Computers and Electrical Engineering 38 (2012) 1433–1443

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Design of multi-path data routing algorithm based on network reliability ☆

Mou Dasgupta *, G.P. Biswas

Department of Computer Science & Engineering, Indian School of Mines, Dhanbad 826 004, India

ARTICLE INFO

Article history:
Received 23 February 2010
Received in revised form 12 April 2012
Accepted 13 April 2012
Available online 28 May 2012

ABSTRACT

Data routing through an interconnected network is important and this paper addresses the design of a multi-path data routing algorithm based on network reliability. Generally, multiple routes for a given source–terminal pair exist in a data network and the best possible route based on network metrics like hop-count, delay, traffic, queue, etc. is selected by a routing algorithm. Since network reliability incorporates all these metrics, the routing decision based on reliability seems to be the best possible option and a distributed routing algorithm based on the source–terminal (s–t) path reliability has been proposed in this work. Each node in the proposed routing generates an adjacency matrix of a network graph by periodically exchanging connection information with the adjacent nodes and selects multiple routes based on reliability of the paths. We propose an implementation of a two-path routing algorithm that instead of one includes two next-hop nodes in each node's routing table. An example is given for further illustration of the proposed algorithm.

1. Introduction

The essence of routing is a simple question – What is the best way to reach from one place to another? But it includes the hurdles embedded in modern intra-domain routing protocols. At the core of this belies the fact of continuous change of network topology. Routing for a static network is trivial. It incorporates a table of routes calculated once for each destination. But the scenario for most of the real networks is different. The network links go up and down at various instances of time resulting to a different topology from the previous one. This requires the nodes to recalculate their routes. This problem in turn can be boiled down as the size of the network grows. With the increase in network size, the change in topology also increases and this change can become problematic, because for a large network the routing updates will be generated more often. This in turn necessitates more frequent route updates and route re-computation. Thus, every router in the network has to bear the cost. This means the most resource constrained router has to effectively determine all the routes. Moreover, the above scenario gets worsen if routing is based on single path. Major routing schemes typically focus on discovering a single optimal path for routing according to some desired metric. These metrics mainly include hop, distance, speed, delay, cost, etc. Accordingly, packets are always routed over a single path, which often results in substantial waste of network resources. An alternate approach is multi-path routing. It distributes the packets among several "good" paths instead of routing all the packets along a single "best" path.

Multi-path routing can be fundamentally more efficient than the currently used single path routing protocols. It can significantly reduce congestion in hot spots by deviating traffic to unused network resources. This improves network utilization and provides load balancing [1], thereby improving the response time. Moreover, congested links usually result in poor performance. For such circumstances multi-path routing can offer steady and smooth data streams [2]. Multi-path routing algorithms that optimally split traffic between a given set of paths have been investigated in the context of flow control in [3–5].

☆ Reviews processed and approved for publication by Editor-in-Chief Dr. Manu Malek.

* Corresponding author. Tel.: +91 9308769191.
E-mail addresses: elle.est.mou@gmail.com (M. Dasgupta), gpbiswas@gmail.com (G.P. Biswas).

0045-7906/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2012.04.013

Yet, the selection of the routing paths is another major design consideration that has a drastic effect on the resulting performance. Although many flow control algorithms are optimal for a given set of routing paths, their performance can significantly differ for different sets of paths. Therefore, two key questions that arise in multi-path routing are how many paths are needed and how to select these paths. Clearly, the number and the quality of the paths selected dictate the performance of the multi-path routing scheme. There are several reasons why it is desirable to minimize the number of paths used for routing. The significant overhead associated with establishing, maintaining and tearing down a path is the prime reason. Also the complexity of the scheme that distributes traffic among multiple paths increases considerably as the number of paths increases.

Previous studies and proposals on multi-path routing have focused on various parameters including random path selection, latency, traffic, throughput and many more. But none of them has considered reliability as one of its parameter whereas reliability is an important performance criterion of communication networks. In [6] Nelakuditi and Zhang have proposed a hybrid approach where traffic is proportional among the paths. Their technique uses globally exchanged link state metrics for identifying a set of good paths and locally collected path state metrics for proportioning traffic among the selected paths.

More recent P2P applications such as Skype use automatic path selection. Skype [7] claims to keep multiple connections open and dynamically chooses the "best" path in terms of latency/quality. Bit torrent [8] maintains four active paths with an additional path periodically chosen at random together with a mechanism that retains the best paths as measured by throughput.

