Journal of Engineering and Applied Sciences 12 (Special Issue 7): 8240-8245, 2017
ISSN: 1816-949X

Performance Enhancement by Traffic Re-scheduling of Network Nodes

[1]Mou Dasgupta, [2]G.P. Biswas and [3]Sangram Ray
[1]Department of Computer Application, National Institute of Technology Raipur,
Chhattisgarh, India
[2]Department of Computer Science and Engineering,
Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand, India
[3]Department of Computer Science and Engineering,
National Institute of Technology Sikkim, Sikkim, India

Abstract: Communication subnets are the main carrier of data communication networks and the overall transmission efficiency depends on the performance of the subnets. There exist some nodes in subnets that forward most of the traffic of a network which may be termed as critical nodes and until some precautions regarding the processing capacity, delay factor, packet discarding, etc. are not taken care of the performance of the subnets will not increase. This study proposes a technique to identify these critical nodes and develops a traffic re-scheduling mechanism that diverts some of data packets from over-used links to unused or less-used links of the critical nodes. A critical threshold is thus assumed that tells either the packets follow the normal routing decision or the diversion based on decision of the traffic equalizer. An example is given to illustrate the proposed traffic re-scheduling technique.

Key words: Critical, nodes, communication, subnets, traffic, re-scheduling

INTRODUCTION

Communication subnets are integral part of all types of communication systems, viz. telecommunication, internet, etc. A communication subnet is a collection of channels/links and nodes where the traffic traversing through a definite channel is processed by the communication nodes. The nodes process the data packets containing routing tags indicative of the output port destinations to which the data packets are addressed and route these packets to the identified output ports. Though this technique is the essence of communication in the subnet, it incorporates various complexities when the channel allocation is not uniform, resulting to what is known as congestion, delay, increased packet loss, decreased throughput, failover, etc. Traffic engineering can be useful when dealing with the said complicacies. It is a technique to optimize network performance by dynamically analysing, predicting and regulating the behaviour of data transmitted over the network, so as to reduce network traffic congestion.

Considerable research has been conducted to model and quantify the performance of network services and technologies (Mortier, 2002; Ismail and Zin, 2009; Bonald and Roberts, 2007). Several network traffic models have been used to study fairness, response times, queue lengths, etc., under different assumptions (Barakat *et al*., 2002). The researcher (Fortz and Thorup, 2002) have proposed a model for optimizing the weights of open shortest path first routing protocol. Their approach considerably increases the demand for traffic. By Mortier (2002), the researcher have developed traffic engineering simulation model to measure the performance of heterogeneous environment over single and multiple links using queuing theory. The essence of every researcher is to reduce congestion and thereby improve the network's performance. By Qin and Zhao (2006), Shi and Mohan (2006), Sharma *et al*. (2008) and Dandamudi (2012), researcher have presented various traffic flow and load balancing algorithms. The researcher (Shi and Mohan, 2006) have designed a traffic flow distribution method based on selecting one of the available label switch paths to carry one aggregated traffic flow. Analyses of load balancing algorithms are done on different parameters (both static and dynamic) by Sharma *et al*. (2008). On the other hand, certain nodes in a network are vulnerable and hence are termed as critical. Identification of such nodes are very important as they may be identified based on several network parameters. By Zhou (2014), the researcher have proposed a heuristic algorithm to find the

---

Corresponding Author: Mou Dasgupta, Department of Computer Application, National Institute of Technology Raipur, Chhattisgarh, India

critical nodes based on two parameters-range and support. The researchers (Asif *et al*., 2016) introduced a new metric to characterize the criticality of a node in a network namely, Combined Banzhaf and Diversity Index (CBDI) that utilizes a diversity index based on the variability of a node's attributes relative to its neighbours and the degree of participation of a node in forming shortest paths.

In contrast to other research in the literature, this study presents a scalable technique that distributes traffic between links, so that, no link is over loaded. In a network of size of the internet implying the growing subnet infrastructure, the important points for any proposed methodology lies in how it utilizes the existing resources and how scalable it is. The objective of this study is rather to highlight the issue of scalability of applying traffic scheduling in communication subnets.

