

# Distributed Control Plane for 4D Architecture

Hammad Iqbal <sup>†</sup>, Taieb Znati <sup>†‡</sup>

<sup>†</sup>School of Information Sciences

<sup>‡</sup>Department of Computer Science

University of Pittsburgh

Email: {hiqbal, znati}@cs.pitt.edu

**Abstract**—We explore the design of a logically centralized but physically distributed control plane for 4D architecture. 4D architecture proposes centralization of network-wide decision making state and logic in a logical control plane to ease the management complexity of data networks. However, the current design implementations of the 4D control plane are limited to physical centralization. We argue that exploration of physically distributed control plane designs would be beneficial to the scalability and practical deployment of the architecture, and present design guidelines for different routing strategies that can be used to optimize the deployment of distributed 4D architecture.

## I. INTRODUCTION

The management complexity of the IP networks arises mainly due to the interaction of ever increasing functionalities, and their required state information, with the distributed nature of routing design, where each router independently computes and maintains the state required for its operation. The original distributed routing design, meant to keep the network simple and robust, is increasingly becoming more and more complex to accommodate new routing protocols and allow a richer set of functionalities as required in meeting traffic engineering, QoS, security, and survivability objectives. However, these improvements make the task of designing and managing these networks more and more difficult, as a network manager has to individually configure routers and switches using low level configuration commands while ensuring that the state computed by the distributed operation of the network is exactly what is needed for the proper operation of the network. This network management problem has so far been addressed, with limited success, by designing incremental tools that help in managing the router configuration tables.

Recently, 4D architecture [1] proposed a new approach where the fundamental cause of this management complexity is addressed. The 4D architecture advocates a new layering design of the IP networks which separates the task of packet forwarding from the control logic required to operate the network elements. This separation of data and control layers is in contrast with the current practice where data forwarding mechanism and control logic are inter-twined inside a network router. This approach has the potential of reducing the cost of network devices as well as reducing the complexity of network management, while maintaining the robustness and resiliency of IP networks.

This research was supported by the NSF under awards 0426886 and 0519728.

The 4D architecture proposes a logically centralized control plane, but so far the investigations of the design have been limited to physical centralization [2], [3] where a Decision Element (DE) collects the required network information, maintains the algorithms required for computing network state, and transmits this state information to the data forwarding elements i.e. routers and switches. The fault tolerance of the control plane is then augmented with multiple stand-by DEs which can takeover in case of failure. While such physical centralization is good as a first order evaluation example, practical deployment of the 4D architecture may be restricted by questions about the overall fault-tolerance, response time, and scalability of the physically centralized control plane.

The first contribution of this paper is that we extend the 4D architecture by exploring the design of a logically centralized but physically distributed 4D control plane, where more than one DEs can collaborate to compute the required network state. We believe this logical extension to the current 4D proposal is necessary to make it scalable with the size of the network, as well as in making it more robust to DE failures. As an example, while a centralized DE design might be attractive for a small to mid-sized campus network, the network latency of a large geographically dispersed enterprise network would result in higher response times in case of failures, making such a choice unattractive. Also, we note that while the 4D control plane might be enrolled in traffic management, threat monitoring, and security tasks, the complexity of even the basic shortest-path reachability computation on a centralized DE rises super-linearly with the size of the Autonomous System (AS) [2], indicating a maximum network size where such a design might be deployed. These observations reinforce the need to extend the centralized 4D control plane but there are also several trade-offs, first being the increase in complexity that results when any distributed computation is employed in the control plane. Recalling that the distributed computation complexity of the present routers is the reason that we are pursuing 4D design in the first place, we note the need to optimize the control plane design so that the overall burden of distributed computation can be minimized. Our second contribution is that we use linear optimization strategies to analyze the deployment strategies for DEs in a distributed 4D architecture. These optimization strategies are decoupled from any particular design of the distributed architecture to ensure the application generality.