Recently a number of distributed routing protocols are proposed and some of them are now introduced. A scalable distributed routing protocol based on data-path services is proposed in [9], where routing is done by dividing services among network routers and establishing an optimal route such that (i) Services for given connection requests are executed in a given order, (ii) source–terminal pair path-cost is minimum and (iii) service processing cost is minimum. It proposes a new network architecture consisting of distributed Service Matrix Data Structure to distribute the services among routers, Distributed Service Matrix Routing (DSMR), Distributed Service Routing Protocol (DSRP) and a control plane for implementation of the routing protocol. A generalized distributed path computation technique is proposed in [10] that removes routing loops and count-to-infinity problem occurring in any distributed routing protocol like distance vector routing protocol. It uses a DIVs (distributed intermediate variables) algorithm, which is not a routing algorithm but runs on the top of any distributed algorithm for enabling distributed, light weight loop-free path computation. The basic idea of DIV algorithm is that each node is assigned a value for each destination of the network based on different schemes, for example, cost-to-destination and each node must select a successor node having lower assigned value (a node, which does not have a path to destination, must assign infinity value). It is claimed that DIV algorithm outperforms many loop preventing techniques and has the ability to maintain path during transition. In [11], an efficient distributed routing algorithm is presented based on locally responsive maximally distributed routing policies, which split the data inflow in a non-destination node among the outgoing links as a function of the current traffic density of the network. For this, it proves the existence of a globally attractive limit flow in a network under locally responsive routing policies and the locally responsive distributed routing polices are maximally robust as the resilience of the network coincides with the min-cut capacity of the network.

In recent times, the cloud-based routing and the application-aware routing are two emerging trends for implementing large scale networks to provide cloud computing and the flexible multimedia services to the users. The cloud computing is a computing environment that provides on-demand computation resources and software through Internet on pay basis and allows users to grow/shrink computation resources as needed and recycles among users. On the other hand, the multimedia services mean to design a multimedia network, where context-aware routing would be made to meet the growing demand for emerging of variety of multimedia end heterogeneous equipments in the network. These fields are new and in the growing phase, some of the routing techniques proposed are described now. An elegant technique based on Transit Portal (TP) to control routing in cloud service is proposed in [12]. The TP provides transparent connectivity between clients and Internet Service Providers (ISPs), where each client network with one layer-2 link and a Border Gateway Protocol (BGP) session to an upstream ISP terminates at TP, and the TP with one layer-2 link and a BGP session terminates to each upstream ISP. The TP acts as a tunnel through which multiplexed data packets and BGP messages for client networks are transmitted. The TP is also useful for application-aware routing, where it can make service-aware routing decision and can re-route clients in different data centers based on the change of users' demand. An application-aware routing protocol (AARP) is proposed in [13] that establishes logical connections through application layer resources like bridges, converters to perform bridging and/or format/rate conversion, respectively, where a logical connection consisting of one or more actual network layer connections formed by network routing algorithms. It proposes AARP architecture that contains application layer resources, AARP server to coordinate set up, modification and release of logical connections and yellow page directory server to track resource utilization in application layer.

In this paper, we propose a distributed multi-path routing algorithm based on reliability of the source–terminal paths. Initially, each node in a network graph gathers connectivity information from its adjacent nodes and forms an adjacency matrix of the graph. It then prepares all possible paths toward each node and calculates the reliability of the paths selected based on number of paths and hop counts (or any other cost parameter may be used). Now two best possible paths based on reliability are selected and the corresponding adjacent nodes are included in its routing table for routing information. Since the routing decision is made based on source–terminal paths, the proposed routing technique is free from loops and the count-to-infinity problem, thus it is better than any distributed routing protocol. The rest of the paper is organized in the following manner. In Section 2 the network model and estimation of reliability is given in detail. Section 3 depicts the explanation of the proposed technique for path selection with an illustrative example. In Section 4 the technique for multi-path routing with numerical results is presented and finally, the conclusion is given in Section 5.

2. Network model and reliability estimation

Networks are modeled by directed graphs (digraphs) in which the communicating entities are vertices (V) or "nodes" and the communication paths are edges (E) or "links". Network links are assumed to be failure-prone components whereas nodes are assumed to be failure-free. Link failures are mutually statistically independent. The links are unidirectional. Bidirectional communication is supported through a parallel link in the opposite direction. The network is free from self loops, i.e., links originating from one node and destined for the same node is not considered as the nodes are failure free.

Network reliability is usually expressed as the probability that a contiguous path exists between a source node and a destination node. This measure of reliability is known as source-to-terminal (s–t) reliability. Since routing decisions are done by a source node for each destination node, so the notion of s–t reliability has been considered instead of overall reliability, which stresses on the fact that every node in a network is connected to every other node.

In order to calculate s–t reliability of a network, a weight is assigned to each edge, where the weight denotes the success probability of the corresponding edge. The vertices are numbered from 1 to N and edges from 1 to n. Two terminal vertices, source and sink (destination), are denoted as s and t, respectively. The terminal pair (source-to-terminal) reliability is denoted as R_{st}. This setup is shown in Fig. A.

$$r_n = 0.9 \text{ for all } n, \quad R_{st} = 0.97119$$

A network can be represented by its adjacency matrix, $G = [g_{ij}]$, where

$$g_{\mathrm{ij}} = \begin{cases} 1, & \text{if an edge exists} \\ 0, & \text{otherwise} \end{cases}$$

