Survey on the OSPF Routing Protocols

Santhosh Pawar
Department of CS
KSWU Vijayapura

Abstract-*Open Shortest Path First (OSPF) is a routing protocol for Internet Protocol (IP) networks. It uses a link state routing (LSR) algorithm and falls into the group of interior gateway protocols (IGPs), operating within a single autonomous system (AS) Open shortest path first (OSPF) is the most commonly used intera-domain routing protocol. Open Shortest Path First (OSPF) is a link state routing protocol. In exiting different type of routing protocols can be extant, but the most well-known routing protocols are Routing Information protocol (RIP) and the Open Shortest Path First (OSPF). In this paper, surveyed OSPF protocols and algorithm. OSPF router saves path of the state of all the various network connections (links) between itself and a network it is trying to send data to. This makes it a link-state routing protocol. Open Shortest Path First (OSPF) is one of the most broadly used intra-domain routing protocol. The OSPF protocol does not provide efficient routing in terms of packet sending to achieve any network optimization neutral. The high cost of network assets and profitable and modest nature of Internet service provisioning. The service providers are absorbed in performance optimization of their networks. This help to reducing congestion network and improving resource consumption across the network, which, in turn, results in an increased income collection. One way of achieving this is through Traffic Engineering.*

Keywords-Open Shortest Path First.

I. INTRODUCTION

The Open Shortest Path First (OSPF) protocol is a link state protocol that handles routing for IP traffic. Its newest implementation, version 2, which is explained in RFC 2328, is an open standard. Open Shortest Path First (OSPF) is an open standard (not proprietary) and it will run on most routers independent of make. Open Shortest Path First (OSPF) uses the Shortest Path First (SPF) algorithm, developed by Dijkstra, to provide a loop-free topology. Open Shortest Path First (OSPF) provides fast convergence with triggered, incremental updates via Link State Advertisements (LSAs). Open Shortest Path First (OSPF) is a classless protocol and allows for a hierarchical design with VLSM and route summarization. As most of us know OSPF stands for Open Shortest Path First. It is an internal routing protocol of the autonomous system based on link state developed by IETF. In IP networks, it dynamically finds and propagates routes by collecting and forwarding autonomous system link state. Each router that runs OSPF protocol always describes the local network connection state (such as valid interface information and reachable neighbor information) with LSA (link state advertisement) and advertises it to the whole autonomous system. Thus, each router receives the LSA generated by all routers within the autonomous system. The LSA collection then forms LSDB (link state database). Because each LSA is the description of the surrounding network topology of a router, the whole LSDB is then the actual reflection of the autonomous system network topology. Based on LSDB, the routers run the SPF (Shortest Path First) algorithm Dijkshtra to be precise. Build a shortest path tree that takes itself as the root, and the tree gives out the route to nodes in the autonomous system. In graph theory, “tree” is a connection figure without loops. Therefore, routes calculated by OSPF are born to be without loops. OSPF can divide the autonomous system into different areas according to the topology. Thus, when the Area Border Router (ABR) transmits routing information to other areas, it generates the brief LSA with the unit of segment. It will decrease the LSA number in the autonomous system and complexity of route calculation.

OSPF adopts four classes of routes that are arranged as follows with priority:

- Internal Routes
- Inter-area Routes
- Type one external Routes
- Type two external Routes

Internal area route and inter-area area route describes the internal network structure of the autonomous system, while external route describes how to choose routes to destinations outside the autonomous system. Generally, type one external routes correspond to the information introduced by OSPF from other internal routing protocols. Costs of these routes and costs of OSFP route itself are comparable. Type two external routes correspond to the information introduced by OSPF from external routing protocols. Costs of these routes are much larger than costs of OSFP route itself, so only external costs are considered for calculation. The main disadvantages of Open Shortest Path First (OSPF) are Open Shortest Path First (OSPF) requires more memory to hold the adjacency (list of OSPF neighbors), topology (a link state database containing all of the routers and their routes), and routing tables, Open Shortest Path First (OSPF) requires extra CPU processing to run the SPF algorithm and Open Shortest Path First (OSPF) is a complex routing protocol.