We present related work in section II. Section III introduces

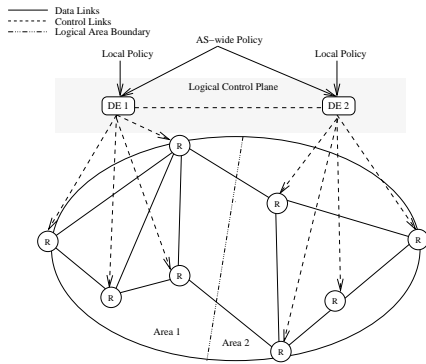


Fig. 1. High-level Overview of the Distributed 4D Design.

the rationale behind the distributed 4D design and presents a linear optimization based framework for optimizing the placement of DEs in an AS. Section IV presents the results of our investigation of optimizing DE placement in rocketfuel topologies. Section V concludes our work.

## II. RELATED WORK

Alternatives to the current IP network’s distributed routing approach were explored early on by several specialized networks. Most notably among them were IBM SNA [4] and TYMNET [5]. Legacy IBM SNA employed dedicated network controllers to compute the routes in a session based host-terminal network. TYMNET used a single “Network Supervisor” to compute the routes for a virtual circuit based network. TYMNET’s use of a centralized Network Supervisor is analogous to using a single DE in the centralized 4D architecture. In TYMNET’s case, the scalability of the network was constrained by the resource bottleneck at the Network Supervisor, limiting the network size to 500-600 nodes. While realizing the technological advances in computation power and bandwidth availability, we believe that a centralized design would still be limited in a maximum network size because of the increase in the routing constraints required by various QoS, robustness, and security objectives, as opposed to the simple connectivity requirement in TYMNET.

Routing Control Platform (RCP) [6], [7] was proposed as a logically centralized point for computing Border Gateway Protocol (BGP) routes and improving the scalability of large networks. However, RCP is limited to BGP route computation and does not extend to the Interior Gateway Protocol (IGP) routes. Recently, IRSCP [8] has been proposed as an intelligent route selector for a network, where it performs computation of BGP routes using not only the IGP information but also input from a intelligent system aware of other aspects of the network such as load conditions and DDoS attacks.

## III. DISTRIBUTED 4D NETWORK DESIGN

Figure 1 shows a high level view of the 4D architecture where the AS network is logically partitioned into two areas, each of which is controlled by a DE. The partitioning is logical because the DEs, grouped together, form the logical control plane; exchanging information with each other needed to

maintain the network-wide control and maintaining consistent decision-making from a router/switch’s perspective. A direct implication of this partitioning is that a DE will have access to full reachability information about its own area, but may have access to only partial and relevant information about other areas. From a DE’s perspective, this means an exchange of the centralized 4D’s global AS-wide network view with a constrained view comprised of full local-area view and a partial view of the peer-area(s). The extent of the peer-area view depends on the control plane task for which it is needed. For example, link status and reachability information provided by the peer areas through a Link State Advertisement (LSA) packet is sufficient for shortest-path routing. On the other hand, the same peer-area view may not suffice for computing optimal reachability when traffic-dependent link weights [9] are used. Other possible control plane tasks such as load balancing, threat monitoring, etc., may require different levels of peer-area views necessary for their operation. Therefore, we note that while the minimization of the inter-DE exchanges is necessary to achieve better scalability and robustness, the minimum level of peer-area view can not be determined *a priori* for all tasks that may involve the 4D control plane. Instead of hardwiring the maximum AS-wide view in the design, we allow more variation by believing that the nature of tasks that are added to the 4D control plane will determine the right balance of peer-area view, and leave the actual split of the AS-wide view to the network designer who is in a better position to determine the necessary peer-area view needed for optimal completion of the control plane task. This modularity of design to accommodate different design preferences is in-line with the principle of modularization along tussle boundaries [10].

Our second enhancement to the centralized 4D design is the addition of local-area policies as input to the 4D control plane. As illustrated in figure 1, AS-wide policy is consistent with all the DEs and may include policies related to security, traffic management, inter-domain routing, etc. This specification of AS-wide policy is one of the design goals of the 4D network, which specifies that network wide policy should be available to the decision/control layer for optimal decision making. However, the network management may also need control over policy issues related to individual area that does not affect the whole AS network. As an example, a planned maintenance event inside an area may not have network-wide implications, if inter-area routing exchange does not change as a result. Such local events may be easily controlled with the help of local-area policy giving some control to local network administrators in policy issues that do not require network-wide coordination. Our proposed architecture is in line with the common observation that most large AS networks are partitioned along divisional and geographical boundaries, with each partition operating with some level of independent control. Therefore the division of network policy into AS-wide and local-area should help in maintaining the natural network organizational structure and result in easier transition to the 4D architecture.

