IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Data Forwarding using Link State Routing Protocol PEFT

Dinesh Kumar.P[1], Krishnapriya.P[2], Mariselvi.J[3] and Janani.B[4]
[1,2,3&4] **Information Technology, Anna University/P.S.R.Rengasamy College Of Engineering For Women , Sivakasi, Tamilnadu, India**

Abstract
Optimal traffic engineering (or optimal multicommodity flow) can be realized using just link-state routing protocols with host by host data transmission. Open Shortest Path First and Intermediate System-Intermediate System (IS-IS), used to split traffic evenly over shortest paths based on link weights. However, computing the link weights for OSPF/IS-IS to the offered traffic is a well-known NP-hard problem and even the best setting of the weights can splits the traffic. we propose a new link-state routing protocol is PEFT(Penalizing Exponential Flow splitting), that splits traffic over multiple paths with an exponential penalty on longer paths. DEFT provably achieves optimal traffic engineering while retaining the simplicity of hop by hop data transmission. The new protocol also used to reducing the time needed to compute the best link weight that is used to identify the traffic distributions.

Key words—Interior gateway protocol, network entropy Maximization, optimization, Open Shortest Path First (OSPF), routing, traffic engineering.

1. Introduction:

Designing a link-state routing protocol has three components. First is *weight computation*: The network-management system computes a set of link weights through a periodic and centralized optimization. The second is *traffic splitting*: Each router uses the link weights to decide traffic-splitting ratios among its outgoing links for every destination. The third is *packet forwarding*: Each router independently decides which outgoing link to forward a packet based only on its destination. A new link-state protocol, Penalizing Exponential Flow-splitting (PEFT), that achieves optimal TE and demonstrating that link-weight computation for PEFT is highly efficient. In addition, the corresponding link weights

can be found efficiently by solving the new optimization problem using the *gradient descent algorithm*. To compute the traffic distribution for PEFT, we should first compute the shortest paths between each pair path should be treated differently based on their path length. In PEFT, *Combinatorial algorithm* is used for split the traffic. It splits the traffic over multiple paths. From the definition of PEFT, more traffic should be sent along an out- going link used by more paths, and the paths should be treated differently based on their path lengths.

In PEFT, packet forwarding is just the same as OSPF: destination-based and hop-by-hop. The key difference is in traffic splitting. OSPF splits traffic *only very* shortest paths, and PEFT splits traffic along all paths, but penalizes longer paths (i.e., paths with larger sums of link weights) exponentially.

2. Architecture:

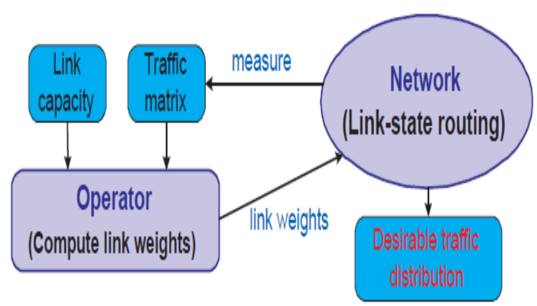

Fig 1: Architecture

Link Capacity: Capacity of the links in which the offered traffic is to be calculated and sent through it.
Operator: Operator which computes the link weights and sent to the network (link state routing).
Network: Link state routing measures the traffic or congestion which occurs and sends the traffic to the traffic matrix.
Traffic Matrix: Traffic matrix has the equation which calculates the congestion and split across the various links. This splitting is done by the operator and again sent it to the link state routing for distribution to the destination.
Desirable Traffic Distribution: Traffic distribution which is desired by the link state routing for sending it to the destination.

3. Existing System:

OSPF (*Open Shortest Path First)*

OSPF split the traffic evenly over shortest path only.

It takes more time for data transmission. OSPF needed the large amount of memory, because it maintains multiple copies of routing information. This protocol has no ability to adjust the splitting percentages. So the optimization problem is difficult.