II. OVERVIEW OF IP TRAFFIC ENGINEERING

If there are several routers on a network, OSPF builds a table (or topography) of the router connections. When data is sent from one location to another, the OSPF algorithm compares the available options and chooses the most efficient way for the data to be sent. This limit unnecessary delays in data transmission and prevents infinite loops.

A. Overview of IP Traffic Engineering

Internet traffic engineering is defined as that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimization of operational IP networks. Traffic engineering encompasses the application of technology and scientific principles to the measurement, characterization, modelling, and control of Internet traffic. Traffic engineering depends on having a set of performance objectives that guide the selection of paths, as well as effective mechanisms for the routers to select paths that satisfy these objectives. Most existing IP networks run Interior Gateway Protocols (IGPs) such as OSPF (Open Shortest Path First) or IS-IS (Intermediate System-Intermediate System) that select paths based on static link weights configured by network operators. Routers use these protocols to exchange link weights and construct a complete view of the topology inside the AS. Then, each router computes shortest paths (as the sum of these weights) and creates a table that controls the forwarding of each IP packet to the next hop in its route. In this paper, we focus on the techniques for selecting the paths rather than the underlying mechanisms for packet forwarding. Traditionally, IP forwarding depends on the destination address in the IP header of each packet. More recently, routers running Multi-Protocol Label Switching (MPLS) can forward packets based on the label in the MPLS header. In either case, we are concerned with how the path is chosen rather than how the packets are forwarded. An important objective of Internet traffic engineering is to facilitate reliable network operations. This can be done by providing mechanisms that network integrity and by embracing policies emphasizing survivability. This results in a minimization of the network to service outages arising from errors, faults and failures occurring within the infrastructure.

III. OPEN SHORTEST PATH FIRST PROTOCOL

Open Shortest Path First (OSPF) is a link-state routing protocol, rather than a distance vector protocol. The main difference here is that a linked-state protocol does not send its routing table in the form of updates, but only shared its connectivity configuration. By collecting connectivity information from all of the devices on the network, OSPF can store all this information in a database and use that information to build a topology map. This information will allow OSPF to identify the best or shortest route to every other network segment on the network. The route selection is based on overall hops to the destination, as well as link speed or link cost. The topology not only includes the best route to the destination as calculated by the Dijkstra algorithm (a search algorithm created by Edsger Dijkstra), but also, when possible, it includes a candidate or backup route to the destination. After creating the topology map, OSPF populates the routing table with the chosen routes to each destination. As traffic passes from router to router, each router evaluates the best path to the destination network. In some cases, this process can lead to routing loops on the network, because each one is evaluating the path based on its own link state database. The OSPF interior network protocol belongs to a single routing domain (or group of routers) known as an Autonomous System (AS). All routers belonging to the same AS share connection information and build their linked-state database from that information. Specifically, with OSPF, as opposed to link-state terminology in general, the primary, or best, route the destination goes through is the Designated Router (DR), although if it fails, the secondary or backup path will be sent to the Backup Designated Router (BDR). OSPF typically uses multicast to share connection information with its neighbors, and this information is sent to the 224.0.0.5 multicast address. OSPF is an open protocol and is defined in RFC2328 for version 2 of the protocol. Version 3 of OSPF has been updated to support IPv6 and is defined in RFC5340. Other than for the newly integrated support for IPv6, no major technical differences exist between version 2 and version 3. OSPF version 2 (OSPFv2) is used with IPv4. OSPFv3 has been updated for compatibility with IPv6's 128-bit address space. However, this is not the only difference between OSPFv2 and OSPFv3. Other changes in OSPFv3, as defined in RFC 2740, include protocol processing per-link not per-subnet addition of flooding scope, which may be link-local, area or AS-wide removal of opaque LSAs support for multiple instances of OSPF per link various packet and LSA format changes (including removal of addressing semantics). Both OSPFv2 and OSPFv3 are fully supported by DC-OSPF.