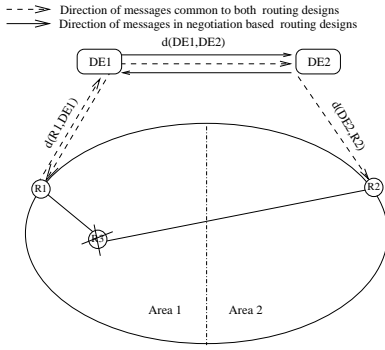


Fig. 2. Update messages triggered by failure of R3.

A DE failure in 4D control plane can have major implications for the network operation. In order to increase the robustness of the network to failures in the control plane, we propose that a goal in designing the 4D network should be  $k$ -coverage - defined for any router/switch in the AS as the number of DEs covering the router in the 4D control plane so that a failure in any one DE should not affect the control plane operation. This measure of 4D control plane robustness is applicable to the centralized 4D design where hot stand-by DEs are used to cover the failure in the primary DE.

#### A. Optimal DE Placement

In this section we examine different strategies for positioning DEs inside an AS network. The optimal positioning of DEs is important as the performance of a distributed 4D architecture is heavily influenced by the placement of DEs within the AS. We can consider several objectives in defining the optimality: minimization of network cost, convergence delay in case of failure, DE response time, and DE-DE delays can each be considered as optimization objectives. However, these objectives taken together can be contradictory; for example network cost is minimized with a centralized DE, as the single central DE is cheaper than multiple local-area DEs and allows us to gain in economy of scale, while the minimization of DE response time suggests a higher number of DEs to minimize the propagation delays between DEs and routers. In our discussion of the placement strategies, we consider minimization of the convergence delay as the primary objective. Minimization of convergence delay is important as it will reduce the time for the routing to stabilize after any topology changes. Since, the actual convergence times are dependent on the routing protocol specifics, we will generalize the worst case convergence delays separately for two different routing strategies: first, where the routing decisions in different areas can be taken independently; and second, those routing strategies that require co-operation among DEs to achieve consistent routing decisions.

Figure 2 illustrates both the cases where the routing decisions in an area are independent or negotiation based. In the first case, failure of router R3 triggers a link-state update from router R1 to DE1, which may involve a timeout at R1 waiting for keep-alive message on link R1-R3. DE1 will then compute

the updated routing tables, presumably after waiting for the expiration of a hold-timer to collect all link-state updates related to the same event, and inform R1 as well as DE2. DE2 will, in turn, compute the updated routing tables for its own area and inform router R2. In this scenario we assumed a routing policy that takes local decisions at each DE based on the available information, without exchanging any more messages than what is needed to disseminate the information about the event. We note that shortest path routing exhibits this property as each entity takes local decisions without negotiating the possible choices with other entities. The total time for achieving convergence in this case will be:

$$t_{total} = t_{timeout}(R1) + d(R1, DE1) + t_{hold}(DE1) + t_{compute}(DE1) + d(DE1, DE2) + d(DE2, R2)$$

The values of  $t_{timeout}$ ,  $t_{hold}$ , and  $t_{compute}$  are protocol-specific and can be assumed to be constant  $c$  for a given network size. Therefore, the worst-case convergence delay happens at the maximum of propagation delays:

$$t_{max} = 2d_{max}(R, DE) + d_{max}(DE, DE) + c \quad (1)$$

In the second case of negotiation based routing, the computation of routing table update requires an exchange of messages between the DEs. Such exchange may be required in routing policies where the objective is to solve a multi-constrained optimization problem and each DE takes part in negotiating the globally optimal solution. In this case, the number of DE-DE exchanges may outnumber the Router-DE messages and so minimizing the convergence delay would require minimizing the aggregate DE-DE delays. We also note that DE-DE exchanges/negotiations may also be necessary in control plane tasks other than routing, and so the applicability of this placement strategy may extend beyond multi-constrained route optimization problems.

