Performance Evaluation of Parallel Waiting Recovery from Wavelength Reservation Failure in Various Topologies of WDM Networks

Naoto Taguchi and Hiroaki Higaki
Department of Computers and Systems Engineering
Tokyo Denki University
E-mail: {taguchi,hig}@higlab.net

In the path multiplexing of wavelength reservation in WDM all-optical networks, the same wavelength is required to be reserved in all the optical links along a message transmission router from a source to a destination edge routers. Especially in a large scale WDM network, message transmission routers are likely to be longer and reservation ratio gets lower. It causes longer reservation time and performance of the WDM network, e.g. end-to-end throughput also gets lower. This paper proposes a parallel waiting recovery method for reservation failure in an optical link and CFW reservation protocol is designed. Simulation experiments show that CFW protocol requires shorter avarage reservation time than the conventional CFD and CFH protocols in various network topologies such as tree-type, mesh-type and BA model which is one of complex network model well modeling real computer networks.

多様なトポロジのWDMネットワークにおけるパラレルウェイト型波長予約プロトコルの性能評価

東京電機大学 理工学部 情報システム工学専攻
田口 直人 桧垣 博章
E-mail: {taguchi, hig}@higlab.net

WDM全光ネットワークにおけるパス多重化による波長予約を行なう場合、送信元エッジルータから送信先エッジルータまでのすべての光リンクで同一の波長を予約することが必要であり、特にネットワーク規模の拡大により平均経路長が大きくなると波長予約成功率が低下する。経路全体を通した波長予約成功率の向上、波長予約成功時の予約時間と波長予約失敗時の回復時間の縮小が平均予約時間の短縮を実現する。本論文では、単一光リンクの波長予約失敗にはタイマによる再試行を導入しつつ、他の光リンクの波長予約を並行に試みるパラレルウェイト型プロトコルを提案し、ツリー型、メッシュ型、BAモデル型のネットワークにおける性能評価を行なう。

1 Introduction

Recently, for supporting network applications that require wide bandwidth and high throughput, development of all-optical networks are required. Here, WDM (Wavelength Division Multiplexing) is one of the most important technologies to realize all-optical networks. An all-optical network consists of a set of edge routers E_i which are gateways between the all-optical network and other networks and a set of core routers C_i which are inner routers connected not to other networks but to each other and to edge routers. In each optical communication link between an edge router and a core router and between two core routers, messages are transmitted on optical signals with multiple different wavelengths simultaneously. There are two multiplexing methods, link multiplexing and pass multiplexing, for WDM networks. In the path multiplexing, an optical signal of the same wavelength is transmitted through all-optical communication links along a message transmission route. On the other hand, in the link multiplexing, an optical signal with different wavelength may be used for message transmission in each optical communication link along the message transmission route. If wavelengths of an input optical signal and an output one are different in a core router, wavelength conversion is required. Here, additional transmission delay is required for wavelength conversion and configuration of wavelength convector module in core routers requires higher cost. In the path multiplexing, there are no additional communication overhead caused by wavelength conversion. However, only when there is a common wavelength which has not yet reserved for another message transmission in every optical communication link along the message transmission route, a common wavelength is reserved in the optical communication links and application messages are transmitted since multiple message transmissions are not allowed to share the same wavelength in an optical communication link.

In order to realize wavelength reservations with the path multiplexing in a WDM all-optical network, some centralized wavelength reservation management methods in which all reservation information is stored in a centralized sever computer have been proposed [2,3,5,7]. However, in a large-scale all-optical network, many message transmission requests are issued and many wavelength reservation and release requests are issued to the centralized server. Hence, the sever computer may be overloaded and networks around the server computer may be congested. In order to solve these problems, distributed wavelength reservation manage-