An elementary event is defined as a particular realization of the network, i.e., it is the specification of the success or failure of each edge. If an edge succeeds then it is represented as '1' and if it fails then it is represented as '−1'. A '0' indicates both '1' and '−1'. An s–t path is a path from source to sink. A shortest s–t path is an s–t path with the least number of edges. An event is a success if the graph realization corresponding to every elementary event constituting that event has at least one s–t path. Otherwise, the event is a failure. If an event E has m elements, $E \equiv \{1, 2, 3, \ldots, m\}$, then the compliment event of E can be given as:

$$\overline{E} = \{\bar{1}\}\ U\ \{1\ \bar{2}\}\ U\ \{1\ 2\ \bar{3}\}\ U \cdots U\{1,\ 2,\ 3,\ \ldots,\ ,\bar{m}\}$$

This follows from De Morgan's laws and the absorption law. Let S be an exhaustive set of disjoint success events, such that the graph realization corresponding to an event contains an s–t path. Let S_i be success event i in the set S. The s–t reliability is the sum of the probability of each event in the set S.

$$R_{\mathrm{st}} = \sum_{i=1}^{|S|} \Pr\{S_i\} \tag{1}$$

The algorithm for reliability calculation is given below.

Algorithm *reli-calc()*

```
Initialization:
R_st ← 0.0
E ← [0 0... 0]
Initialize Q to contain E only
Iteration:
while Q is not empty do
  {
    delete the head event E from Q
    create network G corresponding to E
    call shortest-path-search (G, found)
      {
      if path P exists then
      found ← true
      }
    if found = true then
      {
      E1 = E|P
      call probability (E1, Pr)
      {
      compute the probability of E1
      R_st ← R_st + P_r
      }
      call complement (E1)
      }
  }
```

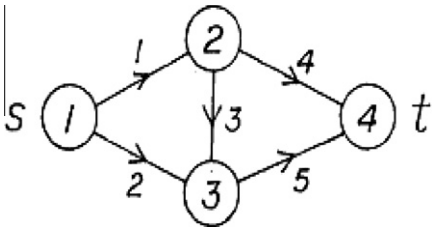

Fig. A. Network 1.

Lemma 1. *A path having p number of hop counts contribute less reliability than a path having q number of hop counts for a given source–destination pair where p > q.*

Proof. Let us consider two paths X and Y for a given source–destination pair, the hop counts are p, q respectively, where $p > q$. Now, the reliability of a path is the product of the success probabilities of its links, where the link's reliability is defined as the success probability of a link. Thus, the reliabilities $R(X)$ and $R(Y)$ for paths X and Y are given as:

$$R(X) = \prod_{i=1}^{p} \Pr(i) \quad \text{and} \quad R(Y) = \prod_{j=1}^{q} \Pr(j)$$

where $\Pr(i)$ is the probability of the link corresponding to the hop i. Each link of a network is assumed to be highly reliable, and for simplicity, if each link has equivalent reliability, then we have $\Pr(i) = r$, where r is the reliability of a link. Then, we can have,

$$R(X) = r^p \quad \text{and} \quad R(Y) = r^q$$

Since $0 < r < 1$ and $p > q$, therefore $r^p < r^q$.

Hence, the Lemma 1 follows. □

3. The proposed technique for path selection

In this section, a path selection algorithm, which is incorporated in the proposed routing technique described in the next section, has been given. As stated earlier, the selection of paths between a source node and a destination node is done based on the reliability of the paths, and the measurement of the same is described in Section 2. Since for a given s–t pair, several paths or routes with varying performance exist in a network, the routes having lower performance may be neglected to reduce the path selection overhead. In order to neglect inefficient s–t paths, we develop a heuristic technique based on comparisons of hop count of the s–t paths with hop count of the longest path or the critical path of the network (any other methods may be followed). Let the set of paths exists for a given s–t pair be P_{st}. We use a superscript to denote the path number for an s–t pair. For example, P_{st}^m denotes the mth path from source s to destination t. The hop count of each path is then calculated and let H_{st}^m denote the hop count for the mth path from s to t. A path is longer than another path if the former has greater hop count than the latter. The next step is to calculate the critical path of the network, where the critical path is defined as the longest path through the network. We consider a threshold, say Φ_T, which is half of the hop count of critical path, and all the paths between a given s–t pair that has hop count greater than Φ_T are discarded. It has been found that removal of paths based on the above technique has negligible effect on the total s–t pair reliability, i.e., reliability of a pair of nodes remains almost same even after discarding the paths having hop count greater than Φ_T. An intuitive explanation of the fact is given as follows. The s–t reliability of a node pair takes into account the number of paths between that pair and the number of hops of all the paths. As the number of paths increases the reliability of that pair also increases. That means the contributions of all the paths are considered. Consequently, all the paths do not contribute equally to the reliability computation. With the increase in hop count the reliability decreases. That is, the paths having less number of hops are more reliable than the paths having larger hop count. It may be noted that in order to identify all possible paths between any s–t pair, each node forms an adjacency matrix of a network graph by periodically exchanging the connection information with the neighboring nodes and updating the current topology information of the network. Since it is distributive in nature, so is applicable for large networks.