IV. RELATED WORKS

W. Ben-Ameur et, al., [1] in this paper introduce the problem of the performance measurement of a routing pattern.

We also define the complexity of a routing pattern as the number of MPLS tunnels needed for its realization. We show how the number of MPLS tunnels that are needed to enhance an IGP routing strategy can be minimized. We compare different routing strategies in IP networks from the two points of view: complexity and performance. We then propose two off-line Traffic Engineering methodologies for IP intra-domain network: the first one is based on an IGP/MPLS architecture; the second one is based only on the IGP routing using an optimized load balancing scheme. S. Fisher et., al., [2] present REPLEX, a distributed dynamic traffic engineering algorithm based on this policy. Exploiting the fact that most underlying routing protocols support multiple equal-cost routes to a destination, it dynamically changes the proportion of traffic that is routed along each path. These proportions are carefully adapted utilising information from periodic measurements and, optionally, information exchanged between the routers about the traffic condition along the path. We evaluate the algorithm via simulations employing traffic loads that mimic actual Web traffic, i. e., burst TCP traffic, and whose characteristics are consistent with self-similarity. B. Fortz and M. Thorup [3] in this paper the problem of optimizing OSPF weights for a given a set of projected demands so as to avoid congestion. We show this problem is NP-hard, even for approximation, and propose a local search heuristic to solve it. We also provide worst-case results about the performance of OSPF routing vs. an optimal multi-commodity flow routing. Our numerical experiments compare the results obtained with our local search heuristic to the optimal multi-commodity flow routing, as well as simple and commonly used heuristics for setting the weights. E. Gourdin and O. Klopfenstein [4] explain about frequent optimization criteria are for instance the so-called Kleinrock function, designed from delay analysis, the minimal residual capacity, maximized to avoid congestion. Many piecewise convex costs on arcs are also used. It appears that most of these functions are basically trying to combine two rather opposite objectives, namely to avoid congestion (or, in other words, to preserve as much residual capacity as possible), and to minimize the length of the routing paths. In this paper, we propose a numerical study giving some insight on the impact of using one objective function rather than another. The impact of piecewise linear costs on the obtained solution is more particularly investigated.

M. Chiang et., al., [5] internet today, traffic engineering is performed assuming that the offered traffic is inelastic. In reality, end hosts adapt their sending rates to network congestion, and network operators adapt the routing to the measured traffic. This raises the question of whether the joint system of congestion control (transport layer) and routing (network layer) is stable and optimal. Using the established optimization models for TCP and traffic engineering as a basis, we find the joint system can be stabilized and often maximizes aggregate user utility. We prove that both stability and optimality of the joint system can be guaranteed for sufficiently elastic traffic simply by tuning the cost function used for traffic engineering. Then, we present a new algorithm that adapts on a smaller timescale to changes in traffic distribution and is more robust to large traffic bursts. Uniting the network and transport layers in a multi-layer approach, this algorithm, distributed adaptive traffic engineering (DATE), jointly optimizes the goals of end users and network operators and reacts quickly to avoid bottlenecks. Simulations demonstrate that DATE converges quickly. M. Suchara et., al., [6] traffic management spans congestion control (at end hosts), routing protocols (on routers), and traffic engineering (by network operators). Historically, this division of functionality evolved organically. In this paper, we perform a top-down redesign of traffic management using recent innovations in optimization theory. First, we propose an objective function that captures the goals of end users and network operators. Using all known optimization decomposition techniques, we generate four distributed algorithms that divide traffic over multiple paths based on feedback from the network links. Combining the best features of the algorithms, we construct TRUMP: a traffic management protocol that is distributed, adaptive, robust, flexible and easy to manage. Further, TRUMP can operate based on implicit feedback about packet loss and delay. We show that using optimization decompositions as a foundation, simulations as a building block, and human intuition as a guide can be a principled approach to protocol design.

