Contention Resolution through Network Global Control in Optical Packet Switching Networks

XIN Ming, CHEN Minghua, CHEN Hongwei, and XIE Shizhong
National Laboratory for Information Science and Technology, Department of Electronic Engineering
Tsinghua University
Beijing, 100084, China
chenmh@tsinghua.edu.cn

***Abstract*—Optical buffering is one of the main obstacles in optical packet switching (OPS) networks. We proposed the network global control strategy to resolve this problem. By introducing the cycled traffic at the edge node and the slot assignment algorithm at the core node, a slotted OPS networks with no contention can be realized. In this network, a good throughput, end to end delay and delay jitter performance can be gotten at high load simultaneously, so it may be a promising candidate for the future optical networks.**

Keywords-optical packet switching (OPS), network global control, slot assignment algorithm, optical buffers

I. INTRODUCTION

It is generally recognized that optical packet switching (OPS) is the key to the success of the next-generation optical networks due to its high data rate, high switching efficiency, transparency to data modulation format and flexible control [1]. However, since it is impossible to construct practical optical RAM buffer nowadays, contention resolution becomes one of the main problems in OPS networks, and seriously limits OPS's practicability in the past decade. Many buffering schemes in the physical layer have been proposed to resolve contention. Buffer based on fiber delay line (FDL) [2] [3] has limited buffering granularity and is too bulky if we want to get an accepted packet loss rate (PLR). Slow light based buffers [4] [5] can offer much smaller buffering granularity, but long delay for high speed signals cannot be realized with a moderate buffer size due to the limited delay-bandwidth product. Until now an efficient and compact optical buffer has not been realized.

While on the other hand, some control mechanisms in the upper layer of OPS have been given to alleviate the difficulty of implementing optical buffer in the physical layer [6] [7] [8]. With these mechanisms, an accepted PLR can be reached, while only a few buffers are needed at each core node. In this paper, we try to give a new network control strategy, aiming to realize an OPS network with zero PLR and no buffers at each core node. Our strategy will be applied to slotted OPS networks called as network global control (NGC) in the following. Firstly, we choose an appropriate time T and make all the torrents (a torrent is some service's traffic between two edge nodes) periodically send their packets with the period of T. Then we use the slot assignment algorithm (SAA) and the transient assignment algorithm (TAA) to decide which torrent's packet should be sent in each slot so that all the potential contentions can be avoided.

The rest of the paper is organized as follows: Section II introduces the network structure of NGC and its working mechanism. Section III gives a detailed explanation of SAA and TAA. Section IV evaluates NGC through simulation analysis. Section V gives some further discussions and finally Section VI concludes this paper.

II. NETWORK STRUCTURE AND WORKING MECHANISM

A. *Traffic Cycling*

The prerequisite of NGC is to introduce the cycled traffic, i.e., let each torrent's packets distribution in the time domain present a certain degree of periodicity. To achieve this goal, first we choose the time T as the global period (GP) for all the torrents in the network. In practice T may need to be optimized according to the network's topology and the traffic's statistical characteristics all over the network. Here, for simplicity, we set T as the average packet transmitting time between every two edge nodes. Then there are $M=T/\Delta T$ slots in each GP, where ΔT is the duration of one slot. For each torrent, suppose that its expected speed is B packets/sec and it has over all N packets. Let $W=BT$ and it will be called as the transmission window (TW) in the following. We mark all the slots with number 1, 2, 3 … in the time order and suppose that the torrent is ready to send its packet at slot 1. Then the traffic cycling can be implemented as follows:

If $N \le W$, choose N slots $S_1, S_2 \ldots S_N$ arbitrarily and send the torrent's packets in these slots, where $1 \le S_1 < S_2 < \ldots < S_N \le M$.

If $N=kW+W_r$, $k>0$, $0 \le W_r < W$, choose W slots $S_1, S_2 \ldots S_W$ arbitrarily and send the torrent's packets in the slots:

$$S_1, S_2 \ldots S_W, S_1+M,$$
$$S_2+M \ldots S_W+M \ldots S_1+(k-1)M, S_2+(k-1)M$$
$$\ldots$$
$$S_W+(k-1)M, S_1+kM, S_2+kM \ldots S_{Wr}+kM,$$

where $1 \le S_1 < S_2 < \ldots < S_W \le M$.

If for each torrent, $N >> W$, after traffic cycling, the packets' distribution in the time domain all over the network will

This work was supported by the National Natural Science Foundation of China under Grant 60632010, the National High-tech Development Project of China under Grant 2006AA01Z237 and 2007AA01Z274, and the National Basic Research Program of China under Grant 2006CB302806.

present very strong periodicity. In fact, if we assume that the average distance between each two edge nodes is at the order of 1000~10000 km, T will be at the order of 0.01~0.1s, since most of the services, such as FTP, E-Mail or media stream need to be transmitted for more than 1 second, for those services $N>10W$ will be satisfied. Fig. 1 gives a schematic illustration about the effect of traffic cycling at some core node. Squares with different colors represent different torrent's packets and the number on each square denotes its output port. Packets from two input ports are trying to output from four output ports. It can be seen that after traffic cycling in the two adjacent GPs T1 and T2 packets have the same distribution, so the contention also appears at the same slots in T1 and T2. If we can appropriately assign the slots of T1 for the torrents so as to eliminate all the contentions, traffic cycling can guarantee that contentions in T2 and several GPs later can also be eliminated. In the following we will give a new network structure and the SAA to show how to eliminate the contentions in T1.