ment methods have also been proposed [4, 6, 8, 10]. Here, reservation information of an optical communication link is managed in an optical router from which the link is issued. For reservation and release of a wavelength in an optical communication link, a request is issued to the optical router managing it. Since reservation information is managed distributedly, a wavelength reservation is not always successful. In a conventional wavelength reservation protocol, a reservation request message *Req* is forwarded from an edge router E_s to another one E_d through a sequence of core routers C_i in a forward reservation method and in an opposite direction in a backward reservation for message transmission from E_s to E_d. On receipt of a *Req* message, each core router C_i reserves one of free wavelengths which have not yet been reserved by another massage transmission. However, it is possible that C_i has no free wavelength which are commonly free in all optical communication links from E_s to E_d. This is called a wavelength reservation failure. Two recovery methods have been proposed. Dropping recovery releases all provisionally reserved wavelengths at the instance when reservation failure is detected. It requires only short period of time for reservation but its probability of successful reservation is low. Holding recovery waits for release of candidate wavelengths for a certain period of time by using a timer before all provisionally reserved wavelengths are released. It achieves higher probability of successful reservation but its time overhead is high especially in a sequential waiting discussed in Section 3. This paper proposes a novel recovery method, parallel waiting to improve the performance of recovery from reservation failure. It takes a tradeoff between reservation time and successful reservation probability.

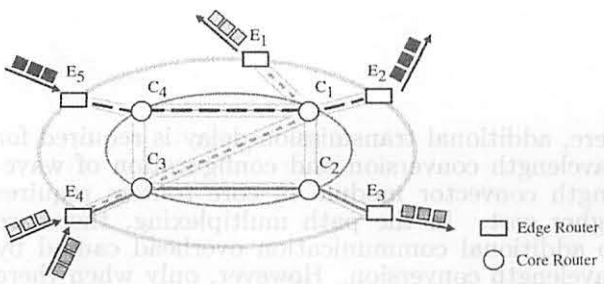


Figure 1: All-optical WDM Network.

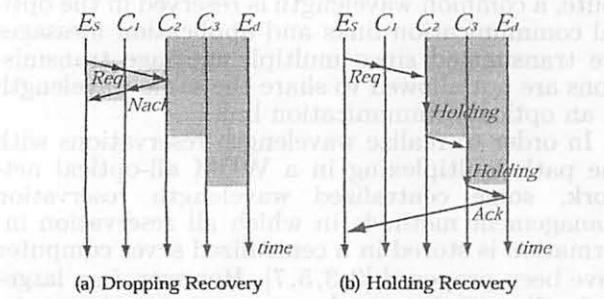


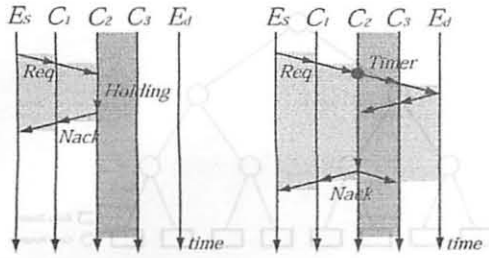
Figure 2: Dropping and Holding Recoveries.

2 Parallel Waiting

In the dropping recovery, if a common wavelength is not reserved in all optical communication links $\langle L_0, \ldots, L_l \rangle$ along a message transmission route from an edge router E_s to another one E_d at the instance when each core router C_i receives a reservation request message *Req*, provisionally reserved wavelengths are soon released in order not to interrupt other reservations sharing some of the optical communication links. However, in a WDM network in which wavelengths are dynamically reserved each time a message transmission is required to be started and released each time a message transmission is finished, it is expected that one of the wavelengths included in *Req.wavelength* piggybacked to the received *Req* message is released for a certain period even if there is no intersection between *Req.wavelength* and a set F_i of wavelengths which have not yet reserved in L_i.