In order to minimize the worst-case convergence delay, we need to compute a DE placement strategy that minimizes the Router-DE and DE-DE delays. We cast this placement problem as a modification of the capacitated  $p$ -median problem [11], where  $p$  is the required number of DEs in the AS. The dissemination plane is assumed to be built from shortest paths, as in the case of centralized 4D design. The computation of  $p$  is an operational decision that is likely to vary from AS to AS, influenced by several factors:

- 1) *AS topology* will influence the number of required DEs in several ways. In AS topologies with large geographical distances between routers, the required number of DEs will be higher to constraint the Router-DE delay. Organizational structure and business division boundaries will affect the number of DEs, while higher density of edges in the topology graph may reduce the required number of DEs.
- 2) *Technological Constraints* include the computational and storage capacities of the DEs. While the computational load for shortest-path routing is within the capacity of a DE build using general-purpose machine [3], the workload on the 4D control plane will increase with the number of control plane tasks.
- 3) *Robustness Objectives* will require redundancy in the 4D control

plane to avoid single points of failure. One such objective is the  $k$ -coverage of the routers and other data plane devices, which requires at least  $k$  DEs in the AS network. Constraints on the maximum length of Router-DE multi-hop path can also be considered to reduce the susceptibility of Router-DE control path to link failures. 4) *Cost* of the 4D network would be largely dependent on the number of DEs and so minimizing the cost will minimize the number of DEs. 5) *Performance Objectives* such as the minimization of convergence delay depend on the value of  $p$ .

The formulation of the problem requires the value of  $p$  as an input. If there is no readily apparent value for  $p$ , a network designer can compute the DE positioning for several values of  $p$  and compare the outcomes on cost vs. benefit.

### B. Formulation of the DE placement problem

Let  $R = \{r_1, r_2, \dots, r_m\}$  be the collection of routers in the AS and  $p$  be the number of DEs to be positioned in the network. Let  $L = \{l_1, l_2, \dots, l_n\}$  be the set of possible locations for the  $p$  DEs. We define  $A$  as the total number of DE-DE adjacencies, that is  $A = p(p-1)/2$ . Let  $d_{ij}$  be the shortest-path delay between router  $r_i$  and a possible DE location  $l_j$ . The delay  $d_{ij}$  may include queuing and transmission delays, in addition to the propagation delay, especially when multi-hop paths are used between a router and a DE. Let  $d_{jk}$  be the delay between locations  $l_j$  and  $l_k$ . Let  $w_i$  be the measure of  $r_i$ 's workload, defined as  $r_i$ 's projected demand on a DE's resources which includes computational, memory, and bandwidth demands. We propose using the size of the routing table as a proxy for the computational demand. Let  $Q_j$  be the maximum workload that a DE at location  $l_j$  is able to sustain.  $x_{ij}$  and  $y_{jk}$  are binary variables with  $x_{ij} = 1$  if  $r_i$  is allocated to the DE at  $l_j$ , and  $y_{jk} = 1$  if a DE-DE adjacency is identified between  $l_j$  and  $l_k$ . The linear programming formulation given below will indicate the position of DE at site  $l_j$  if  $y_j = 1$  using the minimization objective in Eqn. (1).

$$\text{Min} \sum_{j \in L} \left( \sum_{i \in R} 2d_{ij}x_{ij} + \sum_{\substack{k \in L \\ k \neq j}} d_{jk}y_{jk} \right) \quad (2a)$$

subject to:

$$\sum_{j \in L} x_{ij} = 1 \quad i \in R \quad (2b)$$

$$\sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} y_{jk} = A \quad (2c)$$

$$\sum_{\substack{k \in L \\ k < j}} y_{jk} + \sum_{\substack{k \in L \\ k > j}} y_{kj} - (p-1)y_j \leq 0 \quad j \in L \quad (2d)$$

$$\sum_{j \in L} y_j = p \quad (2e)$$

$$\sum_{i \in R} w_i x_{ij} - Q_j y_j \leq 0 \quad j \in L \quad (2f)$$

$$x_{ij}, y_{jk}, y_{kj} \in \{0, 1\} \quad i \in R \quad j, k \in L \quad (2g)$$