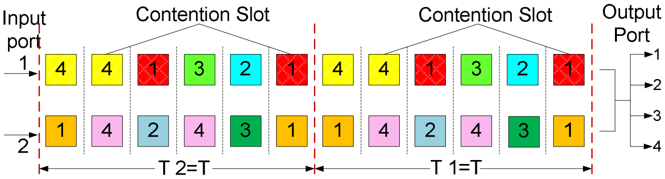


Fig. 1. the schematic illustration of traffic cycling

B. *Network Structure*

The NGC-based OPS network's structure is given in Fig. 2. One core node is chosen as the centre node. Any core node can be the center node, but generally we choose the one closest to the network's topological and geographical center. The center node will be equipped with some powerful electrical processors to calculate and avoid the potential contentions in the network. For each edge and core node, an additional low bandwidth channel is needed to transfer the control information between center and other node. Since deflection routing may disorder packets' arrival at the destination and deteriorate the delay jitter performance a lot [9], each torrent is required to use a fixed route. For example, in Fig. 2, if some torrent needs to be transferred between terminal T_A and T_B, then one fixed route (the dashed red line) will be adopted during the full transmission of this torrent.

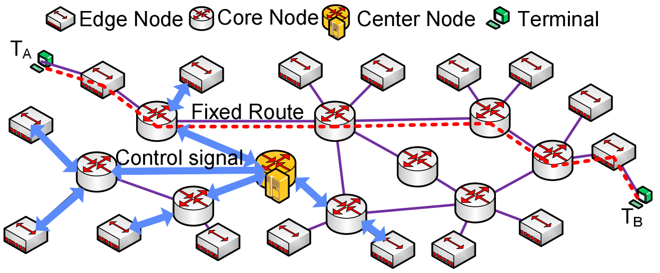


Fig. 2. the NGC-based OPS network structure

C. *Working Mechanism*

The whole process of transmitting torrents can be depicted as a negotiation between edge and center node. First, each edge node sends its torrent statistical information (TSI) to the center node in the control channel (TSI denotes that in one GP, how many torrents need to be transmitted between each two edge nodes, and how many packets in each torrent. More definitely it can be expressed as: in one GP, N_1 packets of torrent 1 are going to be transmitted from edge node 1 to edge node 2, N_2 packets of torrent 2 are from edge node 1 to edge node 3, etc.). After receiving all TSI from each edge node, the center node will first reallocate each torrent's TW and then assign a slot to each packet, based on SAA, which will be given in Section III. Then the center node sends the assignment information back to each edge node, also using the control channel. Based on the returned information, each edge node adjusts its torrents' TWs and sends each packet at its assigned slot. All these procedures above are finished within one GP. Since traffic cycling has guaranteed that the traffic streams at each edge nodes present a strong periodicity in near GPs, it is unnecessary to frequently send TSI to the center node for updating slot assignment. In our scheme, the slot assignment will be updated each 10 GPs.

During each 10 GPs, the edge nodes send different torrent's packets with the period of T, and each slot is used to transmit packet between two fixed edge nodes. For example, if the first and third slot of the first GP at edge node A is assigned to transmit packet going to edge node B, then the following nine GPs' first and third slot will also be used to transmit packet from A to B. If some torrent's transmission is finished, those slots assigned to this torrent in the following GPs will be set idle. If a new torrent from some terminal needs to be added into the network, the terminal will only send a request (including the torrent's target address and its expected TW) to its nearest edge node in the first GP. Then the edge node will allot an appropriate TW to this torrent based on the current situation of slot assignment. If this torrent is urgent and the current available idle slots are too few, the edge node can modify other torrent's TW to guarantee this torrent's quality. So in our scheme, the first GP is used for torrent request, and from the second GP the torrent can obtain the maximum TW the network can afford. This fast start speed feature will be very advantageous when the network encounters large burst traffic.

Each edge node has a real-time traffic stream monitor to record the torrent request change in each 10 GPs. After every 10 GPs, based on the recorded results, the edge node sends a new TSI to the center node to update the slot assignment. With SAA, the center node returns the new slot assignment to each edge node. In order to avoid the contention between the packets with updated slot assignment and the packets of the previous 10 GPs which are still in the network, the new slot assignment cannot be immediately used in the following GP. An extra GP is needed for transition from the old assignment to the new one. In this transition GP, TAA is used to assign the slots. The assigned results of TAA are returned together with those of SAA to each edge node. TAA will also be illustrated in Section III.

