This paper was presented at RNDM 2014 - 6^{th} International Workshop on Reliable Networks Design and Modeling (Nov. 17-19, 2014, Barcelona, Spain)

Geodiverse Routing with Path Jitter Requirement under Regional Challenges

Yufei Cheng*, and James P.G. Sterbenz*†
*Information and Telecommunication Technology Center
Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS, 66045, USA
{yfcheng|jpgs}@ittc.ku.edu
† School of Computing and Communications (SCC) and InfoLab21
Lancaster LA1 4WA, UK
jpgs@comp.lancs.ac.uk
www.ittc.ku.edu/resilinets

***Abstract*—In this paper, we have extended the GeoDivRP geodiverse routing protocol to consider jitter requirements when using multiple geographic paths for telecommunication networks under regional challenges. We have formulated bounded-jitter multipath routing using a multi-commodity flow problem and proposed an integer linear programming formulation to solve it. We have implemented the routing protocol in *ns-3* to employ the optimised paths provided by the bounded-jitter optimisation solution and have demonstrated its effectiveness compared to OSPF in terms of both throughput and overall edge capacity utilisation. GeoDivRP guarantees the jitter constraint provided by the upper layer and satisfies the traffic demand imposed by multiple routing commodities in the telecommunication networks.**

***Index Terms*—network resilience; traffic optimization; multipath routing; geographic diversity; survivable routing; traffic engineering; disaster recovery**

I. Introduction and Motivation

Telecommunication networks rely heavily on physical infrastructure such as optical fibers, amplifiers, routers, and switches to maintain normal operation, and their resilience to various faults and challenges is important to be analysed [1]. Survivable optical networks under random edge and non-correlated failures has been a popular research domain [2], [3]. Recently the research community has become more concerned about the potential damage caused by large-scale challenges and intentional attacks; efficient mechanisms have been proposed to mitigate their impacts [4]–[7]. However, none of these works considers traffic allocation for regional challenges or attacks with a large impact zone, i.e., an earthquake or hurricane that has a challenged radius of up to 500 miles, which can cause failed nodes and links with substantial damage to the normal network communications [8].

It has been observed that a large number of failures in a geographical region can result in catastrophic damage to network communications [7]. When regional challenges or attacks occur, a series of nodes and links in the vicinity can be damaged and removed from the network. Since the challenge effect is frequently long-term [8], a set of backup paths are required for survivable routing. Previous work has studied different vulnerability area identification mechanisms and routing algorithms to reroute around the impact zone with a provided threat model [7]; it has formulated two heuristics for solving the d-distance separation paths (in which any two vertices on disjoint paths are separated by greater than d distance) problem and demonstrated its effectiveness under regional challenges [9]. However, traffic allocation and jitter minimisation are not considered; it is compelling to understand the mechanism to statistically direct the rerouting traffic onto multiple distance d-separated paths and to better cope with network congestion when large-scale challenges occur.

Multipath routing has been widely studied as one effective mechanism to reduce congestion in *hot spots* by deviating traffic to other unused network resources [10]–[12]. When regional challenges occur, the rerouted traffic has the tendency to share common edges in the vicinity of the threat zone and a large amount of rerouting traffic would substantially increase the possibility of congestion. Multipath mechanisms can minimise the after-challenge traffic impact on the *hot* edges as well as the whole network. Furthermore, splitting traffic onto different paths strategically can reduce the congestion level while at the same time provide more throughput.

Most of the traffic allocation and multipath routing studies make an assumption about normal network connectivity [10], [13], in which the traffic allocation follows the widest paths disjoint with respect to the bottleneck edges, in which the bottleneck links from multiple paths are mutually disjoint [14]. Previous work has shown that by deviating from shortest path routing, a multipath protocol can achieve more efficient network utilisation by using optimal traffic distribution mechanisms [13]. One other work has formulated an optimisation problem to model the routing issues in a multi-source-sink multipath routing environment, and it leads to a pseudo-polynomial algorithm based on linear programming in the network with a bounded buffer size and jitter constraint [15], [16]. For jitter traffic analysis in the multipath routing context, previous work calculates the k-shortest paths and selects paths that meet the jitter requirement, which are used to solve the optimisation problem [16]. However, most of them have focused on multipath routing without challenges. With

978-1-4799-7039-1 ©2014 IEEE

increasing requirement on the resilience of the network under large-scale challenges or attacks, it is imperative to analyse the multipath routing efficiency and understand the traffic allocation requirements under these challenges.