The present research first identifies the critical nodes of a subnet and then performs traffic re-scheduling at those nodes, so that, the traffic congestion is reduced and throughput of the subnet increases.

MATERIALS AND METHODS

Proposed technique for traffic re-scheduling and performance enhancement: The notion of traffic re-scheduling is accompanied by three vital inquiries. First one is where to perform traffic re-scheduling? Second when to take decision for traffic re-scheduling? And the third is how to perform traffic re-scheduling? Although, the answer to the first question suffices in itself for traffic scheduling but is not complete without the other two. In the proposed method of traffic re-scheduling, answers to all the three questions are provided. Traffic re-scheduling in this study is referred to re-distribution of traffic over the routing table. The traffic on a node is diverted from the usual routing table to improve its performance. In the first step of the proposed technique, it selects a node called the critical node, from among all other nodes. In a subnet more than one critical node may exist. This node is where the traffic re-scheduling is being done. Thus, a traffic equalizer is placed at this node which takes the decision of traffic re-scheduling. In the next step, traffic balancing is performed. A novel mechanism has been proposed which precisely distributes the traffic to all the outgoing links of the critical node. The proposed technique for traffic re-scheduling has been done in three phases detailed below along with performance analysis.

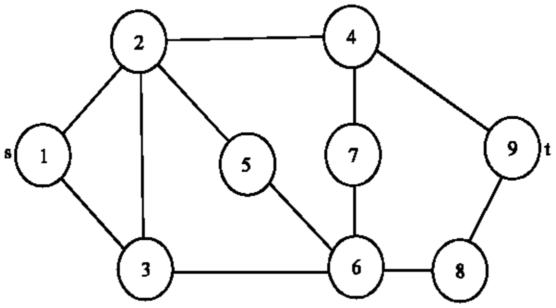

Fig. 1: Graph representation of subnet with 9 nodes

Table 1: Routing table of node v_i

Destination node	Next nodes
$v_{j \neq i}$	v^n_i $n = 1,2, ..., N-1$

Node identification for traffic re-scheduling: A critical node can be defined as the node whose deletion or malicious behavior disconnects or significantly degrades the performance of the network (Dasgupta and Biswas, 2011). In order to detect a critical node a graph theoretic approach of network representation has been considered. Let G (V, E) denote the graph representation of the subnet. $V = \{v_i \mid i \in N\}$ and $E = \{e_l \mid l \in M\}$ are its node set and Edge set where, N is the total number of Nodes and M is the total number of edges in the graph, respectively. The source and destination nodes are connected through a path. For example, in Fig. 1 {1, 2, 5, 6, 8, 9} is a path for source node 1 and destination node 9. For any source node to destination node, one or more paths exist. A routing protocol selects a path based on some metric for a source destination pair and forwards traffic along this path. Let P_{ij} be the path for any source-destination Pair (v_i, v_j) where, $i \neq j$ according to any routing protocol. From the routing table of a node the next node for every path toward the destination is obtained. Let $v_i^1, v_i^2 \ldots v_i^{N-1}$ be the next nodes for source node v_i. The routing table for a node say v_i may be formed as shown in Table 1.

Let v_i^k be the node which appears most of the time as the next node in the routing table for node v_i. That is $v_i^k = \max\{v_i^n\}$.

Let this node be quasi-critical node. In other words, a quasi-critical node is defined as the node which appears maximum times as the next node in a routing table of a node. In this way the quasi-critical nodes for all nodes of the subnet is calculated. Each node then distributes this information to its adjacent nodes. The information includes the source node and its quasi critical node. The frame format of this packet.

