# 向　乔　博士


| | | |
|---|---|---|
| 办公地址: | 51 Prospect Street, Room 204 | 联系电话: 1-313-466-0452 |
| | Department of Computer Science | 电子邮件: qiao.xiang@cs.yale.edu |
| | Yale University | 出生日期: 1985 年 5 月 4 日 |
| | New Haven, Connecticut | 主页: www.cs.yale.edu/homes/qiaoxiang |
| | United States 06511 | 简历日期: 2019 年 12 月 |


## 教育经历

| | | |
|---|---|---|
| 2014 | 计算机科学博士 | 美国韦恩州立大学 Wayne State University, (导师: 张洪伟) |
| 2011 | 计算机科学硕士 | 美国韦恩州立大学 Wayne State University, (导师: 张洪伟) |
| 2007 | 信息安全工学学士 | 南开大学, (导师: 袁晓洁) |
| 2007 | 经济学学士 | 南开大学, (导师: 高洁) |

## 工作经历

| | |
|---|---|
| 2019.02 至今 | 研究助理教授, 美国耶鲁大学计算机科学系, Yale University |
| 2016.02 – 2019.01 | 博士后研究员, 美国耶鲁大学计算机科学系, Yale University |
| 2014.04 – 2015.04 | 博士后研究员, 加拿大麦吉尔大学计算机科学学院, McGill University |
| 2012.05 – 2012.08 | 实习研究员, 三星美国研发中心, Samsung Information Systems America, |

## 研究方向

分布式系统，可编程网络，物联网，机器学习与优化理论

## 学术创新

近年来发表高水平学术论文 37 篇（一作或通信作者共 18 篇），合著专著一部，包括重要国际会议 **IEEE/ACM Supercomputing**, **IEEE INFOCOM**, **AAAI**, **ACM Multimedia**, **IEEE RTSS**, ACM MobiHoc, IEEE/ACM IWQoS，IEEE MASS, IEEE RTCSA 与期刊 **IEEE JSAC**, **IEEE TMC**, **IEEE/ACM ToN**, Computer Networks, FGCS, IEEE TIE, IEEE TCC, IEEE TSG, IEEE Communications Magazine 等。其中 CCF A 类论文 12 篇（一作且通信作者 8 篇），B 类论文 3 篇，C 类论文一作 3 篇，IEEE MASS 会议（CCF C 类）最佳论文奖一作 1 篇，大会特邀论文一作 2 篇，IEEE/ACM Trans 5 篇（一作且通信作者 2 篇）。

**影响力**：(1) 主持设计了基于软件定义网络，机器学习与可信多方安全计算的多管理域科学计算网络细粒度资源管理系统；在 CERN 大型例子对撞机数据传输与分析网络上完成初步部署，实现 100 Gbps 工作流实时最优调度; 系统设计作为互联网草案提交 IETF ALTO 工作组审议；(2) 主持设计了 OpenSDC 可编程数据通路架构，引入数据包网内计算等一系列编程原语，拓展软件定义网络数据通路，提高移动军事网络数据通路可编程性；该架构正在美国陆军实验室 (ARL), 英国国防部实验室 (Dstl) 与国际技术联盟 (International Technology Alliance) 进行测试与标准化；(3) 主持设计了一系列用于下一代车联网的车载通信协议，包括基于数据包价值创新概念的多跳信息广播协议，基于控制理论与机器学习的信息广播联合控制系统，及首个使用调频信号与机器学习的全地形精确车辆定位系统；多项协议被美国通用汽车公司整合入下一代车载通信系统。

**主持/参与项目**

2019.9 至今 Facebook Research Award: Toward Highly Reliable, Programmable, and Efficient Network Control, 共同主持

2017 – 2019 International Technology Alliance Agreement No W911NF-16-3-0002: Software Defined Coalitions, 参与

2016 – 2018 NSF grant #1440745, CC*IIE Integration: Dynamically Optimizing Research Data Workflow with a Software Defined Science Network, 参与

2016 – 2018 Google Research Award, SDN Programming Using Just Minimal Abstractions, 参与

2014 – 2015 NSERC Collaborative Research and Development Grant (CRDJP 418713), 参与

2014 – 2015 CFI Leaders Opportunity Fund 23090, 参与

2014 – 2015 General Motors Research Award, Connected Vehicles Using DSRC, 参与

2011 – 2015 NSF grant CNS-1136007, CPS Medium: A Cross-Layer Approach to Taming Cyber-Physical Uncertainties in Vehicular Wireless Networking and Platoon Control, 参与

2011 – 2015 NSF grant CNS-1054634, CAREER: Taming Uncertainties in Reliable, Real-Time Messaging for Wireless Networked Sensing and Control, 参与

2013 – 2015 GENI-1890, GENI-Enabled Vehicular Sensing and Control Networking: from Experiments to Applications, 参与

2011 – 2014 GENI-1633, WiMAX Prototyping in Metro Detroit: Integrating GENI Engineering with Wireless Network Applications and Science, 参与

2008 – 2011 GENI-fying and Federating Autonomous Kansei Wireless Sensor Networks, 参与

**特邀报告**

2017.11 Unicorn: Unified Resource Orchestration for Multi-Domain Data Analytics, 新加坡国立大学

2016.11 Simplifying SDN Programming using a Data-Driven Function Store: A Journey Originating from Routing State Abstraction, 美国韦恩州立大学

2016.08 Auc2Reserve: A Differentially Private Auction for Electric Vehicle Fast Charging Reservation, IEEE RTCSA 2016 特邀论文, 韩国

2016.07 Toward Real-time, Reliable and Efficient Services in Smart City, 新加坡国立大学

2015.08 Emerging Topics in Wireless Networking, 南开大学

2015.05 Designing Real-Time, Reliable and Efficient Cyber-Physical Systems for Future Smart City, 美国麻省理工学院

2015.04 Towards Real-time, Reliable and Efficient Service in Wireless Cyber-Physical Systems, 美国麦克丹尼尔学院

2014.12 In-Network Processing in Wireless Cyber-Physical Systems, 中国石油大学北京分校

2014.12 In-Network Processing in Wireless Control Systems: Experience and Case Studies, 南开大学

**教学经验**

耶鲁大学 Yale University

2017 秋季学期 助教, 计算机网络 Computer Networks (被学生评为“最佳助教”)

2017 春季学期 首席助教, 计算机编程基础 Introduction to Computer Programming

2016 春季学期 助教, 计算机网络 Computer Networks

韦恩州立大学 Wayne State University

2007 – 2013 讲师: 计算机科学基础 Introduction to Computer Science, 操作系统 Computer Operating Systems-Lab, 计算机体系结构 Computer Architecture and Organization-Lab

2009 – 2013 研究生课程助教: 网络编程 Network, Distributed and Concurrent Programming, 语言与自动机理论 Theory of Languages and Automata, 计算机网络 Data Communication and Computer Networks, 高级计算机网络 Advanced Computer Networking and Seminar in Networking, 分布式与并行系统 Distributed Systems and Parallel Systems

2009 – 2012 本科生课程助教: 算法设计与分析 Algorithm Design and Analysis, 理论计算机基础 Introduction to Theoretical Computer Science, 操作系统 Computer Operating Systems, 计算机网络 Introduction to Computer Networking

南开大学

2006 秋季学期 讲师: 数据库系统实验 Database Systems-Lab, 可视化编程实验 MFC Lab

**学术活动**

ACM SIGCOMM 2020 Workshop on Network Application Integration/CoDesign (NAI 2020), 技术程序委员, 2020

IEEE/ACM International Symposium on Quality of Service (IWQoS), 技术程序委员，2020

IEEE Vehicular Networking Conference (VNC), 宣传主席, 2019

ACM Workshop on Electric Vehicle Systems, Data and Applications (EV-Sys), 技术程序委员, 2017

ARO Workshop on Software Defined Networking for Army Applications (SDNA), 网站主席, 2016

International Journal of Wireless Communication (IJWC), 编辑委员会委员, 2015 至今

ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT), 影子技术程序委员, 2015

Euromicro Conference on Digital System Design (DSD), 技术程序委员, 2015 至今

IEEE International Conference on Computer Communications and Networks (ICCCN), 技术程序委员, 2015

IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 技术程序委员, 2012 至今

IEEE Sensors Applications Symposium (SAS), 技术程序委员, 2013 至今

International Conference on Network-Based Information Systems (NBiS), 技术程序委员, 2015

International Conference on Smart Sensors and Application (ICSSA), 技术程序委员, 2015

### 学术荣誉

| | |
|---|---|
| 2019 | IEEE MASS 大会最佳论文奖 |
| 2013 – 2018 | 学术会议资助 Travel Grants (多次), SIGCOMM, SIGMETRICS, ICNP, eEnergy, IC2E, and Yale University |
| 2016 – 2018 | 博士后 Fellowship, 耶鲁大学 Yale University |
| 2014 – 2015 | 博士后 Fellowship, 麦吉尔大学 McGill University |
| 2007 – 2013 | 助教/助研奖学金, 韦恩州立大学 Wayne State University |
| 2006 | 创业杯挑战大赛一等奖，天津医科大学 |
| 2003 – 2007 | 优秀学生奖学金 (三次)，南开大学 |

### 教学荣誉

| | |
|---|---|
| 2012 – 2013 | 杰出教学奖, 韦恩州立大学 Wayne State University |

### 其他

My Erdös number is 3: Qiao Xiang → James Aspnes → Miklós Ajtai → Paul Erdös

**专著及论文**

专著

2015 1. **Qiao Xiang**, Hongwei Zhang, In-Network Processing in Wireless Sensor Networks, *Handbook of Sensor Networking: Advanced Technologies and Applications, Chapter 4*, CRC Press.

期刊论文

2020 12. Yuwei Xu, Shuai Tong, Tiantian Zhang, Wen Sun, Xiaoyan Hu, **Qiao Xiang**, COMPASS: Directing Named Data Transmission in VANETs by Dynamic Directional Interfaces, to appear in IEEE Access, 影响因子，4.098,

11. Xingjian Lu, Fanxin Kong, Xue Liu, Jianwei Yin, **Qiao Xiang**, Huiqun Yu, Bulk Savings for Bulk Transfers: Minimizing Energy Cost on Inter-Data-Center Traffic, in *IEEE Transactions on Cloud Computing*, 影响因子 5.967.

2019 10. **Qiao Xiang**, Jingxuan Zhang, Xin Wang, Yang Liu, Chin Guok, Franck Le, John MacAuley, Harvey Newman, Yang Richard Yang, Toward Fine-Grained, Privacy-Preserving, Efficient Multi-Domain Network Resource Discovery, in *IEEE Journal on Selected Areas in Communications (**JSAC**)*, 影响因子, 9.302, **CCF A** 类.

9. Kai Gao, **Qiao Xiang**, Xin Wang, Yang Richard Yang, Jun Bi, An Objective-Driven On-Demand Network Abstraction for Adaptive Applications, in *ACM/IEEE Transactions on Networking (**ToN**)*, 影响因子, 3.597, **CCF A** 类.

2018 8. **Qiao Xiang**, Xin Wang, Jingxuan Zhang, Harvey Newman, Yang Liu, Yang Richard Yang, Unicorn: Unified Resource Orchestration for Multi-Domain, Geo-Distributed Data Analytics, in *Future Generation Computer Systems (**FGCS**)*, 影响因子, 5.768, **CCF C** 类.

2017 7. Linghe Kong, Xi Chen, Xue Liu, **Qiao Xiang**, Yi Gao, Noam Ben Baruch, Guihai Chen AdaSharing: Adaptive Data Sharing in Collaborative Robots, in *IEEE Transactions on Industrial Electronics (**TIE**)*, 影响因子 7.168.

2016 6. H. Newman, M. Spiropulu, J. Balcas, J. Bendavid, T. Hendricks, D. Kcira, I. Legrand, A. Mughal, J.R. Vlimant (Caltech/HEP); P. Spentzouris, P. DeMar (Fermilab); I. Monga, C. Guok (ESnet/LBNL); K. Riley, W. Allcock, V. Vishwanath, L. Winkler (Argonne LCF); R.Y. Yang, M. Chen, G. Kai, X. Lin, **Q. Xiang**, J. Zhang (Yale) (alphabetical order except PI), Next Generation Exascale Network Integrated Architecture for HEP and Global Science, Whitepaper for US HPC Leadership.

5. Linghe Kong, Daqiang Zhang, Zongjian He, **Qiao Xiang**, Jiafu Wan, Meixia Tao, Embracing Big Data with Compressive Sensing: A Green Approach in Industrial Wireless Networks, to appear in *IEEE Communications Magazine*, 2016, 影响因子 10.435.

4. Linghe Kong, **Qiao Xiang**, Xue Liu, Xiao-Yang Liu, Xiaofeng Gao, Guihai Chen, Min-You Wu, ICP: Instantaneous Clustering Protocol for Wireless Sensor Networks, *Computer Networks*, special issue on "Internet of Things", 2016, 影响因子 2.516, **CCF B** 类.

2013 3. Xiaohui Liu, Hongwei Zhang, **Qiao Xiang**, Xin Che, Xi Ju, Taming Uncertainties in Real-Time Routing for Wireless Networked Sensing and Control, *IEEE Transactions on Smart Grid (**TSG**)*, special issue on "Smart Grid Communication Systems", 4(1), pp. 288-301, March 2013, 影响因子 6.645.

2011 2. **Qiao Xiang**, Jinhong Xu, Xiaohui Liu, Hongwei Zhang, Loren J. Rittle, When In-Network Processing Meets Time: Complexity and Effects of Joint Optimization in Wireless Sensor Networks, *IEEE Transaction of Mobile Computing (**TMC**)*, 10(10), pp. 1488-1502, October 2011, 影响因子 3.822, **CCF A** 类.

2006 1. Yang Wang, Bo Meng, **Qiao Xiang**, Comparison on Survival Analysis of Traumatic Brain Injury Patients Treated at Normal Temperature and Mild Hypothermia, *Chinese General Practice*, December 2006.

会议论文

2020 29. **Qiao Xiang**, Jensen Zhang, Franck Le, Yang Richard Yang, Toward Programmable Inter-domain Routing , to appear in *2020 ACM/IRTF Applied Networking Research Workshop 2020 (**ANRW'20**)*.

28. Danny Alex Lachos Perez, Christian Esteve Rothenberg, **Qiao Xiang**, Yang Richard Yang, Börje Ohlman, Sabine Randriamasy, Luis M. Contreras, Kai Gao, Multi-Domain Information Exposure using ALTO: The Good, the Bad and the Solution, to appear in *2020 ACM/IRTF Applied Networking Research Workshop 2020 (**ANRW'20**)*.

27. Danny Alex Lachos Perez, **Qiao Xiang**, Christian Esteve Rothenberg, Sabine Randriamasy, Luis M. Contreras, Börje Ohlman, Towards Deep Network & Application Integration: Possibilities, Challenges, and Research Directions, to appear in *ACM SIGCOMM 2020 Workshop on Network Application Integration/CoDesign (**NAI'20**)*.

26. **Qiao Xiang**, Jensen Zhang, Kai Gao, Yeon-sup Lim, Franck Le, Geng Li, Yang Richard Yang, Toward Optimal Software-Defined Interdomain Routing, to appear in *the 39th Annual IEEE International Conference on Computer Communications (**INFOCOM'20**)*. Acceptance rate: $19.8\% = 268/1354$, **CCF A** 类.