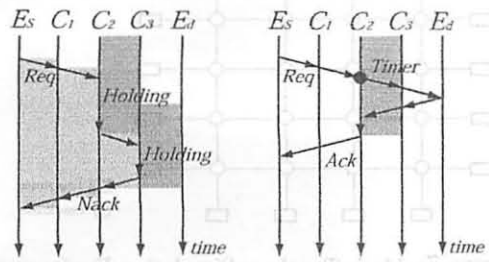
In [4], the holding recovery achieves higher successful probability of wavelength reservation than the dropping one. In the holding recovery, if C_i waits for release of a wavelength in L_i, C_i does not forward a *Req* message to C_{i+1} in the forward reservation and C_{i-1} in the backward one, respectively. That is, wavelengths are never reserved in $\langle L_{i+1}, \ldots, L_l \rangle$ in the forward reservation and in $\langle L_{i-1}, \ldots, L_0 \rangle$ in the backward one, respectively. Such design is based on a consideration that if a *Req* message is forwarded, additional wavelengths are reserved in $\langle L_{i+1}, \ldots, L_l \rangle$ in the forward reservation and $\langle L_{i-1}, \ldots, L_l \rangle$ in the backward one, respectively. If a timer T_i in C_i expires, since these provisionally reserved wavelengths are released before application messages are transmitted through them and these reservations only interrupt ones of other message transmission, the additional reservation causes lower successful probability of wavelength reservations in a WDM network as shown in Figure 3. However, since a *Req* message is not forwarded while a certain core router waits for release of a wavelength in an optical communication link issued from the core router, time duration required for a wavelength reservation is extended in both successful and failed cases. In addition, the provisionally reserved wavelengths interrupt other reservation for longer time and probability of successful wavelength reservations gets lower due to the sequential waiting problem as show in Figures 4. For example in the forward reservation as shown in Figures 4, suppose the case that a core router C_i waits for release of a wavelength λ in L_i after λ is reserved in $\langle L_0, \ldots, L_{i-1} \rangle$. Since a *Req* message is not forwarded through $\langle L_i, \ldots, L_l \rangle$ in the holding recovery, $C_{i+1}, \ldots, C_l$ do not reserve λ even if λ has not yet reserved in $\langle L_i, \ldots, L_l \rangle$. If λ is reserved by a core router C_j $(i+1 \leq j \leq l)$, after λ is released in L_i and C_i reserves λ in L_i for message transmission from E_s to E_d, forwarding of the *Req* message, i.e. reservation procedure for λ along $\langle L_0, \ldots, L_l \rangle$, is also interrupted in C_j. Due to such sequential holdings in multiple core routers, reservation procedure requires longer time duration in both successful and failed wavelength reservations.

Thus, this paper proposes a novel recovery method, a parallel waiting recovery, for a core router detecting reservation failure. Here, when a core router C_i detects reservation failure, i.e. all reserving wavelengths have been reserved for other message transmissions in an optical communication



(a) Holding Recovery (b) Waiting Recovery

Figure 3: Additional Provisional Reservation.



(a) Holding Recovery (b) Waiting Recovery

Figure 4: Shorter Reservation Time Period.

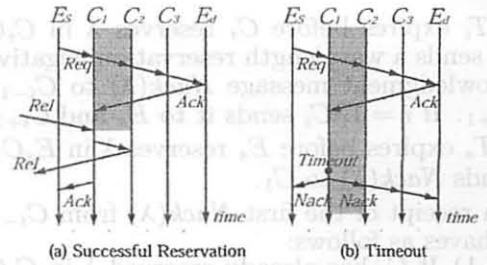
link L_i, C_i sets a timer T_i, waits for release of one of the reserving wavelengths and reserves a wavelength when it is released same as the holding recovery. In addition, C_i sends a *Req* message to C_{i+1} not after C_i reserves the wavelength but at the same time of timer setting. That is, waiting for release in C_i and forwarding of the *Req* message are not sequential but parallel. In addition, multiple core routers may wait for release of wavelength reserved for other message transmissions simultaneously. In case that a core router waits for release and forwards a *Req* message, it is impossible to decide which wavelength in *Req.wavelength* will be released later. Hence *Req.wavelength* piggybacked to the received *Req* message is also piggybacked to the forwarded *Req* message. By applying the proposed parallel waiting recovery, since forwarding of a *Req* message is not blocked as in the dropping recovery, shorter reservation time is achieved and since (parallel) waiting for release provides additional opportunities to reserve wavelength, higher successful wavelength reservation probability is expected.

3 CFW Protocol

We design CFW (Conservative Forward Waiting) protocol for wavelength reservation along a message transmission route from an edge router E_s to anther one E_d thorough a sequence of core routers $C_1, \ldots, C_n$. Here, reservation procedure progresses in a forward direction, i.e. from E_s to E_d without probing as in the backward reservation, each core router and E_s reserves only one wavelength at a time and the proposed parallel waiting recovery is adopted.