Frame format of vi's quasi critical packet:

- Source node: v_i
- Quasi critical node: v_i^k

Each node updates its information after receiving quasi critical packets from its neighbours. In this manner all the nodes gradually get the quasi-critical node of every other node without creating much overhead of distributing the same information to all the nodes. Now, each node checks the highest frequency of quasi-critical node against its critical threshold value. The critical threshold value is given by:

$$C_i^T = \left\lceil \frac{(N\text{-}1)}{O_i} \right\rceil$$

where, O_i is the Outgoing link of node i. If the highest frequency of quasi-critical node is greater than the node's critical threshold value, then this node is the critical node v_c of the subnet. The algorithm 1 for finding quasi-critical node and critical node at every node are given below:

Algorithm QCN:

```
Input: Number of nodes N, next node for each destination D[N-1]
  for i = 1 to N-1 do
     a [i] ← next node at D[i]
     b [i] ← frequency of next node's appearance
     m ← max {b [I]}
  end for
  $v^k$ ← m
Output: Quasi-critical node $v^k$
```

Algorithm CN:

```
Input: Number of nodes N, quasi critical node for each node $v^k$[N], number
of outgoing interfaces l
  for i = 1 to N do
     x [i] ← frequency of $v^k$ [I]
     m ← max {x [I]}
  end for
  $C^T$ ← [(N-1)/l]
  if (m>$C^T$)
    this node is critical node and traffic equalizer is activated
  end if
Output: Critical node $v_C$
```

Activation of traffic equalizer and its activity: Once the critical node has been found out, then the traffic equalizer is activated at this node. Traffic equalizer is a process that distributes traffic evenly to the channels. In a communication subnet the number of critical node is more than one. So, the traffic equalizer is activated at all the nodes that are critical. The intuition behind activating the traffic equalizer at the critical node is that the critical node is the node which receives most of the packets than any other node of the subnet. That is most of the paths include this node as one of its intermediate nodes. In case of heavy load the buffer of this node becomes full resulting in a long queue for the traffic flows. As a consequence, there is congestion in all the outgoing links. Thus, an increased delay of traffic for reaching the destination is inevitable. Also, the packet discard rate will become high. Thus, it is vindicated for placing the traffic equalizer at the critical node for solving the above mentioned problems.

Let C_l be the traffic Capacity on each outgoing edge $e_l, 0 \le l \le N$ of the critical node. Let C_{out} be the total traffic Capacity for all outgoing edges of v_c. Since, the objective is to redistribute the traffic load evenly on all outgoing links, so, the concern is made on the outgoing links only. Also from the law of conservation of traffic, the total traffic entering the critical node must be equal to the total traffic exiting from the outgoing links of the critical node. Let x_l be the amount of traffic on each individual outgoing link, $e_l, 0 \le l \le M$ of the critical node v_c. Let x be the sum of the traffic for all the outgoing links of v_c. Also, the amount of traffic that could pass from each individual link is given by:

$$A_l = \left(\frac{x}{C_{out}} \right) C_l$$

whereas only (x_l/x) fraction is the utilization of the edge e_l. Thus, the difference between maximum traffic and actual traffic on link l is given by:

$$\partial_l = \left[\left(\frac{x}{C_{out}} \right) C_l\text{-}(x_l) \right]$$

It is the amount of traffic that can still pass through each link. For each outgoing link the difference of traffic, ∂_l will vary. The amount of traffic passing through the edges having greater value of ∂_l is less than those edges having smaller ∂_l value. These edges can bear more load that is they can have more traffic pass through them. Whereas if for any link ∂_l is negative then, packets are discarded in other words traffic loss takes place. The role of traffic equalizer is to monitor the value of ∂_l for all the outgoing links and to divert the traffic of the link which has lower value of ∂_l to the link which has greater ∂_l value. As a result, the traffic is uniformly distributed to all the outgoing links. The algorithm for calculating ∂_l is given below:

Algorithm TD:

```
Input: Traffic $x_l$ at each link l,
Capacity $C_l$ of each link,
Number of outgoing links m,
   $x \leftarrow \sum_{l=1}^{m} x_l$
   $C_{out} \leftarrow \sum_{l=1}^{m} C_l$
   for l = 1 to m do
$A_l$←(x/$C_{out}$) $C_l$
$\partial_l$←$A_l$-$x_l$
end for
      Output: $\partial_l$
```