In this paper, we formulate an optimisation problem in a physical network under regional challenges to minimise the delay skew (jitter) among the multiple paths calculated for each vertex pair. The solution provides better edge traffic utilisation and throughput compared to OSPF with shortest path routing. Jitter is used to denote the maximum delay variance between multiple paths for each vertex pair. With the calculated bounded-jitter d-separated paths using the iW-PSP (iterative WayPoint Shortest Path) heuristic for Geo-DivRP [9], our protocol increases the throughput compared to single path routing under large-scale network challenges. Previous works have studied the bounded buffer problem but have assumed a maximum path-length constraint. Our heuristic does not restrict the maximum path length since it may lead to no usable jitter-bounded paths. We argue that for physical topologies, it is not necessary to set an upper bound for path length as the network diameter (longest shortest path for all vertex pairs) is small for a mesh-like topology [17].

We introduce a trade-off parameter δ to control the path stretch, outage risk, and delay variations among multiple paths calculated for the same vertex pair. It balances between short path stretch with high outage risk and the long path stretch with low outage risk. Furthermore, it controls the jitter value between multiple GeoPaths. This is achieved by controlling the d-distance separated paths provided by the GeoDivRP using the iWPSP heuristic. In controlling the delay variation, δ can be either increased or decreased to provide paths with the required jitter value and route around the challenged area.

It is rarely feasible to conduct network experiments on a production network, especially at a national scale. Network researchers resort to simulations to study their ideas and proposals. In this paper, we use *ns-3* [18] simulation software to study our protocol. We use the MATLAB optimisation toolkit [19] for solving the integer programming problem and use real-world network topologies from KU TopView [20] [21]. The same scope of physical challenge results in different damage levels to the network at different locations. Therefore, we choose the challenge locations identified in the previous work [7].

We extend our GeoDivRP routing algorithm to provide distance d-separated paths as well as the optimal traffic allocation on the multiple paths for all the vertex pairs or commodities. We formulate the problem using a linear programming model with the paths provided by a modified iWPSP routing heuristic explained in Section II. When the network is under regional challenges, the rerouted traffic has a limited number of backup paths from which to select, which raises the potential danger for the network to be congested. We consider a multipath scenario and it entails the synchronisation of different paths with a bounded-jitter requirement. We consider the problem of establishing bounded-jitter multiple GeoPaths with a given demand matrix when the challenge occurs. Our model assumes weighted edges and we calculate the maximum throughput the multipath routing can achieve. We have formulated our problem as a multi-commodity flow problem.

A. Multi-commodity flow problem background

The multi-commodity flow problem is a network flow problem with multiple flow demands for vertex pairs between each pair of source and sink vertices. This problem originates from the fact that there are multiple demands to be fulfilled in the network simultaneously and they compete for network resources. In this paper, we use commodities and vertex pairs interchangeably. There are W commodities defined by $W^w = (s^w, t^w, d^w)$, where s^w and t^w are the source and sink of the commodity w, d^w is the traffic demand. We use the Link-Path formulation [22] of the problem with one more jitter constraint and further present an integer linear programming formulation to solve the problem. We do not use the other popular Link-Node formulation since GeoDivRP needs to dictate the distance between different paths, and using the Link-Node formulation would introduce extra complexity when forming paths created with a d-distance separation criteria.

In the following sections, we introduce our model and problem formulation in Section II. We present our simulation results with flow optimisation in Section III-A. Section IV concludes the paper and suggests future work.

II. MODEL AND PROBLEM FORMULATION

Our GeoDivRP fits in the protocol stack as shown in Figure 1 [9]. *Knobs* $\mathbb{K}$ are used by higher layers to influence lower layer operation while *dials* $\mathbb{D}$ are the mechanisms for lower layers to provide feedback to higher layers [23]. The application layer passes a service specification and threat model down to our resilient transport layer protocol ResTP [24], [25]. Upon receiving these parameters, ResTP determines the type of transport service needed (including error control and multipath characteristics) and requests that GeoDivRP calculate geodiverse paths that meet the requirement tuple $(k, d, [h, t])$, where k is the total number of geodiverse paths requested, d is the distance separation criteria, $[h, t]$ are the desired constraints on path stretch h and temporal skew across paths t. ResTP then establishes a multiflow with error control needed for meet the service spec, including the per-subflow error control (ARQ, hybrid ARQ, FEC, or none) and flow bundle (e.g. 2-of-3 erasure code for real-time critical service, or 1+1 redundancy with a hot-standby for delay and loss tolerant service) taking advantage of k d-geodiverse paths $[P]$ provided by GeoDivRP. We apply the multipath algorithm in the context of several real-world service provider networks to analyse the diversity gain and packet delivery ratio when GeoDivRP is considered. We further extend this routing mechanism to establish an efficient algorithm that calculates GeoPaths to satisfy both the bounded-jitter requirement as well as the traffic demand matrix.