DEFT *(Distributed Exponential Flow splitting):*

DEFT split the traffic evenly over multiple paths. It is link based flow splitting, it achieves nearer Optimal value. on convex, on smooth optimization Methods for weight computation and traffic splitting. This method cannot prevent the occurrence of cycle. It is link based flow splitting. So divide the packet over two equal paths

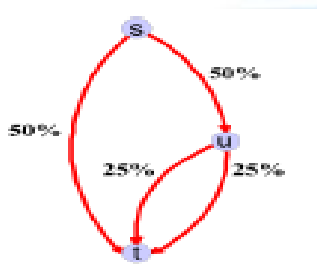

Fig 2: Link Based Flow Splitting

4. Proposed System:

PEFT:

We develop a new link state routing protocol PEFT (**Penalizing** Exponential Flow splitting),that splits traffic over multiple paths with an exponential penalty on longer paths.PEFT achieves optimal TE and demonstrating that link-weight computation for PEFT is highly efficient. we observe a 15% increase in the efficiency of capacity utilization by PEFT over OSPF. Furthermore, an exponential traffic-splitting penalty is the *only* penalty that can lead to this optimality result. The corresponding best link weights for PEFT can be efficiently computed. Optimal traffic distribution is realized by dividing an arbitrary fraction of traffic over many paths. This can be supported by the forwarding mechanism in multiprotocol label switching (MPLS) .It achieves the optimal solution. It reduction in the time needed.

4.1 Modules**:**

A link state routing protocol has three components. First one is Weight Computation. The second is Traffic splitting and the third is data forwarding.

4.2 Weight Computation**:**

Link- weight computation in OSPF can be turned into a convex optimization, which realizes optimal TE with PEFT. To speed up the calculation, weights are used by PEFT. In PEFT, Gradient descent algorithm is used for link weight computation. There are four steps are involved in this algorithm:

There are four steps are involved in the weight computation.

1. Start with a point (guess).
2. Determine a descent direction.
3. Choose the step
.4. Update the link weight.

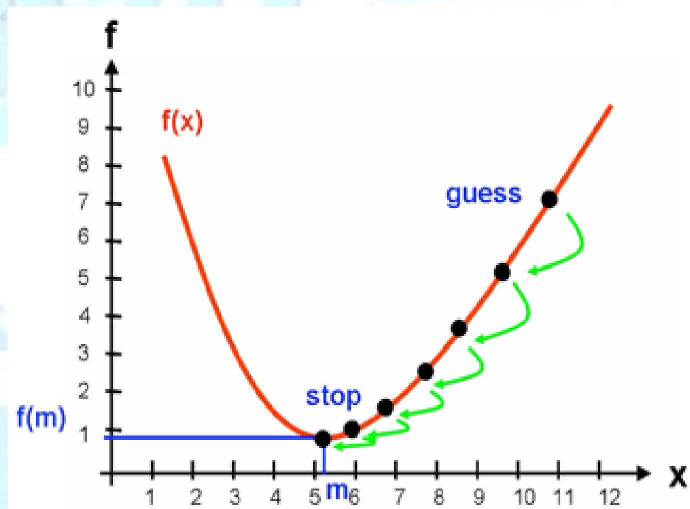

Fig:3 Graph of the Gradient Algorithm

4.3 Traffic Splitting:

To compute the traffic distribution for PEFT, we should first compute the shortest paths between each pair path should be treated differently based on their path length. In PEFT, Combinatorial algorithm is used for split the traffic. It splits the traffic over multiple paths. The traffic-splitting function for PEFT can be calculated by each node autonomously and in polynomial time. From the definition of PEFT, more traffic should be sent along an out- going link used by more paths, and the paths should be treated differently based on their path lengths**.**

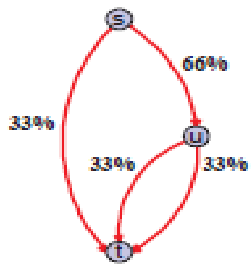

Fig:4 Path Based Flow Splitting

It is path based flow splitting so the data divided into three equal paths. It takes less time for data transmission.

4.4 Packet Forwarding:

Packet forwarding is transmitting data in the form of packets from one node to another node. Each router independently decides which outgoing link to forward the packets based only on destination. Routing is done **packet-by-packet and hop-by-hop**. Each packet is treated independently in each router along the path. At each hop, the router examines the destination IP address for each packet and then checks the routing table for forwarding information. The router will do one of the three things with the packet:

1. Forward it to the next-hop router
2. Forward it to the destination host
3. Drop it.

As an intermediary device, a router processes the packet at the Network layer. However, packets that arrive at a router's interfaces are encapsulated as a Data Link layer (Layer 2) PDU. In the router, the destination address in a packet header is examined. If a matching route in the routing table shows that the destination network is directly connected to the router, the packet is forwarded to the interface to which that network is connected. In this case, there is no next-hop. To be placed onto the connected network, the packet has to be first re-encapsulated by the Layer 2 protocol and then forwarded out the interface. If the route matching the destination network of the packet is a remote network, the packet is forwarded to the indicated interface, encapsulated by the Layer 2 protocol, and sent to the next-hop address. This process may occur a number of times until the packet reaches its destination network. The router at each hop knows only the address of the next-hop; it does not know the details of the pathway to the remote destination host. Furthermore, not all packets going to the same destination will be forwarded to the same next-hop at each router. Routers along the way may learn new routes while the communication is taking place and forward later packets to different next-hops.