III. Algorithm Description

A. *A Simple Example and General Definitions*

We first give a simple example to explain the main idea of SAA. We use the simple topology in Fig. 3: a network with only two core nodes M and N and each core node has two edge nodes A, B and C, D. We assume that there are only 5 slots in one GP, and the packet's transferring time between each two adjacent node is one slot. The TSI of each edge node is also given in Fig. 3. For example, node A's TSI is 2B2C1D, which means that in one GP, there are two packets from A to B, two from A to C, and one from A to D.

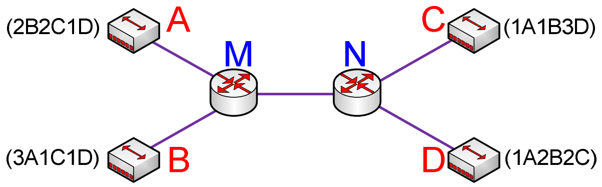


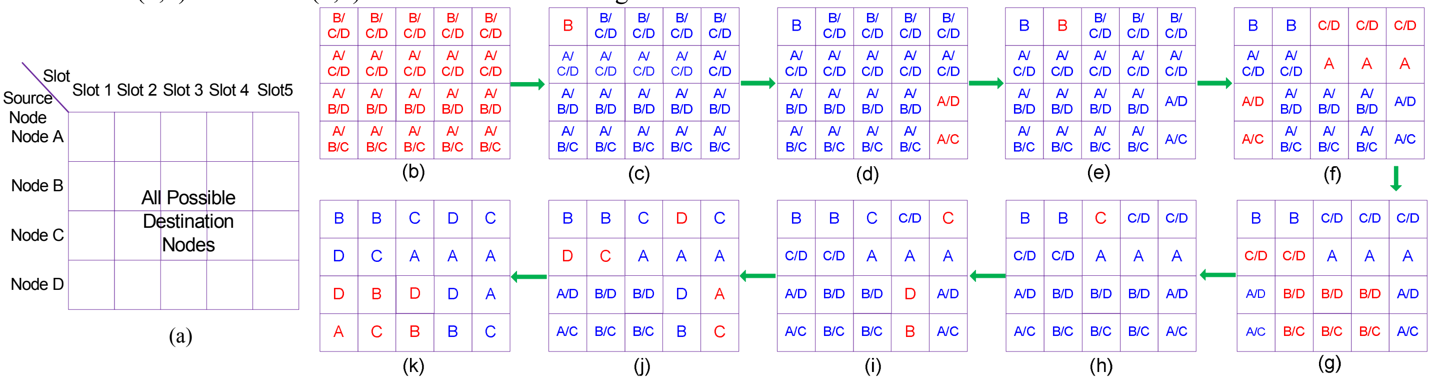
Fig. 3. a simple network topology

A 4×5 table in Fig. 4(a) is used to accomplish the assignment. The four lines denote four edge nodes and five columns denote five slots. Two dimensions element (i, j) is used to denote the cell of line i, column j in the table. For example the cell (1,3) denotes the third slot of node A. In each cell the packet's all possible destination are recorded. For example, since the TSI of node A is 2B2C1D, all of its slots can be assigned to transmit packet to B, C or D, so at the beginning all the cells of line 1 have three options (B/C/D). Similarly other cells can also be assigned with an initial value, as depicted in Fig. 4(b). Our target is to give each cell a unique option. First, we randomly choose one slot and give it a unique value, in Fig. 4(c), the (1,1) cell is chosen and given the value "B". This assignment may affect other cell's options, for example, if the cell (3,5) was assigned with the value "B", the packet in this slot would contend with that in slot (1,1) of the next GP (the packet distribution is cycled) at the core node M (the transmitting time between A and M is one slot, between C and M is two slots, so the packet in C's fifth slot of current GP and the packet in A's first slot of next GP would arrive at node M at the same time). So the option "B" should be discarded from cell (3,5). So is cell (4,5). The modified table is given in Fig. 4(d). After that, we again choose a slot randomly and assign it with a unique value. In Fig. 4(e), the cell (1,2) is chosen and assigned with "B". This assignment can also cause a series of affection to other slots. For instance, the option "B" in cell (3,1) and (4,1) need to be discarded. Since there are only two packets at node A destined to node B, the option "B" in (1,3), (1,4) and (1,5) should also be discarded. This discarding process will continue until no potential contention exists, as depicted in Fig. 4(g). Then a new slot will be assigned and discarding process will be repeated. Finally, in Fig. 4(k), all the slots are assigned with a unique value, so the assignment is finished.

