

Overcoming Failures: Fault-tolerance and Logical Centralization in Clean-Slate Network Management

Hammad Iqbal, Taieb Znati
University of Pittsburgh, USA
{hiqbal, znati}@cs.pitt.edu

Abstract—We investigate the design of a clean-slate control and management plane for data networks using the abstraction of 4D architecture, utilizing and extending 4D’s concept of logically centralized Decision plane that is responsible for managing network-wide resources. In this paper, a dynamically adaptable algorithm for assigning Data plane devices to a physically distributed Decision plane is presented, which enables a network to operate with minimal configuration and human intervention, while providing optimal convergence and robustness against failures. Our work is especially relevant in the context of ISPs and large geographically dispersed enterprise networks, where robust and scalable network-wide control of a large number of heterogeneous devices is desired.

I. INTRODUCTION

The management of today’s typical data networks requires extensive manual configuration of individual protocol parameters that govern the operation of a variety of distributed routing algorithms operating on diverse physical network devices. The distributed and interdependent operation of these protocols makes it very difficult to control their interactions, leaving the networks fragile [1]–[3] and insecure [4]. Incremental solutions to improve network management, including the use of better management tools, have been ineffective as they try to match the pace of changes in various device operations and technical advances.

Effective control and management is especially a challenge for large and geographically dispersed networks, such as first and second tier ISPs, where it is important to efficiently manage the network resources across a large number of heterogeneous network devices, while meeting strict constraints on network availability and reliability.

To tackle the challenge of management complexity, an alternative approach to incremental solutions involves centralization of control state and logic inside a *logically centralized* Decision plane that is responsible for collecting, computing, and maintaining the state required by the network devices to operate. This approach is the basic tenant of the 4D architecture [2], which advocates a new layering design of the IP networks separating the task of packet forwarding, a data layer function, from the task of network control, an operation and management function.

The design of an efficient and robust Decision plane requires careful consideration of the design efficiency and robustness. A physically centralized decision plane design was investigated in [5], [6] where replication of physical Decision Elements (DE) was used to ensure Decision plane robustness to DE

failures. An alternative design approach was identified in [7], where the logical Decision plane was distributed over physically independent DEs. In this design, each DE controls a subset of the whole network, and works collaboratively with other DEs to achieve overall network control. However, it is also important to ensure that the reliability of the physically distributed control approach matches or exceeds the reliability offered by today’s distributed architecture.

Robustness of an architecture to failures is one of the most important factor of its design and we argue that the Decision plane design should be dynamically adaptable to failures at both Decision and Data planes. Furthermore, the DEs and their assigned routers¹ must respond swiftly to events such as failures and traffic surges. This paper addresses these design requirements and presents a Decision plane where a set of DEs, each governing a subset of AS routers, collaboratively maintain a network-wide state to support network-wide routing decisions.

Our work is especially relevant in the context of ISPs and similar large and geographically dispersed networks, where network-wide control is highly desired but any Decision plane design must meet stringent challenges of scalability and robustness. In our design, an individual member of the Decision plane only governs a subset of the total number of routers in the Data plane, and Points Of Presence (POP) in ISP topologies are naturally amenable to such grouping.

Our paper is organized as follows: We describe the network model used in our paper in §II. Trade-offs in the design of assignment algorithm are considered in §III. In §IV, formulation of the router assignment problem is presented along with a novel adaptive algorithm for its solution. We explore related work in §VI and §VII concludes our paper.

II. NETWORK MODEL

We utilize the abstraction of 4D architecture [2] for modeling a *logically centralized Decision plane*, where a set of Decision Elements (DEs) collaborate to perform the function of network-wide control. The high-level design of our network model is shown in Fig. 1 for an ISP topology spanning the continental US with several POPs. The figure illustrates the few basic assumptions taken in our network model:

¹We use “router” as a generic label for routers or switches in the 4D Data plane, while “DE” is used to represent Decision Elements in the 4D Decision plane

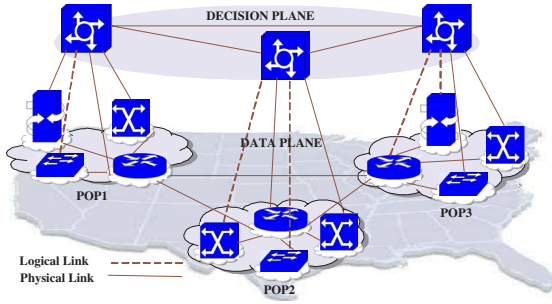


Fig. 1: Overview of the Decision plane design

- The entire network topology is under a single administrative control.
- The Decision plane is fully connected, i.e. there is a path from each DE to all other DEs that is not dependent on the operation of Data plane.
- Positioning of DEs corresponds to the geographical clustering of routers in the Data plane, e.g. within an ISP POP.