The path selection algorithm is given below, where the following notations are used:

s	Source node
t	Destination node
P_{st}	The set of paths from s to t
$P(s, t)$	An element of the set P_{st}

P_{st}^m	The mth path in P_{st}
$H(s, t)$	Hop count of a single path in P_{st}
H_{st}^m	Hop count for mth path in P_{st}
T	Topology of the network at time t
$CP(T)$	Critical path of T
Φ_T	Half of the critical path of T

Algorithm *path-select()*

for all (s, t) ∈ N do
  m = 0
  $P_{st} = \emptyset$
  $P'_{st} = \emptyset$
  while paths exist between s and t
    if P(s, t) then
      $m \leftarrow m + 1$
      $P_{st}^m \subseteq P_s t$
    end if
  end while
  for all P(s, t) ∈ P_{st} do
    $H_{st}^m \leftarrow H(s, t)$
  end for
end for
$CP(T) \leftarrow max\{H_{st}^m\}$
$\Phi_T \leftarrow CP(T)/2$
for all (s, t) ∈ N do
  for all P(s, t) ∈ P_{st} do
    if $(H_{st}^m \leq \Phi_T)$ *then*
      $P_{st}^m \subseteq P'_{st}$
    end if
  end for
end for

Lemma 2. *Algorithm path–select() achieves almost equal reliability of all paths if the paths having hop counts less than the threshold, Φ_T are considered.*

Proof. Let there be two paths S and T for a given source destination node pair, where hop count for S is i and hop count for T is j.

Let $i < \Phi_T$ and $j > \Phi_T$,
where $\Phi_T = 1/2$ (hop count for the critical path of the network) and $i, j \in 1, 2, \ldots, n$ natural numbers.
The reliability for the node pair from Eq. (1) can be written as:

$$R = P_r(S) + P_r(T) \tag{2}$$

Now, since S has less number of hop counts, so it is selected at first for data transfer.
Path T is opted when S in non-operational.
Therefore, Eq. (2) becomes

$$R = P_r(S) + P_r(T|\bar{S})$$

Now,

$$P_r(S) = r^i$$

and

$$P_r(T|\bar{S}) = [1 - P_r(S)]r^j = (1 - r^i)r^j$$

where r is the success probability of each edge and $0 < r < 1$.
The value of $(1 - r^i)$ is very small, so the value of $(1 - r^i)\, r^j$ becomes negligibly small and can be neglected.
Hence,

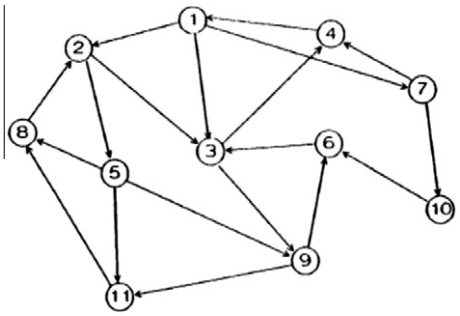

Fig. 1. The topology structure of 11 node network A.

$R \approx P_r(S)$ □

For clarity, the proposed path set up and the efficient path selection techniques are further explained by using an example as given below.

3.1. An illustration of the proposed algorithm

The algorithm *path-select()* explained above is coded in FORTRAN 77 and the simulation is done on a Xeon processor, 3.4 GHz PC with 4 GB RAM. In order to test the performance of the method, a series of computational experiments were performed using this algorithm. We consider a network with 11 nodes and 19 links, shown in Fig. 1 as an example. This network is presented originally by Belovich in [14].

Table 1 shows the adjacency matrix formed at every node. Consider any node arbitrarily, say node 1, of the network as the source node. The number of paths from node 1 to any destination node along with each path's hop count is given in Table 2. The critical path of network *A* is 10. Therefore, Φ_T is 5. Now, for all the destination nodes, node 1 discards those paths having hop count greater than Φ_T. This is depicted in Table 3. This is done for all the nodes in the network. The reliabilities of node 1 as the source to all the destinations before and after path selection are given in Table 4. It is clear from Table 4 that removing the paths having hop count greater than Φ_T has very less significance on the reliabilities of the node pairs. The reliabilities in both the cases are shown in Fig. 2.

Table 1
Input matrix of network A.

Nodes	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	0	0	0	1	0	0	0	0
2	0	0	1	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	1	0	0
4	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	1	0	1
6	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	1	0
8	0	1	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0	0	1
10	0	0	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	1	0	0	0

Table 2
Number of paths and number of hops per path.

Source–destination	1–2	1–3	1–4	1–5	1–6	1–7	1–8	1–9	1–10	1–11
Path number	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4, 5	1, 2, 3	1, 2, 3, 4	1	1, 2, 3, 4, 5, 6	1, 2, 3, 4	1	1, 2, 3, 4, 5
Hop count	1, 5, 8	1, 2, 4, 5	2, 2, 3, 5, 6	2, 6, 9	3, 3, 4, 4	1	3, 4, 4, 5, 5, 7	2, 3, 3, 5	2	3, 3, 4, 4, 6

Table 3
Number of paths and number of hops per path after removal of inefficient paths.