Generally we assume that a slotted OPS network has N nodes, and node 1, 2…N_E are edge nodes, the others are core nodes. Each node at most has K output ports. And there are M slots in one GP. (In NGC, each channel will use the same control mechanism independently, so here we only consider the case with one channel). Let $\Phi_0=\{1,2\ldots N_E\}$ represent the set of all the edge nodes' numbers. Each slot at the edge nodes will be denoted by T_{im}, $i\in\Phi_0$, $0<m\leq M$. The value of T_{im} is some edge node's number, i.e., $T_{im}\in\Phi_0$. If $T_{im}=k$, it means that the packet in the m-th slot at edge node i will be transmitted to edge node k. As depicted in Fig. 4, at the beginning of the SAA, we cannot decide each T_{im}'s value. We only know that $T_{im}\in\Phi_{im}$, where Φ_{im} is a subset of Φ_0 and is decided by edge node i's TSI. Each slot T_{im} also correlated with a contention slot set Γ_{im}, of which the elements have the form of "A-T-B". The value of "A-T-B" is denoted by V(A-T-B):

$$\forall\ \text{A-T-B}\in\Gamma_{im},\ V(\text{A-T-B})>0.$$

And it means that there are only V(A-T-B) elements $k_s\in\Phi_{im}$ (s=1,2,…V(A-T-B)), which satisfies that, the packet in the m-th slot at edge node i and destined to edge node k_s will contend with the packet in the T-th slot at edge node A and destined to edge node B at some node of the network. Each edge node i correlates with a packet number set $\Lambda_i=\{Q_s,\ s\in\Phi_0\}$, where Q_s is the number of packets that destined to edge node s and have not get an assigned slot. During the assignment process, both Γ_{im} and Λ_i need to be timely updated. The whole algorithm can be regarded as to discard elements from each Φ_{im} with the help of Γ_{im} and Λ_i. Finally all Φ_{im} only have one element, so each T_{im}'s value can be decided, the algorithm concludes.



Source Node \ Slot	Slot 1	Slot 2	Slot 3	Slot 4	Slot5
Node A					
Node B			All Possible Destination Nodes		
Node C					
Node D					

(a)

B/C/D	B/C/D	B/C/D	B/C/D	B/C/D
A/C/D	A/C/D	A/C/D	A/C/D	A/C/D
A/B/D	A/B/D	A/B/D	A/B/D	A/B/D
A/B/C	A/B/C	A/B/C	A/B/C	A/B/C

(b)

B	B/C/D	B/C/D	B/C/D	B/C/D
A/C/D	A/C/D	A/C/D	A/C/D	A/C/D
A/B/D	A/B/D	A/B/D	A/B/D	A/B/D
A/B/C	A/B/C	A/B/C	A/B/C	A/B/C

(c)

B	B/C/D	B/C/D	B/C/D	B/C/D
A/C/D	A/C/D	A/C/D	A/C/D	A/C/D
A/B/D	A/B/D	A/B/D	A/B/D	A/D
A/B/C	A/B/C	A/B/C	A/B/C	A/C

(d)

B	B	B/C/D	B/C/D	B/C/D
A/C/D	A/C/D	A/C/D	A/C/D	A/C/D
A/B/D	A/B/D	A/B/D	A/B/D	A/D
A/B/C	A/B/C	A/B/C	A/B/C	A/C

(e)

B	B	C/D	C/D	C/D
A/C/D	A/C/D	A	A	A
A/D	A/B/D	A/B/D	A/B/D	A/D
A/C	A/B/C	A/B/C	A/B/C	A/C

(f)

B	B	C/D	C/D	C/D
C/D	C/D	A	A	A
A/D	B/D	B/D	B/D	A/D
A/C	B/C	B/C	B/C	A/C

(g)

B	B	C	C/D	C/D
C/D	C/D	A	A	A
A/D	B/D	B/D	B/D	A/D
A/C	B/C	B/C	B/C	A/C

(h)

B	B	C	C/D	C
C/D	C/D	A	A	A
A/D	B/D	B/D	D	A/D
A/C	B/C	B/C	B	A/C

(i)

B	B	C	D	C
D	C	A	A	A
A/D	B/D	B/D	D	A
A/C	B/C	B/C	B	C

(j)

B	B	C	D	C
D	C	A	A	A
D	B	D	D	A
A	C	B	B	C

(k)

🟥 Just updated in the last step 🟦 Keep unchanged in the last step

Fig. 4. The procedure of assigning slots in SAA

B. A Necessary and Sufficient Condition

In the example given above, if the TSI is not as appropriate as that in Fig. 3, some slots may have no option after some discarding process. So it is important to distinguish whether a given TSI can finally lead to a no contention assignment before assigning. Now we will give a necessary and sufficient condition for this discrimination.

To guarantee no contention occurred in the network, the condition **S** is necessary and enough:

$$S_{ik} \le M \quad i=1, 2... N \;\; k=1, 2... K$$

Where S_{ik} is the amount of packets that output from the k-th port of node i in one GP.

The necessity is obviously, since there are only M slots in one GP, if more than M packets need to be output, there must be two packets contend the same slot. Now we will give a brief proof for the sufficiency of condition **S**.