We believe these assumptions are easy to meet in any reasonably large network where control and management is presently an issue. The first assumption is necessary for consistent network-wide management and deserves no further explanation. The use of dedicated out-of-band control paths in the second assumption is in contrast with the in-band paths used in current IP networks. Although it is possible to use the same scheme in logically centralized Decision plane design, we have purposely avoided the potential complexity and network fragility introduced by piggy-backing control information over data paths. Our use of out-of-band paths is analogous to the SS7 signaling used in PSTN networks [8] and can be similarly implemented. Finally, our third assumption positions DEs in accordance with the clustering of routers in the underlying Data plane, using the techniques discussed in [7]. This ensures that latency of Decision plane response, and convergence delay in case of failures, is kept close to minimum.

In our design, each DE is only responsible for computing routing tables for the routers under its direct control, i.e. a subset of the total number of routers in a network. We refer to this (sub)set of routers as an *area* and it marks the extent of a DE's direct control over the network. Moreover, DEs exchange reachability information about their areas and utilize this information in establishing routing paths between different areas. In the case of shortest-paths routing, a path between routers in two different areas must travel the inter-area links between them, resulting in optimal routes only under the condition that a similar routing process on the complete topology would have selected the same path. Similar argument also applies to the intra-area routes. It is easy to see that this condition is fulfilled in topologies where distances between routers inside geographical clusters are less than the distance between the clusters. We believe network size and geographical distances in enterprise and ISP networks allow the fulfillment of this condition.

The logically centralized structure of the Decision plane strikes a balance between the extremes of distributed operation of individual routers, as seen in the current data networks, and total centralization, with its inherent scalability and robustness issues. More subtly, it also has the potential to allow easier deployment and transition from a distributed model of operation; as instead of a “forklift” change of the entire networking infrastructure, only a subset of the AS network could be transitioned at a time.

III. TRADE-OFFS IN DECISION PLANE DESIGN

Robustness of the Decision plane is dependent on the mechanisms employed to ensure its continued functioning in case of failures. An approach to this problem was presented in [6], where the Decision plane was designed to be physically centralized and multiple hot-standby DEs were used to increase its robustness in case the current “master” DE fails.

In contrast, a DE in a logically centralized Decision plane is not required to control the entire AS; only a subset of the total number of routers are under the control of a single DE. Any DE failure would therefore orphan the routers under its control. This calls for a scheme that reassigns orphaned routers to the functional DEs so that network control is reinstated. This assignment of routers would involve a trade-off in minimizing routing convergence delay, response time, and load balancing at the Decision layer. The routing convergence delay — transient time period between DE failure and orphaned routers' reception of new routing assignments — represents loss of management control over the orphaned devices, and must be minimized. Similarly, in normal operation the response time of Decision plane also needs to be minimized. In both cases, aggregate router-DE delay provides a metric for the minimization objective. Additionally, large variation in DE loads must be avoided as it can result in slower Decision plane response in parts of the networks and increased potential for DE failures.

Assignment mechanism is also constrained in a unique way as any router assignment must adhere to the underlying physical data plane topology. Specifically, since a DE only controls the routers in its own area, it must avoid any assignment that involves the usage of inter-area paths between routers belonging to the same area. This condition is necessary to ensure that routers in an area can be governed locally without requiring global network knowledge. Therefore, there must be a physical path between routers that are assigned to the same DE that does not involve any links or routers not totally contained within the same area. We refer to this condition as the contiguity constraint.

Trade-offs also exist between complexity of a recovery scheme and the desired level of robustness. For example, we can generalize a simple scheme of using backups as proposed in [6], [9], where each router is statically configured with a primary and an ordered list of standby DEs. However, it is easy to show that this scheme can lead to uneven DE workloads in case of multiple DE failures, potentially causing severe performance degradation. Moreover, a static assignment

will have to be often updated to ensure its applicability and validity with the dynamically changing network topology. These shortcomings suggest that it is desirable to have an adaptive mechanism, that can assign routers to feasible DEs while, 1., balancing the DE workload and, 2., minimizing the physical constraint on Decision plane response time, i.e. the propagation delays between routers and DEs. In the following section we describe our design of such adaptive router assignment mechanism.