2019 25. Tony Wang, **Qiao Xiang**, Jeremy Tucker, Vinod Mishra, Yang Richard Yang, Dandelion: A Novel, High-Level Programming System for Software Defined Coalitions with Local State Sharing, to appear in *the 38th AFCEA/IEEE Military Communications Conference (**MILCOM'19**)*, 高分论文 (review scores: 5, 5, 5, 3).

24. Xi Chen, **Qiao Xiang** (共同一作), Linghe Kong, Xue Liu, RadioLoc: Learning Vehicle Locations with FM Signal in All-Terrain Environments, in *2019 IEEE International Conference on Mobile Ad-Hoc and Smart Systems (**MASS'19**)*, **Best Paper Award (最佳论文奖)**, 1 out of 116 submissions, **CCF C** 类.
23. Danny Alex Lachos Perez, Christian Esteve Rothenberg, **Qiao Xiang**, Yang Richard Yang, Börje Ohlman, Sabine Randriamasy, Farni Boten, Luis M. Contreras, Supporting Multi-Domain Use cases with ALTO, in *2019 ACM/IRTF Applied Networking Research Workshop (**ANRW'19**)*.
22. **Qiao Xiang**, Linghe Kong, Xi Chen, Zhe Wang, Lei Rao, Xue Liu, GreenBroker: Optimal Electric Vehicle Park-and-Charge Control via Vehicle-to-Infrastructure Communication, 大会特邀论文 **Invited Paper**, in *2019 IEEE International Black Sea Conference on Communications and Networking* (**BlackSeaCom'19**),
21. **Qiao Xiang**, Haitao Yu, James Aspnes, Franck Le, Linghe Kong, Yang Richard Yang, Optimizing in the Dark: Learning an Optimal Solution Through a Simple Request Interface, in *2019 AAAI Conference on Artificial Intelligence (**AAAI'19**)*, Acceptance rate: 4.7% (oral) /16.2%, **CCF A** 类.

2018 20. **Qiao Xiang**, Jingxuan Zhang, Xin Wang, Yang Liu, Chin Guok, Franck Le, John MacAuley, Harvey Newman, Yang Richard Yang, Fine-Grained, Multi-Domain Network Resource Abstraction as a Fundamental Primitive to Enable High-Performance, Collaborative Data Sciences, in *2018 ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis (**Supercomputing'18**)*, Acceptance rate: 20%, **CCF A** 类.
19. **Qiao Xiang**, Franck Le, Yeon-sup Lim, Vinod K. Mishra, Christopher Williams, Yang Richard Yang, Hongwei Zhang, OpenSDC: A Novel, Generic Datapath for Software Defined Coalitions, in *the 37th AFCEA/IEEE Military Communications Conference (**MILCOM'18**)*, 最高分论文 (review scores: 5, 5, 5).
18. **Qiao Xiang**, Chin Guok, Franck Le, John MacAuley, Harvey Newman, Yang Richard Yang, SFP: Toward Interdomain Routing for SDN Networks, to appear in *Proceedings of the 2018 ACM SIGCOMM Conference(**SIGCOMM'18**)*, 短文, **CCF A** 类.
17. **Qiao Xiang**, Jingxuan Zhang, Xin Wang, Yang Liu, Chin Guok, Franck Le, John MacAuley, Harvey Newman, Yang Richard Yang, Fine-Grained, Multi-Domain Network Resource Abstraction as a Fundamental Primitive to Enable High-Performance, Collaborative Data Sciences, to appear in *Proceedings of the 2018 ACM SIGCOMM Conference(**SIGCOMM'18**)*, 短文, **CCF A** 类.

2017 16. **Qiao Xiang**, Xin Wang, Jingxuan Zhang, Harvey Newman, Yang Liu, Yang Richard Yang,

Unicorn: Unified Resource Orchestration for Multi-Domain, Geo-Distributed Data Analytics, in *2017 INDIS Workshop*.

15. **Qiao Xiang**, Jingxuan Zhang, Kai Gao, Shenshen Chen, Harvey Newman, Justas Balcas, Yang Richard Yang, ExaO: Multi-Resource Orchestration for Multi-Domain Geo-Distributed Data Analytics (position paper), in *ITA Workshop on Distributed Analytics InfraStructure and Algorithms for Multi-Organization Federations (**DAIS'17**)*.
14. Kai Gao, **Qiao Xiang**, Xin Wang, Yang Richard Yang, Jun Bi, NOVA: Towards On-Demand Equivalent Network View Abstraction for Network Optimization, *the 25th IEEE/ACM International Symposium on Quality of Service (**IWQoS'17**)*, Acceptance rate: $19.9\%$, **CCF B** 类.

2016 13. Fanxin Kong, **Qiao Xiang**, Qinglong Wang, Xue Liu, On-line Event-Driven Scheduling for Electric Vehicle Charging via Park-and-Charge, *the 37th IEEE Real-Time Systems Symposium (**RTSS'16**)*, Acceptance rate: $23\%$, **CCF A** 类.

12. Kai Gao, Chen Gu, **Qiao Xiang**, Xin Wang, Yang Richard Yang, Jun Bi, RSAP: An On-Demand, Minimal Equivalent Routing State Abstraction Protocol, *the 24th IEEE International Conference on Network Protocols (**ICNP'16**)*, 短文, top $30\%$ of all submitted full papers, **CCF B** 类.
11. Kai Gao, Chen Gu, **Qiao Xiang**, Yang Richard Yang, Jun Bi, FAST: Enabling Simplified Programming Abstraction for Complex State-Dependent SDN Programming, *ACM SIGCOMM 2016*, 短文, **CCF A** 类.
10. Xi Chen, Lei Rao, **Qiao Xiang**, Xue Liu, Fan Bai, DRIVING: Distributed Scheduling for Video Streaming in Vehicular Wi-Fi Systems, to appear in *the 24th ACM Multimedia Conference (**MM'16**)*, Acceptance rate: $20\% = 52/260$, **CCF A** 类.
9. **Qiao Xiang**, Linghe Kong, Xue Liu, Jingdong Xu, Wei Wang, Auc2Reserve: A Differentially Private Auction for Electric Vehicle Fast Charging Reservation, 大会特邀论文 **Invited Paper**, *the 22th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (**RTCSA'16**)*, **CCF C** 类.
8. Xi Chen, Linghe Kong, Xue Liu, Lei Rao, Fan Bai, **Qiao Xiang**, How Cars Talk Louder, Clearer and Fairer: Optimizing the Communication Performance of Connected Vehicles via Online Synchronous Control, *the 35th Annual IEEE International Conference on Computer Communications (**INFOCOM'16**)*, Acceptance rate: $18.25\% = 300/1644$, **CCF A** 类.

2015 7. **Qiao Xiang**, Fanxin Kong, Xue Liu, Xi Chen, Linghe Kong, Lei Rao, Auc2Charge: An Online Auction Framework for Electric Vehicle Park-and-Charge, *the sixth International Conference on Future Energy Systems (**ACM eEnergy'15**)*, Acceptance rate: $22.8\% = 16/70$.

6. **Qiao Xiang**, Hongwei Zhang, Jianping Wang, Guoliang Xing, Shan Lin, Xue Liu, On Optimal Diversity in Network-Coding-Based Routing in Wireless Networks, *the 34th Annual IEEE International Conference on Computer Communications (**INFOCOM'15**)*, Acceptance rate: $19\% = 316/1640$, **CCF A** 类.
5. **Qiao Xiang**, Xi Chen, Linghe Kong, Lei Rao, Xue Liu, Data Preference Matters: A New Perspective of Safety Data Dissemination in Vehicular Ad Hoc Networks, *the 34th Annual IEEE International Conference on Computer Communications (**INFOCOM'15**)* Acceptance rate: $19\% = 316/1640$, **CCF A** 类.

2012 4. **Qiao Xiang**, Hongwei Zhang, QoS-Aware In-Network Processing for Mission-Critical Wireless Cyber-Physical Systems, *Doctoral Colloquium on the 10th ACM Conference on Embedded Networked Sensor Systems (DC SenSys'12)*.

3. Xiaohui Liu, Hongwei Zhang, **Qiao Xiang**, Xin Che, Xi Ju, Taming Uncertainties in Real-Time Routing for Wireless Networked Sensing and Control, *the 13th ACM International Symposium on Mobile Ad Hoc Networking and Computing (**MobiHoc'12**)*, Acceptance rate: $20\% = 24/120$, **CCF B** 类.

2011 2. Xiaohui Liu, Hongwei Zhang, **Qiao Xiang**, Towards Predictable Real-Time Routing for Wireless Networked Sensing and Control, *the Cyber-Physical-Systems (CPS) Week Workshop on Real-Time Wireless for Industrial Applications (RealWin'11)*.

2009 1. **Qiao Xiang**, Jinhong Xu, Xiaohui Liu, Hongwei Zhang, Loren J. Rittle, When In-Network Processing Meets Time: Complexity and Effects of Joint Optimization in Wireless Sensor Networks, *the 30th IEEE Real-Time Systems Symposium (**RTSS'09**)*, Acceptance Rate: $< 20\%$, **CCF A** 类.

学位论文及技术报告

2020 4. Shenshen Chen, Geng Li, Kerim Gokarslan, Bin Li, **Qiao Xiang**, Haitao Yu, Franck Le, Yang Richard Yang, Ying Zhang, Carbide: Highly Reliable Networks Through Real-Time Multiple Control Plane Composition, 耶鲁大学计算机系技术报告*, YALEU/DCS/TR1552*, Yale University.

2014 3. In-Network Processing for Mission-Critical Wireless Networked Sensing and Control: A Real-Time, Efficiency, and Resiliency Perspective, 博士论文, Wayne State University.

2011 2. When In-Network Processing Meets Time: Complexity and Effects of Joint Optimization in Wireless Sensor Networks, 硕士论文, Wayne State University.

2009 1. **Qiao Xiang**, QoS-Assured In-Network Processing in Wireless Cyber-Physical Systems: A Survey, 技术报告*, Dependable Networking and Computing Group*, Wayne State University.

# Toward Fine-Grained, Privacy-Preserving, Efficient Multi-Domain Network Resource Discovery

Qiao Xiang, Jingxuan Jensen Zhang, Xin Tony Wang, Yang Jace Liu, Chin Guok, Franck Le, John MacAuley, Harvey Newman, and Y. Richard Yang

***Abstract*— Multi-domain network resource reservation systems are being deployed, driven by the demand and substantial benefits of providing predictable network resources. However, a major lack of existing systems is their coarse granularity, due to the participating networks' concern of revealing sensitive information, which can result in substantial inefficiencies. This paper presents Mercator, a novel multi-domain network resource discovery system to provide fine-grained, global network resource information, for collaborative sciences. The foundation of Mercator is a resource abstraction through algebraic-expression enumeration (*i.e.*, linear inequalities/equations), as a compact representation of multiple properties of network resources (*e.g.*, bandwidth, delay, and loss rate) in multi-domain networks. In addition, we develop an obfuscating protocol, to address the privacy concerns by ensuring that no participant can associate the algebraic expressions with the corresponding member networks. We also introduce a super-set projection technique to increase Mercator's scalability. We implement a prototype Mercator and deploy it in a small federation network. We also evaluate the performance of Mercator through extensive experiments using real topologies and traces. Results show that Mercator 1) efficiently discovers available networking resources in collaborative networks on average four orders of magnitude faster, and allows fairer allocations of network resources; 2) preserves the member networks' privacy with little overhead; and 3) scales to a collaborative network of 200 member networks.**

Manuscript received December 15, 2018; revised June 27, 2019; accepted June 28, 2019. Date of publication July 5, 2019; date of current version August 6, 2019. This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. This work was also supported in part by NSF under Award 1440745, Award 1246133, Award 1341024, Award 1120138, and Award 1659403, in part by Department of Energy (DOE) under Award DE-AC02-07CH11359, in part by DOE/Advanced Scientific Computing Research (ASCR) Project under Grant 000219898, in part by DOE/ASCR under Award DE-SC00155527 and Award DE-SC0015528, and in part by the Google Research Award. *(Corresponding authors: Qiao Xiang; Y. Richard Yang.)*

Q. Xiang, J. J. Zhang, X. T. Wang, and Y. R. Yang are with the Department of Computer Science, Yale University, New Haven, CT 06511 USA (e-mail: qiao.xiang@cs.yale.edu; jingxuan.zhang@yale.edu; xin.wang@yale.edu; yry@cs.yale.edu).

Y. L. Liu is with the Computer Science Department, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: yang.liu5@ucalgary.ca).

C. Guok and J. MacAuley are with the Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mail: chin@es.net; macauley@es.net).

F. Le is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: fle@us.ibm.com).

H. Newman is with the Division of Physics, Mathematics and Astronomy, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: newman@hep.caltech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2019.2927073

***Index Terms*— Multi-domain networks, resource discovery, privacy preserving.**

## I. Introduction

MANY of today's premier science experiments, such as the Large Hadron Collider (LHC) [2], the Square Kilometre Array (SKA) [3], and the Linac Coherent Light Source (LCLS) [4], rely on finely-tuned workflows that coordinate geographically distributed resources (*e.g.*, instrument, compute, storage) to enable scientific discoveries. An example of this is the movement of LHC data from Tier 0 (*i.e.*, the data center at European Organization for Nuclear Research, known as CERN) to Tier 1 (*i.e.*, national laboratories) storage sites around the world. This requires deadline scheduling to keep up with the amount of information that is continually generated by instruments when they are online. Another example is the "superfacility" model being developed by LCLS to allow streaming of data from instruments, across the Wide-Area Network (WAN), directly into supercomputers' burst buffers for near real-time analysis. The key to supporting these distributed resource workflows is the ability to reserve and guarantee network resources (*e.g.*, bandwidth) across multiple network domains to facilitate predictable end-to-end network connectivity. As such, several Research and Education (R&E) networks have deployed inter-domain circuit reservation systems. For example, the Energy Sciences Network (ESnet), a network supporting the LHC experiments, has deployed an On-Demand Secure Circuits and Advance Reservation System called OSCARS [5].

However, due to networks' concern of revealing sensitive information, existing systems do not provide a network interface for users to access network resource information (*e.g.*, network capabilities). Instead, they only allow users to submit requests for reserving a specific amount of resources (*e.g.*, a circuit providing a certain amount of bandwidth and delay), and return either success or failure [5]–[12]. This approach, which we call "probe requests" in the rest of this paper, often results in poor performance and fairness. Specifically, while solutions for reserving resources within a single member network, can be very efficient, solutions for discovering and reserving resources for correlated and concurrent flows across multiple member networks face unique challenges. In particular, solutions to reserving resources within a single administrative domain (*e.g.*, NetStitcher [13], SWAN [14] and B4 [15]) are often provided with the network's topology, and links' availability. In contrast, in a network with multiple administrative domains, because this information is typically considered sensitive, member networks do not reveal internal

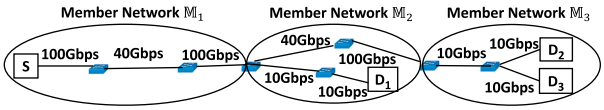



Fig. 1. A motivating example where a user wants to reserve bandwidth for three source-destination pairs: $(S, D_1)$, $(S, D_2)$ and $(S, D_3)$, across 3 member networks $\mathbb{M}_1$, $\mathbb{M}_2$ and $\mathbb{M}_3$.