A network is represented by a connected directed graph $G(V, E)$, where V is the set of vertices and E is the set of edges.

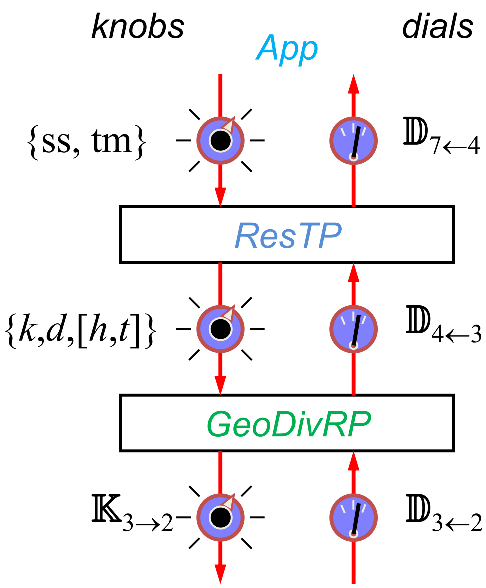

Fig. 1. Layered block diagram of GeDivRP and ResTP

Path is defined as a vector that contains finite sequence of intermediate vertices V and edges E from source s to sink t

$$P = V \cup E \tag{1}$$

We represent a path as the sequence of vertices $P = (v_0, v_1, ...v_h)$, such that, for $0 \leq n \leq h-1, (v_n, v_{n+1}) \in E$. Each path P has an associated cost $c(P)$ which denotes its cost per unit flow. We assume the paths bounded by jitter have the same cost for the same commodity. Each edge is associated with a capacity u_{ij} that denotes the maximum amount of flow on that path and a lower bound l_{ij} that denotes the minimum amount. For most of the cases, the lower bound is 0.

We consider the link-state routing environment, in which each vertex maintains a map of the network edge interconnection, link-state advertisements are flooded throughout the network and all vertices compute their paths based on the updated map. We explore the edge congestion factor to understand how GeoDivRP utilises network resources. Edge flow x_{ij} (when $(i,j) \in E$) is defined as the total flow that has been assigned on this particular edge after optimisation. The value x_{ij}/u_{ij} is the edge congestion factor and the value $\max_{(i,j)\in E} x_{ij}/u_{ij}$ is the network congestion factor. In this paper, we allow the edge congestion factor to have values larger than 100%. The reason is that the simulation model is not using buffers for intermediate vertices, and all the extra data packets assigned to an overloaded edge would be dropped. The assumption facilitates the representation of overloaded edges for OSPF with shortest path routing.

For delay variation or jitter, we only consider the difference in the number of hops among the multiple paths. We argue that the difference for propagation delay on different fiber edges is minimal since the physical topology is more mesh-like and long edges are not favored due to cost [17]. We denote θ as the number-of-hop difference among multiple paths for one vertex pair.

Jitter-Bounded GeoPath Given previous definitions of $P_1(i,j)$ and $P_k(i,j)$ for vertex pair (i,j)

$$L(P_i) - L(P_j) \leq \theta \quad \text{for all } i,j = 1,2,...,K, \tag{2a}$$

$$d_{\text{g}}(P_1(i,j), P_k(i,j)) = d_{\min} \quad \text{for all } k = 1,2,...,K, \tag{2b}$$

where $d_{\min}$ is the minimum distance between any vertex member of the path vector $P_1(i,j)$ and that of $P_k(i,j)$, and K refers to different number of GeoPaths for each commodity. $L(P_i)$ and $L(P_j)$ are the length of the paths. $d(P_i, P_j)$ represents the minimum distance between two paths P_i and P_j.

Equation 2a defines the jitter requirement for k GeoPaths provided by the optimisation problem. Equation 2b defines the geodiversity among those paths.

For each commodity w, let P^w denote the collection of all directed paths from the source vertex s^w to the sink vertex t^w. In the path flow formulation, each decision variable $f(P)$ is the flow on some path P and for the wth commodity. We define this variable for every directed paths in P^w.

$$\eta_{ij}(P) = \begin{cases} 1 & \text{if } (i,j) \in P \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Let $c^w(P)$ be the cost of path $P \in P^w$. Let $\eta_{ij}(P)$ be an edge-path indicator variable, that is, $\eta_{ij}(P)$ is 1 if edge(i,j) is contained in the path P, and is 0 otherwise.

