Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp586-589)

Routing Algorithms in Resilient Packet Rings

UN GI JOO
Department of Knowledge and Industrial Engineering
Sun Moon University
100 Kalsanri, Tangjeong Asan, Chungnam, 336-708
KOREA
http://ie.sunmoon.ac.kr

Abstract: -The Resilient Packet Ring (RPR) is a data ring network, where one of the key issues is on a load balancing for competing streams of elastic traffic. This paper suggests efficient routing algorithms on the RPR. For the algorithms, we evaluate their efficiency via simulation.

Key-Words: - Resilient Packet Ring, Unsplit Loading, Heuristic Algorithm, Simulation

1. Introduction

This paper concerns load balancing problems on a Resilient Packet Ring(RPR), where the RPR is offered by IEEE802.17.[1] The RPR is well suitable for metropolitan area network with two counter-rotating rings that multiple stations share the bandwidth. The load balancing problems can be classified into two kinds of ones with and without demand splitting. A split routing allows the splitting of a demand into two portions to be carried in two directions and an unsplit routing is the one in which each demand must be entirely carried either clockwise or counter-clockwise direction. Since the cost of the ring increases with its capacity, it is desirable to route the demands so as to minimize this maximum load. The Min-max problem is stated as follow: given a set of nodes on a ring network and a set of demands between pairs of nodes, allocate bandwidths to each traffic demand so that the maximum of the loads on the links in the network is as small as possible.[2][3]

For the researches on the unsplit Min-max problem, Cosares and Saniee[4] and Dell'Amico *et al*.[5] studied the problems on bi-directional SONET(Synchronous Optical NETwork) rings. For the split loading problem, Myung *et al*.[6] and Wan and Yang[7] considered the Min-max loading problem on the SONET rings. Most of researches on the loading problems are on the two-fiber bi-directional SONET ring except Wan and Yang[7]. It is noticed that each link on the two-fiber bi-directional SONET ring requires capacity of at least sum of both directional working traffic due to the protection requirement. Wan and Yang[7] considered the Min-max loading problem on a unidirectional SONET ring with two working counter rotated fibers.

This paper develops heuristic algorithms for the unsplit routing problem on the RPR.

2. Problem Description

Consider a RPR network with n nodes sequentially numbered from 1 to n according to clockwise direction, where there exist two counter-rotating data links between two consecutive two nodes. Suppose that there are K types of demands d_k on the ring bandwidth, k=1,2,…, K. For each demand d_k, let O_k and D_k denote the originating and terminating nodes, respectively. It is noticed that demand traffic of the RPR is composed of mainly Internet traffic and measured usually as bits per seconds.

Let x_k be a variable denoting the fraction of the total demand between O_k and D_k routed in the clockwise direction. Thus, x_k=1 and x_k=0 indicate that all the demand d_k are routed in a clockwise and counter-clockwise direction, respectively.

Let L_k^+ denote the set of links contained in the clockwise path from O_k to D_k for a demand type k as depicted in Fig. 1. Similarly, let $L_k^- = L - L_k^+$ denote the set of links contained in the counter-clockwise path from O_k to D_k for a demand type k, where L denotes link set of the given ring. For each link $l \in L$, let $K_l^+ = \{k|\ l \in L_k^+\}$ and $K_l^- = \{k|\ l \in L_k^-\}$ denote the demand index set of origin-destination pairs whose clockwise and counter-clockwise path contains the link l, respectively.

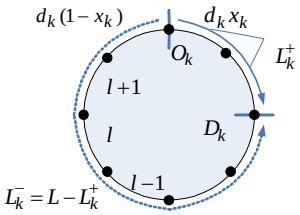


Fig. 1. L_k^+ and L_k^-

Our problem is to find the optimal loading of the RPR without demand splitting. For the problem, let z represent the required ring capacity associated with a loading. Additionally, let denote link loads as R_l^+ and R_l^- for clockwise and counter-clockwise directional link l, respectively. That's to say, $R_l^+ = \sum_{k \in K_l^+} d_k x_k$ and $R_l^- = \sum_{k \in K_l^-} d_k (1 - x_k)$. Then, the splitting problem can be expressed as following 0-1 integer programming.

$$\begin{aligned} &\text{Min } z && (1)\\ &\text{s.t } z \geq R_l^+, \ l=1,2,\ldots, L \\ &\quad z \geq R_l^-, \ l=1,2,\ldots, L \\ &\quad x_k = 0 \text{ or } 1, \ k=1,2,\ldots, K, \end{aligned}$$

where the objective is to find $\{ x_k \}$ which minimizes the ring load z. The first and second constraints denote that the ring load z is the maximum of the clockwise and counter-clockwise link load, respectively. The third constraint implies that the splitting is not allowed for each demand. By the way, if the decision variable x_k is allowed to have any real value on domain [0, 1], then the problem becomes to the spilt routing problem.