Source–destination	1–2	1–3	1–4	1–5	1–6	1–7	1–8	1–9	1–10	1–11
Path number	1, 2	1, 2, 3, 4	1, 2, 3, 4	1	1, 2, 3, 4	1	1, 2, 3, 4, 5	1, 2, 3, 4	1	1, 2, 3, 4
Hop count	1, 5	1, 2, 4, 5	2, 2, 3, 5	2	3, 3, 4, 4	1	3, 4, 4, 5, 5	2, 3, 3, 5	2	3, 3, 4, 4

Table 4
Reliabilities before and after removal of paths.

Source–destination	Previous reliability	New reliability
1–2	0.963354	0.959049
1–3	0.995066	0.995066
1–4	0.979495	0.978873
1–5	0.867018	0.863144
1–6	0.963720	0.963720
1–7	0.900000	0.900000
1–8	0.936882	0.931663
1–9	0.969276	0.969276
1–10	0.810000	0.810000
1–11	0.952021	0.946176

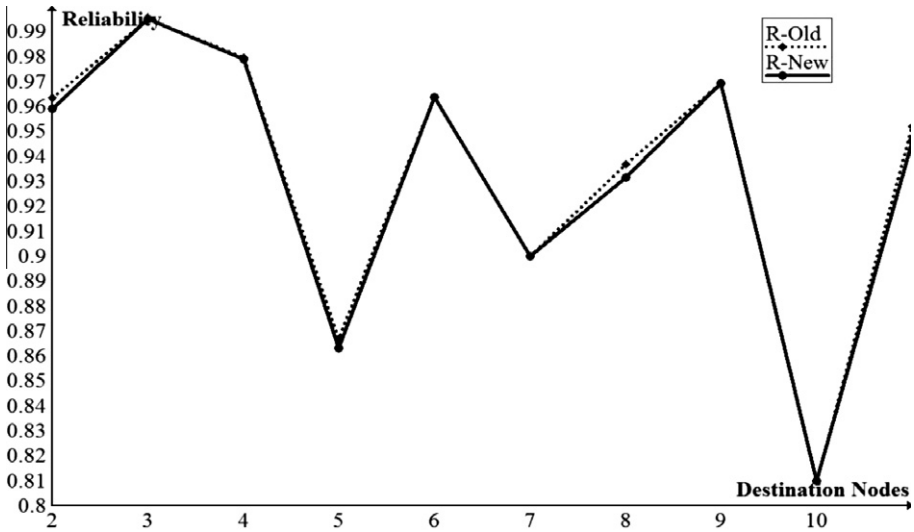

Fig. 2. Reliabilities before and after removal of paths for node 1.

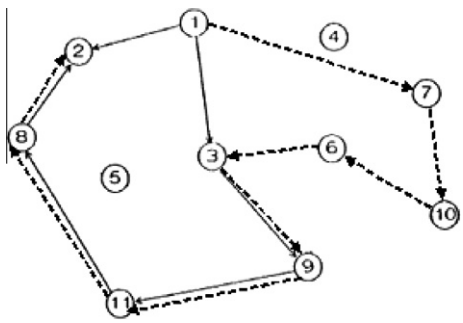

Fig. 3. Paths from node 1 to node 2.

The routes from node 1 to node 2 are shown in Fig. 3. The solid line paths are the selected paths after executing the algorithm where as the dashed line path is not considered for routing.

4. Proposed multi-path routing technique

In this section we propose a multi-path routing technique based on Section 3. First, we show that multi-path routing is better than single path routing on the reliability point of view. The proposed technique in Section 3 selects few good paths

from the set of all paths for any s–t pair and since all the selected paths are equally likely for packet forwarding, then if the packets are forwarded alternatively through any of the selected paths toward destination, then routing performance must be increased. But our path selection algorithm is solely based on reliability constraint. Lemma 1 implies that all the paths of a source to destination do not have equal contribution. Thus, in this case there exists a best path among the selected multiple paths. But if we consider only the best path, then the reliability of that s–t pair falls drastically. This is evident from the experimental results shown in Table 5. In Table 5 the reliabilities of node 1 as source node to all the destination nodes are given, where only one path for each pair is considered. The selected path is the best path for the concerned pair. The best path is selected on the basis of the minimum hop count, since reliability increases with the decrease in the number of hops. That is,

$$Q_{st}^{m} = \min\{\text{all } H(s,t) \text{ in } P_{st}\}$$

The difference of reliabilities for the single path is shown in Fig. 4.

Although it is evident from Figs. 2 and 4 that multi-path is better than single path whenever routing is considered, but there are several factors that render multi-path routing to be incorporated as a full fledged routing protocol. These factors include the overhead associated with the routing table for multiple paths. The complexity for the distribution of the traffic is another major concern. For these reasons all the major routing protocols are based on single path routing. But instead of considering a single path for the routing decision, if one more additional path is taken into account, then the reliability increases at least by a factor of two. Where as the overhead in the routing table does not increase by an equal amount. This is because if two paths are considered for a certain source to destination node pair, then in a particular snapshot, the routing table will include only two next nodes. This is shown in the following table. Table 6 is a generalized single, dual and multiple path(s) routing table.

