

# On the Use of Genetic Algorithms to Fine-tune OSPF Routing

Antoine B. Bagula and Hong F. Wang  
Department of Computer Science  
University of Stellenbosch  
7600 Stellenbosch  
South Africa  
Tel: +27 21 808 4070 Fax: +27 21 808 4416  
Email: bagula@cs.sun.ac.za

**Abstract**— This paper revisits the problem of Traffic Engineering (TE) to evaluate the performance improvements achieved by destination-based TE when OSPF routing is fine-tuned to achieve route optimisation. We build upon the framework provided by evolutionary algorithms to present a TE model where link weights are computed by a genetic algorithm and used as fine-tuning parameters in destination-based path selection. The TE model is based on a hybrid scheme combining offline estimation of the link weights and online path selection. We apply the newly proposed TE model to compute the routing paths for the traffic offered to a 23-node test network. We compare the correlation between the link weights found by the genetic algorithm and the recently proposed Least Path Interference algorithm (LPIA) and evaluate the impact of this correlation on the network performance. We evaluate the routing performance of the TE model using the NS packet level simulator.

**Index Terms**— Open Shortest Path First, Least Path Interference, Traffic Engineering, Genetic Algorithms, Packet-level Simulation.

## I. INTRODUCTION

The Internet has developed beyond a research network initially intended to connect a small number of research institutions into a large decentralised network requiring network management techniques such as Traffic Engineering (TE) to meet the QoS demanded by modern IP applications. TE achieve QoS agreements between the available resources and the current and expected traffic by moving the traffic to where the network capacity is available.

It has been claimed [1] that data networks, including the Internet backbone, are lightly utilised, and will stay that way. This raises the expectation that Open Shortest Path First (OSPF) [2] and IS-IS [3], the most used Interior Gateway Protocols (IGPs) in the current Internet are sufficiently equipped to support QoS and substantial increases of traffic load without the need for sophisticated TE mechanisms. However, these protocols are based on destination-based routing; a packet forwarding paradigm where IP forwarding uses solely the destination address specified in the packet header. Despite its scalability which contributed to the large expansion of the

Internet, destination-based routing leads to opportunistic bandwidth sharing overloading some portions of the network while leaving some others unused. When carrying voice and data, an unbalanced network may result in unattractive behavior such as routing a voice-over-IP call over a high propagation delay path while a low-latency path is available or routing data traffic over high-utilised links while some portions of the network are still under-utilised.

### A. Link weight optimisation

Traffic engineering extensions to OSPF and IS-IS incorporating traffic information into the link-state advertisements and the path selection decisions were proposed in [4], [5]. However, these extensions require modifications to the routers to collect and disseminate information about network load and establish paths based on the load metrics. The link weight optimisation approach proposed in [6] is based on OSPF routing but using a clever selection of link weights to overcome the limitations of the destination-based routing approach. This approach has the advantages of (1) simplicity (2) capability of using diverse performance constraints and (3) compatibility with traditional IGPs. However there are limitations associated with the the weight optimisation approach. These include (1) the necessity for changing routing patterns which leads to routing inconsistencies during the transient behavior from one routing pattern to another (2) the NP-hardness of the problem of finding link metrics which minimise the maximum utilisation and (3) the inability to achieve optimal routing patterns for some networks even with link weight changing. This is the case where equal cost multipath may be applied but the application does not allow traffic splitting.

### B. Contributions and outline

This paper revisits the problem of traffic engineering to assess the relevance of using evolutionary algorithms to complement traditional route optimisation algorithms. The main contributions of this paper are threefold

**Routing optimisation.** We propose a new route optimisation model where optimal link weights are used as fine-tuning parameters in OSPF routing. This model uses a hybrid TE

This work is supported by grant numbers 2054027 and 2677 from the South African National Research Foundation, Siemens Telecommunications and Telkom SA Limited.

scheme based on (1) offline estimation of the link weights using genetic algorithms or a mixed metric computed using a genetic algorithm and a traditional optimisation algorithm and (2) online path selection using the widely known Dijkstra's algorithm.

**Similarity analysis.** The performance of networks is often expressed by system-wide parameters which may hide the strengths and weaknesses of the routing algorithms used to achieve this performance. We compare the correlation between the link weights used by different algorithms and evaluate the impact of this correlation on the network performance. This is achieved using methods borrowed from the time series analysis.