$\mathbb{M}_1$:
$x_1^b + x_2^b + x_3^b \leq 100,$
$x_1^b + x_2^b + x_3^b \leq 40,$
$x_1^b + x_2^b + x_3^b \leq 100,$

$\mathbb{M}_2$:
$x_2^b + x_3^b \leq 40, \quad x_1^b \leq 10,$
$x_2^b + x_3^b \leq 100, \quad x_1^b \leq 10,$

$\mathbb{M}_3$:
$x_2^b + x_3^b \leq 10,$
$x_2^b \leq 10,$
$x_3^b \leq 10,$

Fig. 2. Illustration of resource abstraction for the reservation request from Fig. 1.

network details to external parties. As a result, existing multi-domain reservation systems treat each member network as a black box, probe their available resources by submitting varied circuit reservation requests, and receive boolean responses. In other words, current solutions perform a depth-first search on all member networks, and rely on a trial and error approach: to reserve bandwidth, repeated, and varied attempts may have to be submitted until success.

To illustrate the limitations of existing systems, we consider a collaboration network composed of three member networks running OSCARS [5], as shown in Fig. 1. A user may submit a request to reserve bandwidth for three circuits, from source host $S$ to destination hosts $D_1$, $D_2$ and $D_3$. Given the capabilities of the source host (*e.g.*, the source host may have a 100 Gbps network card), and to ensure fairness across the circuits, the user may request 33.33 Gbps for each circuit. Upon receiving this request, OSCARS processes the circuits sequentially, for example, in the order of $(S, D_1)$, $(S, D_2)$ and $(S, D_3)$. For each circuit, it uses a depth-first search approach to probe if each member network can provide the requested bandwidth. In this example, there is no path with 33.33 Gbps of bandwidth from $S$ to $D_1$, and hence OSCARS notifies the user that this request fails.

The user can then adjust the requested bandwidth. However, with the limited feedback in OSCARS, the user does not know the amount of available bandwidth from $S$ to $D_1$. Consequently, the user may use a cut-to-half-until-reserved search strategy. As a result, after 12 attempts, the networks allocate 8.33 Gbps ($33.33 \rightarrow 16.67 \rightarrow 8.33$) for $(S, D_1)$, 8.33 Gbps ($33.33 \rightarrow 16.67 \rightarrow 8.33$) for $(S, D_2)$ and 1.04 Gbps ($33.33 \rightarrow 16.67 \rightarrow 8.33 \rightarrow 4.17 \rightarrow 2.08 \rightarrow 1.04$) for $(S, D_3)$. In addition to requiring a large number of search attempts, the approach may obtain a bandwidth allocation that is far from optimal. For example, given the links' capacities and availability, a fair optimal bandwidth allocation is actually 5 Gbps for each circuit. Without a network interface to provide network resource information, designing an algorithm using existing systems to identify this solution can lead to substantially more complexity and churns.

Driven by the benefits of providing network resource information to users to improve the performance of network resource reservation systems, industry and academia have spent substantial efforts on designing network resource discovery systems to provide such information (*e.g.*, [16]–[25]). However, designing a network resource discovery system is a non-trivial task that requires addressing a series of challenges. First, the interface should provide a unified, accurate representation of the availability and sharing of multiple properties of network resources (*e.g.*, bandwidth, delay and loss rate) from multiple networks. Second, it should protect the privacy of networks by not exposing their private information (*e.g.*, topology, policy and capacity region). Third, it should not introduce too much computation and communication overhead to networks, and should scale to large multi-domain networks.

Existing resource discovery systems do not fully address these challenges. For example, resource discovery systems in grid-computing [16]–[23] only focus on the discovery of end-point resources (*i.e.*, computation and storage resources) and their availability for different services. Resource discovery systems in cloud computing (*e.g.*, CloudMirror [26], Pretium [27] and Amoeba [28] adopt a network-does-all approach, in which users are provided with a more expressive interface for specifying requirements on data transfers and the network orchestrates resources between different user requests. Though this approach protects the privacy of the network, the network can only provide elastic resource reservations for user requests (*i.e.*, some requests may be preempted or rejected). Some recent systems (*e.g.*, the ALTO protocol [24], [25], [29] and the SENSE project [30], [31]) provide users the information of certain properties of network resources using the one-big-switch abstraction. While this approach protects the privacy of network, it cannot provide the accurate information of network resource sharing between flows (*e.g.*, bandwidth), which is critical for optimizing the emerging use cases (*e.g.*, large-scale collaborative data sciences).

In this paper, we present Mercator, a novel multi-domain network resource discovery system designed to address the limitations of current reservation systems and optimize multi-domain workflows. Mercator copes with the three aforementioned challenges for providing network resource information through three main components. The first and core component of Mercator is a resource abstraction through algebraic-expression enumeration (*i.e.*, linear inequalities and equations), which provides a compact, unifying representation of multiple properties (*e.g.*, bandwidth, delay and loss rate) of multi-domain network resources. For example, considering the same example of Fig. 1, the resource abstraction captures the constraints of bandwidths from all networks using the set of linear inequalities depicted in Fig. 2. Specifically, the variables $x_1^b$, $x_2^b$, $x_3^b$ represent the available bandwidth that can be reserved for $(S, D_1)$, $(S, D_2)$ and $(S, D_3)$, respectively. Each linear inequality represents a constraint on the reservable bandwidths over different shared resources by the three circuits. For example, the inequality $x_1^b + x_2^b + x_3^b \leq 100$ indicates that all three circuits share a common resource and that the sum of their bandwidths can not exceed 100 Gbps. With this set of linear inequalities, the user does not need to repeatedly probe the domains, but can immediately derive the bandwidth allocation to satisfy its own objective (*e.g.*, same rate for each transfer, different ratios according to demand ratios, or a fairness allocation such as max-min fairness).

Second, Mercator introduces a resource abstraction obfuscating protocol to ensure that member networks and other external parties cannot associate an algebraic expression with a corresponding member network, leading to a complete unified aggregation of multiple domains, appearing as much as possible as a single (virtual) network. Although such complete integration may not be needed in all settings, it can be highly beneficial in settings with higher privacy or security concerns. For example, in the scenario of Fig. 1, this protocol ensures that (1) the user cannot infer that the constraint $x_2^b + x_3^b \leq 10$ comes from network $\mathbb{M}_3$, and (2) that neither network $\mathbb{M}_1$ nor $\mathbb{M}_2$ knows the existence of this constraint. Finally, Mercator also introduces a super-set projection technique, which substantially improves the scalability and performance of Mercator through pre-computation and projection.

The main contributions of this paper are as follows:

- We identify the fundamental reason of the poor performance of current reservation systems for multi-domain data transfers as the lack of visibility of network information (*e.g.*, topology and link availability) of each member network, and design Mercator, a novel multi-domain network resource discovery system, to address this issue;
- In Mercator, we propose a novel, compact resource abstraction to represent the network resource availability and sharing (*e.g.*, bandwidth, delay and loss rate) among virtual circuit requests through algebraic-expression enumeration;
- We design a resource abstraction obfuscating protocol to prevent the user from associating the received algebraic expressions with their corresponding member networks;
- We develop a super-set projection technique to substantially improve the scalability of Mercator;
- We fully implement Mercator, deploy it in a small federation network, and also conduct extensive experiments using real network topologies and traces. Results show that Mercator (1) efficiently discovers available networking resources in collaborative networks on average six orders of magnitude faster, and allows fairer allocations of network resources; (2) preserves the member networks' privacy with little overhead; and (3) scales to a collaborative network of 200 member networks.

The remaining of this paper is organized as follows. We give an overview of Mercator in Section II. We give the details of the algebraic-expression-based resource abstraction in Section III. We discuss the resource abstraction obfuscating protocol and the super-set projection technique in Section IV and Section V, respectively. We introduce the implementation and deployment of Mercator in Section VI. We present the evaluation results of Mercator in Section VII. We discuss the related work in Section VIII and conclude the paper in Section IX.

## II. Mercator Overview

This section presents the basic workflow and the architecture of Mercator, and a brief overview of its three main components: the resource abstraction through algebraic-expression enumeration, the resource abstraction obfuscating protocol and the super-set projection technique.

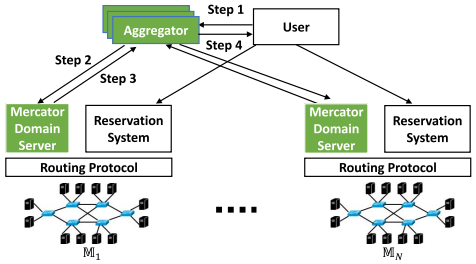



Fig. 3. The architecture and basic workflow of Mercator.

### A. Basic Workflow

Mercator introduces and relies on a logically centralized aggregator, and a Mercator domain server in each member network. Consider a multi-domain network of $N$ member networks $\mathbb{M}_i$, where $i = 1, \ldots, N$ (Fig. 3). The basic workflow of Mercator to discover the multi-domain network bandwidth availability and sharing for a set of requested circuits is:

- *Step 1:* A user (*e.g.*, an application) submits a resource discovery request for a set of circuits to the aggregator by specifying the source and destination endpoints of each circuit, and what properties of network resources he/she wishes to discover (*e.g.*, bandwidth, delay and loss rate).
- *Step 2:* After authenticating and verifying the authorization of the request, the aggregator determines the member networks that the circuits traverse, and queries the Mercator domain server in each of these member networks to discover their resource abstractions. The determination of the relevant member networks for the aggregator to contact is further described in Section II-B.
- *Step 3:* Upon receiving the query from the aggregator, each Mercator domain server computes the resource abstraction (Section II-C, Section III) of the corresponding member network, and executes an obfuscating protocol (Section II-C, Section IV) to send the obfuscated resource abstraction to the aggregator.
- *Step 4:* The aggregator collects the obfuscated resource abstractions from the relevant member networks, and derives the original resource abstractions to present to the user. Based on the received information, the user determines the bandwidth allocation for each circuit, and sends a reservation request to the underlying reservation system.

The above workflow illustrates the main steps for a user to discover the available network bandwidth and properties for a set of circuits traversing multiple member networks. To further improve the scalability of Mercator, Section V introduces the super-set projection technique. It allows the aggregator to proactively discover the resource abstractions for a set of circuits between every pair of source and destination member networks, and project the pre-computed result to get the resource abstraction when receiving actual requests from users. The super-set projection technique can significantly reduce the delay, as well as number of messages, of resource discovery, and allows the aggregator to process multiple requests concurrently.

### B. Architecture

This section describes the roles of the aggregator and Mercator domain servers in further details (Fig. 3).

*1) Aggregator:* The aggregator is the main interface of Mercator. It is responsible for authenticating and verifying the authorization of users' resource discovery requests (*e.g.*, through PKI [32]), querying Mercator domain servers in member networks to discover network resource information, and returning the collected abstractions to users. Depending on the specific requirements of different multi-domain networks, Mercator may adopt different authentication/authorization systems, *e.g.*, OpenID [33] and SAML [34]. We leave the detailed investigation of this issue in Mercator as future work.

The aggregator has connections to Mercator domain servers in all member networks. It also acts as a Border Gateway Protocol (BGP) [35] speaker, and has BGP sessions to all member networks. Consequently, given a request for a set of circuits $F$, the aggregator can infer the member-network path for each circuit, *i.e.*, the list of member networks a circuit will traverse, and the ingress points of the circuits to each member network[1] (as described in Step 1 of workflow). As such, for this request, the aggregator can also infer the set of circuits traversing and consuming resources in each $\mathbb{M}_i$, denoted as $F_i$. It can then queries the Mercator domain servers at each $\mathbb{M}_i$ by providing $F_i$ and their ingress points to enter $\mathbb{M}_i$.

*2) Mercator Domain Server:* Given a Mercator domain server in member network $\mathbb{M}_i$, its primary role is to compute the resource abstraction of $\mathbb{M}_i$. To achieve it, Mercator follows the layering design principle to separate the routing protocol and the available network resources. In this way, given a set of circuits sent by the aggregator, their routes in $\mathbb{M}_i$ are computed and provided by the routing protocol in $\mathbb{M}_i$. The Mercator domain server in $\mathbb{M}_i$ takes these routes as inputs, and derives the available bandwidth and shared properties for the requested flows along those routes. After computing the abstraction, the Mercator domain server executes an obfuscating protocol to send the obfuscated resource abstraction to the aggregator, which addresses member networks' privacy concern.

### C. Key Design Points

Having illustrated the high-level workflow of Mercator, we next give a brief overview on its key design points.

*1) Resource Abstraction Through Algebraic-Expression Enumeration (Section III):* Mercator follows two important principles in human-computer interaction, *familiarity* and *uniformity*, to design a unifying abstraction that captures the properties (*e.g.*, available bandwidth, delay and loss rate) of resources shared – within and between member networks – by a set of requested circuits. This novel, compact resource abstraction is the core component of Mercator, and relies on algebraic expressions (*i.e.*, linear inequalities/equations), a concept familiar to scientists and network engineers [37], to express the available bandwidth sharing for a set of requested circuits to be reserved.

Existing resource abstractions, including graph-based abstractions [38], [39] and the one-big-switch abstractions [24], [25], either fail to protect the private, sensitive information of each member network, or fail to capture the accurate resource availability and sharing between virtual circuit requests. In contrast, the resource abstraction of Mercator, expressed through algebraic-expression enumeration, naturally and accurately captures different properties (*e.g.*, bandwidth, delay and loss rate) of shared resources of a set of circuits without requiring member networks to reveal their network topology. Compared with the Boolean response of current resource reservation systems such as OSCARS, the user receives the complete resource feasible region of the collaboration networks for the requested circuits represented through algebraic expressions. A point in that feasible region represents a feasible allocation of resources for the different circuits in the request. In other words, the user can choose any point in the returned region as the parameters for the circuits to be reserved, using his own resource allocation strategy (*e.g.*, max-min fairness), and get predictable performance guarantee (*e.g.*, bandwidth, delay and loss rate).

*2) Resource Abstraction Obfuscating Protocol (Section IV):* The algebraic-expression-based abstraction provides a compact, unifying representation of the multi-domain network resource information. It does not require member networks to reveal their network topologies and link availabilities. However, it does expose the resource feasible region of each member network (illustrated by the examples in Section I and Section III). Some member networks might prefer not to expose such information, as malicious parties may use it to identify links where to launch attacks (*e.g.*, DDoS). To address this issue, we develop a resource abstraction obfuscating protocol, which prevents the resource discovery aggregator from identifying the source of each received resource constraint. Specifically, the key idea consists of having each Mercator domain server obfuscate its own set of linear inequalities as a set of linear equations through a private random matrix of its own and a couple of random matrices shared with few other Mercator domain servers from other member networks (*e.g.*, through a consensus protocol), and then sends the obfuscated set of linear equations back to the aggregator using symmetric-key encryption, *e.g.*, Advanced Encryption Standard (AES) [40]. We demonstrate that from the received obfuscated equations, the aggregator can retrieve the actual resource feasible region for the circuits across member networks, but cannot associate any linear inequality with its corresponding member network. As a result, even if a malicious party obtains the resource feasible region across member networks, launching attacks to all member networks is much harder than attacking a particular member network.