The algorithm for two-path routing is given below:

Algorithm *two-path-routing()*

call algorithm path-select() //outputs selected path-set P'_{st}
for $(s,t) \in N$ *do*
  select two paths from P'_{st} *based on reliability*
  identify next node for each path
  insert them in routing table for same destination
end for

Thus, implementing algorithm *two-path-routing()* implies that the inclusion of an additional path minimizes congestion and delay along with the improvement in the traffic flow, throughput, reliability and hence, the overall performance. Table 7 depicts the increase in reliability when an additional path is considered to a single path routing taking node 1 as the source node.

The difference of reliabilities for two paths considered is shown in Fig. 5.

For performance analysis we consider reliability of the paths between a node pair and reliability of the node pair itself. Here, we show that the two paths selected yield the highest reliability than any other path combination for that node pair. Lemma 1 implies that a path having greater hop count has less reliability than a path having less number of hop count. Now in the proposed technique the first path has the minimum number of hop count, so it has highest reliability than all other paths. The second path chosen has the next minimum hop count hence, it has second highest reliability.

The calculation of s–t reliability requires non-operation of the first path when the second path is selected and non-operation of the first and second paths when the third path is selected and so on. Let $a, b, c, \ldots$ be the hop counts for paths $A, B, C, \ldots$ of an s–t pair, where $a < b < c < \cdots$. According to Lemma 2 the s–t reliability is given by

Table 5
Reliabilities for the best single paths.

Source–destination	Reliability
1–2	0.900000
1–3	0.900000
1–4	0.810000
1–5	0.810000
1–6	0.729000
1–7	0.900000
1–8	0.729000
1–9	0.810000
1–10	0.810000
1–11	0.729000

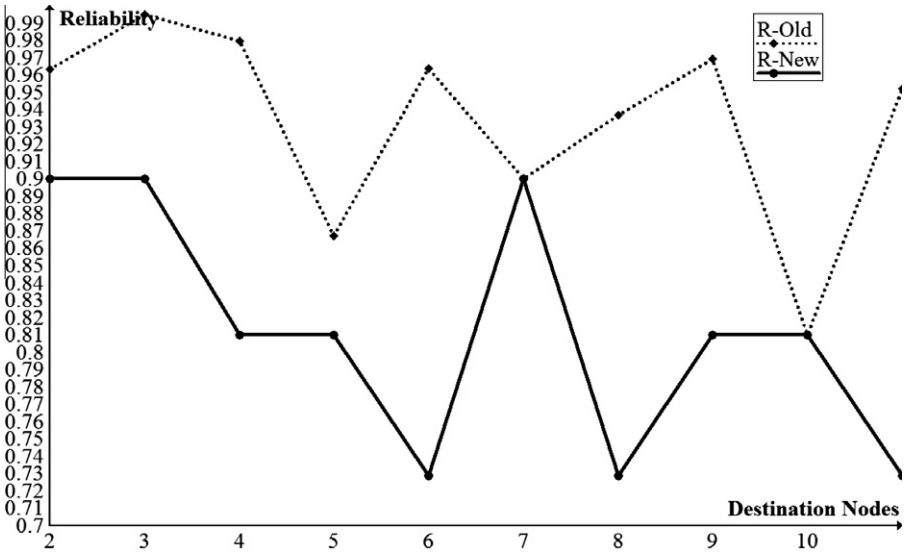

Fig. 4. Reliability difference for single paths of node 1.

Table 6
Routing table.

Routing	Destination node	Next node	Hop count
Single path	X	Y	Z
2-Paths	X	Y_1	Z_1
		Y_2	Z_2
K-Paths	X	Y_1	Z_1
		Y_2	Z_2
		.	.
		.	.
		Y_k	Z_k

$$R = Pr(1^{st}\ path) + Pr(2^{nd}\ \text{path provided non-operation of } 1^{st}\ \text{path}) + Pr(3^{rd}\ \text{path provided non-operation of } 1^{st} \text{ and } 2^{nd}\ paths) + \cdots \quad (3)$$

Let r be the success probability of each edge, then r^a, r^b, r^c, ... are the reliabilities of individual paths.

Let $X_1 = Pr(1\text{st path})$, $X_2 = Pr(2\text{nd path provided non-operation of 1st path})$ and so on. X_2 requires the edges of the 1st path to fail independently. That is, X_2 includes the term $(1 - r)$ which is very small in itself assuming that the success probability of each edge is very high. And $X_1 = r^a$. Therefore, $X_1 > X_2$. As we move further in the hierarchy the term $(1 - r)$ increases because all the previous paths should be non-operational for the next path to be chosen and hence, their values become less.

Table 7
Reliabilities for two paths.