**Performance evaluation.** We evaluate the performance achieved by the route optimisation model when finding the routing paths for the traffic offered to a 23-node test network. We use the NS packet level simulator to compare the performance improvements achieved by the genetic link weight optimisation scheme, **OSPF** routing and the recently proposed Least Path Interference Algorithm (**LPIA** [7]). We draw some preliminary conclusions and present guidelines for future work.

The remainder of this paper is organised as follows. Section II presents the routing and rerouting problems and describes the routing algorithms considered. An application of the different algorithms to compute paths for the flows offered to a 23-node network is presented in section III. Our conclusions are presented in section IV.

## II. THE ROUTING OPTIMISATION MODEL

Consider a network represented by a directed graph  $(\mathcal{N}, \mathcal{L})$  where  $\mathcal{N}$  is a set of nodes and  $\mathcal{L}$  is a set of links. Let  $C_\ell$  denote the capacity of link  $\ell$  and let  $\mathcal{P}_{i,e}$  denote the set of paths connecting the ingress-egress pair  $(i, e)$ . Assume that a request to route a flow of  $d_{i,e}$  bandwidth units between an ingress-egress pair  $(i, e)$  is received and that future demands concerning IP flow routing requests are not known.

Let  $L_p = \sum_{\ell \in p} L_\ell(n_\ell, r_\ell)$  denote the cost of path  $p$  where  $L_\ell(n_\ell, r_\ell)$  is the cost of link  $\ell$  when carrying  $n_\ell$  flows and  $r_\ell$  is the total bandwidth reserved by the IP flows traversing link  $\ell$ .

The flow routing problem consists of finding the best feasible path  $p \in \mathcal{P}_{i,e}$  where

$$L_p = \min_{k \in \mathcal{P}_{i,e}} L_k \quad (1)$$

$$d_{i,e} < \min_{\ell \in p} (C_\ell - r_\ell) \quad (2)$$

Equations (1) and (2) express respectively the optimality of the routing process and the feasibility of the flows.

We consider a route optimisation model based on (1) fine-tuning the **OSPF** routing metric to achieve bandwidth usage maximisation and (2) using Dijkstra routing algorithm to find the shortest paths followed by the traffic.

### A. Fine-tuning the **OSPF** metric

Routing metrics are weights which are assigned to the links of the network to express the relative preference of a link during path computation. They may be classified into (1) additive/non-additive depending on how the path metric is calculated and (2) static/dynamic depending on their variability with time.

**OSPF**, the most used protocol for intra-domain routing uses by default a static routing metric which may lead to network un-balancing as previously stated. As proposed by CISCO, **OSPF** routing uses a link metric which is the inverse of the link capacity. This link metric is expressed by

$$L_p(\text{OSPF}) = \sum_{\ell} 1/C_\ell \quad (3)$$

where  $L_p(x)$  and  $L_\ell(x)$  are the path and link metrics associated with the routing algorithm  $x$  for a path  $p$  traversing a link  $\ell$ .

Our route optimisation model is based on fine-tuning the **OSPF** routing metric using weights computed by a genetic algorithm referred to as **GENA** or a traditional optimisation algorithm such as the Least Path Interference Algorithm (**LPIA**). The fine-tuned routing metric is expressed by

$$L_p(\text{GENA}) = \sum_{\ell} w_\ell^\alpha / C_\ell^{1-\alpha} \quad (4)$$

Where  $\alpha$  is a calibration parameter and  $w_\ell$  is the fine-tuning parameter (weight) computed by **GENA** or the path interference on link  $\ell$  computed by **LPIA**. Note that a linear combination of the two link weights may also be considered. It is expressed by the mixed weight metric

$$w_\ell = \beta\theta_\ell + (1 - \beta)\gamma_\ell \quad (5)$$

where  $\theta_\ell$  is the weight generated by **GENA**,  $\gamma_\ell$  is the path interference (weight) generated by **LPIA** on link  $\ell$  and  $\beta$  is a calibration parameter expressing the importance given to each of the two optimisation techniques:  $\beta \approx 0$  for traditional route optimisation using **LPIA**,  $\beta \approx 1$  for evolutionary optimisation using **GENA** and  $0 < \beta < 1$  for a route optimisation integrating traditional and evolutionary optimisation.