Constraint (2b) ensures that a router is assigned to exactly one DE. Constraint (2c) is used to guarantee the correct number of DE-DE adjacencies in the objective function. Constraint (2d) forbids adjacencies for locations where a DE is not present. Constraint (2e) limits the total number of DEs to  $p$ . Finally, by Constraint (2f) we ensure that the total assigned workload at a location does not exceed the available capacity at that location. We observe that this formulation's objective is more sensitive to the aggregate delays between routers and DEs, in comparison to the DE-DE delays, as the number of routers is greater than the DEs. Therefore, there will be more terms where  $d_{ij}x_{ij}$  is positive as compared to terms where  $d_{jk}y_{jk}$  is positive. This will increase the sensitivity of this formulation to router-DE delays. For the routing strategies where the routing decision at a DE may not be taken independently, we minimize the DE-DE delay while bounding the maximum router-DE delay by a constant  $B$ . Our new LP formulation, with the router-DE delay bounded by  $B$  in constraint (3g), is:

$$\text{Min} \sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} d_{jk}y_{jk} \quad (3a)$$

subject to:

$$\sum_{j \in L} x_{ij} = 1 \quad i \in R \quad (3b)$$

$$\sum_{j \in L} \sum_{\substack{k \in L \\ k \neq j}} y_{jk} = A \quad (3c)$$

$$\sum_{\substack{k \in L \\ k < j}} y_{jk} + \sum_{\substack{k \in L \\ k > j}} y_{kj} - (p-1)y_j \leq 0 \quad j \in L \quad (3d)$$

$$\sum_{j \in L} y_j = p \quad (3e)$$

$$\sum_{i \in R} w_i x_{ij} - Q_j y_j \leq 0 \quad j \in L \quad (3f)$$

$$d_{ij}x_{ij} \leq B \quad i \in R \quad j \in L \quad (3g)$$

$$x_{ij}, y_{jk}, y_{kj} \in \{0, 1\} \quad i \in R \quad j, k \in L \quad (3h)$$

These linear programs can be solved using either stand-alone LP solvers e.g. CPLEX [12] and GLPK [13], or by using approaches discussed in [14], [15], and references therein. In §IV we discuss results obtained for real AS topologies using GLPK solver. Since the 4D architecture proposes the separation of data and dissemination paths, the bound on maximum delay ( $B$ ) and DE work-load ( $K_j$ ) provided in the two formulations will not be affected by the dynamic routing choices in normal conditions. However, failures of dissemination paths, due to failures in either control or data planes, can lead to discovery of new dissemination paths that violate the bounds on  $B$  and  $K_j$ . The magnitude of deviation from these bounds will depend on the connectivity of the AS topology graph.

## IV. DESIGN EVALUATION

In this section, we provide computational results for the optimal DE placement problem given in § III. We investigated

five different AS topologies from the rocketfuel project [16], and utilized the `glpsol` utility in `GLPK` [13] to solve the linear programs for DE placement. Rocketfuel reports latencies and inferred weights between pair of vertices (routers), and we used the latency values between vertices  $i$  and  $j$  as the measure of shortest-path delay  $d_{ij}$  in our model. Since rocketfuel measurements were made online, this measure of delay contains average queuing delay between the pair of vertices in addition to the propagation delay. The AS topologies were checked for connectivity and the largest connected component was utilized when full connectivity was not found in the instance graph. The number of routers  $m$ , the maximum shortest-path delay  $d_{max}$ , and the average shortest-path delay  $d_{avg}$  for the instances are shown in table I.

TABLE I  
ROCKETFUEL TOPOLOGY SUMMARY

AS Number	Vertices	$d_{max}(ms)$	$d_{avg}(ms)$
1221	104	54	7.82
1755	87	47	6.25
3257	161	83	7.77
3967	79	105	11.93
6461	138	137	17.43

To limit the size of the problem, we considered  $n = 10, 15, 20$  most central vertices as possible locations  $L$  for the DEs. We used the “betweenness” of a vertex as the measure of centrality and goodness of choice when a DE is located at that vertex. Betweenness is a measure of centrality, commonly used in social networks and network survivability analysis, that values those vertices more which occur on shortest paths (geodesic) between many other vertices. Betweenness is formally defined for a vertex  $v$  as [17]:

$$C_{B(v)} = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where  $\sigma_{st}$  is the number of shortest paths from  $s$  to  $t$  and  $\sigma_{st}(v)$  is the number of shortest paths that  $v$  lies on. Figures 3 shows the plot of the average delay between a router and a DE for different rocketfuel topologies as a function of  $p$  for  $n = 10$  possible DE locations. The plots for  $n = 15$  and  $20$ , not included here for brevity, showed similar results indicating that investigation of only a small subset of most central locations is sufficient in locating optimal DE placement. It can be seen from the figure that the knee of the average delay contours occurs around  $p = 3$  to  $4$  in the tested topologies. Reduction in the average delay is evident in comparison to the observed delays in table I. This shows that a distributed control plane with even a few DEs will give much better route convergence delays as compared to a centralized design.