[CFW Protocol]

1) An edge router E_s checks whether there are any wavelengths which have not yet reserved in a communication link E_sC_1 for other communications.
   1–1) If a wavelength λ has not yet reserved, E_s reserves λ in E_sC_1.
   1–2) Otherwise, i,e, if all possible wavelengths in E_sC_1 have been already reserved for other communications, E_s sets a timer T_s and waits for release of one of the reserved wavelength λ in E_sC_1. If λ is released before T_s expires, E_s reserves λ in E_sC_1.
2) E_s sends a wavelength reservation request message $Req(\lambda)$ to a next hop core router C_1.
3) On receipt of $Req(\lambda)$, a core router C_i checks whether λ has been reserved for another communication in a communication link C_iC_{i+1}.
   3–1) If λ has not yet reserved, C_i reserves λ in C_iC_{i+1}.
   3–2) Otherwise, i.e. if λ has already reserved in C_iC_{i+1} for another communication, C_i sets a timer T_i and waits for release of λ in C_iC_{i+1}. If λ is released before T_i expires, C_i reserves λ in C_iC_{i+1}.
4) C_i sends $Req(\lambda)$ to C_{i+1}. If $i = n$, C_i sends $Req(\lambda)$ to an edge router E_d.
5) On receipt of $Req(\lambda)$, E_d sends back a wavelength reservation acknowledgment message $Ack(\lambda)$ to C_n.
6) On receipt of $Ack(\lambda)$, C_i checks whether it has already reserved λ in C_iC_{i+1}.
   6–1) If C_i has already reserved λ in C_iC_{i+1} according to 3–1) or 3–2), C_i sends back $Ack(\lambda)$ to C_{i-1}. If $i = 1$, C_i sends back $Ack(\lambda)$ to E_s.
   6–2) Otherwise, i.e. if C_i has not yet reserved λ in C_iC_{i+1} and T_i has not yet expired, C_i waits for release of λ in C_iC_{i+1}. If λ is released before T_i expires, C_i reserves λ in C_iC_{i+1} and sends back $Ack(\lambda)$ to C_{i-1}. If $i = 1$, C_i sends back $Ack(\lambda)$ to E_s.
7) On receipt of $Ack(\lambda)$, E_s checks whether it has already reserved λ in E_sC_1.
   7–1) If E_s has already reserved λ in E_sC_1 according to 1–1) or 1–2), E_s starts transmission of application messages along a transmission route $\|E_s, C_1, \ldots, C_n, E_d\|$ by using optical signal of λ.
   7–2) Otherwise, i.e. if C_i has not yet reserved λ in E_sC_1 and T_s has not yet expired, E_s waits for release of λ in E_sC_1. If λ is released before T_s expires, E_s reserves λ in E_sC_1 and starts transmission of application messages along a transmission route $\|E_s, C_1, \ldots, C_n, E_d\|$ by using optical signal of λ.



(a) Successful Reservation (b) Timeout

Figure 5: CFW Protocol.

8) If T_i expires before C_i reserves λ in C_iC_{i+1}, C_i sends a wavelength reservation negative acknowledgment message $Nack(\lambda)$ to C_{i-1} and C_{i+1}. If $i = 1$, C_i sends it to E_s and C_{i+1}.
9) If T_s expires before E_s reserves λ in E_sC_1, E_s sends $Nack(\lambda)$ to C_1.
10) On receipt of the first $Nack(\lambda)$ from C_{i-1}, C_i behaves as follows:
    10–1) If C_i has already reserved λ in C_iC_{i+1}, C_i releases λ and sends $Nack(\lambda)$ to C_{i+1}. If $i = n$, C_i does not send $Nack(\lambda)$.
    10–2) Otherwise, i.e. if C_i still waits for release of λ, C_i resets T_i and sends $Nack(\lambda)$ to C_{i+1}. If $i = n$, C_i does not send $Nack(\lambda)$.
11) On receipt of the first $Nack(\lambda)$ from C_{i+1}, C_i behaves as follows:
    11–1) If C_i has already reserved λ in C_iC_{i+1}, C_i releases λ and sends $Nack(\lambda)$ to C_{i+1}. If $i = 1$, C_i sends $Nack(\lambda)$ to E_s.
    11–1) Otherwise, i.e. if C_i still waits for release of λ, C_i resets T_i and sends $Nack(\lambda)$ to C_{i-1}. If $i = 1$, C_i sends $Nack(\lambda)$ to E_s.
12) On receipt of $Nack(\lambda)$ from C_1, E_s behaves as follows:
    12–1) If E_s has already reserved λ in E_sC_1, E_s releases λ.
    12–2) Otherwise, i.e. if E_s still waits for release of λ, E_s resets T_s. □