### B. The **GENA** Algorithm

Genetic algorithms use concepts from genetics to evolve solutions to problems. They are based on an evolutionary paradigm where each iteration of the algorithm transforms one population of individuals into a new generation, using some pre-determined fitness measure for an individual. In applications of genetic algorithms, potential solutions must be represented and encoded in terms of a *genome*. Each problem generally has its own genome representation, and more than one representation could be used for a given problem. The fitness measure or *fitness function* determines how good the solution represented by some genome is. The appropriate fitness function is determined by the problem and the genome representation. Genetic algorithms are based on a population selection model where the evolution of one generation into

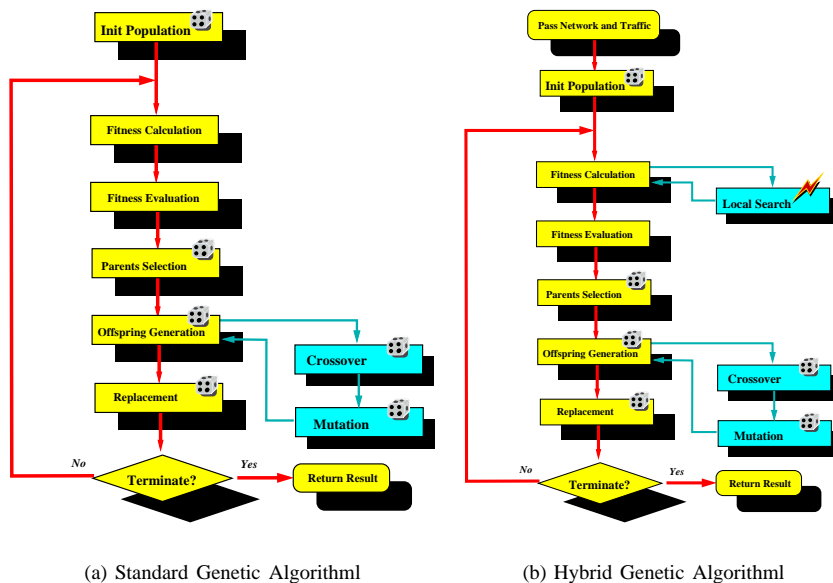


Fig. 1. “The GENA algorithm”

another relies on three genetic operations: (1) **reproduction** (2) **crossover** and (3) **mutation**. These operations are assigned probabilities.

**Reproduction** is a direct copying of a member of the current generation into the next generation. **Crossover** is the combination of two genomes from the current generation into two different genomes in the next generation. Crossover attempts to combine good solutions to find potentially better solutions. **Mutation** is the random permutation of one of the tokens in the genome representation of a member of the current generation. By introducing new solutions at each stage of the algorithm, mutation ensures that the evolution process does not get stuck at a local optimum.

**Probabilities** are associated with the crossover, reproduction and mutation operations. These probabilities are denoted  $P_c$ ,  $P_r$  and  $P_m$  respectively, and  $P_c + P_r + P_m = 1$ . In general,  $P_m \ll P_c$  and  $P_c \approx P_r$ . The **selection** of candidates for the genetic operations is random, but the selection is *fitness-proportionate* to ensure the survival of good solutions over generations. The conditions for the **termination** of the algorithm are problem-specific, although for practical reasons one often limits the number of iterations.

#### Standard genetic algorithm

A standard genetic algorithm depicted by Figure 1 (a) executes the following six steps [8]:

- Step 1.** Randomly generate an initial population  $x(0) = \{x_1(0), \dots, x_n(0)\}$
- Step 2.** Compute the fitness  $f(x_i(t))$  of each individual  $x_i(t)$  of the current population  $x(t)$
- Step 3.** Generate an intermediate population  $x_r(t)$  by applying the reproduction operator

**Step 4.** Generate population  $x(t+1)$  by applying some other operators to  $x_r(t)$ .

**Step 5.** Stop if the *EndTest* is true

**Step 6.** Set  $t := t + 1$  and goto Step 2

#### Application to link weight estimation

We apply the genetic algorithm in an IP routing context to find a set of optimal link weights leading to reduced link utilisation. These weights are used as fine-tuning parameters in **OSPF** routing. We consider a hybrid genetic model where the standard genetic algorithm described above is complemented by a local search to improve the fitness of individuals. This hybrid model is depicted by Figure 1 (b).

**Population.** In our model a population is expressed by a set of links represented by a vector where each position holds a link weight.

**Fitness function.** The fitness function is expressed by  $(1/\max_{\ell} \mu_{\ell})^p$  where  $\mu_{\ell}$  is the utilisation of link  $\ell$  and the power value is set to  $p = 1.5$  when the fitness is improved or  $p = 0.5$  if the fitness is decreased.