*3) Super-Set Projection (Section V):* To improve the scalability of Mercator, we introduce the super-set projection technique. The main idea consists of having the aggregator periodically query Mercator domain servers to discover the

[1]In BGP glossary, such a path is also called an autonomous-system-path, or an AS-path, which is announced in BGP update messages along BGP sessions. The Route View Project [36] relies on a similar architecture with BGP speakers establishing sessions with hundreds of peering networks to collect BGP updates, and provides a real time monitoring infrastructure. In particular, we observe that the AS path for each destination prefix is currently already collected and made publicly available. As such, Mercator does not introduce additional privacy issues.

resource abstraction for a set of circuits between every pair of source and destination member networks. With these precomputed abstractions, when a user submits a resource discovery request, the aggregator does not need to query the Mercator domain servers to compute the abstraction for each received request. Instead, the aggregator performs a projection on the precomputed abstractions based on the source and destination member networks of each circuit in the actual user request, to get the abstraction for this request. For example, consider a network of 2 member networks $\mathbb{M}_1$ and $\mathbb{M}_2$. Using super-set projection, the aggregator queries the Mercator domain servers at both member networks about the bandwidth properties for a set of 2 circuits, one from $\mathbb{M}_1$ to $\mathbb{M}_2$ and the other from $\mathbb{M}_2$ to $\mathbb{M}_1$, and gets a set of linear inequalities $\{x^b_{12} + x^b_{21} \leq 100,\ x^b_{12} \leq 50\}$. Suppose later a user submits a request for 1 circuit, with the source being an endpoint in $\mathbb{M}_2$ and the destination being an endpoint in $\mathbb{M}_1$, to the aggregator. The aggregator projects the precomputed set of linear inequalities by removing all variables that are not $x^b_{21}$, and returns the result $\{x^b_{21} \leq 100\}$ to the user.

Such projection is much more efficient than having Mercator domain servers compute the abstraction for each received circuit request. With this technique, when a user submits a resource discovery request to the aggregator, the aggregator does not need to query Mercator domain servers (Step 2 in Section II-A), and the Mercator domain servers do not need to compute and obfuscate the resource abstraction for the request (Step 3 in Section II-A). Only when the user fails to reserve the resource based on the projected abstraction will the aggregator query the Mercator domain servers to obtain an up-to-date abstraction for the user. As such, servers in the aggregator pool can process requests concurrently (*e.g.*, using optimistic concurrency control), significantly improving the scalability, fault-tolerance, and performance of Mercator.

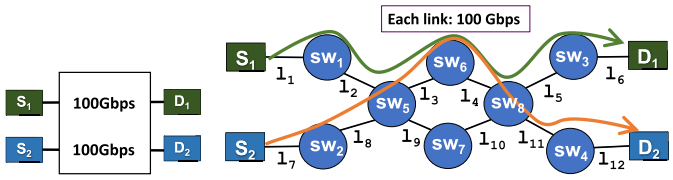
After an overview of the key design points in Mercator, we discuss these designs in detail in the next few sections.

## III. Resource Abstraction Through Algebraic-Expression Enumeration

In this section, we give the details of the resource abstraction through algebraic-expression enumeration, the core component of Mercator. We first discuss the limitations of existing design options. Next we give the specifications of this abstraction, and how it handles important use cases, *e.g.*, multicast, multi-path routing and load balancing, using the bandwidth property as an example. Then we discuss how the resource abstraction can represent other important properties of network resources (*e.g.*, delay and loss rate), and how the resource abstractions from different member networks are aggregated to provide a unified representation of network resources.

### A. Basic Issue

As illustrated by the example in Section I, the fundamental reason for the poor performance of existing circuit reservation systems is they are lack of the visibility of properties, *e.g.*, bandwidth, of shared network resources for a set of circuits to be reserved. One may think of a strawman to let each member network provide the full topology information to the aggregator in a graph-based abstraction [38], [39]. This design, however, exposes all the sensitive, private information of each member network, *i.e.*, network topology and links' availability, to external parties, leading to security breaches.




**(a)** The one-big-switch shows that each circuit can get a 100 Gbps bandwidth.

**(b)** The physical topology shows that the route of two circuits share bottleneck links, *i.e.*, $l_3$ and $l_4$, hence they can only collectively get a 100 Gbps bandwidth.

Fig. 4. A running example for illustrating the inefficiency of one-big-switch abstraction and the basic idea of resource abstraction through algebraic-expression enumeration, where two circuits $(S_1, D_1)$ and $(S_2, D_2)$ need to be reserved.

A second strawman is to use a one-big-switch abstraction to provide simplified views of network information [24], [25], which protects the privacy of each member network. However, this abstraction fails to capture the information of shared resource among virtual circuit requests and thus is inaccurate. Consider the example in Fig. 4, where the user wants to reserve two circuits from $S_1$ to $D_1$ and $S_2$ to $D_2$, respectively. Using the one-big-switch abstraction in the P4P system [25], the user will get the information that each circuit can reserve a bandwidth up to 100 Gbps (Fig. 4a). However, the routes for the two circuits – computed by the underlying routing protocol – share common links $l_3$ and $l_4$ (Fig. 4b), making it infeasible for both circuits to each reserve a 100 Gbps bandwidth.

In some recent studies [41], [42], a variation of the one-big-switch abstraction was proposed to define the resource sharing among different traffic flows as operations defined in different algebra fields. However, this abstraction is too complex and can only handle single-path routing policies.

### B. Basic Idea

Different from the graph-based abstraction and the one-big-switch abstraction, the basic idea of the resource abstraction in Mercator is simple yet powerful: *given a set of requested circuits to be reserved, capture the properties (e.g., available bandwidth) of relevant shared resources, through a set of algebraic expressions.*

Specifically, suppose the Mercator domain server at a member network receives the resource discovery request of a set of circuits $F$ entering this member network. For each circuit $f_j \in F$, we use $x^b_j$ to denote the available bandwidth the user can reserve for this circuit. Upon receiving this request, the Mercator domain server first checks the intradomain route of each circuit $f_j$. Then the server enumerates all the links in the member network. For each link $l_u$, it generates a linear inequality:

$$\sum x^b_j \leq l_u.bandwidth, \quad \forall f_j \text{ that uses link } l_u \text{ in its route.}$$

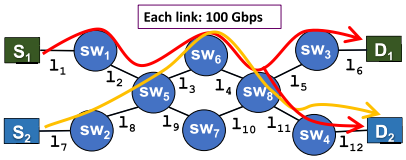



Fig. 5. A running example illustrating how the resource abstraction handles multicast through algebraic-expression enumeration, where two circuits $(S_1, \{D_1, D_2\})$ and $(S_2, D_2)$ need to be reserved.

Revisit the example in Fig. 4, the Mercator domain server will generate the following set of linear inequalities $\Pi(F)$:

$$\begin{aligned} x_1^b &\le 100 \quad \forall l_u \in \{l_1, l_2, l_5, l_6\}, \\ x_2^b &\le 100 \quad \forall l_u \in \{l_7, l_8, l_{11}, l_{12}\}, \\ x_1^b + x_2^b &\le 100 \quad \forall l_u \in \{l_3, l_4\}, \end{aligned} \tag{1}$$

which accurately captures the bandwidth sharing among two circuits' routes.

### C. Removing Redundant Linear Inequalities

Observe the set of linear inequalities in the above example. One may realize that this set has redundancies, *e.g.*, there are 4 same inequalities $x_1^b \le 100$ in this set. Given $\Pi(F)$, a linear inequality $y \in \Pi(F)$ is redundant if and only if the optimal solution of any optimization problem with $\Pi(F)$ as the constraint is the same as that with $\Pi(F) - \{c\}$ as the constraint. In our system, the Mercator domain server adopts a classic compression algorithm [43] to remove the redundant linear inequalities. In this example, the compressed $\Pi(F)$ will only contain one inequality, *i.e.*, $x_1^b + x_2^b \le 100$.

Through algebraic-expression enumeration, the resource abstraction can handle not only unicast, as shown above, but many other settings. Below we show how resource abstraction handles three important use cases in collaborative data sciences.

### D. Use Case 1 - Multicast

Consider the example in Fig. 5, where the first circuit is a multicast circuit from $S_1$ to $D_1$ and $D_2$, and the second one is a unicast circuit from $S_2$ to $D_2$. The routes for these circuits, computed by the underlying routing protocol, are marked in red and yellow, respectively. The resource abstraction captures the bandwidth sharing between these two circuits by introducing auxiliary variables $x_{11}^b$ and $x_{12}^b$ for the multicast circuit. Because the traffic duplication for the first circuit happens at switch 8, we use $x_{11}^b$ to represent the traffic from switch 8 to $D_1$, and $x_{12}^b$ to represent the traffic from switch 8 to $D_2$. In this way, the Mercator domain server will generate the following set of linear inequalities:

$$\begin{aligned} x_{11}^b &= x_1^b, \quad x_{12}^b = x_1^b, \\ x_1^b &\le 100 \quad \forall l_u \in \{l_1, l_2\}, \\ x_{11}^b &\le 100 \quad \forall l_u \in \{l_5, l_6\}, \\ x_2^b &\le 100 \quad \forall l_u \in \{l_7, l_8\}, \\ x_1^b + x_2^b &\le 100 \quad \forall l_u \in \{l_3, l_4\}, \\ x_{12}^b + x_2^b &\le 100 \quad \forall l_u \in \{l_{11}, l_{12}\}, \end{aligned} \tag{2}$$

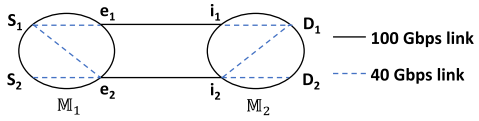



Fig. 6. A running example illustrating how resource abstraction handles complex routing and traffic engineering policies through algebraic-expression enumeration and how resource abstractions from different member networks are stitched, where two circuits $(S_1, D_1)$ and $(S_2, D_2)$ need to be reserved.

### E. Use Case 2 - Multi-Path Routing

Consider the example in Fig. 6, where the user wants to discover the bandwidth sharing for two circuits $f_1 : (S_1, D_1)$ and $f_2 : (S_2, D_2)$, and $\mathbb{M}_1$ uses multi-path routing for the circuit $f_1$, *i.e.*, routing to two egresses $e_1, e_2$.

In particular, the Mercator domain server at $\mathbb{M}_1$ introduces variables $x_{11}$ and $x_{12}$ to represent the available bandwidth from $S$ to egresses $e_1$ and $e_2$, respectively, and share the introduction of these variables to $\mathbb{M}_2$. Then $\mathbb{M}_1$ independently adds an equation $x_1 = x_{11} + x_{12}$ into its set of linear inequalities $\Pi_1(F)$. The resulting resource abstraction at both member networks are then expressed as

$$\begin{aligned} \Pi_1(F): x_1^b = x_{11}^b + x_{12}^b, \quad & \Pi_2(F): x_{11}^b \le 40, \\ x_{11}^b \le 40, \quad & x_{12}^b \le 40, \\ x_{12}^b \le 40, \quad & x_2^b \le 40. \\ x_2^b \le 40, & \\ x_{11}^b \le 100, & \\ x_{12}^b + x_2^b \le 100. & \end{aligned} \tag{3}$$

Using $\Pi_1(F)$ and $\Pi_2(F)$ as the constraint, the user can then make reservation requests based on the optimization of her own objective function. For example, to achieve the max-min fairness between two circuits, the user will reserve $x_1^b = 80$ Gbps for $(S_1, D_1)$ and $x_2^b = 40$ Gbps for $(S_2, D_2)$, where internally $\mathbb{M}_1$ can allocate $x_{11}^b = x_{12}^b = 40$ Gbps.

### F. Use Case 3 - Load-Balancing

In the same example in Fig. 6, assume $\mathbb{M}_1$ uses weighted-cost-multi-path (WCMP) and has an internal policy to allocate bandwidth for the circuit $(S_1, D_1)$ along two path $S_1 \to e_1$ and $S_1 \to e_2$ in a ratio of 1:2. With this policy, the previous reservation request with $x_1^b = 80$ Gbps and $x_2^b = 40$ Gbps is no longer valid as $x_{11}^b$ and $x_{12}^b$ cannot reach 40 Gbps simultaneously. To capture this policy so that the user does not make the invalid reservation request, the Mercator domain server at $\mathbb{M}_1$ introduces an additional equation $x_{12}^b = 2x_{11}^b$ into $\Pi_1(F)$ and sends to the user. And the user can compute the valid, optimal reservation decisions, *e.g.*, $x_1^b = 60$ Gbps and $x_2^b = 40$ Gbps, to achieve max-min fairness.

### G. Resource Abstraction for Other Properties

The algebraic-expression-based resource abstraction provides a generic representation for different properties of network resources. We now illustrate this generality by showing how it can represent two other important properties of network resources, delay and loss rate.

Specifically, suppose the Mercator domain server at a member network receives the resource discovery request of the delay and loss rate of a set of circuits $F$ entering this network. For each circuit $f_j \in F$, we use $x_j^d$ to denote the delay of this circuit in the network. The Mercator domain server checks the intradomain route of each $f_j$, and generates the following linear expression:

$$x_j^d = \sum_u l_u.delay, \quad \forall l_u \text{ in the route of} f_j.$$

Similarly, for the property of loss rate, we use $x_j^r$ to denote the loss rate of circuit $f_j$, and the Mercator domain server generates the following linear expression:

$$x_j^r = 1 - \prod(1 - l_u.lossrate), \quad \forall l_u \text{ in the route of} f_j.$$

As such, the algebraic-expression-based resource abstraction is a unified representation of different properties of network resources for a set of circuits. Given $\Pi_i(F)$, the resource abstraction of $\mathbb{M}_i$ for a set of $F$ circuits, from the geometric perspective, represents the resource feasible region of $\mathbb{M}_i$ for providing bandwidths, delays and loss rates to this set of circuits.

### H. Aggregation of Multi-Domain Resource Abstraction

Given a set of $F$ circuits spanning over $N$ member networks, the resource abstractions $\Pi_i(F_i)$ from all the Mercator domain servers in the member networks can be aggregated into a unified, aggregated representation of multi-domain network resources $\oplus\Pi_i(F_i)$, where $\oplus$ is a property-specific operator.

Specifically, for the bandwidth property, the $\oplus$ operator is $\cup$, *i.e.*, the union of multiple sets of linear inequalities. Geometrically speaking, $\cup\Pi_i^b(F_i)$ represents the intersection of the bandwidth feasible region of all member networks. For the delay property, the $\oplus$ operator is $\sum_d$, *i.e.*, the sum of delays from different networks. In $\sum_d \Pi_i^d(F_i)$, for each circuit $f_j$ and all member networks $\mathbb{M}_i$, $x_j^d = \sum delay_{ji}$, where $delay_{ji}$ is the delay of circuit $f_j$ in network $\mathbb{M}_i$. For the loss rate property, the $\oplus$ operator is $\sum_r$. In $\sum_r \Pi_i^r(F_i)$, for each circuit $f_j$ and all member networks $i$, $x_j^r = 1 - \prod(1 - lossrate_{ji})$, where $lossrate_{ji}$ is the loss rate of circuit $f_j$ in network $\mathbb{M}_i$.

## IV. Privacy-Preserving Resource Abstraction

Given a member network, the algebraic-expression-based resource abstraction specified in Section III accurately captures different properties of the available network resources among virtual circuits without exposing its network topology and links' availability. However, a resource abstraction still represents the resource feasible region of the corresponding member network for a set of circuits. Such information is still private and sensitive, and a malicious party who acquires it may use it to launch attacks to the corresponding member network. To address the privacy challenge for network resource discovery and preserve the privacy of resource feasible region of member networks while still providing the accurate network resource information for circuits, we extend the base resource abstraction to develop an obfuscating protocol in Mercator.

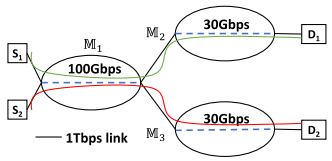



Fig. 7. A running example to illustrate the resource abstraction obfuscating.

In this section, we first formally define the privacy-preserving resource abstraction problem. Next, we present the details of our protocol and conduct a rigorous analysis.