It is noticed that the researches of [4], [5] and [6] are on the problems of $\underset{x_k}{Min} \{ z \mid z \geq R_l^+ + R_l^- \}$ and there is no previous research on problem (1) except Wan and Yang[7], where Wan and Yang suggested heuristic algorithms for a unidirectional SONET ring and evaluated their worst case performance. This paper considers an unsplit routing problem on the RPR and develops heuristic algorithms with average case performance evaluation.

3. Heuristic Algorithms

It is noticed that the unsplit loading problem is a NP-complete so that a heuristic algorithm is required. This section suggests two heuristic algorithms for the RPR networks and evaluates their average performance.

We can consider the shortest-distance routing as a heuristic algorithm, denoted as H1.

Heuristic algorithm H1: For each k, if 2 ($D_k - O_k$) > n when $O_k < D_k$ or 2 ($D_k - O_k$) > – n when $O_k > D_k$, set $x_k = 0$. Otherwise, set $x_k = 1$.

The H1 routes the traffic k by using its O_k and D_k without considering its amount of demand d_k. Therefore, we suppose that the performance of H1 may be not good with respect to ring load and need to develop other algorithms for the better performance. We will develop another heuristic algorithm by using the characterization the split routing problem. It is noticed that the split problem has efficient polynomial algorithm since the problem is a Linear Programming(LP) one. However, the routing problem is only a sub-problem within the comprehensive ring planning and the problem has to be solved multiple times in practical applications. Therefore, efficiency of a routing algorithm has a big effect on the overall performance of the planning procedure.

The following property is derived by Wan and Yang[7].

Property 1. The ring load z of the split problem is a convex function with respect to $\sum_{k=1}^{K} d_k x_k$.

Even though Property 1 characterizes the convexity of the ring load z, finding the optimal loading is difficult since there are many alternatives $\{ x_k \}$ with the same value of $\sum_{k=1}^{K} d_k x_k$. Therefore, we need to further characterize the optimal routing of the split problem. For the characterization, let define additional notations x_k^i and δ_i such that $\delta_i = \underset{1 \leq l \leq n}{Max} \{ \sum_{k \in K_l^+} d_k x_k^i \} - \underset{1 \leq l \leq n}{Max} \{ \sum_{k \in K_l^-} d_k (1 - x_k^i) \} = \underset{1 \leq l \leq n}{Max} \{ R_l^+ \} - \underset{1 \leq l \leq n}{Max} \{ R_l^- \}$, where x_k^i denotes the variable x_k at the i-th iteration. Let $x_j^{i+1} = x_j^i - \Delta_i / d_j$ for a demand type j and $x_k^{i+1} = x_k^i$ for all k, k ≠ j, where Δ_i =Min{ δ_i /2, d_j }. Then, δ_i has the following property.

Property 2. The values of $\{ \delta_i \}$ has a relationship of $\delta_i \geq \delta_{i+1}$ for each i, i=1,2,…, K-1.

Proof. Since $\underset{1 \leq l \leq n}{Max} \{ R_l^+ \}$ = Max{ $\max_{l \in L_j^+} \{ R_l^+ \}$, $\max_{l \in L_j^-} \{ R_l^+ \}$ } and $\underset{1 \leq l \leq n}{Max} \{ R_l^- \}$ = Max{ $\max_{l \in L_j^+} \{ R_l^- \}$, $\max_{l \in L_j^-} \{ R_l^- \}$ }, δ_i = Max{ $\max_{l \in L_j^+} \{ R_l^+ \}$, $\max_{l \in L_j^-} \{ R_l^+ \}$ } – Max{ $\max_{l \in L_j^+} \{ R_l^- \}$,

$\max_{l\in L_j^-}\{R_l^-\}$ } and δ_{i+1} = Max{ $\max_{l\in L_j^+}\{R_l^+\} - \Delta_i$, $\max_{l\in L_j^-}\{R_l^+\}$ }– Max{ $\max_{l\in L_j^+}\{R_l^-\}$, $\max_{l\in L_j^-}\{R_l^-\} + \Delta_i$ }, where $R_l^+ = \sum_{k\in K_l^+} d_k x_k^i$, $R_l^- = \sum_{k\in K_l^-} d_k(1-x_k^i)$ and Δ_i =Min{ δ_i /2, d_j }. There are four cases of situations such that (1) Case 1: $\max_{l\in L_j^-}\{R_l^+\} \geq \max_{l\in L_j^+}\{R_l^+\} - \Delta_i$ and $\max_{l\in L_j^+}\{R_l^-\} \geq \max_{l\in L_j^-}\{R_l^-\} + \Delta_i$; (2) Case 2: $\max_{l\in L_j^-}\{R_l^+\} \geq \max_{l\in L_j^+}\{R_l^+\} - \Delta_i$ and $\max_{l\in L_j^+}\{R_l^-\} < \max_{l\in L_j^-}\{R_l^-\} + \Delta_i$; (3) Case 3: $\max_{l\in L_j^-}\{R_l^+\} < \max_{l\in L_j^+}\{R_l^+\} - \Delta_i$ and $\max_{l\in L_j^+}\{R_l^-\} \geq \max_{l\in L_j^-}\{R_l^-\} + \Delta_i$; (4) Case 4: $\max_{l\in L_j^-}\{R_l^+\} < \max_{l\in L_j^+}\{R_l^+\} - \Delta_i$ and $\max_{l\in L_j^+}\{R_l^-\} < \max_{l\in L_j^-}\{R_l^-\} + \Delta_i$. For each case, we can show that the relationship $\delta_i \geq \delta_{i+1}$ is satisfied. This completes the proof.