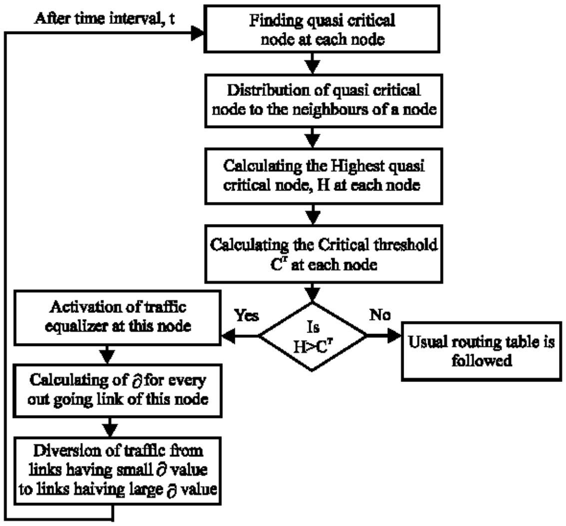

Fig. 2: The schematic diagram of traffic re-scheduling

Forwarding the traffic: Activation of the traffic equalizer implies that a process is being activated that will do the traffic re-scheduling task by redistributing the traffic. Until and unless this process is activated traffic follow the usual routing table. When the traffic arrive the critical node, the node receives all the routing information. An input buffer at that node is maintained. All the incoming traffic that arrive the critical node is stored in the buffer. From the routing table the next node of each packet is fetched. Thus, the amount of traffic on each outgoing link, x_l is known. If the critical threshold value is greater than the highest frequency of quasi-critical node of a node then, the traffic equalizing process is not activated. When the critical threshold value falls below the highest frequency of quasi-critical node an interrupt is generated. This triggers the interrupt handler to begin execution. The interrupt handler in turn triggers the traffic equalizer and traffic re-scheduling takes place. As a result, the interface of the traffic flow changes but on the next node it follows the usual routing table to reach the destination node. In this approach the traffic gets diverted by one hop only from the usual direction toward destination. The new next node receives the traffic as it receives from other nodes and the routing protocol at that node selects the best possible path for forwarding traffic Fig. 2. The algorithm for traffic equalizer activation is given below:

Algorithm TE:

```
Input: Critical threshold value $C^T$, highest frequency of quasi-critical node H
  if ( $H>C^T$) then
    TE = 1
  else
    TE = 0
  end if
  return TE
```

Performance analysis: As one of the objectives of this study is to improve the performance of the critical node, the approach considered here is to minimize time required for forwarding the traffic. Now, the total traffic forwarding time of a node v_i is given by:

$$\sum_{l=1}^{m} \frac{x_l}{C_l}$$

where, m is the number of outgoing links of the node. Let $C_1<C_2<C_3<, ..., <C_m$. This order is taken for ease of understanding though any order is viable. According to the condition mentioned in study when this node becomes critical the traffic equalizer is activated at this node. From study loads from over utilized link is diverted to underutilized link of the critical node. This implies that for given x, the link having largest difference between x_l and C_l receives load form the link having smallest amount of this difference. Let Kth link has largest difference between x_k and C_k and jth link has smallest difference between x_j and C_j. When the traffic equalizer is not activated the traffic forwarding time of this node is:

$$T_1 = \sum_{l=1}^{m} \frac{x_l}{C_l}$$

$$\text{i.e.}\, T_1 = \frac{x_1}{C_1}+\frac{x_2}{C_2}+\frac{x_3}{C_3}+\ldots+\frac{x_j}{C_j}+\frac{x_k}{C_k}+,\ldots,+\frac{x_m}{C_m}$$

$$\text{i.e.}\, T_1 = \sum_{l=1,\, l\neq j,k}^{m} \frac{x_l}{C_l}+\frac{x_j}{C_j}+\frac{x_k}{C_k}$$

After the activation of the traffic equalizer, link k receives link j's traffic. Then the response time of the node becomes:

$$T_2 = \sum_{l=1,\, l\neq j,\, k}^{m} \frac{x_l}{C_l}+\frac{x_j+x_k}{C_k}$$

Since:

$$Cj<Ck$$

Therefore:

$$\frac{x_l}{C_j}>\frac{x_j}{C_k}$$

Hence, $T_1>T_2$. Thus, the proposed technique improves the performance of the critical node by reducing its traffic forwarding time. This directly implies that the delay incurred by packets in queues of overloaded links is minimized.