4 Evaluation

In this section, we evaluate performance of our proposed parallel waiting recovery from reservation failure in WDM all optical networks. Here, required time duration for wavelength reservation including time for recovery from reservation failure and for re-reservation until finally a wavelength is reserved along a predetermined message transmission route from a source edge router to a destination one is evaluated. A base WDM wavelength reservation protocol is a forward reservation protocol and for recovery from failure our proposed parallel waiting recovery, dropping recovery and holding recovery are applied and achieved performance is compared.

Two types of network topologies are assumed; tree-type and mesh-type. In the former, core routers are configured in a binary tree and edge routers are connected to the binary tree as leaf nodes (Figure 6). With depth N of the tree, there are $2^N - 1$ core routers and 2^N edge routers. On the other hand, in the latter, core routers are configured in a $N \times N$ square and $4N$ edge routers are connected to the square along edges of the square (Figure 7).

In addition to the topology assumptions, the following simulation parameters are assumed:

- Each optical links has 16 different wavelengths.
- Timeout interval for release of wavelength reservation in the holding and parallel waiting recovery is 50 msec.
- Message transmission requests arrives according to Poisson process with an average interval of 100–200 msec.

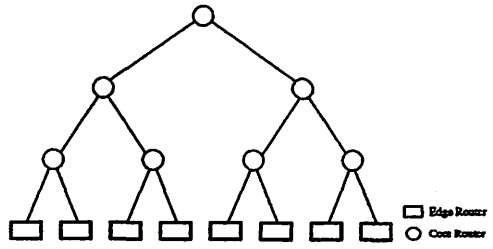


Figure 6: Tree Topology Simulation Environment.

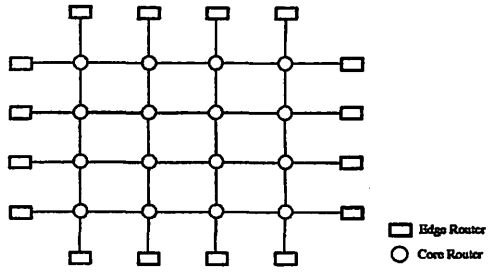


Figure 7: Mesh Topology Simulation Environment.

- Message transmission interval is according to an exponential distribution with an average of 100 msec.
- Processing time in each router for a wavelength reservation control message is 1 msec.
- A simulation interval is 100,000 msec (100 sec).

Figures 9–14 show the simulation results. Here, CFD, CFH and CFW represent the dropping recovery, holding recovery and parallel waiting recovery, respectively. In tree-type topology network environments, our proposed CFW achieves 4.5% ($N := 5$) and 3.1% ($N := 6$) shorter reservation time than CFD and 3.5% ($N := 5$) and 8.7% ($N := 6$) shorter reservation time than CFH. However, when $N := 7$, 180% and 37% longer reservation time is required in CFW than CFD and CFH, respectively. In our simulation, a pair of source and destination edge routers are selected by using randomness of unique distribution. Hence, about half of message transmissions are through the root core router in the tree-type network topology. Therefore, optical communication links connected to the root core router are highly congested and it becomes to be difficult to reserve wavelength especially with a large N. In mesh network topology environments, there are no such 'hotspot' and loads of wavelength reservation requests are more averagely distributed. In mesh-type topology environments, CFW achieves 3.6% ($N := 4$), 4.5% ($N := 6$) and 6.4% ($N := 8$) shorter average reservation time than CFD and 2.0% ($N := 4$), 2.9% ($N := 6$) and 4.3% ($N := 8$) shorter than CFH.