According to Property 2, if we find out a solution with δ_i = 0 then the current solution is an optimal solution of the split problem. Based upon Property 2, we develop another heuristic algorithm, denoted as H2.

Heuristic Algorithm H2

Step 1. Let i = 1 and x_k^i = 1 for all k, k=1,2,…, K. Calculate each initial clockwise link load as $R_l^+ = \sum_{k\in K_l^+} d_k x_k^i$ and let $R_l^- = 0$ for all links l.

Step 2. Compute δ_i, $\delta_i = \underset{1\leq l\leq n}{Max}\{R_l^+\} - \underset{1\leq l\leq n}{Max}\{R_l^-\}$. If $\delta_i >$ 0, go to Step 3. Otherwise, go to Step 4.

Step 3. (1) For a link l_1 such that $\underset{1\leq l\leq n}{Max}\{R_l^+\} = R_{l_1}^+$, find a demand type j with maximum clockwise load on the link l_1 , i.e., $j = \arg\max_{k\in K_{l_1}^+}\{d_k x_k^i\}$. (2) Calculate Δ_i for the demand j as Δ_i =Min{ δ_i /2, d_j }. (3) Let $x_j^{i+1} = x_j^i - \Delta_i / d_j$ and $x_k^{i+1} = x_k^i$ for all k, k=1,2,…, K, k$\neq j$. (4) For each link l, let $R_l^+ = R_l^+ - \Delta_i$, if $l \in L_j^+$ and let $R_l^- = R_l^- + \Delta_i$ otherwise, go to Step 2 with i= i+1 (until $i \leq K$). If i =K+1, go to Step 4.

Step 4. If all x_k^K 's are integer, stop. Otherwise, let S = { $x_k^K \mid 0 < x_k^K < 1$ }, $R_l^+ = \sum_{k\in K_l^+} d_k x_k^K$, $R_l^- = \sum_{k\in K_l^-} d_k(1-x_k^K)$, and go to Step 5.

Step 5. For each k, $k \in S$, set the value of x_k^K zero or one as follows.

(1) Calculate λ_1 = Max{ $\max_{l\in L_k^+}\{R_l^+\} - d_k x_k^K$, $\max_{l\in L_k^-}\{R_l^-\} + d_k x_k^K$ } and λ_2 =Max{ $\max_{l\in L_k^+}\{R_l^+\} + d_k(1-x_k^K)$, $\max_{l\in L_k^-}\{R_l^-\} - d_k(1-x_k^K)$ }.

(2) If $\lambda_1 < \lambda_2$, then set $R_l^+ = R_l^+ - d_k x_k^K$ for $l \in L_k^+$, $R_l^- = R_l^- + d_k x_k^K$ for $l \in L_k^-$, and x_k^K =0. Otherwise, set $R_l^+ = R_l^+ + d_k(1-x_k^K)$ for $l \in L_k^+$, $R_l^- = R_l^- - d_k(1-x_k^K)$ for $l \in L_k^-$, and x_k^K = 1.

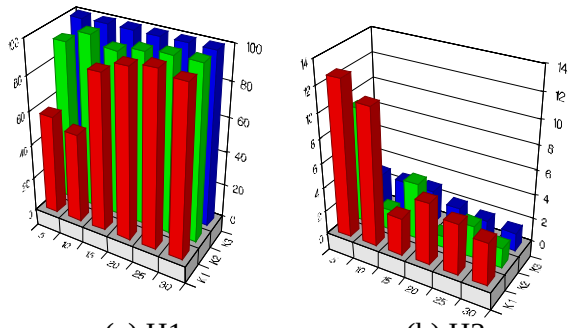
(3) Go to Step 2(1) unless all the k's in S are considered.

