PL(i, r) that represent m CMs, and the slave arbiter SL(i, j, r) is equivalent to the projection of the master...
arbiter in each VOQ group \( G(i, j) \), each master arbiter \( M_i(i, r) \) has its own slave arbiter \( S_{l}(i, j, r) \) in each \( G(i, j) \), so each \( G(i, j) \) has \( m \) slave arbiters. The first iteration of the CMSD algorithm in IM is shown as Figure 4.

The non-empty VOQ \( (i, j, h) \) in \( G(i, j) \) sends a request to all slave arbiters \( S_{l}(i, j, r) \) within this \( G(i, j) \). At the same time, \( G(i, j) \) containing non-empty VOQ sends requests to all master arbiters. Like CRRD, the master arbiter \( M_{i}(i, r) \) round-robin \( P_{ML}(i, r) \) position to give a grant for slave arbiter \( S_{l}(i, j, r) \), and the authorized slave arbiter \( S_{l}(i, j, r) \) select a VOQ request by round-robin. VOQ accept a grant and return accept information to \( S_{l}(i, j, r) \) and \( M_{i}(i, r) \). The above process is iterated many times until all output ports or all non-empty VOQs are matched. In the phase 1, the relationship between the VOQ and the output port within the IM is established, phase 2 the pointer update method is the same as the CRRD algorithm.

The matching process of the SRRD algorithm is the same as the CMSD algorithm, except that improvements made in the initialization and the way of update of the pointer. The initial value of the pointer is set to, \( P_{V}(i, j, h) = h \), \( P_{SL}(i, j, r) = r \), \( P_{ML}(i, r) = (i + r) \%k \), if \( P_{ML}(i, r) = j \), then \( P_{C}(r, j) = i \). When the pointer is updated, \( P_{ML}(i, r) \) and \( P_{C}(r, j) \) are always incremented by one, \( P_{V}(i, j, h) \) and \( P_{SL}(i, j, r) \) remain unchanged. Due to the static and full synchronization settings of the pointers in the SRRD algorithm, the performance of the SRRD algorithm is much better than that of the CRRD and CMSD algorithms.

According to the matching process of the CRRD algorithm, it can be seen that multiple iterations are required to obtain the maximum match in IM, and if there is output port contention in CM, the established connection in IM will be invalid, resulting in failure to fully utilize the link resources. So, many improved algorithms have been proposed, which can be divided into two categories:

1) The first type of algorithm research the aspect that a wasted certain \( L_{C}(r, j) \) caused by output port contention in the phase 2 of CRRD algorithm. The main idea of the literature [18] is that the unmatched \( L_{C}(r, j) \) arbiter returns the idle link information to each IM in phase 2. Literature [5] proposed the concept of collision domain, that is, the set of VOQ groups \( G(i, j) \) in each IM that compete for each CM \( L_{C}(r, j) \) which matches problem in the Clos switching network is convert to the problem of selecting cells for each CM in the collision domain. A time slot is divided into max \((m, k)\) sub-slots, and the VOQ and \( L_{C}(r, j) \) between IM and CM in the sub-slots are matched many times to avoid the link waste problem of the CRRD algorithm.

2) The second type of algorithm is to add the queue status counter to the CM. The CM is equivalent to the input queue (IQ) crossbar model. The mature algorithm in the crossbar is used to obtain the match in the CM to guide the transmission of cells in the IM. This type of algorithm is also called the reverse dispatching algorithm ([RDA] [6, 19], the settings of IM and CM are shown in Figure 5. This type of algorithm can evenly send incoming cells’ requests to each CM in a round-robin manner at phase 1. After obtaining the matching result within the CM, VOMQ sends the corresponding cell in the next time slot. Those algorithms can make use of the rich research results of IQ-crossbar, and the number of inter-stage communication is small, but the internal acceleration of IM is relatively high, and it may be necessary to simultaneously write \( n \) cells to the same VOMQ or read \( m \) cells from VOMQ at the same time.

![Figure 5. Internal structure of IM and CM of RDA](image-url)

In addition to the above two main type improved algorithms, some scholars have proposed the orthogonal routing configuration algorithm [22]. This type of algorithm does not require multiple
iterations in IM. By statically setting the link connection configuration for each slot, each cell does not conflict at the CM.

3. Conclusions
A good scheduling algorithm can give full play to the advantages of the Clos network. This paper analyzes and summarizes research methods of the existing scheduling algorithm of MSM Clos network. The centralized scheduling algorithm is complex and difficult to apply in practice, while the distributed scheduling algorithm is simple, which is a research trend, but there are links bandwidth resource wasting, and the improved algorithm requiring inter-stage communication or high internal speedup are not expected. And there are fewer algorithms that take service or traffic priority into account, IM's queue organization and how to avoid starvation of low-priority services still need to be studied in this situation.