Real network environments are considered to be modeled by certain complex networks with relatively shorter distance between two edge routers, high clustering indicator and exponential distribution of degrees of nodes, i.e. routers. The BA model is one of the complex network model which has similar property of real computer networks [1]. Hence, we evaluate the performance of our CFW protocol

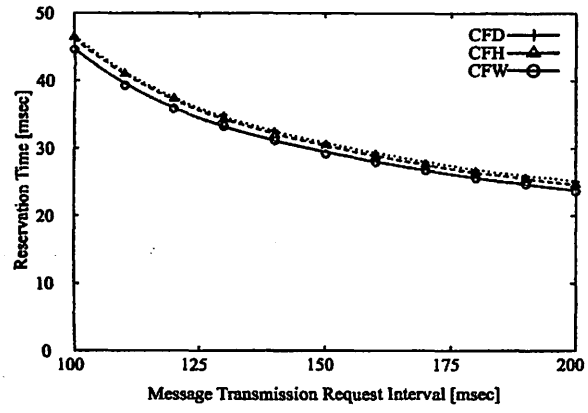


Figure 8: Average Reservation Time (Tree $N := 5$).

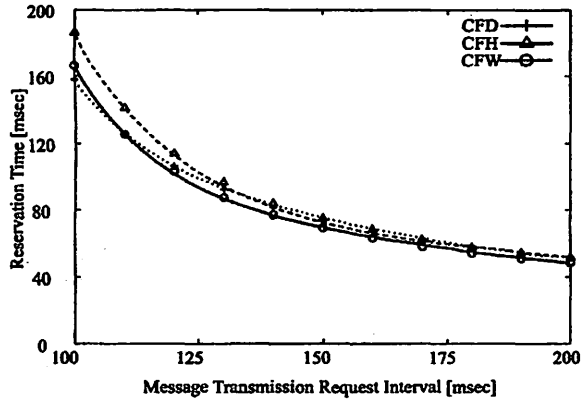


Figure 9: Average Reservation Time (Tree $N := 6$).

in the BA model network. Figures Figures 15,16 and 17 show results of performance comparison of CFD, CFH and CFW in 30, 60 and 120 routers in the BA model networks. Here, CFW requires the shortest average reservation time. By comparison with CFD, 3.2%, 4.3% and 6.4% shorter reservation time is required in the 3 different scales of networks. In addition, by comparison with CFH, reservation time is 1.5%, 2.5% and 4.5% shorter in simulation results. Hence, CFW achieves averagely 3.7% higher reservation performance than the conventional methods.

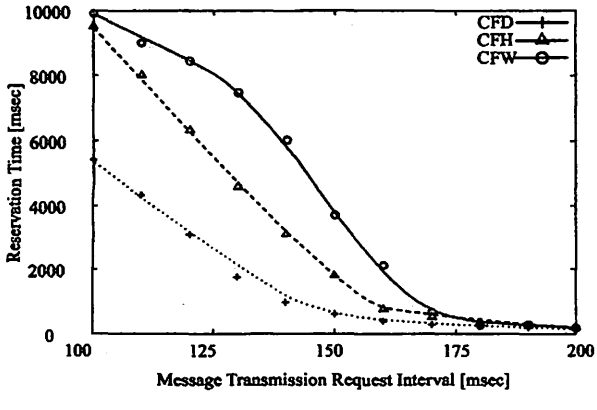


Figure 10: Average Reservation Time (Tree $N := 7$).

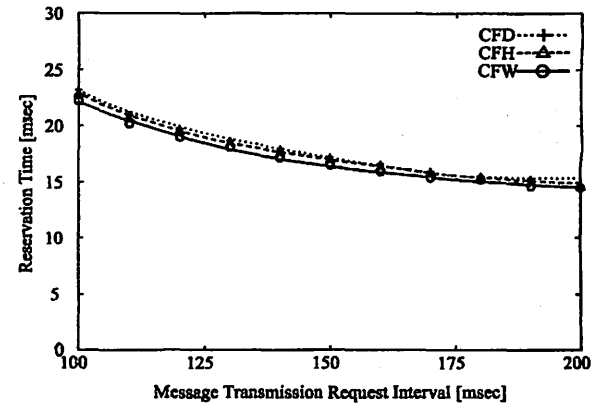


Figure 11: Average Reservation Time (Mesh $N := 4$).