IV. ADAPTIVE ASSIGNMENT OF DATA PLANE DEVICES

Let $R = \{r_1, r_2, \dots, r_m\}$ be the collection of routers in a AS, assumed to be homogeneous in terms of their demands of Decision plane resources, and $E = \{e_1, e_2, \dots, e_n\}$ be the set of n functional DEs in the network. For any r_i , $N(r_i)$ denotes the set of routers in physical open neighborhood of r_i , i.e. r_i and all of its physically adjacent routers. We define $A(e_j)$ to be the set of routers assigned to e_j and A as the adjacency matrix of router assignments for all DEs in E , which is the output of the assignment problem. Let $x(r_i, e_j)$ be a binary indicator variable defined as $x(r_i, e_j) = 1 \iff e_j \leftarrow r_i$. Let $d(r_i, e_j)$ be the minimum delay between router r_i and a DE e_j , and $D[d(r_i, e_j)]_{m \times n}$ be the matrix of all such delays. Let $L_j = \sum_{r_i \in R} x(r_i, e_j)$ be the load on DE e_j and Q_j be the capacity, i.e. the maximum number of routers, that e_j is able to govern.

We assume that information about the network topology, specifically router adjacencies and delay, would be available to the Decision plane as part of the service offered by the Discovery and Dissemination planes of 4D architecture. Use of source routes [6], [9] is one method by which such information can be collected. However, the design specifics of Discovery and Dissemination planes are beyond the scope of this work.

A. ILP Formulation

From the discussion of the previous section, the objective of the assignment problem is to assign routers in R to DEs in E in such a way that aggregate delay between routers and their assigned DEs is minimized, while ensuring that the DE workload is balanced. Formally, we define our objective function as $\sum_{e_j \in E} \sum_{r_i \in R} d(r_i, e_j) x(r_i, e_j)$ and introduce a constraint to balance the loads using the average load L_{avg} , and a load balancing parameter $\Delta \geq 1$.

$$L_{avg} = m / \sum_{e_j} Q_j \quad 0 < L_{avg} \leq 1$$

The optimization problem can be formulated as the following ILP:

$$\text{Minimize } \sum_{e_j \in E} \sum_{r_i \in R} d(r_i, e_j) x(r_i, e_j) \quad (1)$$

s.t.

$$\sum_{e_j \in E} x(r_i, e_j) = 1 \quad \forall r_i \in R \quad (2)$$

$$\sum_{r_i \in R} x(r_i, e_j) - Q_j \leq 0 \quad \forall e_j \in E \quad (3)$$

$$L_j \leq [\Delta L_{avg} Q_j] \quad \forall e_j \in E \quad (4)$$

$$\sum_{r_k \in N(r_i)} x(r_k, e_j) \geq x(r_i, e_j) \quad |A(e_j)| \geq 1, \forall r_i \in R \quad (5)$$

$$x(r_i, e_j) \in \{0, 1\} \quad \forall r_i \in R, e_j \in E \quad (6)$$

The objective function minimizes aggregate delay between routers and their assigned DEs. Constraint (2) ensures that each router in R is assigned, (3) ensures that the DE workload capacities are not violated, and (5) imposes the contiguity requirement.

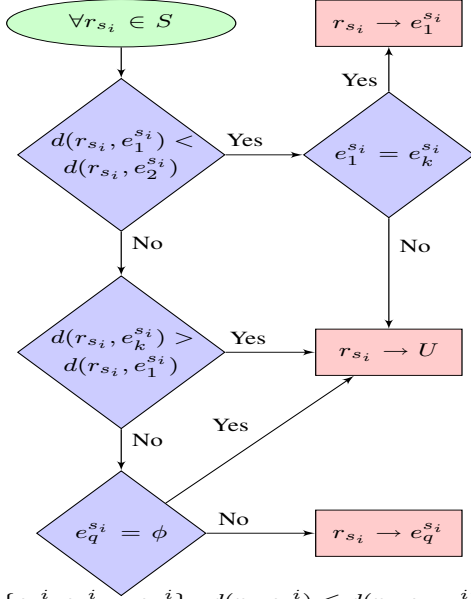
The load balancing constraint (4) is weighted by a parameter, Δ , which controls the maximum deviation of a DE's normalized workload from the average normalized workload for all DEs. Setting $\Delta = 1$ would force workloads of all DEs to be exactly equal to the average normalized workload, or in other words each DE will have the same fractional utilization of its capacity as all others. In case of homogeneous DE capacities this translates to an equal workload for all DEs. On the other hand, $\Delta > 1$ allows the normalized workload of at least one DE to be higher than the average by $(\Delta - 1) * 100$ percentage.