References
[1] Giovane L, Wang J, Murty M R, et al. Volume Manufacturable High speed 850nm VCSEL for 100G Ethernet and Beyond[C]// Optical Fiber Communications Conference & Exhibition. IEEE, 2016.
[2] S. Alanazi and B. Hamdaoui, "CAFT: Congestion-Aware Fault-Tolerant Load Balancing for Three-Tier Clos Data Centers," 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 2020, 1746-1751.
[3] C. C. Wu, L. Qiao and Q. Chen, "Design of a 640-Gbps Two-Stage Switch Fabric for Satellite On-Board Switches," in IEEE Access, 2020, 8: 68725-68735.
[4] Liu Kai, Yan Jian, Cui Siqian, et al. Onboard parallel wavelength Clos-network [J]. Journal of CAEIT, 2019, 14(1): 30-34.
[5] Zhang Maosen, Qiu Zhiliang, Gao Ya, et al. Divide-and-conquer dispatching scheme for satellite Clos-network switches [J]. Journal of electronics & information technology, 2012, 34(11): 2734-2740.
[6] Zhang Maosen, Qiu Zhiliang, Gao Ya, et al. Reverse dispatching scheme for satellite Clos-network switches [J]. Journal of Xidian university, 2013, 40(4): 96-101.
[7] Yang Jungang, Bao Minquan, Liu Zengji, et al. A distributed scheduling algorithm maintaining cells order for three-stage Clos networks [J]. Chinese journal of computers, 2008, (3): 467-475.
[8] Xin Shanshan. Study on switching and routing algorithms based on multistage switching networks in large-scale data center [D]. Tianjin university, 2016.
[9] Lee H Y, Hwang F K. A new decomposition algorithm for rearrangeable Clos interconnection networks [J]. IEEE Transactions on Communications, 1996, 44(11): 1572-1578.
[10] Hwang, F. K. A modification to a decomposition algorithm of Gordon and Srikanthan [J]. Computers IEEE Transactions on, 1997, 46(8):958-960.
[11] Liu Xiaofeng, Zhao Youjian. A parallel matrix decomposition algorithm for routing on rearrangeable Clos-network switches [J]. Journal of internet technology, 2017, 18(2):229-236.
[12] Hopcroft J E, Karp R M. An $N^{5/2}$ algorithm for maximum matching in bipartite graphs [C]. 12th Annual Symposium on Switching and Automata Theory, East Lansing, Michigan, USA, 1971, October 13-15.
[13] L. Wang, T. Ye, T. T. Lee. A parallel route assignment algorithm for fault-tolerant Clos networks in OTN switches [J]. IEEE Transactions on Parallel and Distributed Systems, 1 May 2019, 30(5): 977-989.
[14] Hwang, F. Control algorithms for rearrangeable Clos networks [J]. IEEE Transactions on Communications, 1983, 31(8):952-954.
[15] Eiji, Oki, Zhigang, et al. Concurrent round-robin-based dispatching schemes for Clos-network switches [J]. IEEE/ACM Transactions on Networking, 2002.
[16] Pun K, Hamdi M. Static round-robin dispatching schemes for Clos-network switches[C]// Workshop on High Performance Switching & Routing. IEEE, 2002.

[17] Zheng Zhen, Qiao Lufeng, Chen Qinghua, et al. A concurrent round-robin-based variable-length packets dispatching scheme for satellite onboard clos-network switches [C]// International Congress on Image & Signal Processing. IEEE, 2016.

[18] Liu Xiaofeng, Zhao Youjian, Chen Guo. Heuristic concurrent dispatching algorithm for MSM Clos-network switches [J]. Journal of software, 2015, 26(10): 2644-2655.

[19] Gao Ya, Qiu Zhiliang, Zhang Maosen, et al. Distributed weight matching dispatching scheme in MSM Clos-network packet switches [J]. IEEE Communications Letters, 2013, 17(3): 580-583.

[20] M. Hall, Combinatorial Theory [M]. Waltham, MA: Blaisdell, 1967.

[21] Paull M C. Reswitching of Connection Networks [J]. Bell Labs Technical Journal, 2013, 41(3): 833-855.

[22] L. Li, L. Wu, J. Wang, H. Liu and X. Li, "Orthogonal-path schedule algorithm for MSM Clos-network switching fabric [C]," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, 1442-1447.