We include the important parameters used in Table I:

The optimisation problem using linear programming given the previous definition is shown below:

$$\text{minimize} \sum_{1\leq w\leq W} \sum_{P\in P^w} c^w(P) f(P) \tag{4}$$

subject to

$$\sum_{1\leq w\leq W} \sum_{P\in P^w} \eta_{ij}(P) f(P) \leq u_{ij} \quad \text{for all } (i,j) \in E, \tag{5a}$$

$$\sum_{P\in P^w} f(P) = d^w \quad \text{for all } w = 1,2,...,W, \tag{5b}$$

$$f(P) \geqslant 0 \quad \text{for all } w = 1,2,...,W, \text{ and all } P \in P^w, \tag{5c}$$

$$L(P_1(s^w,t^w)) - L(P_k(s^w,t^w)) \leq \theta; \ \forall P_k(s^w,t^w) \text{ and all } w = 1,2,...,W, \text{ and all } k = 1,2,...,K, \tag{5d}$$

TABLE I
PARAMETER LIST

Parameter	Description
$G(V,E)$	input graph $\|G\|$ with a set of vertices V
P	Path
$c(P)$	associated cost per unit flow
u_{ij}	maximum amount of flow on path
l_{ij}	minimum amount of flow on path
x_{ij}	total flow assigned on edge ij
$L(P)$	length of the path P
S	source vertex
D	destination vertex
S_n	neighbor vertex chosen by source vertex
D_n	neighbor vertex chosen by destination vertex
k	number of geodiverse path requested
d	distance separation between each and every vertex in different disjoint paths
δ	delta distance when selecting waypoint vertex (threat model)
$f(P)$	flow on path P
d^w	demand of commodity w

The objective function shown in Equation 4 minimises the cost of flows over different paths for all of the commodities. We assume the cost is the same on different jitter-bounded GeoPaths for the same commodity, which means if the GeoPaths satisfy the jitter constraint in Equation 5d, then $c^w(P)$ is the same for all the paths in that commodity. Equation 5a is the edge capacity utilisation constraint for each edge (i,j) which states that the sum of the path flows passing through the edge is at most the capacity of that edge, u_{ij}. Equation 5b is the traffic demand for all the vertex pairs. For each commodity w, it states that the total flow on all the paths connecting the source vertex s^w and sink vertex t^w must equal to the demand d^w. Equation 5c rules out non-feasible flows, and restricts all variables to be non-negative. Equation 5d ensures that the paths satisfy the jitter requirement for that commodity.

All the candidate paths for the optimisation problem are provided by the iWPSP routing heuristic. The paths returned from the iWPSP are simple paths, with the parameter δ controlling the jitter result for different GeoPaths. The jitter requirement θ is passed in along with the other requirement tuples. As shown in Figure 2, when $k=2$, iWPSP first selects neighbor vertices S_1 and D_1 that are d-distance separated from source vertex S and destination vertex D, respectively (for simplicity in this presentation we assume that such vertices exist; otherwise the vertices with the greatest distance will be chosen, iterating until vertices d apart are located). Assuming the shortest path connecting S and D is L, iWPSP selects waypoint vertices m' and m'' in the opposite direction that are distance $d+\delta$ apart from the middle vertex m in the shortest path, where the segment $m'mm''$ intersects the shortest path. Dijkstra's algorithm is performed for the two branches S_1m' and D_1m'. By connecting the shortest path returned from the two branches, the heuristic obtains the first geodiverse path P_1. The same mechanism repeats for waypoint vertex m'' for the second geodiverse path. The variable d is a user-chosen parameter based on the threat model, and δ is experimentally chosen

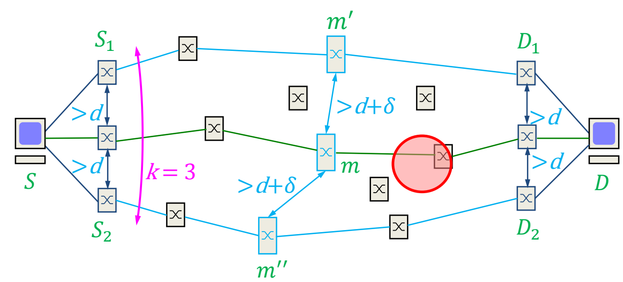

Fig. 2. Iterative waypoint shortest path heuristic

for different network topologies to increase the probability of the heuristic to return a d-separation path. The δ parameter is also introduced to prevent the edges in each of the two paths from interleaving and creating routing loops. By tweaking the value of δ, the heuristic can select a nearby waypoint vertex if the previous one fails running Dijkstra's algorithm. When it cannot selects paths within the jitter bound θ, it increases or decreases the value of δ accordingly. The code of iWPSP is shown in Algorithm 1.