S. Srivastava et., al., [7] an important traffic engineering problem for OSPF networks is the determination of optimal link weights. Certainly, this depends on the traffic engineering objective. Regardless, often a variety of performance measures may be of interest to a network provider due to their impact on the network. In this paper, we consider different objectives and discuss how they impact the determination of the link weights and different performance measures. In particular, we propose a composite objective function; furthermore, we present a Lagrangian relaxation-based dual approach to determine the link weight system. We then consider different performance measures and discuss the effectiveness of different objectives through computational studies of a variety of network topologies. We find that our proposed composite objective function with Lagrangian relaxation-based dual approach is very effective in meeting different performance measures and is computationally very fast. D. Xu et., al., [8] Network operators control the flow of traffic through their networks by adapting the configuration of the underlying routing protocols. For example, they tune the integer link weights that interior gateway protocols like OSPF and ISIS use to compute shortest paths. The resulting optimization problem -to find the best link weights for a given topology and traffic matrix -is computationally intractable even for the simplest objective functions, forcing the use of local-search techniques. The optimization problem is difficult in part because these protocols split traffic evenly along shortest paths, with no ability to adjust the splitting percentages or direct traffic on other paths. In this paper, we propose an extension to these protocols, called Distributed Exponentially-weighted Flow Splitting (DEFT), where the routers can direct traffic on non-shortest paths, with an exponential penalty on longer paths. DEFT leads not only to an easier-to-solve optimization problem, but also to weight settings that provably perform no worse than OSPF and IS-IS. Furthermore, in our optimization problem, both link weights and flows of traffic are integrated as optimization variables into the formulation and jointly solved by a two-stage iterative method. Our novel formulation leads to a much more efficient way to identify good link weights than the local-search heuristics used for OSPF and IS-IS today. DEFT retains the simplicity of having routers compute paths based on configurable link weights, while approaching the performance of more complex routing protocols that can split traffic arbitrarily over any paths. K. Xu et., al., [9] the general form of multipath utility optimization is not strictly concave, making its solution quite unstable. Decomposition-based techniques like Traffic-management Using Multipath Protocol (TRUMP) alleviates the instability, but their convergence is not guaranteed, nor is their optimality. They are also inflexible in differentiating the control at different links. In this paper, we address the above issues through a novel logarithm-barrier-based approach. Our approach jointly considers user utility and routing/congestion control. It translates the multipath utility maximization into a sequence of unconstrained optimization problems, with infinite logarithm barriers being deployed at the constraint boundary. We demonstrate that setting up barriers is much simpler than choosing traditional cost functions and, more importantly, it makes optimal solution achievable. We further demonstrate a distributed implementation, together with the design of a practical Logarithm Barrier-based-Multipath Protocol (LBMP). We evaluate the performance of LBMP through both numerical analysis and packet-level simulations. The results show that LBMP achieves high throughput and fast convergence over diverse representative network topologies. Such performance is comparable to TRUMP, and is often better. Moreover, LBMP is flexible in differentiating the control at different links, and its optimality and convergence are theoretically guaranteed. F. P. Tso et., al., [10] Equal cost multiple path (ECMP) forwarding is the most prevalent multipath routing used in data center (DC) networks today. However, it fails to exploit increased path diversity that can be provided by traffic engineering techniques through the assignment of nonuniform link weights to optimize network resource usage. To this extent, constructing a routing algorithm that provides path diversity over nonuniform link weights (i.e., unequal cost links), simplicity in path discovery and optimality in minimizing maximum link utilization (MLU) is nontrivial. In this paper, we have implemented and evaluated the Penalizing Exponential Flow-splitting (PEFT) algorithm in a cloud DC environment based on two dominant topologies, canonical and fat tree. In addition, we have proposed a new cloud DC topology which, with only a marginal modification of the current canonical tree DC architecture, can further reduce MLU and increase overall network capacity utilization through PEFT routing. Z. Shao et., al., [11] in this paper explain about ensemble routing was proposed to achieve management scalability and robustness by using Virtual Local Area Networks (VLANs) and operating on the granularity of flow ensembles, i.e. group of flows. The key challenge of intra-data-center traffic engineering with ensemble routing is the combinatorial optimization of VLAN assignment, i.e., optimally assigning flow ensembles to VLANs to achieve load balancing and low network costs. Based on the Markov approximation framework, we solve the VLAN assignment problem with a general objective function and arbitrary network topologies by designing approximation algorithms with close-to-optimal performance guarantees. We study several properties of our algorithms, including performance optimality, perturbation bound, convergence of algorithms and impacts of algorithmic parameter choices. Then we extend these results to variants of VLAN assignment problem, including interaction with TCP congestion and QoS considerations.