RESULTS AND DISCUSSION

Illustrative example: Consider a subnet of 11 nodes and 19 links (Belovich, 1995) shows in Fig. 3. For finding the

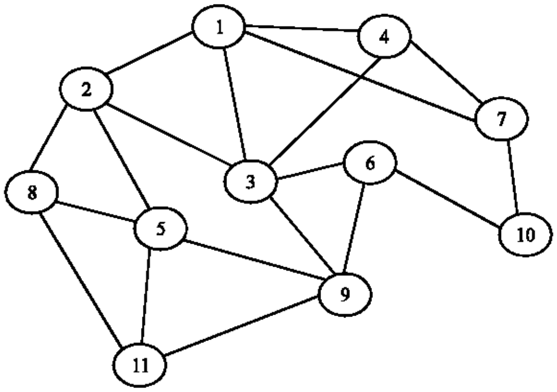

Fig. 3: The topology structure of 11 nodes subnet

Table 2: Quasi critical nodes of source node 1

Destination	Next node
2	2
3	3
4	3
5	2
6	3
7	7
8	2
9	3
10	7
11	3

Quasi critical node is 3

Table 3: Quasi critical nodes for all source nodes

Source nodes	Quasi critical nodes
1	3
2	3
3	4
4	1
5	8
6	3
7	4
8	2
9	11
10	6
11	8

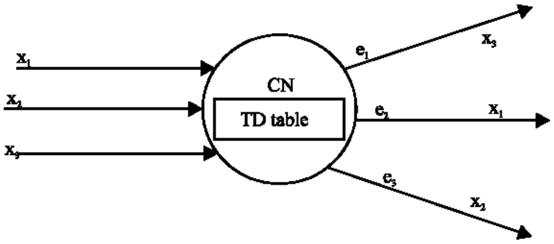

Fig. 4: Traffic interface before traffic re-scheduling

Data	Next node	Next link
x_1	v_5	e_2
x_2	v_8	e_3
x_3	v_2	e_1

Capacity of the edges (Mbps)	
C_1	32
C_2	16
C_3	64

Traffic of the links (Mb)	
x_1	10
x_2	15
x_3	25

Fig. 5: Traffic interferance after traffic re-scheduling

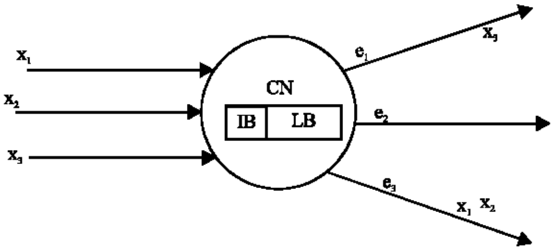

Fig. 6: Traffic interface after traffic re-scheduling

Table 4: Critical threshold values for all nodes

Nodes	Critical thresholds
1	3
2	3
3	2
4	4
5	3
6	4
7	4
8	4
9	3
10	5
11	4

critical node of the subnet first the quasi-critical node for every node is to be found. Here, distance vector routing protocol is used to obtain the routing table for next node column. The traffic considered is artificial. Table 2 shows the next node of the routing table of node 1 and its quasi critical node. Similarly, the other nodes also calculate their quasi critical nodes and distribute this information to their neighbours. Gradually all the nodes receive the quasi critical nodes of every node. The quasi-critical nodes for all nodes are depicted in Table 3. From Table 3, it is evident that node 3 has the highest frequency (equal to three) as the quasi critical node for all source-destination pairs. Now, the critical threshold at every node is calculated. Table 4 shows the critical threshold value of every node. It is seen that node 3's critical threshold value falls below the highest frequency of the quasi critical node. Hence, traffic equalizer is activated at node 3 which is the critical node for the subnet in Fig. 2.

The traffic re-scheduling takes place at the critical node. Figure 4 explains how the traffic is traversing through the outgoing links when there is no traffic equalizer. The traffic data, capacity of each link and the traffic diversion (Fig. 5).