We define that a network is in the harmonic state if $\forall i \in \Phi_0$, the following three conditions are satisfied:

i) $\forall\, \Phi_{im},\ |\Phi_{im}| > 0$

ii) $\forall Q_s \in \Lambda_i$, if $Q_s = 0$, $\forall 0 < m \le M$, if $|\Phi_{im}| > 1$, $s \notin \Phi_{im}$

iii) $\forall$ A-T-B$\in \Gamma_{im}$, $V(\text{A-T-B}) < |\Phi_{im}|$

For instance, in Fig. 4, (b), (d), (g) and (k) are in the harmonic states.

If either ii) or iii) is not satisfied, some elements in some Φ_{kl}, ($k \in \Phi_0$, $0<l\le$M) need to be discarded. If ii) is not satisfied, i.e.,

$$\exists Q_s \in \Lambda_i,\ Q_s = 0,\ \exists 0 < m \le M,\ |\Phi_{im}| > 1,\ s \in \Phi_{im},$$

then Q_s=0 means that all the packets destined to edge node s has been assigned with slots, so s should be discarded from Φ_{im}. If iii) is not satisfied, i.e.,

$$\exists \text{A-T-B} \in \Gamma_{im},\ V(\text{A-T-B}) = |\Phi_{im}|.$$

This means that the packet in the T-th slot at edge node A and destined to edge node B will contend with the packet in the m-th slot at edge node i and destined to any possible target edge node. Then in order to avoid this potential contention, B should be discarded from Φ_{AT}. We will call the two kinds of discarding above as reasonable discarding in following.

We now prove two propositions:

(1) If the network is in a harmonic state, then $\forall \Phi_{im},\ |\Phi_{im}| > 1$, we arbitrarily choose $k \in \Phi_{im}$ and let $T_{im}=k$, after a series of reasonable discarding, the network can return to another harmonic state.

(2) If the network's TSI satisfies with condition **S**, then after initializing each Φ_{im} and some reasonable discarding, the network can be set in a harmonic state.

For (1), we adopt the reduction of absurdity. In fact, if $\exists \Phi_{im},\ |\Phi_{im}| > 1$, after we choose some $k \in \Phi_{im}$, let $T_{im}=k$ and implement a series of reasonable discarding, the network cannot return to the harmonic state, there must $\exists\, \Phi_{jl},\ |\Phi_{jl}| = 0$. And this exactly means that at the beginning when the network is still in the harmonic state, the element i-m-k should belongs to Γ_{jl} and V(i-m-k)=$|\Phi_{jl}|$, which contradicts to the harmonic state's condition iii). So an absurdity is get and (1) is proved.

For (2), each Φ_{im} is only decided by Λ_i when initializing, so at the beginning we have $\Phi_{i1}= \Phi_{i2}=\ldots=\Phi_{i\text{M}}$ (as depicted in Fig. 4(b)). Furthermore, since the traffic is cycled, Γ_{im} has a cyclic symmetry, i.e., if A-T-B$\in \Gamma_{im}$ then $\forall 0 < l \le M$,

$$\text{A-(T}+l\text{)-B} \in \Gamma_{i(m+l)} \text{ and } V(\text{A-T-B}) = V(\text{A-(T}+l\text{)-B}),$$

where "T+l" and "m+l" should be performed modulus operation with M when they are larger than M. So if some element s in Φ_{im} is reasonably discarded when condition ii) or iii) is not satisfied, due to Λ_i's independence with m and Γ_{im}'s cyclic symmetry, s should also be discarded from any other Φ_{il} ($0<l\le$M). So $\Phi_{i1}= \Phi_{i2}=\ldots=\Phi_{i\text{M}}$ will be always right during the reasonable discarding process. If the network cannot be set in a harmonic state, it means that after some reasonable discarding process, $\exists \Phi_{jl},\ |\Phi_{jl}| = 0$, then $\forall 0 < k \le M,\ |\Phi_{jk}| = 0$. Suppose Torrent A originates from node j, then even one packet of Torrent A cannot be transmitted by node j, and this means that at some port of some node in Torrent A's route, there will be more than M packets need to be transmitted in one GP, which contradicts with condition **S**. So the suppose is not right and (2) gets proved.

Based on (1) and (2), so long as condition **S** is satisfied in the network, we can assign T_{im} one by one and maintain the network in the harmonic state by reasonable discarding, and finally a no contention assignment can be get. So the sufficiency of condition **S** is also proved.

C. Computation Model

We adopt the asynchronous parallel random access machine (APRAM) model to process SAA and TAA. In this parallel computation model, a main processor P_M and N_E sub processors P_i ($0<i\le N_E$) are working simultaneously. Each P_i is corresponding with an edge node in the network and they can be activated (i.e., awaked from the idle state to process some programs) by each other or by P_M. Each P_i has a local RAM, which stores T_{im}, Φ_{im}, Γ_{im} ($0<m\le$M) and Λ_i.