This heuristic naturally controls the jitter for different paths in different commodities with the introduction of δ. By slightly increasing or decreasing the δ value along one direction of path calculation, we can alter the jitter value of the returned GeoPaths. If the paths returned are not bounded by the provided jitter requirement, iWPSP uses a different δ value to find another set of GeoDiv paths.

A. Complexity analysis

We study the complexity for calculating d-distance separated paths as well as the traffic optimisation problem. iWPSP has a complexity of $2c^2n^2\log n$, where c is the average number of neighbors for vertices, the complexity for choosing the waypoint vertex is $O(n)$, where n represents the number of vertices, and $2n\log n$ is for Dijkstra's algorithm to calculate the two shortest paths. Therefore, the worst case scenario is

Functions:
Calculate k number of geographically d-distance separated jitter-bounded paths
Input:
G_i:= input graph
S_k:= source vertex
D_k:= destination vertex
δ:= delta distance when selecting waypoint vertex
$[d, T, b, \theta]$:= requirement tuple
begin
segment T connecting S and D, with its middle point m;
choose neighbor vertex S_k, D_k that is at least d distance from S_{k-1}, D_{k-1}, respectively;
if *k is odd number* **then**
choose two vertices m_1 and m_2 that are separated by $d + \delta$ on each direction of T, where $m_1 m m_2$ is perpendicular bisector of T;
P_1 = SourceTree$_{DS}$ $\leftarrow$ Dijkstra(D, S);
$k- = 3$;
else
choose two vertices m_1 and m_2 that are separated by $d/2 + \delta$ on each direction of T, where $m_1 m m_2$ is perpendicular bisector of T;
$k- = 2$;
end
$p_{m_1 S_1}$ = SourceTree$_{S_1 m_1}$ $\leftarrow$ Dijkstra(m_1, S_1);
$p_{m_2 S_2}$ = SourceTree$_{S_2 m_2}$ $\leftarrow$ Dijkstra(m_2, S_2);
$p_{m_1 D_1}$ = SourceTree$_{D_1 m_1}$ $\leftarrow$ Dijkstra(m_1, D_1);
$p_{m_2 D_2}$ = SourceTree$_{D_2 m_2}$ $\leftarrow$ Dijkstra(m_2, D_2);
while $k > 0$ **do**
segment T = newest established path;
choose one vertex m_k that is separated by distance $d + \delta$ from T on the farther direction from the absolute shortest path;
$p_{m_k S_k}$ = SourceTree$_{m_k S_k}$ $\leftarrow$ Dijkstra(m_k, S_k);
$p_{m_k D_k}$ = SourceTree$_{m_k D_k}$ $\leftarrow$ Dijkstra(m_k, D_k);
$k- = 1$;
end
if *k is odd number* **then**
$P_2 = p_{m_1 S_1} + p_{m_1 D_1}$;
$P_3 = p_{m_2 S_2} + p_{m_2 D_2}$;
...
$P_k = p_{m_{k-1} S_{k-1}} + p_{m_{k-1} D_{k-1}}$;
remove path that fails the jitter requirement.;
else
$P_1 = p_{m_1 S_1} + p_{m_1 D_1}$;
$P_2 = p_{m_2 S_2} + p_{m_2 D_2}$;
...
$P_k = p_{m_k S_k} + p_{m_k D_k}$;
remove path that fails the jitter requirement.;
end
return $(P_1, P_2, ..., P_k)$
end
Algorithm 1: Iterative waypoint shortest path heuristic

$O(n^2 \log n)$ while the best case scenario is $O(n \log n)$. Most of the physical topologies have an average degree between two and three [17]. This means that c in our complexity analysis is a small constant. This reduces the best case time complexity of iWPSP to $O(n \log n)$. The complexity for solving the linear program is polynomial. Therefore, the complexity of the GeoDivRP routing is dominated by the complexity of GeoPath calculation.

III. Model Implementation and Simulation Results

We have used *ns-3* [18] to implement the GeoDivRP routing protocol and GeoPaths calculated from the routing protocol are passed to the optimisation toolkit in MATLAB [19]. After solving the optimisation problem, the GeoPaths and flow allocation information on each path are returned to the *ns-3* simulation. These paths are used for the network data transmission and guarantee the traffic demand for all commodities. This mechanism ensures that the paths can achieve the optimised throughput and bounded-jitter for data transmission.