The value of Δ also dictates the trade-off between the objectives of minimum aggregate delay and load balancing as it changes the feasible set of solutions. A large value of Δ optimizes a solution for the objective of minimizing aggregate delay, while a tighter constraint will show significant trade-off in favor of load balancing.

Our approach is different from the traditional load balancing method of minimizing the maximum load, and provides better control to a network operator while ensuring robust and efficient operation of the Decision plane. The sub-problem with only the minimum delay objective and (2), (3) and (6) is commonly referred to as Terminal Assignment Problem [10].

We construct a two-phase exact algorithm to solve the optimization problem. The first phase of the algorithm constructs an ordering of routers, S , where S is the sorted order of minimum delay assignments for each router, and greedily assigns routers in the order of S to their closest (min-delay) feasible DEs, if such assignments are possible. To meet the contiguity constraint (5), a router r_i 's assignment is made to the closest DE e_j if $d(r_i, e_j)$ is strictly less than the delay between r_i and any other DE and e_j has slack capacity. On the other hand, if there are other DEs at same delay from r_i as e_j , r_i is assigned to a feasible DE that has an existing assignment in $N(r_i)$. Otherwise, r_i is kept unassigned.

The goal of the first phase of algorithm is to make all feasible lowest-cost assignments that can be made without changing any previously made assignments. This phase constructs an optimal solution for the assigned routers. Any router that remain unassigned after the first phase are assigned by the second phase using a branch exchange algorithm that iteratively accommodates previously unassigned routers, while maintaining feasibility of the solution. Our solution is $O(m^2n)$ in the worst case, and finds optimal solution to the assignment problem if it exists.



$$E^i = \{e_1^i, e_2^i, \dots, e_n^i\} : d(r_i, e_j^i) \leq d(r_i, e_{j+1}^i)$$

$$S = \{r_{s_1}, r_{s_2}, \dots, r_{s_m}\} : d(r_{s_i}, e_1^{s_i}) \leq d(r_{s_{i+1}}, e_1^{s_{i+1}})$$

$$U = \{\text{Set of unassigned routers}\}$$

$$k = \text{Index of the first feasible DE in } E^i$$

$$e_q^{s_i} \in E^{s_i} \quad k \leq q < n :$$

$$\exists r_a \in N(r_{s_i}), A(e_q^{s_i})$$

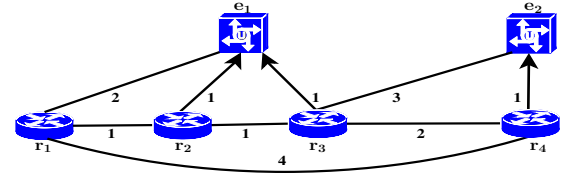
$$d(r_{s_i}, e_q^{s_i}) = d(r_{s_i}, e_1^{s_i})$$

Fig. 2: Greedy Phase Algorithm

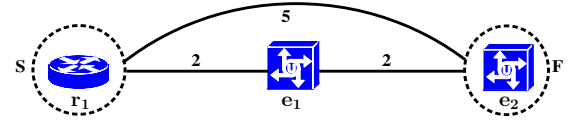
1) *Greedy Phase*: We utilize a greedy heuristic to assign routers to DEs while maintaining the feasibility of solution. Since, a greedy approach does not make any changes to its local decisions, the order in which decisions are taken becomes important. Our approach considers routers in the order of lowest assignment costs for each router. Assignments are made only with a feasible min-delay DE, where feasibility is determined by the constraints given in §IV-A. Fig. 2 describes the definitions and operation of this phase.

2) *Exchange Phase*: The greedy phase makes all the feasible min-cost router assignments that can be made without changing any existing assignment. Assignment of an unassigned router after the greedy phase's completion may involve a trade-off between sub-optimal assignment to available DEs or reassignment/exchange of already assigned routers to allow a lower cost assignment. Therefore, in order to ensure optimality of the solution, the assignment mechanism must be able to find the lowest-cost set of exchanges that allow the assignment of an unassigned router. This mechanism is provided by the exchange phase, which utilizes a branch-exchange algorithm, similar in design to the method described in [10], to construct an auxiliary graph of the network and uses shortest path algorithm for computing lowest-cost assignments.