D. Slot Assignment Algorithm

With all the preparations above, the outline of SAA can be given in Fig. 5. First, the TSI at edge nodes is input into APRAM. Then P_M calculates and modifies the TSI to guarantee that at each node condition **S** can be satisfied. After that, P_M will randomly choose a T_{im} and assign it a value. Then P_M will activate P_i to update its Λ_i, Γ_{in}, Φ_{in} ($0<n\le$M). P_i may also activate other sub processors when updating its Γ_{in}. When all activated sub processors finish their updating task, P_M again assigns a slot randomly and activate the corresponding sub processor. Finally all T_{im} are assigned and these results can be output to the edge nodes.

E. Transient Assignment Algorithm

TAA is more or less the same with SAA. There are two main differences: first, since TAA is used between the previous and the next 10 GPs, when initializing Φ_{im}, Γ_{im} and Λ_i in current GP, the slots' assignment of the previous GP and the next GP are all known, so some discarding process should also be introduced due to those assignment's affection; secondly, the network's initial state is not a harmonic state due to the affection of the assignment of the previous and next GP, during the reasonable discarding process some Φ_{im}'s elements may all be discarded. In this case, some torrent's TW will be reduced to guarantee no contention occurs. The detail of TAA is given in Fig. 6.

```
Slot Assignment Algorithm
Input:   The TSI at each edge node
Output: An assignment for all the slots with no contention in the network
Begin
    $P_M$:  Receive TSI
            Modify TSI to make it satisfy condition S
           Initialize $\Phi_{im}$, $\Gamma_{im}$, $\Lambda_i$ (0<i≤N, 0<m≤M),   $B_i$←1 (0<i≤$N_E$),  Num←0
            while Num<$N_E$·M do
              if $\prod_{i=1}^{N_E} B_i$==1  then //the network is in the harmonic state
                randomly choose i, m and  k∈$\Phi_{im}$, $T_{im}$←k, $\Phi_{im}$←{k}
                $B_i$←0, Activate ($P_i$, k, m), Num←Num+1;
              endif
            end while
      Output $T_{im}$ (0<i≤$N_E$, 0<m≤M).
      for all activated $P_i$ (0<i≤$N_E$) do
       if activated by $P_M$
         (k, m)=ActivateParameter, update $\Gamma_{im}$, $Q_k$←$Q_k$-1 //update $\Lambda_i$
         if $Q_k$==0 then for n=1 to M
             if |$\Phi_{in}$|>1 then delete k from $\Phi_{in}$, update $\Gamma_{in}$ endif
             end for
         endif
        end if
       if activated by $P_{sub}$
         A-T-B=ActivateParameter   delete B from $\Phi_{iT}$, update $\Gamma_{iT}$
       endif
       for n=1 to M and all A-T-B∈$\Gamma_{in}$
           if  V(A-T-B)==0 then delete A-T-B from $\Gamma_{in}$ endif
           if  V(A-T-B)==|$\Phi_{in}$| then
             delete A-T-B from $\Gamma_{in}$, $B_A$←0, Activate($P_A$, A-T-B)
           endif
       endfor
       $B_i$←1
      end for all
End
```

Fig. 5. Pseudo code for the Slot Assignment Algorithm

```
Transient Assignment Algorithm
Input:   The slot assignment in the previous and next GP
Output: The slot assignment in current GP
Begin
    $P_M$:  Initialize $TP_{im}$, $TN_{im}$ (0<i≤$N_E$, 0<m≤M)
         // TP and TN are the slot assignment in the previous and next GP
           Initialize $\Phi_{im}$, $\Gamma_{im}$, $\Lambda_i$ based on $TP_{im}$, $TN_{im}$ (0<i≤$N_E$, 0<m≤M)
           $B_i$←1 (0<i≤$N_E$),  Num←0
           while Num<$N_E$·M do
             if  $\prod_{i=1}^{N_E} B_i$==1   then
                randomly choose i, m and  k∈$\Phi_{im}$, $T_{im}$←k, $\Phi_{im}$←{k}
                $B_i$←0, Activate ($P_i$, k, m), Num←Num+1;  endif
           end while
      Output $T_{im}$ (0<i≤N, 0<m≤M).
      for all activated $P_i$ (0<i≤$N_E$) do
       if activated by $P_M$
         (k, m)=ActivateParameter, update $\Gamma_{im}$, $Q_k$←$Q_k$-1
         if $Q_k$==0 then for n=1 to M
             if |$\Phi_{in}$|>1 then delete k from $\Phi_{in}$, update $\Gamma_{in}$ endif  end for
       endif end if
       if activated by $P_{sub}$
         A-T-B=ActivateParameter   delete B from $\Phi_{iT}$,
         if |$\Phi_{iT}$|==0 then  Num←Num+1 endif   update $\Gamma_{iT}$
       endif
       for n=1 to M and all A-T-B∈$\Gamma_{in}$
           if  V(A-T-B)==0 then delete A-T-B from $\Gamma_{in}$ endif
           if  V(A-T-B)==|$\Phi_{in}$| then
             delete A-T-B from $\Gamma_{in}$, $B_A$←0, Activate($P_A$, A-T-B) endif
       endfor
       $B_i$←1
      end for all
End
```