The steps for the routing algorithm to calculate the jitter-bounded GeoPaths is shown as follows:

- obtain the GeoPaths using iWPSP routing heuristic for each vertex pair that satisfies the jitter constraint and d-distance separation criteria
- multi-commodity flow optimisation using the linear programming formulation (ILP) in MATLAB
- GeoPaths returned from optimisation are used for data transmission in *ns-3* network simulations

A. Simulation Results

We have presented the simulation result of GeoDivRP with optimised jitter-bounded GeoPaths using the Level 3 [26] and Sprint [17] physical networks, and compared with the performance of OSPF. We carry out the simulation once for each topology since there is no randomness in a given service provider topology. The capacity for all the edges is 5 Gb/s, and we use CBR (constant bit rate) traffic, sending from each vertex to all the others at a data rate varying uniformly from 1 Mb/s to 12 Mb/s as the traffic demand. The varying demand in different networks is to evaluate and demonstrate the maximum traffic demand GeoDivRP can provide. We use the same challenge area at Kansas City identified in our vulnerability area identification mechanism [7] with a 300 km challenge range.

The Level 3 physical network contains 99 vertices and 132 edges. In Figure 3, we present the throughput improvement from traffic optimisation using GeoDivRP. The x-axis shows the traffic demand in Mb/s, and the y-axis presents throughput for each vertex-pair delivered in the simulation. When the traffic demand for each vertex pair is below 6 Mb/s, each edge has more than enough bandwidth than required; the shortest path algorithm for OSPF can deliver that demand for all the commodities. However, when the demand increases to 7 Mb/s, OSPF cannot provide the increasing traffic demand and the throughput drops. This is due to the fact that the shortest path routing tends to over-utilise edges around the challenge

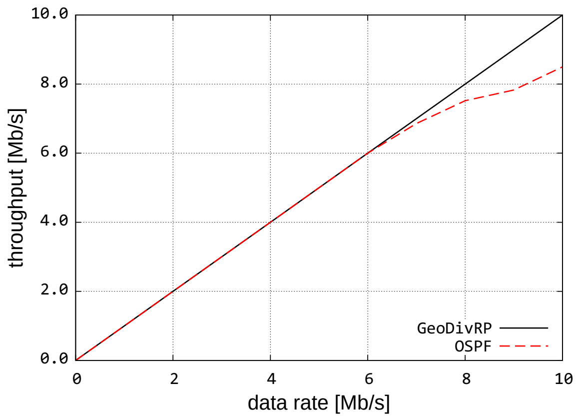

Fig. 3. Level 3 network average throughput

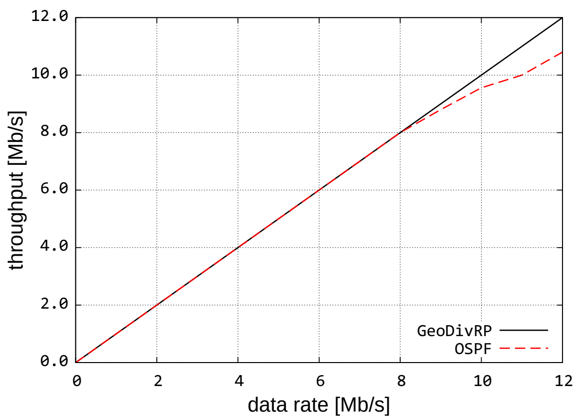

Fig. 4. Sprint network average throughput

area, which causes network congestion. On the other hand, GeoDivRP can provide the traffic demand up to 10 Mb/s for each commodity. The optimisation formulation cannot provide optimal solution beyond 10 Mb/s for Level 3. The reason is that the demand cannot be met with the current network conditions. We are planning to develop a traffic allocation heuristic for the demand beyond the optimisation range in our future work.

As shown in Figure 4, the throughput improvement for the Sprint network shows a similar trend as Level 3. When the traffic demand is 12 Mb/s, the throughput improvement for GeoDivRP compared to OSPF is 1.2 Mb/s for each commodity. The Sprint network contains 77 vertices and 114 edges. If we consider all the commodities of $77 \times 76 = 5852$ vertex pairs, the total throughput improvement for all vertex pairs provided by GeoDivRP is 7022.4 Mb/s greater than that for OSPF.