In simple terms, auxiliary graph represents the feasible combinations of router assignment exchanges between DEs, weighed by the cost of such exchanges. The min-cost path through the graph represents the min-delay assignment for an unassigned router. Therefore, edges of the graph represent pos-



(a) Topology with r_1 unassigned.



(b) Auxiliary graph where $(S, e_1) = e_1 \leftarrow r_1$ and $(e_1, F) = e_2 \leftarrow r_3$

Fig. 3: Operation of the exchange phase on a network example where $\Delta = 1$ and edges are annotated with delay values. The min-cost assignment is along $(S, e_1), (e_1, F)$

sible feasible exchanges (and new assignments) between DEs which, themselves, are represented by the graph's vertices. Similar to the greedy phase, feasibility of any exchange or new assignment depends on conformance to the constraints presented in §IV-A. Auxiliary graph is constructed according to the following rules:

- There are two special vertices S and F that represent the source and destination vertices for the shortest path computation. The shortest path from S to F , at each iteration of exchange phase, provides the lowest cost assignment of one unassigned router.
- There are additional vertices, $Y = Y_1, Y_2, \dots, Y_k$, each corresponding to a fully loaded DE.
- There is an edge (S, Y_k) corresponding to potential assignment of an unassigned router $Y_k \leftarrow r_i : \exists r_a \in A(Y_k), r_a \in N(r_i)$ with an edge weight $d(r_i, Y_k)$.
- There is an edge (Y_k, Y_l) corresponding to a router r_i at the border of Y_k and Y_l 's areas, such that $x(r_i, Y_k) = 1, \exists r_a \in A(Y_l), r_a \in N(r_i)$ and the weight $d(r_i, Y_l) - d(r_i, Y_k)$ is positive.
- There is an edge (Y_k, F) corresponding to a router r_i 's feasible re-assignment from Y_k to a DE e_j with slack capacity. The weight of this edge is $d(r_i, e_j) - d(r_i, Y_k)$.
- There is an edge (S, F) with weight $d(r_i, e_j)$ for $e_j \leftarrow r_i$.

Dijkstra's shortest path algorithm is used to compute the shortest path from S to F on the directed auxiliary graph. This shortest path represents the minimum cost set of exchanges that are needed to assign a previously unassigned router. The auxiliary graph is updated after the assignment and the process repeated until all routers have been assigned. Fig. 3(a) shows the operation of the exchange phase on a simple example and Fig. 3(b) shows the construction of its auxiliary graph.

3) *Proof*: Omitted due to space limitations.

4) *Complexity*: The greedy phase of the algorithm is $O(m)$. The exchange phase's complexity is dependent on the shortest path computation, with worst case complexity of $O(n^2)$. The exchange phase calls Dijkstra's algorithm for each unassigned router, resulting in an overall worst case complexity

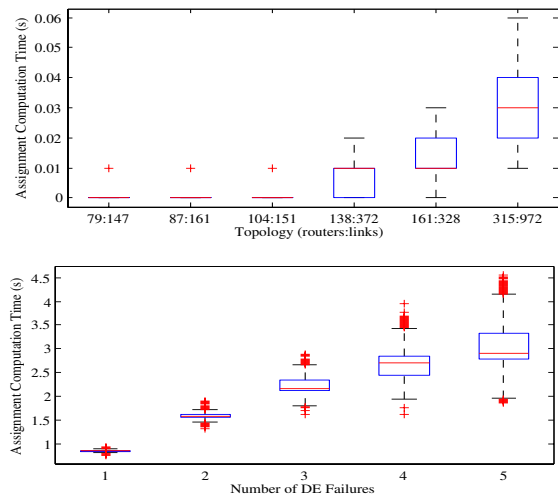


Fig. 4: Box Plot of the computation time for router re-assignment with $\Delta = 1$. The box shows the first and third quartile along with the median. Whiskers show the min. and max. values, while the outliers are plotted as “+”. Top: (a) Rocketfuel backbone topologies, Bottom: (b) BRITE topologies with $m = 1500$

of $O(mn^2)$. In reality, the greedy phase assigns most of the routers, and the few unassigned routers in tightly-constrained DE failure scenarios each require one iteration of the exchange phase. This results in average-case complexity of $\Theta(m+kn^2)$, where $k \ll m$. Also, since the number of routers in a network are expected to be much higher than the number of DEs, i.e. $m \gg n$, complexity of the scheme is dominated by the complexity of greedy phase.