Fig. 6. Pseudo code for the Transient Assignment Algorithm

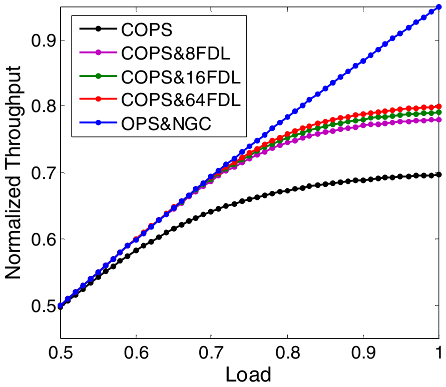


Fig. 7. Throughput vs. Load plot

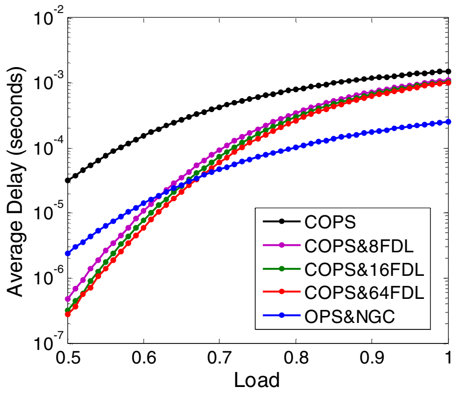


Fig. 8. Average Delay vs. Load plot

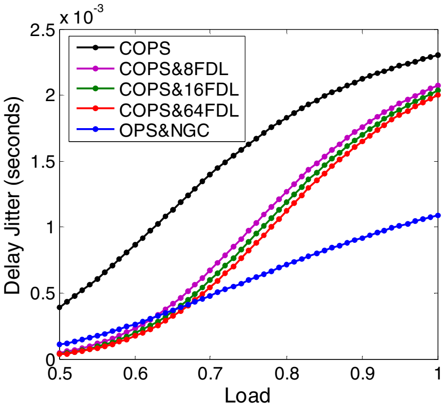


Fig. 9. Delay Jitter vs. Load plot

IV. Simulation Results

In order to compare the performance of NGC with conventional OPS, a simulation model is constructed. We use a network with complete symmetry topology (In a network with complete symmetry topology, all links' status is equal, so there is no such link that acts as an artery and may limit the network's throughput due to its limited bandwidth. The topology of real OPS networks such as 14-node NSFNET and 16-node Pan-European network are all nearly completely symmetric). Each node has 8 input/output ports, and each port has 16 channels. Based on the real Internet situation [10], the route between each two edge nodes is assumed to be 12 hops and 1000km long on average, and the GP in NGC is 5 milliseconds. Two cases will be considered: conventional OPS network without NGC (COPS) and OPS network with NGC (OPS&NGC). In COPS, FDL-based feed-forward buffering structure is used at each core node and the first in, first out (FIFO) buffering strategy is adopted; the FDL number at each core node is changed from 0,8,16 to 64.

In Fig. 7, the Normalized Throughput with increased Load in each case is compared. For OPS&NGC, a nearly ideal throughput can be get since all the potential contentions can be eliminated by SAA. TW Reducing in TAA is the main reason for the deviation from ideal case. While in COPS, due to the low buffering efficiency of FDL, the throughput increases slowly as the number of FDL buffers rises. Even with large number of FDLs such as 64, the throughput is still much lower than that in OPS&NGC at high load.

In Fig. 8 and Fig. 9 the average end to end delay and delay jitter in five cases are compared. Here the transporting time has been excluded and we only consider the delay induced by TW Reducing in OPS&NGC or buffering and retransmitting for lost packet in COPS. It can be seen that both average delay and delay jitter performance in OPS&NGC is the best when load is above 0.7. At low load, since the delay and jitter is much shorter, a little worse performance of OPS&NGC does not make serious effect on the whole network.

Above all, with NGC, we can not only overcome the problem of unaccepted performance at high load in conventional OPS due to the lack of appropriate optical buffers, but also get a nearly ideal performance in throughput, so we think NGC is a promising control mechanism for future OPS networks.

V. Further discussion

Right now our scheme only applies to slotted OPS. Although slotted OPS needs synchronization, this synchronization only needs to be accurate to the order of the packet length, if a variable delay line is equipped at each input port of the core nodes. The delay time of each delay line does not need to vary frequently since the physical link's length is seldom changed. So a commercial manual variable delay line is enough. Besides, the Multi-Wavelength OPS proposed recently [11] [12] also makes it possible for slotted OPS to support variable packet length. So we think slotted OPS with NGC nowadays is a good substitute for asynchronous OPS.

The robustness of NGC can be improved by equipping another core node as a backup center node, which can be put into work when the computation equipment at the main center node is shutdown. The original center can forward the control information to the reserved one before the edge nodes know that center is changed, so there is little influence on the slot assignment during the fault recovering time.