**Acceptance probability.** The population acceptance probability is increased upon fitness improvement or decreased when the fitness has worsened to ensure the survival of good populations over generations.

**Probabilities.** As depicted by the dices in Figure 1, the different genetic operations are performed based on given probabilities.

#### C. The LPIA algorithm

The Least Path Interference Algorithm uses the interference among competing flows referred to as path interference as fine-tuning parameter for the **OSPF** routing metric. The path interference is a measure of the probability for the traffic carried by a path to traverse a link. This parameter may be used

as a penalty to avoid the creation of bottleneck in a network by moving the traffic away from overcrowded links. This is implemented in this paper by a node-disjoint extension of the widely known  $K$ -shortest Path (**KSP**) algorithm.

Let  $\mathcal{K}_{i,e}$  denote the set of node disjoint paths from the ingress node  $i$  to the egress node  $e$  and  $\mathcal{K} = \cup_{i,e} \mathcal{K}_{i,e}$  denote the set of node disjoint paths for all  $(i, e)$  pairs. Let  $\mathcal{N}(p)$  denote the set of nodes of a path  $p$  and  $\mathcal{V}(i)$  denote the set of nodes which are neighbor (adjacent) to node  $i$ . The path interference is expressed by

$$w_\ell = \sum_{p \in \mathcal{K}} \delta_{\ell,p} \quad (6)$$

The estimation of the path interference  $w_\ell$  is based on the following algorithm:

- Step 1.** for each link  $\ell \in \mathcal{L}$  set  $n_\ell = 0$   
**Step 2.** for each I-E pair  $(i,e)$   
for each node  $v \in \mathcal{V}(i)$
- find the shortest path  $p \in \mathcal{K}_{i,e}$  from  $i$  to  $e$  traversing  $v$ .
  - for each link  $\ell \in p$  set  $n_\ell = n_\ell + 1$
  - for each node  $k \in \mathcal{N}(p)$  and each node  $k_n \in \mathcal{V}(k)$  set the link cost  $L_{k,k_n} = \infty$ .
- Step 3.** for each link  $\ell \in \mathcal{L}$  set  $w_\ell = n_\ell$ .

### III. AN IMPLEMENTATION

This section presents simulation experiments conducted using a 23-node test network to compare the performance of 3 different algorithms: (1) Open Shortest Path First (**OSPF**) (2) Genetic Algorithm (**GENA**) and (3) Least Path Interference Algorithm (**LPIA**). The 23-node network illustrated in Figure 2 is a fictitious representation of the USA network which includes 253 ingress-egress pairs and 102 uni-directional links capacitated with 5,289,780 units of bandwidth. In the rest of this paper 15 of the 308 I-E pairs will be considered.

#### A. Performance parameters

We considered several performance parameters some expressing the network efficiency while others were borrowed from the time series analysis to evaluate the correlation between the different routing metrics used to compute the paths and the performance values achieved.

##### Efficiency

We considered three main parameters for evaluating the efficiency of a network. These include (1) the maximum utilisation (2) the path length  $LEN$  and (3) the delay  $DEL$ .

##### Correlation

The time series analysis provides a framework for comparing two  $T$ -dimensional data vectors  $\tilde{x} = (x_0, x_1, \dots, x_{T-1})$  and  $\tilde{y} = (y_0, y_1, \dots, y_{T-1})$  using the linear correlation coefficient  $R(\tilde{x}, \tilde{y})$  of the two vectors. The coefficient  $R(\tilde{x}, \tilde{y})$

measuring the linear correlation between two sets of values represented by the two vectors  $\tilde{x}$  and  $\tilde{y}$  is given by

$$R(\tilde{x}, \tilde{y}) = \frac{N(\tilde{x}, \tilde{y})}{\sqrt{(D(\tilde{x}) - D(\tilde{y}))}} \quad (7)$$

where

$$N(\tilde{x}, \tilde{y}) = T \sum_{t=0}^{T-1} x_t y_t - \sum_{t=0}^{T-1} x_t \sum_{t=0}^{T-1} y_t$$

$$D(\tilde{x}) = T \sum_{t=0}^{T-1} x_t^2 - (\sum_{t=0}^{T-1} x_t)^2$$

$R(\tilde{x}, \tilde{y})$  yields

- $R(\tilde{x}, \tilde{y}) = 1$  if there is a strong correlation between  $\tilde{x}$  and  $\tilde{y}$  ( $\tilde{y} = a\tilde{x}$ ).
- $R(\tilde{x}, \tilde{y}) = 0$  if there is no correlation between  $\tilde{x}$  and  $\tilde{y}$  ( $\tilde{y} \neq a\tilde{x}$ ).