### A. Privacy-Preserving Resource Abstraction Problem

*1) Basic Issue:* We use the example in Fig. 7 to illustrate the privacy concern of the resource abstraction, where Mercator tries to discover the shared bandwidth of two virtual circuits $(S_1, D_1)$ and $(S_2, D_2)$ across 3 member networks. In this example, all links in black line are 1 Tbps aggregating links. The inter-member-network-paths of two circuits are $[\mathbb{M}_1, \mathbb{M}_2]$ and $[\mathbb{M}_1, \mathbb{M}_3]$, respectively. And two circuits share the same intra-domain path in $\mathbb{M}_1$.

When receiving the resource discovery request, the Mercator domain server at each member network will abstract the bandwidth sharing of both circuits into a set of linear inequalities. After removing the redundant inequalities of each member network, the resource abstraction of each member network's bandwidth is:

$$\begin{aligned} &\Pi_1^b(F_1) : \{x_1^b + x_2^b \leq 100\} \\ &\Pi_2^b(F_2) : \{x_1^b \leq 30\} \\ &\Pi_3^b(F_3) : \{x_2^b \leq 30\}. \end{aligned} \tag{4}$$

If each Mercator domain server directly sends its own resource abstraction to the aggregator, the aggregator will have the knowledge of the resource feasible region of each individual member network. This makes the whole collaboration network vulnerable because the aggregator is a single point of failure possessing the private information of all member networks. In other words, if an attacker gains the control to the aggregator, he can leverage such specific information to attack any member network.

*2) Problem Definition:* To make Mercator functional and secure, therefore, we need a solution that provides the accurate network resource information for the set of virtual circuits to be reserved, and at the same time protects each member network from exposing its private resource feasible region. To this end, we first give a formal definition of privacy-preserving, equivalent resource abstraction:

***Definition** 1 (Equivalent, Privacy-Preserving Resource Abstraction): Given a set of circuits $F$ that span over $N > 1$ member networks, the resource abstraction $\Pi_p(F)$ collected by the aggregator is equivalent and privacy-preserving if for all network resource properties (e.g., bandwidth, delay and loss rate), (1) the resource feasible region represented by $\Pi_p(F)$ is the same as that represented by $\oplus\Pi(F_i)$ where $i = 1, 2, \ldots, N$; and (2) for any linear inequality $c \in \Pi_p(F)$, the aggregator cannot associate it with a particular member network.*

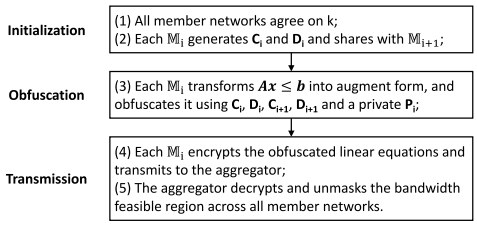



Fig. 8. The resource abstraction obfuscating protocol.

With this definition, we further define the privacy-preserving resource abstraction problem:

***Problem 1*** *(Privacy-Preserving Resource Abstraction Problem): Given a set of circuits $F$ that span over $N > 1$ member networks, design a security protocol in the resource discovery system to ensure that for all network resource properties (e.g., bandwidth, delay and loss rate), (1) the aggregator receives the equivalent, privacy-preserving resource abstraction $\Pi_p(F)$; and (2) for any $\mathbb{M}_i$, it does not know any linear inequality from any other $\Pi_j(F_j)$, where $j \neq i$.*

*3) Security Model:* In this paper, we assume a *semi-honest* security model, *i.e.*, the aggregator and all member networks will not deviate from the security protocol, but merely try to gather information during the execution of the protocol [44]. This is sufficient for collaboration science networks where member networks share resources to collaboratively conduct common tasks such as data transfers, storage and analytics.

### *B. Resource Abstraction Obfuscating Protocol*

There are different design options for Problem IV-A.2, *e.g.*, garbled circuit based protocols [45]. However, these designs would incur expensive computation and communication overhead, hence are not suitable for the need of multi-domain resource discovery. In this paper, we tackle this problem by designing a novel resource abstraction obfuscating protocol that only requires simple operations on matrices, *i.e.*, addition and multiplication.

*1) Basic Idea:* Our protocol leverages random matrix theory [46], [47]. In particular, each $\mathbb{M}_i$ independently computes and sends to the aggregator a set of disguised *linear equations*, which are derived from the private $\Pi_i(F_i)$, a random matrix $\mathbf{P_i}$ known only to $\mathbb{M}_i$, two random matrices $\mathbf{C_i}$ and $\mathbf{D_i}$ known only to $\mathbb{M}_i$ and $\mathbb{M}_{i-1}$, and two random matrices $\mathbf{C_{i+1}}$ and $\mathbf{D_{i+1}}$ known only to $\mathbb{M}_i$ and $\mathbb{M}_{i+1}$.

*2) Protocol:* The protocol is composed of three phases: initialization, obfuscation and transmission, as shown in Fig. 8. For the simplicity of presentation, we let $m_i = |\Pi_i(F_i)|$, *i.e.*, the number of linear inequalities in $\Pi_i(F_i)$ after redundancy removal, and $M_i = \sum_{j=1}^{i} m_j$. And for each circuit $f_j$, we also omit the superscript representing different properties in the corresponding $x_j$. As such, the resource abstraction of a member network $i$ is written as $\Pi_i(F_i) = \mathbf{A_i x} \leq \mathbf{b_i}$.

During the *initialization phase*, all member networks agree on a common $k > \sum m_i$. For each $\mathbb{M}_i$ where $i = 1, 2, \ldots, N-1$, it generates a $k$-by-$(|F| + m_i + m_{i+1})$ random matrix $\mathbf{C_i} = [\mathbf{C_i}^{|\mathbf{F}|}\ \mathbf{C_i}^{m_i}\ \mathbf{C_i}^{m_{i+1}}]$, and a $k$-by-1 random matrix $\mathbf{D_i}$, and sends to $\mathbb{M}_{i+1}$. And we define $\mathbf{C_0}$, $\mathbf{D_0}$, $\mathbf{C_N}$ and $\mathbf{D_N}$ as zero matrices. As we will illustrate in the remaining of this section, these zero matrices are used for presentation completeness and will not affect the correctness of the obfuscating protocol.

During the *obfuscation phase*, each $\mathbb{M}_i$ introduces $m_i$ slack variables, denoted by $\mathbf{x_i^s}$, to transform $\Pi_i(F_i) = \mathbf{A_i x} \leq \mathbf{b_i}$ from the standard form to the augment form and gets the following equivalent linear system:

$$\left[\mathbf{A_i} \quad \mathbf{I_{m_i}}\right]\left[\mathbf{x},\ \mathbf{x_i^s}\right] = \mathbf{b_i}. \tag{5}$$

We then add slack variables introduced by all other member networks with zero coefficients into the linear system in Equation (5) and get the following equivalent linear system:

$$\left[\mathbf{A_i} \quad \mathbf{0_{M_{i-1}}} \quad \mathbf{I_{m_i}} \quad \mathbf{0}\right]\left[\mathbf{x},\ \mathbf{x_1^s},\ \ldots,\ \mathbf{x_i^s},\ \ldots,\ \mathbf{x_N^s}\right] = \mathbf{b_i}. \tag{6}$$

Next, each $\mathbb{M}_i$ generates a private random matrix $\mathbf{P_i} \in R^{k \times m_i}$, and left-multiplies both sides of Equation (6) to get:

$$\left[\mathbf{P_i A_i} \quad \mathbf{0_{M_{i-1}}} \quad \mathbf{P_i} \quad \mathbf{0}\right]\left[\mathbf{x},\ \mathbf{x_1^s},\ \ldots,\ \mathbf{x_i^s},\ \ldots,\ \mathbf{x_N^s}\right] = \mathbf{P_i b_i}. \tag{7}$$

Then each $\mathbb{M}_i$ adds

$$\left[\mathbf{C_i^{|F|}} - \mathbf{C_{i-1}^{|F|}} \quad \mathbf{0_{M_{i-2}}} \quad -\mathbf{C_{i-1}^{m_{i-1}}} \quad -\mathbf{C_{i-1}^{m_i}} + \mathbf{C_i^{m_i}} \quad \mathbf{C_i^{m_{i+1}}} \quad \mathbf{0}\right],$$

to the coefficient matrix of the left-hand-side (LHS) of Equation (7), and adds $-\mathbf{D_{i-1}} + \mathbf{D_i}$ to its right-hand-side (RHS) to get Equation (8), as shown at the bottom of this page, where it can be observed that for each $\mathbb{M}_i$, the coefficient matrix of LHS of Equation (8) is of dimension $k$-by-$|F| + M_N$, and the RHS is of dimension $k$-by-1.

In the *transmission phase*, each $\mathbb{M}_i$ encrypts the set of linear equations in Equation (8) using a symmetric-key algorithm, *e.g.*, AES, and sends the cypher text to the aggregator. After collecting the linear equations from all member networks, the aggregator decrypts them and computes the sum of all LHS matrices and RHS matrices of all member networks, respectively. After simple elimination, the LHS sum is expressed as:

$$\left[\sum \mathbf{P_i A_i} \quad \mathbf{P_1} \quad \ldots \quad \mathbf{P_N}\right].$$

Similarly, the sum of all RHS matrices of all member networks can be expressed as $\sum \mathbf{P_i b_i}$. Denoting $[\mathbf{x_1^s}, \ldots, \mathbf{x_N^s}]$ as $\mathbf{x^s}$, the aggregator can get the privacy-preserving abstraction $\Pi_p(F)$:

$$\left[\sum \mathbf{P_i A_i}\ \mathbf{P_1} \quad \ldots \quad \mathbf{P_N}.\right]\left[\mathbf{x},\ \mathbf{x^s}\right] = \sum \mathbf{P_i b_i}. \tag{9}$$

*3) Example:* We use the example in Fig. 7 to illustrate the resource abstraction obfuscating protocol. For simplicity,

$$\left[\mathbf{P_i A_i} + \mathbf{C_i^{|F|}} - \mathbf{C_{i-1}^{|F|}}\ \mathbf{0_{M_{i-2}}}\ -\mathbf{C_{i-1}^{m_{i-1}}}\ \mathbf{P_i} - \mathbf{C_{i-1}^{m_i}} + \mathbf{C_i^{m_i}}\ \mathbf{C_i^{m_{i+1}}}\ \mathbf{0}\right] \cdot \left[\mathbf{x},\ \mathbf{x_1^s},\ \ldots,\ \mathbf{x_i^s},\ \ldots,\ \mathbf{x_N^s}\right] = \mathbf{P_i b_i} - \mathbf{D_{i-1}} + \mathbf{D_i} \tag{8}$$

we assume three member networks agree on $k = 4$. The private random matrices $\mathbf{P_1}$, $\mathbf{P_2}$ and $\mathbf{P_3}$ are generated as $\mathbf{P_1} = [11, 49, 95, 34]$, $\mathbf{P_2} = [58, 22, 75, 25]$, and $\mathbf{P_3} = [50, 69, 89, 95]$. The obfuscated resource abstractions computed by each network are:

$$\begin{aligned}
15x_1 + 14x_2 + 15x_{11}^s + 4x_{21}^s + 0x_{31}^s &= 1130,\\
53x_1 + 50x_2 + 53x_{11}^s + 2x_{21}^s + 0x_{31}^s &= 4910,\\
96x_1 + 97x_2 + 96x_{11}^s + 4x_{21}^s + 0x_{31}^s &= 9540,\\
38x_1 + 37x_2 + 38x_{11}^s + 1x_{21}^s + 0x_{31}^s &= 3420,\\
-2x_1 + 47x_2 + 0x_{11}^s + -3x_{21}^s + 47x_{31}^s &= 1470,\\
-4x_1 + 68x_2 + 0x_{11}^s + -4x_{21}^s + 68x_{31}^s &= 2040,\\
-4x_1 + 85x_2 + 0x_{11}^s + -3x_{21}^s + 86x_{31}^s &= 2630,\\
-4x_1 + 91x_2 + 0x_{11}^s + -2x_{21}^s + 94x_{31}^s &= 2810,
\end{aligned}$$

and

$$\begin{aligned}
56x_1 + 0x_2 + -4x_{11}^s + 57x_{21}^s + 3x_{31}^s &= 1740,\\
22x_1 + 0x_2 + -4x_{11}^s + 24x_{21}^s + 1x_{31}^s &= 680,\\
78x_1 + 2x_2 + -1x_{11}^s + 74x_{21}^s + 3x_{31}^s &= 2250,\\
25x_1 + 0x_2 + 0x_{11}^s + 25x_{21}^s + 0x_{31}^s &= 770.
\end{aligned}$$

Summing these obfuscated resource abstractions together, the resulting resource abstraction $\Pi_p(F)$ collected by the aggregator is:

$$\begin{aligned}
69x_1 + 61x_2 + 11x_{11}^s + 58x_{21}^s + 50x_{31}^s &= 4340,\\
71x_1 + 118x_2 + 49x_{11}^s + 22x_{21}^s + 69x_{31}^s &= 7630,\\
170x_1 + 184x_2 + 95x_{11}^s + 75x_{21}^s + 89x_{31}^s &= 14420,\\
59x_1 + 129x_2 + 34x_{11}^s + 25x_{21}^s + 95x_{31}^s &= 7000,
\end{aligned}$$

where $x_{11}^s$, $x_{21}^s$ and $x_{31}^s$ are slack variables. Assume the user's objective is to maximize the throughput, *i.e.*, $x_1 + x_2$. Using this set of linear inequalities as the constraint, it can get the optimal solution where $x_1 = x_2 = 30$ Gbps, the same as when using Equation (4) as the constraint.

### C. Analysis

We conduct rigorous analysis on different properties of the proposed obfuscating protocol.

*1) Correctness:* We first study the correctness of this protocol. In particular, we prove the correctness of this protocol for different properties in the following propositions.

***Proposition** 1 (Bandwidth Resource Abstraction Equivalence): If the resource abstraction $\mathbf{Ax} \leq \mathbf{b}$, where $\mathbf{A} = [\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_N}]$ and $\mathbf{b} = [\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b_N}]$, represents the bandwidth property for a set of circuits $F$ over $N$ member networks. Using the proposed obfuscating protocol, the bandwidth feasible region of represented by Equation (9) is the same as the bandwidth feasible region represented by $\mathbf{Ax} \leq \mathbf{b}$.*

*Proof:* To prove this proposition, we first observe that the bandwidth feasible region of $\mathbf{Ax} \leq \mathbf{b}$ is the same as that of

$$\begin{bmatrix}\mathbf{A} & \mathbf{I_{M_N}}\end{bmatrix}\begin{bmatrix}\mathbf{x},\ \mathbf{x^s}\end{bmatrix} = \mathbf{b} \tag{10}$$

Representing $\mathbf{P} = [\mathbf{P_1}, \ldots, \mathbf{P_N}] \in R^{k \times M_N}$, we first observe that $\begin{bmatrix}\sum \mathbf{P_i A_i}\ \mathbf{P_1}\ \ldots\ \mathbf{P_N}\end{bmatrix} = \mathbf{P}\begin{bmatrix}\mathbf{A}\ \mathbf{I_{M_N}}\end{bmatrix}$, and that $\sum \mathbf{P_i b_i} = \mathbf{Pb}$ [47]. It is easy to see that when $\begin{bmatrix}\mathbf{x}\ \mathbf{x^s}\end{bmatrix}$ satisfies Equation (10), it also satisfies Equation (9).