M. Chiesa et., al., [12] configuring static link weights and splitting traffic between the resulting shortest-paths via the Equal-Cost-Multi-Path (ECMP) mechanism. Yet, despite its vast popularity, crucial operational aspects of TE via ECMP are still little-understood from an algorithmic viewpoint. We embark upon a systematic algorithmic study of TE with ECMP. We consider the standard model of TE with ECMP and prove that, in general, even approximating the optimal link-weight configuration for ECMP within any constant ratio is an intractable feat, settling a long-standing open question. We establish, in contrast, that ECMP can provably achieve optimal traffic flow for the important category of Clos data center networks. We last consider a well-documented shortcoming of ECMP: suboptimal routing of large (“elephant”) flows. We present algorithms for scheduling “elephant” flows on top of ECMP (as in, e.g., Hedera [1]) with provable approximation guarantees. Our results complement and shed new light on past experimental and empirical studies of the performance of TE with ECMP. V. Foteinos et., al., [13] obviously, tackling such growth requires sophisticated Traffic Engineering (TE) and associated management schemes. On the one hand, TE mechanisms should be intelligent and self-adaptive so that to take fast and reliable decisions with respect to traffic allocation into network paths. On the other hand, the management of this intelligence cannot rely on the traditional command and control paradigm. Contrarily, it needs to be based on systems that hide technology complexity from the operator and relax him from the rather slow and error prone task of manual configuration. Accordingly, in this work, we present an operator-friendly management framework that is used to drive the decisions of an autonomous algorithm for TE in IP/MPLS core networks. Through the framework, the operator is able to select from a set of high level policies, which the proposed TE algorithm needs to take into account while seeking for routing configurations during its autonomous operation. The behaviour of the proposed TE algorithm under the operator choices is experimented through numerous simulations and extensive test cases. Results showcase the efficiency and optimal performance of the algorithm, compared to other TE solutions proposed in literature, while at the same time they validate the framework's friendliness towards operator. B. Movsichoff et., al., [14] this work provides a family of optimal adaptation laws. These laws enable each node in the network to independently distribute traffic among any given set of next hops in an optimal way, as measured by a given global utility function of a general form. This optimal traffic distribution is achieved with minimum information exchange between neighboring nodes. Furthermore, this approach not only allows for optimal multiple forwarding paths but also enables multiple classes of service, e.g., classes of service defined in the differentiated services architecture. Moreover, the proposed decentralized control scheme enables optimal traffic redistribution in the case of link failures. Suboptimal control laws are also presented in an effort to reduce the computational burden imposed on the nodes of the network. W. Su et., al., [15] In this paper, we put forward an integrated traffic control structure and the associated control laws for multidomain networks. This control structure performs per-edge-to-edge-based multi-next-hop or multipath rate adaptation and load balancing among domain edge nodes in a multidomain network. This control structure is underpinned by a large family of distributed control laws, with provable convergence and optimality properties. With any user-defined global design objective, a set of control laws can be selected from this family of control laws that track an operational point where the global design objective is achieved, while providing traffic engineering (TE) and fast failure recovery (FFR) features for class-of-service (CoS)-aware flow aggregates. The structure allows the user to have full control over how the domains should be created and whether to use point-to-multipoint and/or point-to-point multipath. The flexibility and versatility of the control structure makes it an ideal theoretical underpinning for the development of integrated traffic control solutions for large-scale networking systems, in particular, software-defined networks in which the data plane is fully programmable via a well-defined south-bound interface, such as Open Flow. The simulation testing demonstrates the viability of the solution in providing TE, FFR, and CoS features.