4.5 Comparison between Existing system and Proposed system:

Table: 14.5 Comparison between Existing system and proposed system

Contents	Link State Routing		
	OSPF	DEFT	PEFT
Traffic Splitting	Even among shortest paths	Exponential	Exponential
Number of Iterations	High	High	Low
Optimal TE	No	Nearer optimal value	Optimal value
Time Consumption	High	high	Low

5. CONCLUSION AND FUTURE WORK:

Commodity-flow-based routing protocols are optimal for any Convex objective in Internet TE, but introduce much configuration Complexity. Link-state routing is simple, but prior work Suggests it does not achieve optimal TE. This paper proves that Optimal traffic engineering, in fact, *can* be achieved by link-state routing with hop-by-hop forwarding, and the right link weights *can* be computed efficiently, as long as flow splitting on nonshortestpaths is allowed but properly penalized. We also show uniqueness of the exponential penalty in Achieving optimal TE and discuss interpretations of NEM from The viewpoints of statistical physics and combinatory. Before concluding this paper, we would like to highlight that Optimization is used in three different ways in this paper. First and obviously, it is used when developing algorithms to solve the link-weight computation problem for PEFT.

In a more interesting way, the level of difficulty of optimizing Link weights for OSPF is used as a hint that perhaps we need to revisit the standard assumption on how link weights should be used. In this approach of "Design for Optimizability," sometimes a restrictive assumption in the protocol can be perturbed at low "cost" and yet turn a very hard network-management problem into an efficiently solvable one. In this case, better (and indeed the best) TE and faster weight computation are simultaneouslyachieved.In yet another way, optimization in the form of NEM is introduced as a conceptual framework to develop routing protocols. The NEM framework for distributed routing also leads to several interesting future directions, including extensions to robust and to the interactions between congestion controls at sources with link-state routing in the network.

6. REFERENCES:

[1] D. Xu, M. Chiang, and J. Rexford, "DEFT: Distributed exponentiallyweightedflow splitting," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 71–79.
[2] B. Fortz and M. Thorup, "Increasing Internet capacity using local search," *Compute. Optimiz. Appl.*, vol. 29, no. 1, pp. 13–48, 2004.
[3] D. Awduche, "MPLS and traffic engineering in IP networks," *IEEECommun. Mag.*, vol. 37, no. 12, pp. 42–47, Dec. 1999.
[4] Swung, Y.Wang, and L. Zhang, "Internet traffic engineering without full mesh overlaying," in *Proc. IEEE INFOCOM*, Anchorage, AK,2001, vol. 1, pp. 565–571.
[5] A. Sridharan, R. Guerin, and C. Idiot, "Achieving near-optimal trafficengineering solutions for current OSPF/IS-IS networks," *IEEE/ACMTrans. New.* vol. 13, no. 2, pp. 234–247, Apr. 2005.
[6] S. Srivastava, G. Agrawal, M. Pioro, and D. Medhi, "Determining link Weight system under various objectives for OSPF networks using a Lagrangian relaxation-based approach," *IEEE Trans on Network & Service Management*, vol. 2, no. 1, pp. 9–18, Nov. 2005.
[7] J. H. Fong, A. C. Gilbert, S. Kannan, and M. J. Strauss, "Better alternatives to OSPF routing," *Algorithmic*, vol. 43, no. 1-2, pp. 113–131,2005.
[8] W. R. Blunden, *Introduction to Traffic Science*. London, U.K.: Printer hall, 1967.
[9] J. A. Tomlin and S. G. Tomlin, "Traffic distribution and entropy," *Nature*, vol. 220, pp. 974–976, 1968.
[10] J. A. Tomlin, "A new paradigm for ranking pages on the world wideweb," in *Proc. 12th WWW*, New York, 2003, pp. 350–355.