Next, from the results in [46] and that $\mathbf{P} \in R^{k \times M_N}$, we have $rank(\mathbf{P}) = M_N < k$. As a result, $\mathbf{P}$ has a left inverse matrix $\mathbf{P}_{left}^{-1}$ where $\mathbf{P}_{left}^{-1}\mathbf{P} = \mathbf{I}_{M_N}$. Hence when $\begin{bmatrix}\mathbf{x}\ \mathbf{x^s}\end{bmatrix}$ satisfies Equation (9), *i.e.*, $\mathbf{P}\begin{bmatrix}\mathbf{A}\ \mathbf{I_{M_N}}\end{bmatrix}\begin{bmatrix}\mathbf{x},\ \mathbf{x^s}\end{bmatrix} = \mathbf{Pb}$, we have

$$\mathbf{P}_{left}^{-1}\mathbf{P}\begin{bmatrix}\mathbf{A} & \mathbf{I_{M_N}}\end{bmatrix}\begin{bmatrix}\mathbf{x},\ \mathbf{x^s}\end{bmatrix} = \mathbf{P}_{left}^{-1}\mathbf{Pb},$$

which then transforms into Equation (10). Therefore, Equations (9) and (10) represent the same bandwidth feasible region, which completes the proof. ■

Next, we give the correctness proof for delay and loss rate resource abstraction equivalence.

***Proposition** 2 (Delay/Loss Rate Resource Abstraction Equivalence): If the resource abstraction $\mathbf{Ax} \leq \mathbf{b}$, where $\mathbf{A} = [\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_N}]$ and $\mathbf{b} = [\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b_N}]$, represents the delay or loss rate property for a set of circuits $F$ over $N$ member networks. The aggregator can compute the aggregated multi-domain delay or loss rate resource abstraction $\oplus\Pi_i(F_i)$, where $\Pi_i(F_i) = \mathbf{A_i}x \leq \mathbf{b_1}$ using the set of linear equations in Equation (9).*

*Proof:* From the proof of Proposition 1, we know that $\mathbf{P}$ is full column rank. Observe $\begin{bmatrix}\sum \mathbf{P_i A_i}\ \mathbf{P_1}\ \ldots\ \mathbf{P_N}\end{bmatrix}$, the coefficient matrix on the LHS of Equation (10), we can find that each column of $\sum \mathbf{P_i A_i}$ can be linearly expressed by the columns in $\mathbf{P}$. As a result, we can dissect $\begin{bmatrix}\sum \mathbf{P_i A_i}\ \mathbf{P_1}\ \ldots\ \mathbf{P_N}\end{bmatrix}$ into $\mathbf{P}\begin{bmatrix}\mathbf{A}\ \mathbf{I_{M_N}}\end{bmatrix}$ through Gaussian Elimination and learn $\mathbf{A}$. Similarly, we can learn $\mathbf{b}$. As such, we can reconstruct $\mathbf{Ax} \leq \mathbf{b}$ and then compute the aggregated multi-domain delay or loss rate resource abstraction $\oplus\Pi_i(F_i)$. ■

*2) Security:* Next, we give the following proposition on the privacy-preserving property of the proposed protocol.

***Proposition** 3 (Resource Abstraction Privacy-Preserving): In the semi-honest security model, the proposed resource abstraction obfuscating protocol ensures that (1) the aggregator cannot associate any linear equation it receives in $\Pi_p(F)$ with any particular member network, and (2) for any $\mathbb{M}_i$, it does not know any linear inequality from any other $\Pi_j(F_j)$ $(j \neq i)$.*

*Proof:* From the description of the resource abstraction obfuscation proof, we see that each $\mathbb{M}_i$ directly sends its own set of disguised linear equations back to the aggregator, hence it does not know any linear inequality from any other member network. Furthermore, even though Proposition 2 shows that it is possible for the aggregator to compute $\mathbf{Ax} \leq \mathbf{b}$, the aggregator cannot associate any $\mathbf{A_i}$ or $\mathbf{b_i}$ to any particular $\mathbb{M}_i$ because $\mathbf{P_i}$ is also disguised by matrices $\mathbf{C_i}$, $\mathbf{C_{i-1}}$, $\mathbf{C_{i+1}}$, $\mathbf{D_i}$, $\mathbf{D_{i-1}}$ and $\mathbf{D_{i+1}}$ before sending back to the aggregator. ■

Even with Proposition 2 and the inter-member-network-path information of each circuit, the aggregator still cannot associate any linear inequality in $\mathbf{Ax} \leq \mathbf{b}$ with the corresponding member network or any networking device (*i.e.*, switch or link). This is because (1) the set of linear equations sent by each member network do not represent its original feasible region, and (2) the inter-member-network-path does not reveal any topology information inside member networks.

With Propositions 1, 2, and 3, we can get the following theorem.

***Theorem** 1: Given a set of circuits $F$ that span over $N$ member networks, the proposed resource abstraction obfuscating protocol ensures that the aggregator receives equivalent, privacy-preserving resource abstraction and each member network only knows its own resource feasible region.*

As stated in Section IV-A, the resource abstraction obfuscating protocol was designed for the semi-honest security model. Next, we analyze the privacy-preserveness of our obfuscating protocol in a collusion security model, a more adversarial model in which some member networks may share their resource abstractions with the aggregator. Specifically, we prove the following proposition.

***Proposition** 4 (Resource Abstraction Privacy-Preserving against Collusion): Assume some member networks may collude with the aggregator to share their resource abstractions $\Pi_j(F_j)$. Given a non-colluding member network $\mathbb{M}_i$, the resource abstraction obfuscating protocol ensures that the aggregator cannot associate any linear equation it receives in $\Pi_p(F)$ with $\mathbb{M}_i$ unless all other $N-1$ member networks choose to share their resource abstractions $\Pi_j(F_j)$, where $j \neq i$, with the aggregator.*

*Proof:* The proof of this proposition follows the proof of Proposition 2 and Proposition 3. Essentially, in order to associate any linear equation in $\Pi_p(F)$ to a non-colluding member network $\mathbb{M}_i$, the aggregator needs to know that this linear equation does not belong to any other member network. Given that the colluding member networks only share their own resource abstractions with the aggregator, this can only be achieved for all other $N-1$ networks to share their own resource abstractions $\Pi_j(F_j)$, where $j \neq i$, with the aggregator. ■

This proposition indicates that the obfuscating protocol can preserve the privacy of a member network against the collusion between the aggregator and up to $N-2$ member networks.

*3) Efficiency:* We next analyze the efficiency of our protocol at different phases. During the initialization phase, the main overhead comes from the process each member network agreeing on $k$, and each $\mathbb{M}_i$ share $\mathbf{C_i}$ and $\mathbf{D_i}$ with $\mathbb{M}_{i+1}$. The first part can be efficiently realized using leader-election algorithms in ring topology or pre-configured. For the second part, it can be efficiently realized by sharing random seeds between $\mathbb{M}_i$ and $\mathbb{M}_{i+1}$. In the obfuscating phase, the computation overhead is also low because it only involves simple, cheap matrix operations, *e.g.*, addition and multiplication.

One may have concern on the transmission overhead of our protocol in the transmission phase because we disguise the set of linear inequalities of each member network into a larger set of linear equations. As such, we quantify the transmission overhead of our obfuscating protocol as follows:

***Proposition** 5 (Transmission Overhead): Given a resource discovery procedure for a set of circuits $F$ spanning over $N$ member networks, the transmission overhead of the resource abstraction obfuscating protocol at each member network is $O(k|F|)$, where $k > \sum m_i$.*

*Proof:* Observing the set of equations sent by each $\mathbb{M}_i$ in Equation (8), we can see that most of the columns of the LHS coefficient matrix are zero-columns. Therefore, each $\mathbb{M}_i$ only needs to send nonzero-columns to the aggregator and specifies the indice of these columns. As such, the number of elements to be sent to the aggregator is bounded by $O(k|F|)$, where $k > \sum m_i$. This substantially reduces the amount of data needs to be transmitted from $\mathbb{M}_i$ to the aggregator. ■

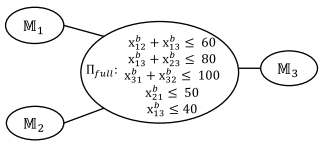



Fig. 9. An illustrating example of super-set projection.

## V. Super-Set Resource Abstraction Projection

As pointed out in Section I, the third challenge for resource discovery is efficiency and scalability, as the number of resource discovery requests may be large in collaboration networks and each request could trigger a resource discovery procedure. This procedure requires the communication between the aggregator and the user, and between the aggregator and every Mercator domain server in member networks. Furthermore, the introduction of resource abstraction obfuscating in Section IV may also increase the communication and computation overhead of resource discovery. To address the efficiency and scalability issue, we develop a novel super-set projection technique, which does not require any change to the resource abstraction design in Section III or the extended obfuscating protocol in Section IV. In this section, we first describe its basic idea, and then give the details of this mechanism.

### A. Basic Idea

The intuition of super-set projection is simple: to have the aggregator proactively discover the resource abstraction for a set of circuits between every pair of source and destination member networks, and use these pre-computed abstractions to quickly project to get the resource abstraction for user's requests.

In particular, in a collaboration network of $N$ member networks, the super-set projection technique first simulates the need of $N(N-1)$ artificial circuits, where each circuit $f_{ij}$ represents an artificial circuit from $\mathbb{M}_i$ to $\mathbb{M}_j$. With this artificial resource discovery request, the aggregator follows the normal resource discovery process to discover the shared bandwidth of all these $N(N-1)$ circuits across the whole collaboration network, represented by $\Pi_{full}$. When a user sends an actual resource discovery request for a set of $F$ circuits, the aggregator checks the source and destination member networks of each circuit, and uses the stored $\Pi_{full}$ to derive $\Pi(F)$ by removing unrelated inequalities and unrelated artificial circuits, instead of starting a new resource discovery procedure. In this way, the overhead of resource discovery is reduced to a single round of message exchange between the aggregator and the user.

*Example:* Consider an example of 3 member networks in Fig. 9. With the super-set projection, the aggregator discovers

the bandwidth sharing of all $3 \times 2 = 6$ network-to-network artificial circuits as $\Pi_{full}$ in the figure. When a user submits a resource discovery request for two circuits $(S_1, D_1)$ and $(S_2, D_2)$, where $S_1$ is in $\mathbb{M}_1$, $S_2$ and $D_1$ are in $\mathbb{M}_2$ and $D_2$ is in $\mathbb{M}_3$. The aggregator first maps the $(S_1, D_1)$ to the artificial circuit from $\mathbb{M}_1$ to $\mathbb{M}_2$, and $(S_2, D_2)$ to the artificial circuit from $\mathbb{M}_2$ to $\mathbb{M}_3$. Next, it projects $\Pi_{full}$ to these two circuits to get the resource abstraction for these two circuits by (1) removing all linear inequalities that do not contain $x^b_{12}$ or $x^b_{23}$, and (2) for every remaining linear inequality, remove all the items on the LHS that are not $x^b_{12}$ or $x^b_{23}$. Finally, it returns the resource abstraction: $\{x^b_{12} \leq 60,\ x^b_{23} \leq 80\}$, to the user.

### B. Update of $\Pi_{full}$

We ensure the freshness of $\Pi_{full}$ via two mechanisms. First, the Mercator domain servers at member networks periodically send updated information to the aggregator. Second, when the reservation system receives and successfully executes a resource reservation request from the user, it sends a notification to the aggregator with the reservation details so that the aggregator can update $\Pi_{full}$. The aggregator will only query the Mercator domain servers to obtain an up-to-date abstraction for the user when the user fails to reserve the resource based on the projected abstraction.

### C. Handling Heterogeneous Flows

One may notice that the super-set projection technique is designed based on the assumption that given a source-destination member network pair, all the traffic flows between these two member networks will be treated homogeneously by all other member networks. In practice, flows between the same source-destination member network pair may be handled differently by other member networks, *i.e.*, they are heterogeneous flows. To address this limitation, we use traffic classes to differentiate heterogeneous flows. In particular, for each source-destination member network pair with $G$ different traffic classes, the super-set projection technique considers these classes as $G$ separate artificial circuits and proactively discovers the bandwidth sharing among these $G$ circuits and other artificial circuits.

## VI. Implementation and Deployment

In this section, we describe the implementation of the Mercator and the recent deployment of Mercator in a small federation network to orchestrate large-scale science dataset transfer between two major cities in the United States.

### A. Implementation

Figure 10 shows the Mercator domain server implementation, including the Mercator domain server and the aggregator.

*1) Mercator Domain Server:* We build the Mercator domain server on top of the OpenDaylight Software Defined Network controller [48]. Essentially, the Mercator domain server collects the network state information from the OpenDaylight controller, *e.g.*, topology, policy and traffic statistics, processes the collected information into resource abstraction, and sends the abstraction back to the aggregator.

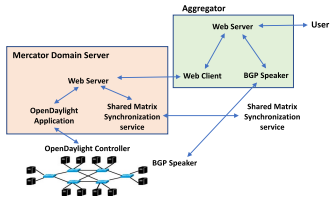



Fig. 10. The Mercator implementation.

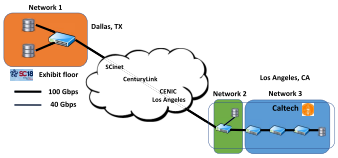



Fig. 11. Deployment of Mercator on a small federation network at Dallas, Texas and Los Angeles, California.

The Mercator domain server has three modules: an OpenDaylight application running in a Karaf container, and a web server accepting the resource discovery from the aggregator and responding with the resource abstraction, and a synchronization service communicating with neighbor domain servers to exchange shared-random matrices that are used for abstraction obfuscating.

*2) Aggregator:* The aggregator has three modules: a web server, a web client and a BGP speaker. The web server provides interfaces for the user to submit a resource discovery request for a set of circuits in a format specified by the ALTO protocol [24]. The web client communicates with Mercator domain servers in different member networks by sending resource discovery requests. In addition, the BGP speaker maintains BGP sessions with the border routers or route servers at member networks to collects inter-member-network paths information.

### B. Deployment

We deploy Mercator in a small federation network shown in Figure 11. Specifically, this federation is composed of three member networks. Network 1 is in Dallas, Texas, and Network 2 and network 3 are in Los Angeles, California. Network 1 is connected to network 2 through a layer-2 WAN circuit with a 100 Gbps bandwidth, provisioned by several providers such as SCinet, CenturyLink and CENIC. Network 1 is a temporal science network in the CMS experiment [49], while network 2 and 3 are long-running CMS Tier-2 sites. In this federation, users need to reserve network resources to transfer large-scale science datasets (*e.g.*, with a size of hundreds of PB) between networks.

In our deployment, a Mercator domain server is deployed in each network, and the aggregator is deployed in Dallas. We also deploy SFP, a BGP-compatible routing protocol providing fine-grained routing information [50] in the federation. Upon receiving a user's request, *e.g.*, to discover network resources for circuits from network 1 to network 2 and 3, the aggregator in Mercator contacts the SFP speaker at different networks to discover the interdomain routes for

these circuits, and then sends resource discovery requests to the Mercator domain servers at different networks. After collecting the resource abstraction from Mercator domain servers, the aggregator assembles them and returns to the user. The user then uses such information to compute the optimal amount of network resources to reserve for each circuit and send to the underlying reservation to reserve the resources.

*1) Performance:* We evaluate the accuracy and latency of Mercator for discovering network resources in this network. During our evaluation, Mercator accurately discovers the network resource information for a large amount of circuits reservation requests with a very low discovery latency. Specifically, for all the reservation requests, Mercator always provides the accurate information of available bandwidth sharing in the network (*i.e.*, a 100% accuracy), with an average discovery latency of $\sim$100 milliseconds, and a worst latency of less than 1 second. With the discovered network resource information, users can transmit large-scale science datasets at a speed up to 100 Gbps, (*i.e.*, the theoretical maximal throughput). A demonstration of the Mercator deployment in this federation network can be found at [51].