## V. CONCLUSION

We explored the design of a logically centralized and physically distributed 4D control plane and presented guidelines for optimizing the physical design of the same. Our analysis shows that the distributed 4D design can efficiently provide

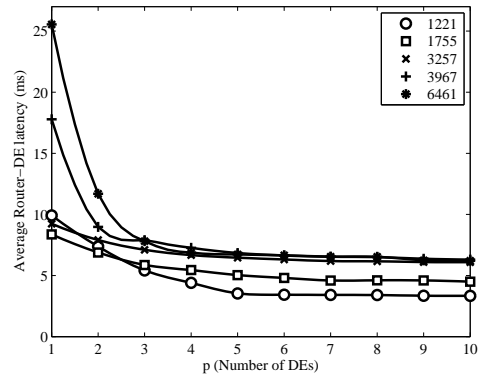


Fig. 3. Plot of average Router-DE delay for  $n = 10$

optimal network control for shortest-path routing, while the optimality for more sophisticated routing approaches needs to be investigated. We believe that the distributed control of the 4D architecture can enhance its scalability and we hope that our work will encourage future research efforts on its design and adaptation.

## REFERENCES

- [1] A. Greenberg, G. Hjalmtysson, D. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, “A clean slate 4D approach to network control and management,” *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 41–54, 2005.
- [2] A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, “Refactoring network control and management: A case for the 4D architecture,” Carnegie Mellon University, Tech. Rep. CMU-CS-05-117, Sept 2005.
- [3] H. Yan, D. Maltz, T. Ng, H. Gogineni, H. Zhang, and Z. Cai, “Tesseract: A 4D network control plane,” in *USENIX Symposium on Networked Systems Design and Implementation (NSDI '07)*, April 2007.
- [4] D. Lynch, J. Gray, and E. Rabinovitch, Eds., *SNA and TCP/IP Enterprise Networking*. Prentice Hall, 1997.
- [5] L. Tymes, “Routing and flow control in TYMNET,” *IEEE Transactions on Communications*, vol. 29, pp. 392–398, Apr 1981.
- [6] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. Merve, “The case for separating routing from routers,” in *ACM SIGCOMM Workshop on Future Directions in Network Architecture*, 2004.
- [7] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. der Merwe, “Design and implementation of a routing control platform,” in *Networked Systems Design and Implementation (NSDI)*, 2005.
- [8] J. V. der Merwe et al., “Dynamic connectivity management with an intelligent route service control point,” in *SIGCOMM workshop on Internet network management (INM)*, 2006.
- [9] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1992.
- [10] D. Clark, J. Wroclawski, K. Sollins, and R. Braden, “Tussle in cyberspace: defining tomorrow’s internet,” *IEEE/ACM Transactions on Networking*, vol. 13, pp. 462–475, 2005.
- [11] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley-Interscience, 1995.
- [12] *CPLEX System User Guide, Version 10.0*, ILOG Inc, 2006.
- [13] *GLPK (GNU Linear Programming Kit) Reference Manual*, Free Software Foundation, Inc, 2006.
- [14] E. Senne and L. Lorena, *Computing Tools for Modeling, Optimization and Simulation*. Kluwer Academic Publishers, 2000, ch. Lagrangean/Surrogate Heuristics for p-Median Problems.
- [15] C. Alberto and R. Giovanni, “A branch-and-price algorithm for the capacitated p-median problem,” *Networks*, vol. 45, no. 3, 2005.
- [16] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with rocketfuel,” in *Proceedings of SIGCOMM*. ACM Press, 2002.
- [17] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, vol. 40, pp. 35–41, 1977.