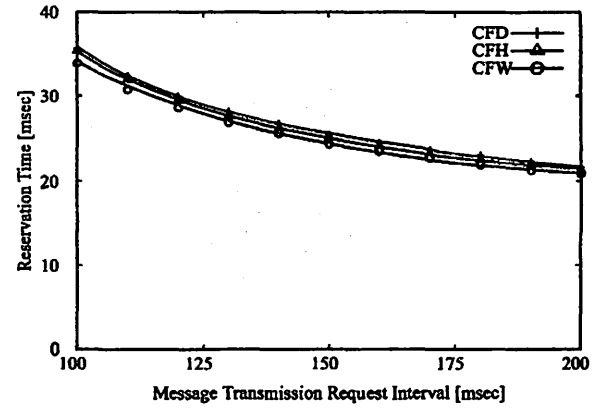


Figure 12: Average Reservation Time (Mesh $N := 6$).

5 Conclusion

This paper proposes a novel recovery method, parallel waiting, for WDM wavelength reservation and designes a CFW reservation protocol. In simulation experiments, CFW achieves higher performance than the conventional protocols CFD and CFH which are based on dropping and holding recovery, respectively. CFW protocol works well in less congested environments and in BA conmplex network model which has same property of real computer networks, i.e. higher clustering indicator and exponential distribution of degrees of routers, averagely 3.7% shorter reservation time is required than the others even with network hubs around which network congestions are likely to be occurred.

References

[1] Barabasi, A.L. and Albert, R., "Emergence of Scaling in Random Networks," Science, No. 286, pp.509–512 (1999).

[2] Chlamtac, I., Ganz, A. and Karmi, G., "Ligthpath Communications : An Approach to High-Bandwidth Optical WANs," IEEE Transactions on Conumunications, vol. 40, No.7, pp. 1171–1182 (1992).

[3] Lee, K.C. and Li, O.K., "A Circuit Rerouting Algorithm for All-optical Widearea Networks," Proceedings of the 13th IEEE INFOCOM, pp. 954–961 (1994).

[4] Mei, Y. and Qiao, C., "Efficient Distributed Control Protocols for WDM All-Optical Networks" Proceedings of the 6th International Conference

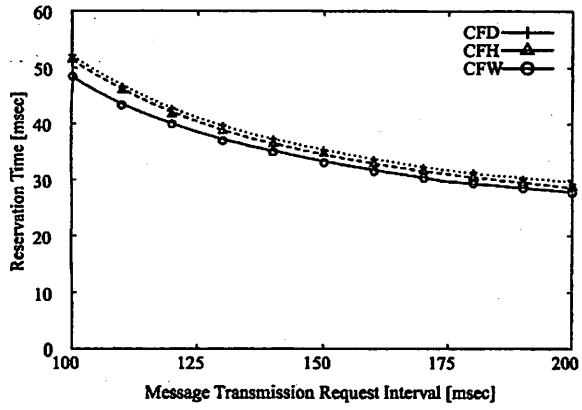


Figure 13: Average Reservation Time (Mesh $N := 8$).

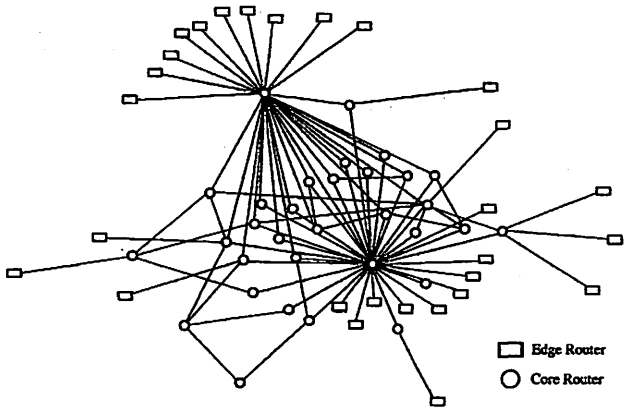


Figure 14: Real Network Topology Simulation Environment.

on Computer Communications and Networks, pp. 150–153 (1997).