## VII. Evaluation

We implement Mercator on commodity servers (*i.e.*, equipped with Intel(R) Xeon(R) E5-2609 2.50GHz 4-core CPU and 32 GB memory) and evaluate its performance based on a member-network-level topology from a large federation of networks supporting large-scale distributed science collaborations, and using real traffic traces from recent science experiments. After describing our experimental setup, we first demonstrate the benefits of resource abstraction through algebraic-expression enumeration. Second, we demonstrate the efficiency of the proposed resource abstraction obfuscation protocol. Finally, we demonstrate that the super-set projection technique substantially increases the scalability of Mercator.

### A. Experimental Setup

We evaluate Mercator on the member-network-level topology from LHC Open Network Environment (LHCONE), a global science network consisting of 62 member networks, where scientists conduct large-scale distributed analytics. Because inter-member-network routing typically is not based on shortest path routing, but follows business relationships (*e.g.*, customer, peer, provider), we label the connections between every pair of connected member networks with their business relationship using the CAIDA network relationships dataset [52], and we compute the inter-member-network paths according to conventional policies for selecting and exporting routes. For member networks' intradomain topologies, we randomly select a topology for each network from the Topology Zoo [53], which provides a collection of real intradomain topologies. The topology of transit member networks varies from 31 switches/routers with 33 links to 49 switches/routers with 85 links. The topology of stub member networks (*e.g.*, campus science networks) ranges from 7 switches/routers with 6 links to 21 switches/routers with 44 links.

### B. Benefits of Resource Abstraction Through Algebraic-Expression Enumeration

The first set of experiments demonstrate the benefits of the resource abstraction through algebraic-expression enumeration. We show that this abstraction reduces the time to discover network resources by up to six orders of magnitude, and allows fairer allocations of network resources.

*1) Methodology:* To evaluate the benefits of this resource abstraction, we replay the trace from a large-scale distributed experiment, and submit network resource reservations for the corresponding flows. More specifically, we use the actual trace from the CMS experiment [54], a major scientific experiment in LHC, and a main source of traffic in LHCONE. We extract the traffic flows, with their source member network, destination member network and the time. We focus on the 7-day trace starting from September 30, 2018 to October 6, 2018, and slice the data trace into 24 continuous 2-hour time windows. We apply the resources reservation once every time window. In other words, resources for traffic flows starting at the same time window are reserved in the same request, and we assume all resources will be released in the next time window.

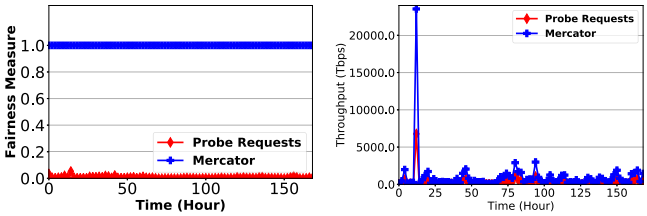
We compare the performance of Mercator with that of existing reservation systems. In particular, for existing systems, we consider one that adopts a probe-requests based approach:

- **Mercator**: As described in Section II, for every resource discovery request, the aggregator queries the relevant member networks for their resource abstraction, and then derives the feasible bandwidth allocation region.
- **Probe requests**: As described in Section I, existing resource reservation systems such as OSCARS process each circuit in the request one at a time and in a sequential order. For each circuit, the resource reservation system initiates a depth-first search to probe if each member network can provide the requested bandwidth. We set the initial requested bandwidth for a circuit as $C/N$ where $C$ is the source host's capacity, and $N$ is the number of flows from that host. In the event of a failure, the resource reservation system performs a binary search of the available bandwidth repeatedly halving the requested bandwidth until success. The process is repeated for each circuit in the request.

*2) Results:* First, we consider that the goal of the resource allocation policy is to maximize the minimum throughput of all the requested flows (max-min fairness). Such a policy is commonly desired as it ensures high throughput and fairness across the circuits. We compare the fairness of the network resource allocations obtained with Mercator to that obtained with the probe-requests based solution. We adopt Jain's fairness index [55] to measure the fairness [56]:

$$J(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} {x_i}^2}$$

where $x_i$ is the ratio of the actual allocation and the optimal fair allocation for a single flow. 12a shows that with resource abstraction, Mercator can always compute the optimal max-min fairness allocation. Hence its fairness index is always 1. In contrast, the highest fairness index the probe-requests can get is 0.05, with most of the slots even smaller than 0.01.




**(a)** Obj. = max-min fairness **(b)** Obj. = max total throughput

Fig. 12. Comparison of performance between the probe-requests approach and Mercator in different objectives.

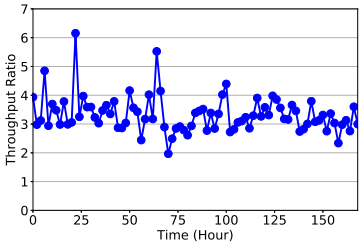



Fig. 13. Ratio of throughput between Mercator and the probe-requests approach when the objective is to maximize the total throughput.

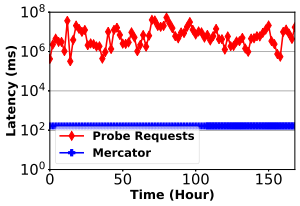



Fig. 14. Resource discovery latency of the probe-requests approach and Mercator.

Second, we consider the case where the objective is to maximize the total throughput. 12b shows that the total throughput of Mercator is larger than that obtained by the probe-requests based solution throughout the whole experiment. Fig. 13 shows that the ratio of throughput of Mercator over that of probe-requests based solution is by 3.47x on average, and up to 6.2x. The results are noteworthy given that Mercator assumes the routes for each circuit to be completely determined by the underlying intradomain routing protocol. In contrast, the probe-requests approach sequentially explores every possible route for each circuit until it finds an available one. In other words, even with much less exploration, Mercator still outperforms the probe requests significantly. Allowing Mercator to consider not only the routes provided by the underlying routing protocols, but also all other available routes, could lead to significant additional improvements. We leave the extension of Mercator to consider all possible routes in the network as future work.

Fig. 14 presents the total resource discovery latency for completing all circuits resource reservations in a time window. We assume the aggregator to be in New York, and consider network latencies as measured in [57]. The figure shows the total resource discovery latency with Mercator can reduce the time to discover network resources by four orders of magnitude on average and up to six orders of magnitude at times.

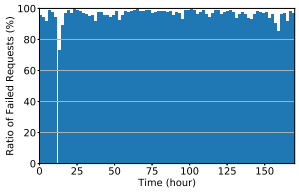



Fig. 15. Ratio of failed requests in the probe-requests approach.

This is because resource abstraction allows users to query the information from different member networks in parallel. In contrast, existing probe-requests based solutions process requests sequentially, and continuously probe to discover the available network resources.

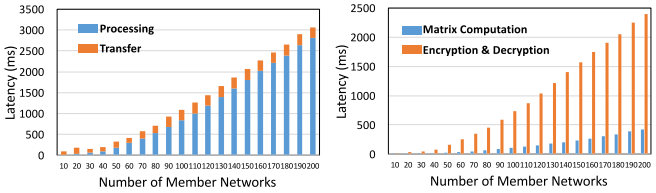
Finally, we highlight that the probe-requests based solution suffers high request failure ratio, *i.e.*, a large number of requests cannot succeed: We define a failure of a request as the inability to reserve resource for the circuit, due to the lack of remaining capacity despite the gradually decreasing requested bandwidth. Fig. 15 shows that during the 7-day period Mercator is running, the probe-requests based solution has an average request failure ratio of 87%. In other words, more than 80% of the circuits cannot reserve network resources. This is because the probe-requests approach processes the request for each circuit sequentially. Therefore, the first few circuits may successfully reserve network resources and saturate the network. As such, the majority of the latter requests may fail as the links do not have any spare resources. In contrast, the request failure ratio of Mercator is null because Mercator returns a feasible region for the set of circuits so that the user can make optimal reservation decisions for all circuits.

### *C. Efficiency of Resource Abstraction Obfuscating Protocol*

This second set of experiments evaluate the performance of the resource abstraction obfuscating protocol. We show that this protocol efficiently scales for collaboration networks of 200 member networks, with a maximal overall latency around 3 seconds and an average data transmission overhead between the aggregator and member networks of only around 180 KB.

*1) Methodology:* We conduct our experiment by using the member-network-level topology from the LHC Open Network Environment (LHCONE). In each round of the experiment, we randomly select a set of member networks from the topology. For each chosen member network, we randomly select a set of $m$ linear inequalities, where $m$ is randomly chosen between 5 and 15, to represent the bandwidth feasible reason for 10 circuits in this member network. For the encryption and decryption operations in the obfuscating protocol, we use the AES algorithm, provided by the Python Cryptography Toolkit (pycrypto) [58]. The parameters $k$, $\mathbf{C_i}$ and $\mathbf{D_i}$ are pre-configured as discussed in Section IV-C.

We consider two metrics, *i.e.*, the latency and the data transmission overhead of the resource abstraction obfuscating protocol. First, the overall latency of the protocol is measured from the beginning of the obfuscation phase, when each member network independently starts to obfuscate its own set of linear inequalities, to the end of the transmission process,




**(a)** Overall latency. **(b)** Processing latency.

Fig. 16. The latency of the resource abstraction obfuscating protocol.

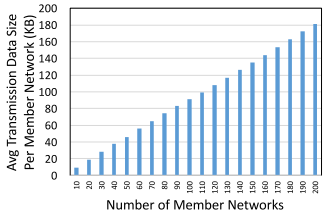



Fig. 17. The data transmission overhead of the resource abstraction obfuscating protocol.

when the aggregator obtains $\sum \mathbf{P_i A_i x} = \mathbf{b}$. We use the field statistic results measured in [57] as the communication latencies between the aggregator and the Mercator domain servers at the different member networks. Second, the data transmission overhead is measured as the size of the set of encrypted, obfuscated linear equations transferred from each member network to the aggregator. We vary the number of member networks from 10 to 200, in a step size of 10. For each number of member networks, we repeat the experiment 10 times and measure the average values of these metrics.

*2) Results:* We present the results of our experiments in Fig. 16 and Fig. 17. In particular, Fig. 16a shows the overall latency of the obfuscating protocol under different numbers of member networks, together with a break down on processing delay and transmission latency. We observe that even for a large collaboration network with 200 member networks, which is larger than most existing operational collaboration networks, the overall latency of the resource abstraction obfuscating protocol is only slightly over 3 seconds, which demonstrates that the latency of this protocol is reasonably low. We also observe that the processing latency takes a much higher percentage than the transmission latency and that the processing latency has a linear growth as the number of member networks increases. We further plot the breakdown of the processing latency. Fig. 16b shows that both the cryptography operations of AES and the matrix operations in the resource abstraction obfuscating protocol increases linearly as the number of member networks increases, but the AES encryption and decryption operations are the most expensive operations in the protocol (*i.e.*, up to 2.4 seconds for federations of 200 member networks). More importantly, although the obfuscating protocol may take over 3 seconds for a federation of 200 member networks, we emphasize that with the super-set projection technique, the Mercator domain servers do not need to execute the obfuscating protocol for each individual request.

Next, we present the average data transmission overhead of the obfuscating protocol at each member network in Fig. 17.

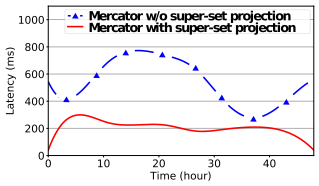



Fig. 18. Comparison of latency between Mercator with and without super-set projection.

We see in this figure that even after the encryption, the size of data to be transmitted from member networks to the aggregator is still very small. For example, for a collaboration network with 200 member networks, the average size of data transmitted from a member network to the aggregator is only 180 KB. As discussed in Section IV-C, this is because most of the columns of the LHS coefficient matrix are zero-columns and each member network only needs to send nonzero-columns to the aggregator. The linear scaling of the data transmission overhead (*i.e.*, the ciphertext) at each member network comes from the linear increase of the number of disguised linear equations (*i.e.*, the plaintext), which is caused by the linear increase of $k$ due to the increased number of member networks. This is consistent with Proposition 5 in Section IV-C.

### *D. Efficiency of Super-Set Projection*

In this experiment, we evaluate the efficiency of the super-set projection technique in improving the scalability of Mercator. We show that this mechanism improves the resource discovery delay of Mercator by 2 times, and that its update latency is within seconds in a collaborative network with 200 member networks.

*1) Methodology:* We conduct our experiments by using the same settings as in Section VII-B.1. We focus on two metrics. The first one is the resource discovery latency. When Mercator uses super-set projection, the resource discovery latency is reduced to only the round-trip time from the user to the Mercator aggregator because the aggregator can derive the resource abstraction for a request from the precomputed $\Pi_{full}$.

To have a comprehensive understanding on the scalability of super-set projection, we are also interested in a second metric, the update latency. This is measured as the resource discovery latency of from the time the aggregator starting the artificial resource abstraction discovery procedure to the time the aggregator receives the latest $\Pi_{full}$. In particular, we measure this latency under different collaboration scales by varying the number of member networks and the number of stub member networks in the collaborative network. For each setting, we repeat the experiment 10 times and compute the average update latency. In each repetition, we also randomly choose different sizes of intradomain topologies from the Topology Zoo dataset for each member network.

*2) Results:* Fig. 18 compares the resource discovery latency of Mercator with and without super-set projection for a 48-hour period in the LHCONE trace. The results for the whole 7-day period is similar, and hence is omitted.

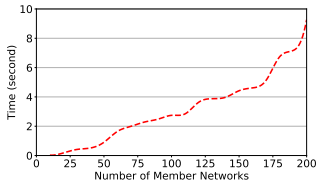



Fig. 19. Update latency of super-set projection.

We observe that the super-set projection technique decreases the average resource discovery latency by around 2 times. Fig. 19 presents the update latency of this mechanism. It shows that even in a collaborative network with 200 member networks, the update latency of $\Pi_{full}$ is still less than 10 seconds. Most importantly, although computing $\Pi_{full}$ may take up to ten seconds for a federation of 200 member networks, we emphasize that resource discovery requests do not get blocked at the aggregator because servers from the aggregator pool can still process incoming requests using the previously computed resource abstraction, which is continuously locally updated (*e.g.*, available resources are continuously reduced as incoming requests reserve resources).

## VIII. Related Work

### A. Resource Reservation

Network resource reservation systems are deployed driven by the demand and substantial benefits of providing predictable network resources [5]–[11], [13]–[15]. Systems running in a single administrative domain (*e.g.*, NetStitcher [13], SWAN [14] and B4 [15]) are often provided with detailed network information, such as the topology and links' availability. Therefore, optimizing resource reservations in a single administrative domain can be very efficient. In contrast, in a multi-domain network (*e.g.*, LHC), due to networks' concern of revealing sensitive information, resource reservation systems only allow users to submit requests for reserving a specific amount of resources (*e.g.*, a circuit providing a certain amount of bandwidth and delay), and return either success or failure [5]–[11]. Without an interface to provide network resource information, optimizing resource reservations in a multi-domain network requires a complex, time-consuming trial-and-error process.

### B. Resource Discovery

Multiple multi-domain resource discovery systems (*e.g.*, [16]–[20]) are designed to discover endpoint resources (*i.e.*, computation and storage resources) and their availability for different services across multiple domains. In contrast, there has been little progress on multi-domain network resource discovery systems that provide fine-grained, global network resource information, to support high-performance, collaborative data sciences.