The scalability of NGC OPS can be implemented by adopting a hierarchical network architecture, in which the network can be divided into many autonomous systems (AS) from top to bottom. Each AS has its own center node and different network levels use different light wavelength for communication. Then the control within each AS and between different AS can be carried on independently, and NGC can be straightly applied on each hierarchical level.

Finally we give a simply analysis about the time complexity of SAA and TAA. Since we adopted the APRAM computation model, the computing task has been allocated to each sub processor. The randomly assignment procedure in P_M will be at most processed for $N_E \cdot M$ times, each activated P_i, at most needs to update M Γ_{im} and Φ_{im}, so the upper limit of time complexity of SAA or TAA is $O(N_E \cdot M^2)$. In real OPS network, N_E~100, GP~1ms, one slot~0.1us, so M~1e4, $O(N \cdot M^2)$~1e10, The computation task should be finished within one GP, so the processing speed should be at the order of 1e13 times/sec, this can be achieved by a middle powerful super electronic computer nowadays such as IBM's ASC Purple - eServer [13] and may be much easier to realize in the near future.

VI. Conclusion

By utilizing the NGC mechanism and SAA/TAA, we have realized a slotted OPS network without the need of buffering at core node. This network can give a much better throughput, end to end delay and delay jitter performance compared with the conventional FDL-based OPS at high load. Our future work will be engaged in optimizing TAA and improving the robustness and scalability of NGC-based OPS network.

References

[1] S. J. B. Yoo, "Optical packet and burst switching technologies for the future photonic internet", *J. Lightw. Technol.*, vol. 24, no. 12, pp. 4468–4492, Dec. 2006.

[2] Y. Yeo, J. Yu, and G. Chang, "Performance of DPSK and NRZ-OOK signals in a novel folded-path optical packet switch Buffer", *Proc. of Opt. Fiber Commun. (OFC)*, paper OWK3, Anaheim, USA, 2005.

[3] N. Chi, Z. Wang, S. Yu, "A large variable delay, fast reconfigurable optical buffer based on multi-loop configuration and an optical crosspoint switch matrix", *Proc. of Opt. Fiber Commun. (OFC)*, paper OFO7, Anaheim, USA, 2006.

[4] H. Yang and S. J. B. Yoo, "All-optical variable buffering strategies and switch fabric architectures for future all-optical data routers", *J. Lightw. Technol.*, vol.23, no.10, pp. 3321–3330, Oct. 2005.

[5] J. Yang, A. Karalar, S. Djordjevic, N. Fontaine, C. Yang , W. Chen, S.Chu, B. Little, and S.J. B. Yoo, "Variable slow light buffers in all-optical packet switching routers", *Proc. of Opt. Fiber Commun. (OFC)*, paper OTuF2, San Diego, USA, 2008.

[6] F. Xue, and S. J. B. Yoo, "TCP-aware active congestion control in optical packet-switched networks", *Proc. of Opt. Fiber Commun. (OFC)*, paper MF108, Atlanta, USA, 2003.

[7] Z. Lu, D. K. Hunter, and I. D. Henning, "Congestion control scheme in optical packet switched networks", *Proc. of Opt. Fiber Commun. (OFC)*, paper OThM4, Anaheim, USA, 2006.

[8] N. Beheshti, Y. Ganjali, R. Rajaduray, D. Blumenthal, N. McKeown, "Buffer sizing in all-optical packet switches", *Proc. of Opt. Fiber Commun (OFC)*, paper OThF8, Anaheim, USA. 2006.

[9] J. Jue, "An algorithm for loopless deflection in photonic packet-switched networks" *Proc. of IEEE International Conference on Communications (ICC)*, vol. 5. pp 2776-2780, New York, USA, 2002.

[10] P. V. Mieghem, *Performance Analysis of Communications Networks and Systems*, Cambridge University Press, 2006 (ISBN-13: 9780521855150 | ISBN-10: 0521855152), pp 358.

[11] H. Onaka, Y. Aoki, K. Sone, G. Nakagawa, Y. Kai, S. Yoshida, Y. Takita, K. Morito, S. Tanaka, and S. Kinoshita, "WDM optical packet interconnection using multi-gate SOA switch architecture for peta-flops ultra-high-performance computing systems," *Proc. of 34th Europe Conference on Optical Communication (ECOC)*, paper Tu4.6.6, Cannes, France, 2006.

[12] H. Furukawa, N. Wada, H. Harai, M. Naruse, H. Otsuki, M.Katsumoto, T. Miyazaki, K. Ikezawa, A. Toyama, N. Itou, H.Shimizu, H. Fujinuma, H. Iiduka, G. Cincotti, and K.Kitayama, "Field Trial of IP over 160 Gbit/s Colored-Optical-Packet Switching Network with Transient-Response-Suppressed EDFA and 320 Gbit/s through put Optical Packet Switch Demonstrator," *Proc. of Opt. Fiber Commun. (OFC)*, paper PDP4, Anaheim, USA, 2007.

[13] www.top500.org/system/8128