#### B. Simulation experiments

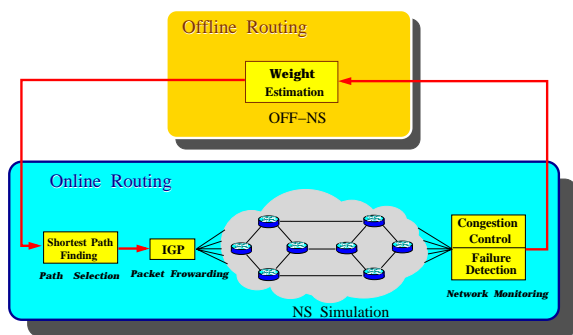
We conducted simulation experiments to analyse the similarity between the different algorithms and evaluate their performance. The simulation model is illustrated by Figure 2 (a) where OFF-NS calculation and NS simulation processes are interacting in a two-layer architecture to find a set of optimal paths to forward the traffic to. The main component of the OFF-NS calculation process is the link weight calculation. In the NS simulation process are performed (1) the path selection (2) the packet forwarding and (3) the network monitoring used in congestion control and recovery from failure. We considered CBR traffic over the UDP protocol for all our simulation experiments.

##### Path length and delay

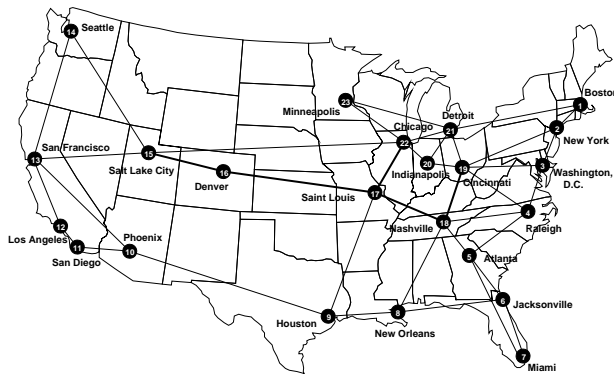
We compared the path length and delay for the three algorithms: **OSPF**, **GENA** and **LPIA**. The path length gives an indication on the resource consumption of the algorithm since a long path ties up many resources while a shorter path uses less resources. The path delay gives an indication of how quickly the packets routed over that path are transmitted between the two end-points. It can be used to express the QoS achieved by the applications in a mixed routing environment where best effort and real-time applications are competing for bandwidth on the network. The results presented in table I show that (1) **LPIA** and **OSPF** achieve the same performance in terms of path length (2) **LPIA** achieves the best path delay and (3) **GENA** performs worse. We note that **LPIA** and **OSPF** have the same path length but differ in the delay. We conducted further experiments which showed that the difference was a result of routing the traffic offered to the I-E pair (5, 13) on two different equal length paths: the path (5, 7, 17, 16, 15, 14, 13) for OSPF routing and path (5, 4, 17, 16, 15, 14, 13) for LPIA routing.

##### Link utilisation

We conducted a set of experiments to compare the maximum link utilisation achieved by each of the three algorithms. The maximum link utilisation gives an indication of the potential for traffic growth: lower maximum utilisation is an indication of better bandwidth usage leading to a higher



(a) Simulation model



(b) USA network

Fig. 2. “The test network and simulation model”

potential for traffic growth. The results revealed the following performance in terms of maximum link utilization: (1) 33% maximum utilisation for **GENA** (2) 42% maximum utilisation for **OSPF** and (3) 42% maximum utilisation for **LPIA**. These results show that **GENA** achieves the lowest maximum utilisation though **LPIA** and **OSPF** achieve the best performance in terms of delay and path length. This leads to a trade-off between the three performance indexes.

### Similarity

We compared the correlation between the link weights and utilisation achieved by the three different algorithms to evaluate (1) how the routing parameters produced by the two route optimisation algorithms **GENA** and **LPIA** are related to **OSPF** routing and (2) how they relate to each other. The results presented in table I reveals that **LPIA** is more strongly correlated to **OSPF** routing than **GENA** in terms of both the link weights and the link utilisation. However previous results have shown that **GENA** and **LPIA** achieve the same maximum link utilisations. This reveals that the two algorithms may achieve approximatively the same performance (in terms of link utilisation) though using different methods (different fine-tuning parameters). The delay difference between the two algorithms is probably a result of the difference between the methods used.