Source–destination	Reliability
1–2	0.959049
1–3	0.981000
1–4	0.963900
1–5	0.863144
1–6	0.926559
1–7	0.900000
1–8	0.794610
1–9	0.882900
1–10	0.810000
1–11	0.926559

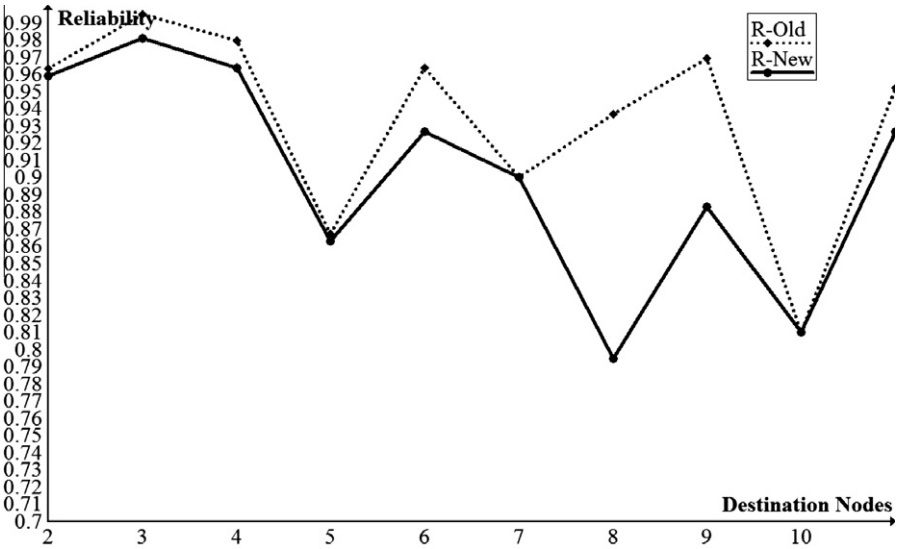

Fig. 5. Reliability difference for two paths of node 1.

Also, since $a < b < c < \cdots$ and $0 < r < 1$ therefore, $r^a > r^b > r^c > \cdots$. Hence, the two paths of the proposed technique yield the highest reliability for a node pair than any other path combination of that pair.

It may be noted that the proposed multi-path routing algorithm is distributive in nature because each node forms an adjacency matrix of a network graph by exchanging connection information with the adjacent nodes, sets up all paths for a given s–t pair using the adjacency matrix, calculates reliability of the selected paths and prepares the routing table for multi-path routing. The routing decision is made based on the measurement of s–t pair path reliability and includes two (or more) best possible paths in the routing table as the next nodes towards terminal. Since the proposed routing is s–t path-based routing, thus the loop and count-to-infinity problem as happens to be in distributed routing algorithms can be avoided. The proposed routing algorithm as stated supports the addition of two or multiple paths in the routing table of each node, thus it can reduce the node delay, congestion and balance the network loads.

Thus, the main significant contributions of the proposed multi-path routing algorithm are (i) It is distributive in nature; (ii) It provides s–t path-based routing based on the measurement of path reliability; (iii) It avoids loops and count-to-infinity problems; and (iv) It provides multiple routes between any s–t pair.

Although our routing algorithm has been proposed for routing in an AS (autonomous system), with some modifications it can be used in the cloud based and application aware routings as well. The routing in cloud based services means to provide a single best path to the users for all network services and it is done mainly by connecting the users to the different ISPs by tunneling through BGP session [15]. Since BGP is used for inter-AS routing based on the paths and network policies, our proposed routing algorithm is applicable in developing BGP session if the border routers of ASes instead of routers inside an AS execute it and instead of next hop, the whole path is included in the routing table in terms of next-node manner. On the other hand, since the application aware routing protocols consists of one or more network-layer connections routed through application-layer resources, such as bridges and/or format/rate converters, our proposed multi-path routing algorithm can be used to set up multiple network layer connections as needed in application aware routing technique through the use of best reliable source–destination paths.

The proposed multi-path routing is also applicable in software defined networking (SDN). In SDN [22,23] approach, two separate planes called control plane and the data plane are used, where the control plane contains different modules corresponding to the different network applications such as routing, QoS, and traffic engineering and the data plane comprises network hardware components like routers, switches, links, bridges, etc. and their interconnectivity as a communication network. Note that the control plane is connected to the data plane through an interface called OpenFlow. For instance, in case of routing for a given source–destination pair, the control plane performs and finalizes the routing decision, and the data plane transmits data from the source to the destination based on the routing decision made in the control plane, whereas in standard networking both the routing decision and the data forwarding are done in network layer that corresponds to the data plane in SDN. In fact, this layering abstraction introduces greater flexibility into the network and simplifies management, provisioning, experimentation and configuration of networks and/or devices. Besides, the new modules can be added to the control plane of SDN and the virtualization of the underlying networks can be done in the control plane of SDN that supports further manipulation through logical mapping and others.

Now, our proposed multi-path routing scheme can be incorporated in SDN, where a sub-module under the routing module in the control plane of SDN can be developed. This sub-module then executes and decides the multi-path routing for a

given source–destination pair and forwards the routing decisions to the NOS (network operating system) through the interface OpenFlow for establishing and forwarding the data packets over the data plane of the SDN. Since our routing scheme is designed based on the source–destination path reliability, thus if this routing scheme is incorporated in SDN, the reliability of SDN must be enhanced. It may also be noted that since the routing decision and its uses are made through control plane and the data plane of SDN, then the existence of the transport layer and its uses become unimportant, and thus the routing scheme becomes a multi-network layer routing, which finds efficient applications in cloud, application-aware and data center routing as specified earlier.