V. CONCLUSION

Intradomain routing protocols such as OSPF and IS-IS have been deployed in a large number of networks throughout the Internet for many years. In this paper provide a survey of existing load balanced routing techniques to improve the network performance. Most existing work however do not address the interaction among traffic classes. Review of exiting algorithm and techniques used in open shortest path first routing protocol. As such, previous TE methods are not efficient.

REFERENCES

[1] W. Ben-Ameur, ´ E. Gourdin, B. Liau, and N. Michel, “Routing strategies for IP networks,” Telektronikk Mag., vol. 97, no. 2/3, pp. 145-158, 2001.

[2] S. Fisher, N. Kammenhuber, and A. Feldmann, “REPLEX: dynamic traffic engineering based on wardrop routing pplicies,” in ACM CoNEXT, 2006. [3]B. Fortz and M. Thorup, “Increasing Internet capacity using local search,” Computational Optimization and Applications, vol. 29, no. 1, pp. 13-48, 2004.

[3] E. Gourdin and O. Klopfenstein, “Comparison of different QoS-oriented objectives for multicommodity flow routing optimization,” in 13th International Conference on Telecommunications, 2006.

[4] J. He, M. Chiang, and J. Rexford, “Towards robust multi-layer traffic engineering: optimization of congestion control and routing,” IEEE Journal on Selected Areas in Commun., vol. 25, no. 5, pp. 868-880, June 2007.

[5] J. He, M. Suchara, and M. Chiang, “Rethinking Internet traffic management: from multiple decompositions to a practical protocol,” in ACM CoNEXT, 2007.

[6] S. Srivastava, G. Agrawal, M. Pioro, and Medhi, “Determining link weight system under various objectives for OSPF networks using a Lagrangian relaxation-based approach,” IEEE Transactions on Network and Service Management, vol. 2, no. 1, pp. 9-18, Nov. 2005.

[7] D. Xu, M. Chiang, and J. Rexford, “DEFT: Distributed exponentiallyweighted flow splitting,” in IEEE INFOCOM, 2007

[8] K. Xu, H. Liu, J. Liu, and J. Zhang, “LBMP: a logarithm-barrier-based multipath protocol for Internet traffic management,” IEEE Trans. on Para. and Dist. Systems, vol. 22, no. 3, pp. 456-470, Mar. 2011.

[9] F. P. Tso and D. P. Pezaros, “Improving Data Center Network Utilization Using Near-Optimal Traffic Engineering,” IEEE Trans. on Para. and Dist. Systems, vol. 24, no. 6, pp. 1139-1148, 2013 [11] Z. Shao, X. Jin, W. Jiang, M. Chen, and M. Chiang, “Intra-Data-Center Traffic Engineering with Ensemble Routing,” IEEE INFOCOM, 2013.

[10] M. Chiesa, G. Kindler, and M. Schapira, “Traffic Engineering with Equal-Cost-MultiPath: An Algorithmic Perspective,” IEEE INFOCOM, 2014.

[11] V. Foteinos, K. Tsagkaris, P. Peloso, L. Ciavaglia, and P. Demestichas, “Operator-Friendly Traffic Engineering in IP/MPLS Core Networks,” IEEE Trans. on Netw. and Serv. Manag., vol. 11, no. 3, pp. 333-349, 2014.

[12] B. Movsichoff, C. Lagoa, and H. Che, “Decentralized optimal traffic engineering in connectionless networks, IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 293-303, Feb 2005.

[13] W. Su, C. Liu, C. Lagoa, H. Che, K. Xu, and Y. Cui, “Integrated, Distributed Traffic Control in Multidomain Networks,” IEEE Trans. on Control Systems Technology, to appear.