We record the time for solving the optimisation problem in different physical topologies. It is calculated when the challenge happens in the Kansas City area with a traffic demand of 10 Mb/s for each commodity. As shown in Table II, the maximum time for the optimisation is for the Sprint

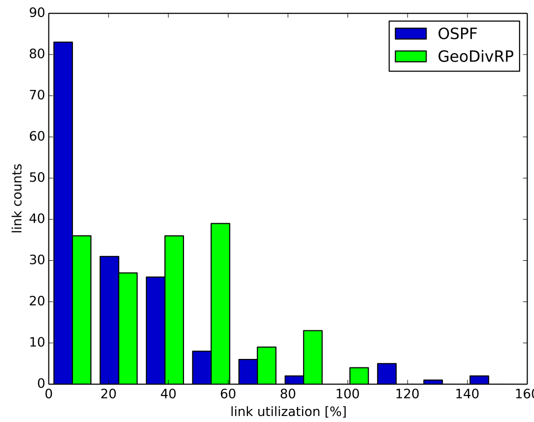

Fig. 5. Edge utilisation in Level 3 network

network is about 7 seconds, while most of the others take less than 1 second. The evaluation is on a Linux machine with a 3.16GHz Core 2 Duo CPU and 4GB memory.

We further compare GeoDivRP to OSPF in terms of the overall edge congestion factor. We define the edge congestion factor and network congestion factor as the percentage of the bandwidth that has been taken by the network flows and network congestion factor is the maximum edge congestion factor for all the edges [13] [27], which has proven to be a good indicator for network congestion. The optimisation problem we have introduced is not specifically minimising the network congestion factor. Therefore, some edges are still being used to 100% edge capacity. However, since we specify the capacity upper bound on path flows, the traffic does not congest any network edge and uses the network resources efficiently. For OSPF, on the other hand, it always chooses the shortest path, and does not consider the remaining network resources on the any network edge, which causes network congestion.

In Figure 5, we present the edge congestion factor for the Level 3 network when the demand is 10 Mb/s for each vertex pair. GeoDivRP has distributed the traffic load to all the network edges and does not overload the edges. However, OSPF has used some edges up to 140%, which means that for each of the overloaded edges, the data traffic for 40% of the edge capacity will be dropped. We allow the links to exceed the edge capacity simply to compute the dropped packets. Since the edge capacity is 5 Gb/s, 2 Gb of traffic is dropped each second on these edges. This causes significant traffic loss to the network, and it is especially damaging when the network is under large-scale challenges. The dropped traffic could be buffered but the end-to-end delay would increase exponentially. The network congestion factor for GeoDivRP is 100% while that of OSPF is 140%.

This edge congestion analysis has demonstrated that GeoDivRP can allocate traffic to edges efficiently and avoid overloading any network edge, while OSPF over-utilises edges causing the data packets to either dropped or buffered suffering

TABLE II
TIME FOR OPTIMISATION ALGORITHM

Network	Number of Nodes	Number of Links	Number of Failed Nodes	Optimisation Time (s)
CORONET	75	99	2	0.62
Internet2	57	65	1	0.04
Level 3	99	132	4	2.06
Sprint	77	114	3	6.96
TeliaSonera	18	21	1	0.02

increased end-to-end delay.

IV. Conclusion and Future Work

We have proposed the GeoDivRP routing protocol with traffic demand and the bounded-jitter requirement. We have generated an Integer Linear Programming (ILP) formulation of the problem and solved it. We have incorporated the optimised GeoPaths in the GeoDivRP and have compared our protocol performance with OSPF in terms of both throughput and overall edge congestion. Our protocol shows considerably better performance than OSPF.

We argue that GeoDivRP is adequate in face of large-scale challenges. First, the iWPSP routing heuristic returns d-distance separated paths with controlled algorithm and time complexity. Second, the jitter-bounded requirement guarantees the optimised traffic allocation along different paths with little extra complexity and delay. Finally, our previous paper [9] presents improved packet delivery ratio (PDR) and delay when compared to OSPF, this paper presents better edge utilisation compared to OSPF.

For future work, we plan to incorporate the non-linear optimization problem for controlling the average network congestion and delay. We will explore different capacity planning mechanisms for regional network challenges. We will extend the GeoDivRP optimisation algorithm to return the best allocation result even when the ILP formulation has no solution. We are planning to introduce more queuing mechanisms to *ns-3* and examine the GeoDivRP performance with the delay function introduced.

Acknowledgment

The authors would like to thank the members of the ResiliNets group for discussions which led to this work. This research was supported in part by NSF grant CNS-1219028 (Resilient Network Design for Massive Failures and Attacks).