V. NUMERICAL EVALUATION

In this section we provide the results of our performance evaluation of the assignment algorithm. Our first set of data consists of the ISP backbone topologies collected by Rocketfuel project [11]. The second set is comprised of artificial two-tiered hierarchical topologies generated by BRITE [12], which is used to model a large-sized ISP topology consisting of 1500 routers and 15 DEs.

Each failure in the Decision plane triggers the re-computation of the router assignments and we measured the time taken for each run of the assignment algorithm on a 64 bit 3.6 Ghz machine. The computation time required to run each iteration of the algorithm is plotted in Fig. 4 for both sets of topologies, with a worst-case DE capacity constraint of $\Delta = 1.0$. The plot shows that even in case of very large network topologies and worst-case constraints on load-balancing router assignment algorithm converges to a solution within very reasonable times.

VI. RELATED WORK

Several recent studies have embraced centralization of network logic as a way of overcoming management complexity or providing new services that are presently difficult to implement. Greenberg et al. [2] provide a comprehensive survey of the issues in network control and management, and propose the architectural vision embraced and extended in this paper.

Centralized control has been explored in BGP design where RCP [1], [13] was proposed as a logically centralized point for computing BGP routes and improving the scalability of large networks. However, RCP is limited to BGP route computation and does not extend to Interior Gateway Protocol (IGP) routes.

Several efforts in open router design [9], [14] have also advocated migration of control functions away from routers to reduce their complexity, where they utilize “control elements” for the implementation of distributed network algorithms, and design protocols to enable communication between different network elements. In contrast, our work uses 4D’s approach of network-wide decision making and presents a robust and scalable design for the Decision plane that is not limited to the implementation of current distributed algorithms.

VII. CONCLUSION

We presented the design of a clean-slate control and management plane, using the abstraction of 4D architecture, to simplify the management complexity in large enterprise and ISP networks. Our work focused on increasing the robustness and reliability of the Decision plane and included a novel method of adaptive assignment of routers to the logically centralized Decision Elements (DE). Evaluation of our algorithm on different topologies show its computational efficiency in reaching at optimal solution, and support the case for its use in network management.

ACKNOWLEDGEMENTS

This research was sponsored by the NSF under Award Nos. 0426886 and 0519728.

REFERENCES

- [1] N. Feamster *et al.*, “The case for separating routing from routers,” in *ACM SIGCOMM FDNA Workshop*, 2004.
- [2] A. Greenberg *et al.*, “A clean slate 4D approach to network control and management,” *SIGCOMM CCR*, vol. 35, no. 5, 2005.
- [3] D. A. Maltz, G. Xie, J. Zhan, and H. Zhang, “Routing design in operational networks: A look from the inside,” in *Proc. ACM SIGCOMM*. ACM Press, 2004.
- [4] A. Wool, “A quantitative study of firewall configuration errors,” *IEEE Computer*, vol. 37, no. 6, 2004.
- [5] A. Greenberg *et al.*, “Refactoring network control and management: A case for the 4D architecture,” Carnegie Mellon University, Tech. Rep. CMU-CS-05-117, Sept 2005.
- [6] H. Yan *et al.*, “Tesseract: A 4D network control plane,” in *USENIX NSDI*, 2007.
- [7] H. Iqbal and T. Znati, “Distributed control plane for 4D architecture,” in *Globecom 2007*. IEEE, 2007.
- [8] I. T. Union, “ITU-T recommendation Q.700: Introduction to CCITT Signalling System No. 7,” 1993.
- [9] T. V. Lakshman *et al.*, “The SoftRouter architecture,” in *HotNets-III*, 2004.
- [10] A. Kershenbaum, *Telecomm. network design algorithms*. McGraw-Hill, Inc., 1993.
- [11] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, “Inferring link weights using end-to-end measurements,” in *In ACM SIGCOMM IM Workshop*, 2002.
- [12] A. Medina, A. Lakhina, I. Matta, and J. Byers, “BRITE: An approach to universal topology generation,” in *MASCOTS*, 2001.
- [13] M. Caesar *et al.*, “Design and implementation of a routing control platform,” in *USENIX NSDI*, 2005.
- [14] L. Yang, R. D. T. Anderson, and R. Gopal, “RFC 3746 Forwarding and Control Element Separation Framework,” 2004.