[5] Ramasawami, R. and Sivarajan, K.N., "Optical Routing and Wavelength Assignment in All–optical Networks," Proceedings of the 13th IEEE INFOCOM, pp. 534–543 (1994).

[6] Ramaswami, R. aud Segall, A., "Distributed Network Control for Wavelength Routed Optical Network," Proceeding of the 15th IEEE INFORCOM, pp. 138–147 (1996).

[7] Subramaniam, S., Azizoglu, M. and Somani, A.K., "Connectivity and Soarse Wavelength Conversion in Wavelength–routing Networks," Proceedings of the 15th IEEE INFOCOM, pp. 148–155 (1996).

[8] Qiao, C. and Mei, Y., "Wavelength Reservation Under Distributed Control," Proceedings of the IEEE/LEOS Summer Topical Meeting on Broadband Optical Networks, pp. 45–46 (1996).

[9] Yuan, X., Gupta, R. and Melhem, R., "Distributed control in optical WDM networks," in Proceedings of IEEE Conference on Military Communications (MILCOM'96), Vol. 3, pp. 100–104 (1996).

[10] Yuan, X., Melhem, R. and GuptaGupta, R., "Distributed Path Reservation Algorithm for Multiplexed All-Optical Inferconuection Networks," Proceedings of the 3rd IEEE Symposium on High Performance Computer Architecture, pp. 38–47 (1997).

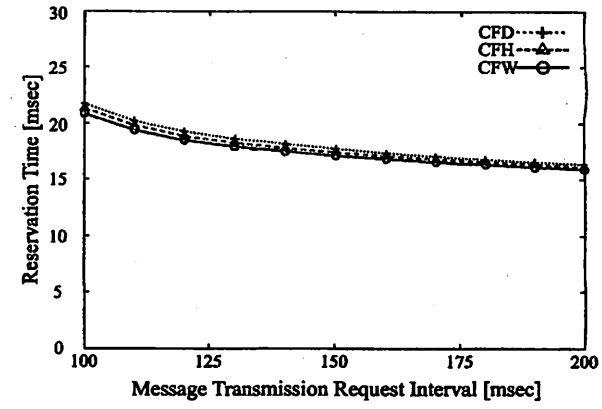


Figure 15: Average Reservation Time (BA $N := 30$).

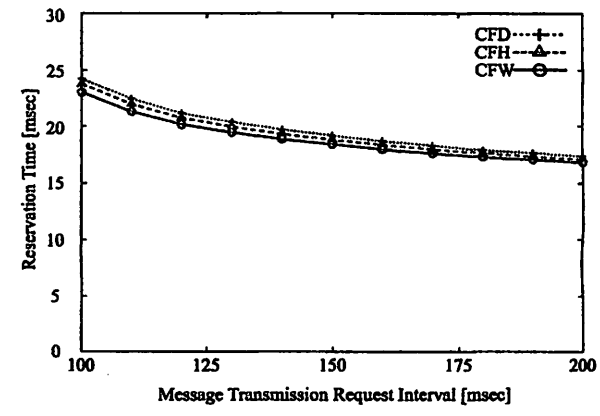


Figure 16: Average Reservation Time (BA $N := 60$).

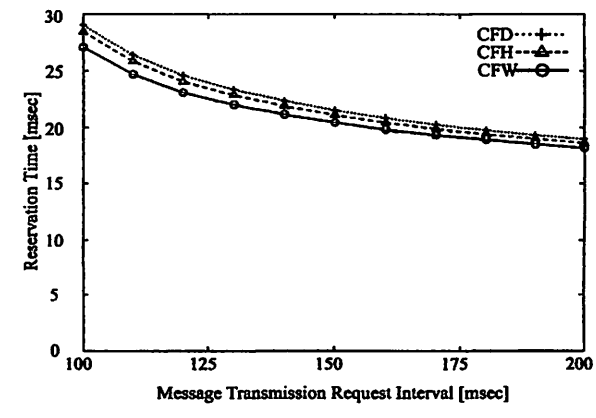


Figure 17: Average Reservation Time (BA $N := 120$).