References

[1] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, “Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines,” *Computer Networks*, vol. 54, no. 8, pp. 1245–1265, 2010.

[2] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, “Resilience of the Internet to Random Breakdowns,” *Phys. Rev. Lett.*, vol. 85, pp. 4626–4628, November 2000.

[3] D. Magoni, “Tearing down the internet,” *IEEE J.Sel. A. Commun.*, vol. 21, pp. 949–960, September 2006.

[4] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, “Assessing the Impact of Geographically Correlated Network Failures,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pp. 1–6, 2008.

[5] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, “Assessing the Vulnerability of the Fiber Infrastructure to Disasters,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1610–1623, 2011.

[6] S. Neumayer and E. Modiano, “Network reliability with geographically correlated failures,” in *Proc. of IEEE INFOCOM*, pp. 1–9, March 2010.

[7] Y. Cheng, J. Li, and J. P. G. Sterbenz, “Path Geo-diversification: Design and Analysis,” in *Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, (Almaty), September 2013.

[8] M. J. F. Denise M. B. Masi, Eric E. Smith, “Understanding and Mitigating Catastrophic Disruption and Attack,” *Sigma Journal*, pp. 16–22, September 2010.

[9] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, and J. P. Sterbenz, “Optimised heuristics for a geodiverse routing protocol,” in *Proceedings of the IEEE 10th International Workshop on the Design of Reliable Communication Networks (DRCN)*, (Ghent, Belgium), April 2014.

[10] C. C. Aggarwal and J. B. Orlin, “On multiroute maximum flows in networks,” *Networks*, vol. 39, no. 1, pp. 43–52, 2002.

[11] B. H. Shen, B. Hao, and A. Sen, “On multipath routing using widest pair of disjoint paths,” in *High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on*, pp. 134–140, 2004.

[12] C. Hopps, “Analysis of an Equal-Cost Multi-Path Algorithm.” RFC 2992 (Informational), Nov. 2000.

[13] R. Banner and A. Orda, “Multipath routing algorithms for congestion minimization,” *Networking, IEEE/ACM Transactions on*, vol. 15, pp. 413–424, April 2007.

[14] S. Nelakuditi and Z.-L. Zhang, “On selection of paths for multipath routing,” in *Proceedings of the 9th International Workshop on Quality of Service*, IWQoS '01, (London, UK, UK), pp. 170–186, Springer-Verlag, 2001.

[15] T. Anjali, A. Fortin, and S. Kapoor, “Multi-commodity network flows over multipaths with bounded buffers,” in *Communications (ICC), 2010 IEEE International Conference on*, pp. 1–5, May 2010.

[16] T. Anjali, A. Fortin, G. Calinescu, S. Kapoor, N. Kirubanandan, and S. Tongngam, “Multipath network flows: Bounded buffers and jitter,” in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–7, March 2010.

[17] E. K. Çetinkaya, M. J. F. Alenazi, Y. Cheng, A. M. Peck, and J. P. G. Sterbenz, “On the Fitness of Geographic Graph Generators for Modelling Physical Level Topologies,” in *Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, (Almaty), pp. 38–45, September 2013.

[18] “The ns-3 network simulator.” http://www.nsnam.org, July 2009.

[19] “Mathworks - matlab and simulink for technical computing.” http://www.mathworks.com, 1994.

[20] J. P. Rohrer, M. J. F. Alenazi, and J. P. G. Sterbenz, "ResiliNets Topology Map Viewer," January 2011.

[21] J. P. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, Q. Shi, and J. P. Rohrer, "Evaluation of Network Resilience, Survivability, and Disruption Tolerance: Analysis, Topology Generation, Simulation, and Experimentation (invited paper)," *Telecommunication Systems*, vol. 52, no. 2, pp. 705–736, 2013.

[22] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[23] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Redundancy, Diversity, and Connectivity to Achieve Multilevel Network Resilience, Survivability, and Disruption Tolerance (invited paper)," *Springer Telecommunication Systems*, 2012. (accepted April 2012).

[24] J. P. Rohrer, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "Airborne telemetry networks: Challenges and solutions in the ANTP suite," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, (San Jose, CA), pp. 74–79, November 2010.

[25] J. P. Rohrer, R. Naidu, and J. P. G. Sterbenz, "Multipath at the transport layer: An end-to-end resilience mechanism," in *Proceedings of the IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, (St. Petersburg, Russia), pp. 1–7, October 2009.

[26] "Level 3 network map." http://maps.level3.com.

[27] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS." RFC 2702 (Informational), Sept. 1999.