Figure 6 shows the flow of traffic after the activation of the traffic equalizer. The calculation of A_l, ∂_l and the

Data	Next node	Next link
x_1	v_5	e_3
x_2	v_4	e_3
x_3	v_2	e_1

Calculation of A_1	
A_1	14.2
A_2	7.1
A_3	28.5

Calculation of ∂_1	
∂_1	13.7
∂_2	6.9
∂_3	28.2

Fig. 7: A_1, ∂_1 and modified traffic diversion table

Table 5: Traffic forwarding time at critical node

Links	Traffic forwarding time (sec) without traffic re-scheduling	Traffic forwarding time (sec) with traffic re-scheduling
e_1	0.781	0.781
e_2	0.625	0
e_3	0.234	0.390
Total	1.640	1.17

modified traffic diversion (Fig. 7). For performance analysis the traffic forwarding time at the critical node with and without traffic-rescheduling are compared as shown in Table 5.

Advantages: A new technique to increase the performance of computer subnets has been proposed. It has several advantages, especially, the following:

Scalability: The proposed technique is scalable. It accomplishes traffic re-scheduling only on the critical nodes of a subnet which constitutes a very small percent of the total nodes.

Overhead and associated cost: Obtaining the critical nodes require some simple calculations based on the information available from the routing table of a node. Only the distribution of quasi critical nodes occupies some resources but it still is minimized because a node distributes information merely to its neighbor. Associated costs are also less as the proposed technique does not include extra operations and message exchange other than trivial information for processing.

Serviceability: As the critical node has the maximum incoming traffic, so, traffic re-scheduling at this node results in better quality of service at network level which indirectly helps the upper layers.

CONCLUSION

A technique has been presented for increasing the performance in communication subnets by re-scheduling traffic. For which an algorithm for identifying critical nodes of a subnet is proposed where a traffic equalizer process is placed to divert the traffic from heavily used links to least used links depending on the critical threshold value, otherwise normal routing table is followed. The method shows effective results with respect to a node's traffic forwarding time.

REFERENCES

Asif, W., H.K. Qureshi, M. Rajarajan and M. Lestas, 2016. Combined Banzhaf and Diversity Index (CBDI) for critical node detection. J. Netw. Comput. Appl., 64: 76-88.

Barakat, C., P. Thiran, G. Iannaccone, C. Diot and P. Owezarski, 2002. A flow-based model for internet backbone traffic. Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, Marseille, France, November 6-8, 2002, ACM, New York, USA., pp: 35-47.

Belovich, S.G., 1995. A design technique for reliable networks under a nonuniform traffic distribution. IEEE. Trans. Reliab., 44: 377-387.

Bonald, T. and J.W. Roberts, 2007. Scheduling network traffic. Sigmetrics Perform. Eval. Rev., 34: 29-35.

Dandamudi, S., 2012. Sensitivity evaluation of dynamic load sharing in distributed systems. Master Thesis, Carleton University, Ottawa, Canada.

Dasgupta, M. and G.P. Biswas, 2011. Relieving critical nodes and designing of fault tolerant data networks. Int. J. Eng. Sci. Technol., 3: 41-51.

Fortz, B. and M. Thorup, 2002. Optimizing OSPF/IS-IS weights in a changing world. IEEE J. Selected Areas Communi., 20: 756-767.

Ismail, M.N. and A.M. Zin, 2009. Traffic engineering: Simulation model and real network environment over single and multiple links. Eur. J . Sci. Res., 25: 54-67.

Mortier, R., 2002. Internet traffic engineering. Ph.D Thesis, University of Cambridge, Cambridge, England.

Qin, Y. and C. Zhao, 2006. Efficient split scheduling scheme for traffic balancing on multi-link networks. Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06) Vol. 2, June 20-24, 2006, IEEE, Hanzhou, Zhejiang, China, ISBN:0-7695-2581-4, pp: 180-185.

Sharma, S., S. Singh and M. Sharma, 2008. Performance analysis of load balancing algorithms. World Acad. Sci. Eng. Technol., 38: 269-272.

Shi, T.J. and G. Mohan, 2006. An efficient traffic engineering approach based on flow distribution and splitting in MPLS networks. Comput. Commun., 29: 1284-1291.

Zhou, X., 2014. A novel critical node in generalized networking. Proceedings of the 20th International Conference on Automation and Computing (ICAC), September 12-13, 2014, IEEE, Cranfield, England, ISBN: 978-1-4799-6274-7, pp: 73-78.