## IV. CONCLUSION

This paper presents a genetic algorithm and evaluates the performance improvements achieved by this algorithm compared to traditional optimisation in a hybrid offline+online setting based on offline link weight optimisation and online path selection. We present a genetic algorithm (**GENA**) where routing optimisation is performed to minimise the link utilisation. We compare this algorithm to the Least Path Interference Algorithm (**LPIA**) where the routing objective is to minimize the interference among competing flows. Preliminary simulation results reveal that (1) **OSPF** and **LPIA** achieve the best performance in terms of resource consumption (2)

**GENA** improves **OSPF** routing in terms of bandwidth utilisation while **LPIA** improves **OSPF** routing in terms of delay and (3) **GENA** outperforms **LPIA** in terms of bandwidth usage but performs poorer in terms of delay and path length. This leads to a trade-off that needs to be investigated in future work.

Genetic algorithms have been recently widely deployed in the academia as efficient heuristic solutions to the NP-hardness property occurring in many routing problems. We have assessed the performance improvements generated by a genetic algorithm on a 23-node test network using the path length, the link maximum utilisation and the average delay. This study needs to be extended using different performance parameters for different network topologies, under different traffic profiles and for different TE needs before deploying the algorithms presented in this paper as efficient TE tools on the service provider backbone.

## REFERENCES

- [1] A.M. Odlyzko, “Data networks are lightly utilized, and will stay that way”, *Review of Network Economics*, Vol.2, Issue 3 September 2003.
- [2] J. Moy, “OSPF Version 2”, *RFC1583*, <http://www.ietf.org/rfc/rfc1583.txt>, March 1994.
- [3] J. Parker, “Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)”, *RFC3719*, <http://www.ietf.org/rfc/rfc3719.txt>, February 2004.
- [4] D. Katz, D. Yeung, K. Kompella. “Traffic engineering extensions to OSPF”, *Work in progress, IETF draft draft-katz-yeung-ospf-traffic-06.txt*, November 2001.
- [5] T. Li, H. Smit. “IS-IS extensions for traffic engineering”, *IETF draft draft-ietf-isis-traffic-04.txt*, February 2001.
- [6] B. Fortz, M. Thorup, “Internet Traffic Engineering by Optimizing OSPF Weights”, *Proceedings of IEEE INFOCOM*, March 2000.
- [7] Antoine B. Bagula. “Hybrid Traffic Engineering: The Least Path Interference Algorithm”. In *Proceedings of the Southern African Institute for Computer Scientists and Information Technologists, Cape Town, South Africa. ACM International Conference Proceedings Series, ISBN: 1-58113-982-9, pp 89–96*, October 2004.
- [8] W. H. Steeb, Y. Hardy and R. Stoop, “The nonlinear workbook”, *World Scientific, second edition, ISBN 981-238-230-5*, 2002.

TABLE I  
CORRELATION, PATH LENGTH AND DELAY

Algorithm	Link weight and utilisation correlation					
	Weight			Utilisation		
	OSPF	GENA	LPIA	OSPF	GENA	LPIA
OSPF	-	0.02	0.52	-	0.29	0.96
GENA	0.02	-	0.03	0.29	-	0.24
LPIA	0.52	0.03	-	0.96	0.24	-

I-E	Path length and delay					
	LEN			DEL		
	OSPF	GENA	LPIA	OSPF	GENA	LPIA
[0, 8]	4	5	4	224	182	234
[1, 9]	5	6	5	275	425	243
[2, 10]	6	6	6	288	350	257
[3, 11]	6	6	6	285	350	96
[4, 12]	5	6	5	434	415	243
[5, 13]	6	6	6	500	96	257
[6, 14]	5	6	5	271	98	241
[7, 20]	3	5	3	48	235	58
[15, 1]	4	5	4	57	406	67
[16, 2]	3	4	3	46	396	56
[17, 22]	3	4	3	85	235	56
[18, 9]	4	5	4	62	413	32
[19, 6]	4	4	4	61	237	72
[21, 3]	3	3	3	45	383	56
[22, 10]	5	5	5	25	256	35
Maximum	6	6	6	500	425	257
Average	4	5	4	179	268	134