5. Conclusions

A multi-path distributed routing algorithm has been presented which calculates reliability of the selected paths for a given s–t pair for taking routing decisions, where all the paths having the hop count less than one half of the critical path of the network are identified and considered as selected paths. By doing so the reliability constraint is maintained, that is, exclusion of larger hop paths has a very negligible impact on reliability of that s–t pair. Each node identifies all possible paths of a given s–t pair based on the adjacency matrix of the network, which is formed by exchanging the topology information with the neighboring nodes only. It has also been observed that for a given s–t pair if only one best path out of all possible paths is selected as it is done in existing routing algorithms even then the reliability of that pair is considerably reduced. It has further been observed that even if two routes, instead of one or all routes, are considered for routing, the reliability for an s–t pair is considerably increased. Thus, in the present work a multi-path routing algorithm with two paths has been designed and proposed that not only enhances the routing of data packets, but also keeps the routing table size as minimum as possible. Also since the proposed technique is s–t path-based, so it guarantees that the loops and/or count-to-infinity problems do not exist. As an illustration of the proposed routing technique a network of 11 nodes and 19 links has been tested, in which multi-path routing based on the quantitative measure of reliability has been shown.

References

[1] Iyer S, Bhattacharyya S, Taft N, McKeoen N, Diot C. A measurement based study of load balancing in an IP backbone, SprintATL. Tech. Rep. TR02-ATL-051027, May 2002.
[2] Bertsekas D, Gallager R. Data networks. Englewood Cliffs, NJ: Prentice-Hall; 1992.
[3] Han H, Shakkottai S, Hollot CV, Srikant R, Towsley D. Multipath TCP: a joint congestion control and routing scheme to exploit path diversity in the Internet. IEEE/ACM Trans Netw 2006;14(6):1260–71.
[4] Kelly FP, Maulloo AK, Tan DKH. Rate control in communication networks: shadow prices, proportional fairness and stability. J Oper Res Soc 1998;49:237–52.
[5] Kelly FP, Voice T. Stability of end-to-end algorithms for joint routing and rate control. Comput Commun Rev 2005;35(2):5–12.
[6] Nelakuditi S, Zhang ZL. On selection of paths for multipath routing, vol. 92. Berlin/Heidelberg: Springer; 2001. p. 170–84.
[7] Skype web site. <http://skype.com/products/explained.html>.
[8] Cohen B. Incentives built robustness in BitTorrent. In Proceeding of P2P economics, workshop, June 2003.
[9] Huang Xin, Ganapathy Sivakumar, Wolf Tilman. A scalable distributed routing protocol for networks with data-path services. In: IEEE international conference on network protocols, ICNP 2008. p. 318–27.
[10] Ray Saikat, Guérin Roch, Kwong Kin-Wah, Sofia Rute. Always acyclic distributed path computation. IEEE/ACM Trans Network 2010;18(1):307–19.
[11] Como Giacomo, Savla Ketan, Acemoglu Daron, Dahleh Munther A, Frazzoli Emilio. Robust distributed routing in dynamical flow networks – Part I: Locally responsive policies and weak resilience. Cornell University Library; 2011. p. 1–32.
[12] Valancius V, Feamster N, Rexford J, Nakao A. Wide-area route control for distributed services. In Proceedings of USENIX annual technical conference. Boston, MA, June 2010.
[13] Veeraraghavan M, Pancha P, Eng KY. Application-aware routing protocol. In: Proceedings of second IEEE symposium on computers and communications; 1997. p. 442–8.
[14] Belovich SG. A design technique for reliable networks under a non-uniform traffic distribution. IEEE Trans Reliab 1995;44(3):377–86.
[15] Valancius Vytautas, Kim Hyojoon, Feamster Nick. Transit portal: BGP connectivity as a service. In: Proceedings of the ACM SIGCOMM; 2010.

Mou Dasgupta has obtained B.Sc. degree in Economics (honours) from Calcutta University in 2004 and Master of Computer Application from West Bengal University of Technology in 2007. Currently she is a senior research fellow in the Department of Computer Science and Engineering of Indian School of Mines, Dhanbad, and pursuing Ph.D. in Computer Science in the same university. Her main research interests include data communication networks and multi-objective optimization.

G.P. Biswas is a Professor in the Department of Computer Science & Engineering, Indian School of Mines (ISM), Dhanbad. He has around 14 years of teaching and research experience, where he teaches a number of UG/PG papers including Theory of Computation, Computer Org/Arch., Computer Networks, VLSI Designs, Cryptography and Network Security, etc. he has published around 50 research papers in Journals and Conference/Seminar Proceedings. Prof. Biswas has experience of guiding B.Tech (CSE), M.Tech (CA) and Ph.D. students. His research interest includes Cryptography, Network security, Cellular Automata (CA), VLSI designs, Computer Networks and Data Communication.