Steps 1 through 3 find an optimal solution of the split problem and the remaining steps modify the resulted solution to the one of unsplit problem. Step 3 improves the current solution { x_k^i } by rerouting the amount of Δ_i traffic demand of demand type j, where the rerouting demand j makes that either resulting solution { x_k^{i+1} } satisfies $\underset{1\leq l\leq n}{Max}\{R_l^+\} = \underset{1\leq l\leq n}{Max}\{R_l^-\}$ or all the demand d_j is routed in the counter-clockwise direction. Since Step 3 is activated only when δ_i >0 for a demand type j, $j \in K_{l_1}^+$, $\delta_i > \delta_{i+1}$ and the algorithm terminates with δ_i =0. Step 5 reroutes the demand type k in the set S to one of the two directions in which would result in the smaller increase of ring load, where λ_1 and λ_2 denote the amount of increasing ring load resulted from entirely rerouting the demand k in counter-clockwise and clockwise directions, respectively.

The computational load of H2 is determined by Steps 3 and 5. The rerouting procedure of Step 3 requires $O(n)$ for each demand type k and there are K demand types, thus the computational time complexity of Step 3 is $O(n \cdot K)$. Similarly, Step 5 has computational load of $O(n \cdot K)$ since at most K variables are to be checked with computational load of $O(2n)$. Therefore, the overall computational time complexity of H2 is $O(n \cdot K)$.

For the performance evaluation of H1 and H2, we consider 18 combinations of (n, K), (n,K) =

{(5,6), (5,8), (5,10), (10,12), (10,23), (10,45), (15,25), (15,50), (15,105), (20,40), (20,95), (20,190), (25,60), (25,150), (25,300), (30,90), (30,200), (30,435)}. For each instance (n, K), we randomly generate ten problems with traffic demands between 5 and 100 for randomly selected pairs of originating and terminating nodes, and find out the resulted average ring load(Load) and calculate the average computation time(Time) of H1 and H2 by using Visual C/C++ code on Pentium IV PC (1.0GHz, window XP). For the evaluation the solution quality, we calculate the relative deviations as [load of H1 or H2 – optimal load of the split problem][100]/[optimal load of the split problem] since the optimal ring load of the split problem is a lower bound of the optimal solution of the unsplit problem. Fig. 2 depicts the relative deviations. We can observe that the deviation of H1 is increased as (n,K) increasing and the mean deviation of H1 is 93.44%. By the way, the deviation of H2 is not increased as the (n,K) increasing and H2 has mean deviation of 4.15%.



(a) H1 (b) H2

Fig.2. Relative deviations of H1 and H2

Fig. 3 shows the computational time of H1 and H2. Even if the CPU time of both algorithms is increasing as the (n,K) increasing, the time is not large. Therefore, we can conclude that the heuristic algorithm H2 can be used for good load balancing even when the ring has large value of (n,K).

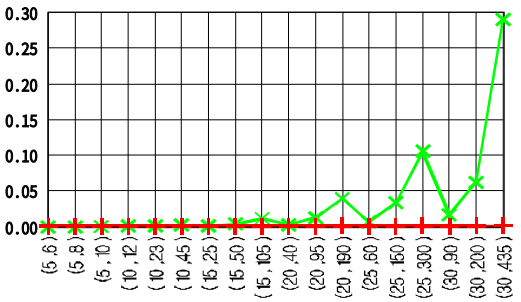


Fig. 3. Computational time of H1 and H2

4. Conclusion

This paper considers a routing problem without load splitting on a RPR. For the unsplit routing problem, we suggest two construction type heuristic algorithms based upon characterization of the split problem and the short-way routing concept and show their efficiency by using various numerical examples. We expect that our algorithms will be used for efficient ring loading on a RPR or a uni-directional dual ring network to improve the ring utilization. However, development of improvement type algorithms for the unsplit problem is remained as a further study.

References:

[1] *A Summary and Overview of the IEEE 802.17 Resilient Packet Ring Standard*, RPR Alliance, 2004.

[2] L. Massoulie and J. Roberts, "Bandwidth Sharing: Objectives and Algorithms", *INFOCOM'99*, 1999, pp.1395-1403.

[3] H-S Lee *et al.*, "Optimal Time Slot Assignment Algorithm for Combined Unicast and Multicast Packets," *ETRI Journal*, Vol. 24, No. 2, 2002, pp. 172-175.

[4] S. Cosares and I. Saniee, "An Optimization Problem related to Balancing Loads on SONET Rings", *Telecommunications System*, Vol.3, 1994, pp.165-181.

[5] M. Dell'Amico *et al.*, "Exact Solution of the SONET Ring Loading Problem", *Operations Research Letters*, Vol.25, 1999, pp.119-129.

[6] Y.-S. Myung *et al.*, "Optimal Load Balancing on SONET Bidirectional Rings", *Operations Research*, Vol.45, 1997, pp.148-152.

[7] P.-J. Wan and Y. Yang, "Load-Balanced Routing in Counter Rotated SONET Rings", *Networks*, Vol.35, 2000, pp.279-286.