Many cluster/grid resource management systems [38], [39], [59]–[64] adopt a graph-based abstraction to discover and manage network resources. This abstraction is designed for single administrative domains (*e.g.*, a company or a university) to manage their own network, where they do not need to preserve the privacy of network. If this abstraction is directly ported to a multi-domain collaborative network, it would expose the private information (*e.g.*, the network topology) of member networks, leading to security breaches.

Some systems in cloud computing [26]–[28] adopt a network-does-all approach, in which users are provided with a more expressive interface for specifying requirements on data transfers and the network orchestrates resources between different user requests. Though this approach protects the privacy of the network, the network can only provide elastic resource reservation for user requests (*i.e.*, some requests may be preempted or rejected). Some recent systems (*e.g.*, the ALTO protocol [24], [25], [29] and the SENSE project [30], [31]) provide users the information of certain properties of network resources using the one-big-switch abstraction. While this approach protect the privacy of network, it cannot provide accurate information of network resource sharing between flows (*e.g.*, bandwidth), which is critical for optimizing the emerging use cases (*e.g.*, large-scale collaborative sciences).

Some recent studies [37], [41], [42], [65], [66] propose variations of the one-big-switch abstraction to represent the resource availability and sharing among different data traffic flows using operations defined on different algebra fields. However, this abstraction (1) cannot handle complex routing and traffic engineering policies, *e.g.*, WCMP, and (2) will raise security concern when applied to multi-domain science collaborations. In contrast, Mercator provides fine-grained, global network resource information, to support high-performance, collaborative data sciences, through a unifying representation and composition framework to reveal compact, complete multi-domain network resource information.

## IX. Conclusion

Existing multi-domain network resource reservation systems often operate on coarse-grained or localized information, resulting in substantial inefficiencies. To address this issue, We present Mercator, a novel multi-domain network resource discovery system to provide fine-grained, global network resource information, to support high-performance, collaborative data sciences. The core of Mercator is a unifying representation resource abstraction using algebraic expressions to represent multi-domain network resources. We develop a resource abstraction obfuscating protocol and a super-set projection technique to ensure the privacy-preserving and the scalability of Mercator. Evaluation using real data shows that Mercator discovers fine-grained network resources by up to six orders of magnitude, allows fairer allocations of network resources, and scales to a collaborative network of 200 member networks.

## Acknowledgment

The authors thank Haizhou Du, Kai Gao, Linghe Kong, Geng Li, Yeon-sup Lim, Alan Liu, Ennan Zhai, and Yan Zhu for their help during the preparation of this paper.

## References

[1] Q. Xiang *et al.*, "Fine-grained, multi-domain network resource abstraction as a fundamental primitive to enable high-performance, collaborative data sciences," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2018, pp. 58–70.

[2] *The Large Hadron Collider (LHC) Experiment*. Accessed: Dec. 2018. [Online]. Available: https://home.cern/topics/large-hadron-collider

[3] *The Square Kilometre Array*. Accessed: Dec. 2018. [Online]. Available: https://www.skatelescope.org/

[4] *The Linac Coherent Light Source*. Accessed: Dec. 2018. [Online]. Available: https://lcls.slac.stanford.edu/

[5] *Oscars: On-Demand Secure Circuits and Advance Reservation System*. Accessed: Dec. 2018. [Online]. Available: https://www.es.net/engineering-services/oscars/

[6] M. Campanella *et al.*, "Bandwidth on demand services for European research and education networks," in *Proc. IEEE 1st Int. Workshop Bandwidth Demand*, Nov. 2006, pp. 65–72.

[7] C. Guok and D. Robertson, "ESnet on-demand secure circuits and advance reservation system (OSCARS)," Internet2 Joint, 2006, vol. 92.

[8] W. Johnston, C. Guok, and E. Chaniotakis, "Motivation, design, deployment and evolution of a guaranteed bandwidth network service," in *Proc. TERENA Netw. Conf.*, 2011, pp. 1–14.

[9] B. Riddle, "BRUW: A bandwidth reservation system to support end-user work," in *Proc. TERENA Netw. Conf.*, Poznan, Poland, 2005.

[10] J. Sobieski, T. Lehman, and B. Jabbari, "DRAGON: Dynamic resource allocation via GMPLS optical networks," in *Proc. MCNC Opt. Control Planes Workshop*, Chicago, IL, USA, 2004.

[11] X. Zheng, M. Veeraraghavan, N. S. V. Rao, Q. Wu, and M. Zhu, "CHEETAH: Circuit-switched high-speed end-to-end transport architecture testbed," *IEEE Commun. Mag.*, vol. 43, no. 8, pp. S11–S17, Aug. 2005.

[12] Q. Xiang, H. Yu, J. Aspnes, F. Le, L. Kong, and Y. R. Yang, "Optimizing in the dark: Learning an optimal solution through a simple interface," in *Proc. AAAI*, Nov. 2018, pp. 1–8.

[13] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 74–85, 2011.

[14] C.-Y. Hong *et al.*, "Achieving high utilization with software-driven WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 15–26, 2013.

[15] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.

[16] Y. Deng, F. Wang, and A. Ciura, "Ant colony optimization inspired resource discovery in P2P grid systems," *J. Supercomput.*, vol. 49, no. 1, pp. 4–21, 2009.

[17] A. Iamnitchi and I. Foster, "A peer-to-peer approach to resource location in grid environments," in *Grid Resource Management*. Springer, 2004, pp. 413–429.

[18] T. Kocak and D. Lacks, "Design and analysis of a distributed grid resource discovery protocol," *Cluster Comput.*, vol. 15, no. 1, pp. 37–52, 2012.

[19] I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, and F. Wurthwein, "The pilot way to grid resources using glideinWMS," in *Proc. IEEE CSIE*, Mar./Apr. 2009, pp. 428–432.

[20] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: The condor experience," *Concurrency Pract. Exper.*, vol. 17, nos. 2–4, pp. 323–356, 2005.

[21] R. Ahmed, N. Limam, J. Xiao, Y. Iraqi, and R. Boutaba, "Resource and service discovery in large-scale multi-domain networks," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 4, pp. 2–30, 4th Quart., 2007.

[22] A. Hameurlain, D. Cokuslu, and K. Erciyes, "Resource discovery in grid systems: A survey," *Int. J. Metadata, Semantics Ontologies*, vol. 5, no. 3, pp. 251–263, 2010.

[23] N. J. Navimipour, A. M. Rahmani, A. H. Navin, and M. Hosseinzadeh, "Resource discovery mechanisms in grid systems: A survey," *J. Netw. Comput. Appl.*, vol. 41, pp. 389–410, May 2014.

[24] R. Alimi, Y. Yang, and R. Penno, *Application-Layer Traffic Optimization (ALTO) Protocol*, document RFC 7285, 2014.

[25] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4P: Provider portal for applications," *ACM SIGCOMM*, vol. 38, no. 4, pp. 351–362, Aug. 2008.

[26] J. Lee *et al.*, "Application-driven bandwidth guarantees in datacenters," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 467–478, 2014.

[27] H. Zhang *et al.*, "Guaranteeing deadlines for inter-data center transfers," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 579–595, Feb. 2017.

[28] V. Jalaparti, I. Bliznets, S. Kandula, B. Lucier, and I. Menache, "Dynamic pricing and traffic engineering for timely inter-datacenter transfers," in *Proc. ACM SIGCOMM Conf.*, 2016, pp. 73–86.

[29] D. Perez and C. Rothenberg, "ALTO-based broker-assisted multi-domain orchestration," in *Proc. IETF Draft*, Mar. 2019.

[30] H. Newman *et al.*, "Next-generation exascale network integrated architecture for global science," *J. Opt. Commun. Netw.*, vol. 9, no. 2, pp. A162–A169, 2017.

[31] I. Monga *et al.*, "SDN for end-to-end networked science at the exascale (SENSE)," in *Proc. IEEE/ACM INDIS*, Nov. 2018, pp. 33–44.

[32] V. Welch, M. Thompson, D. E. Engert, S. Tuecke, and L. Pearlman, *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*, document RFC 3820, RFC Editor, Jun. 2004, p. 37. [Online]. Available: https://rfc-editor.org/rfc/rfc3820.txt. doi: 10.17487/RFC3820.

[33] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, "Open ID connect core 1.0," OpenID, San Ramon, CA, USA, Tech. Rep., Nov. 2014.

[34] OSST Committee. (2012). *Security Assertion Markup Language (SAML) 2.0*. [Online]. Available: http://www.oasis-open.org/committees/tchome.php

[35] Y. Rekhter, S. Hares, and D. T. Li, *A Border Gateway Protocol 4 (BGP-4)*, document RFC 4271, Jan. 2006. [Online]. Available: https://rfc-editor.org/rfc/rfc4271.txt

[36] *Route Views Project*. Accessed: Dec. 2018. [Online]. Available: http://www.routeviews.org/routeviews/

[37] V. Heorhiadi, M. K. Reiter, and V. Sekar, "Simplifying software-defined network optimization using SOL," in *Proc. NSDI*, 2016, pp. 223–237.

[38] B. Hindman *et al.*, "Mesos: A platform for fine-grained resource sharing in the data center," in *Proc. NSDI*, 2011, p. 22.

[39] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in *Proc. ACM EuroSys*, 2015, p. 18.

[40] W. Stallings, "The advanced encryption standard," *Cryptologia*, vol. 26, no. 3, pp. 165–188, Jul. 2002. doi: 10.1080/0161-110291890876.

[41] K. Gao, C. Gu, Q. Xiang, X. Wang, Y. R. Yang, and J. Bi, "ORSAP: Abstracting routing state on demand," in *Proc. IEEE ICNP*, Nov. 2016, pp. 1–2.

[42] K. Gao, Q. Xiang, X. Wang, Y. R. Yang, and J. Bi, "NOVA: Towards on-demand equivalent network view abstraction for network optimization," in *Proc. ACM/IEEE IWQoS*, Jun. 2017, pp. 1–10.

[43] J. Telgen, "Identifying redundant constraints and implicit equalities in systems of linear constraints," *Manage. Sci.*, vol. 29, no. 10, pp. 1209–1222, 2002.

[44] M. Raykova, *Secure Computation in Heterogeneous Environments: How to Bring Multiparty Computation Closer to Practice?*. Columbia Univ., 2012.

[45] A. C.-C. Yao, "How to generate and exchange secrets," in *IEEE FOCS 1986*.

[46] X. Feng and Z. Zhang, "The rank of a random matrix," *Appl. Math. Comput.*, vol. 185, no. 1, pp. 689–694, Jan. 2007.

[47] O. L. Mangasarian, "Privacy-preserving horizontally partitioned linear programs," *Optim. Lett.*, vol. 6, no. 3, pp. 431–436, 2012.

[48] *The Opendaylight Project*. Accessed: Dec. 2018. [Online]. Available: https://www.opendaylight.org

[49] The CMS Collaboration *et al.*, "The CMS experiment at the CERN LHC," *J. Instrum.*, vol. 3, no. 8, pp. S08004–S08004, 2008.

[50] Q. Xiang, C. Guok, F. Le, J. MacAuley, H. Newman, and Y. R. Yang, "SFP: Toward interdomain routing for SDN networks," in *Proc. ACM SIGCOMM Conf. Posters Demos*, 2018, pp. 87–89.

[51] *A Demonstration of Mercator*. Accessed: Dec. 2018. [Online]. Available: https://youtu.be/kUK78gHlQDI

[52] (2016). *The CAIDA AS Relationships Dataset*. [Online]. Available: http://www.caida.org/data/as-relationships/

[53] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011.

[54] *CMS Task Monitoring*. Accessed: Dec. 2018. [Online]. Available: http://dashb-cms-job.cern.ch/

[55] R. Jain, D.-M. Chiu, and W. R. Hawe, "A quantitative measure fairness discrimination for resource allocation shared computer system," Eastern Res. Lab., Digit. Equip. Corp., Hudson, MA, USA, Tech. Rep., 1984, vol. 38.

[56] J. Y. Boudec, "Rate adaptation, congestion control and fairness: A tutorial," EPFL, Lausanne, Switzerland, Tech. Rep., Oct. 2000.

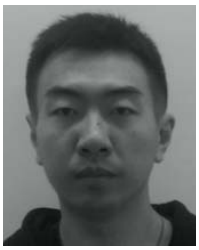
[57] (2018). *Global Ping Statistics—WonderNetwork*. [Online]. Available: https://wondernetwork.com/pings/

[58] *Python Cryptography Toolkit*. Accessed: Dec. 2018. [Online]. Available: https://pypi.python.org/pypi/pycrypto

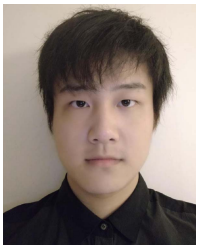
[59] *Under the Hood: Scheduling MapReduce Jobs More Efficiently With Corona*. Accessed: May 9, 2017. [Online]. Available: http://on.fb.me/TxUsYN
[60] E. Boutin *et al.*, "Apollo: Scalable and coordinated scheduling for cloud-scale computing," in *Proc. OSDI*, 2014, pp. 285–300.
[61] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair scheduling for distributed computing clusters," in *Proc. IEEE Int. Conf. Recent Trends Inf. Syst.*, 2009, pp. 261–276.
[62] Q. Pu *et al.*, "Low latency geo-distributed data analytics," in *Proc. ACM SIGCOMM*, 2015, pp. 421–434.
[63] R. Viswanathan, G. Ananthanarayanan, and A. Akella, "CLARINET: WAN-aware optimization for analytics queries," in *Proc. Usenix Conf. Operating Syst. Design Implement.*, 2016, pp. 435–450.
[64] A. Vulimiri, C. Curino, B. Godfrey, K. Karanasos, and G. Varghese, "WANalytics: Analytics for a geo-distributed data-intensive world," in *Proc. CIDR*, 2015.
[65] Q. Xiang *et al.*, "Unicorn: Unified resource orchestration for multi-domain, geo-distributed data analytics," in *Proc. IEEE SmartWorld, DAIS Workshop*, Aug. 2017, pp. 1–6.
[66] Q. Xiang, X. Wang, J. Zhang, H. Newman, Y. R. Yang, and Y. J. Liu, "Unicorn: Unified resource orchestration for multi-domain, geo-distributed data analytics," in *Proc. IEEE INDIS Workshop*, 2017.

**Qiao Xiang** received the bachelor's degree in information security and the bachelor's degree in economics from Nankai University in 2007, and the master's and Ph.D. degrees in computer science from Wayne State University in 2012 and 2014, respectively. From 2016 to 2018, he was a Post-Doctoral Fellow with the Department of Computer Science, Yale University. From 2014 to 2015, he was a Post-Doctoral Fellow with the School of Computer Science, McGill University. He is currently an Associate Research Scientist with the Department of Computer Science, Yale University. His research interests include software defined networking, resource discovery and orchestration in collaborative data sciences, interdomain routing, and wireless cyber-physical systems.

**Jingxuan Jensen Zhang** received the bachelor's degree in engineering from the Department of Computer Science and Engineering, Tongji University, in 2015. He is currently Visiting Ph.D. with the Department of Computer Science, Yale University. His research focuses on network resource discovery, abstraction and programming consistency for large-scale data analytics systems. He is also an Active Member of the IETF ALTO Working Group and the OpenDaylight Open Source Community.



**Xin Tony Wang** received the bachelor's degree in engineering from the Department of Computer Science and Engineering, Tongji University, in 2014. He is currently Visiting Ph.D. with the Department of Computer Science, Yale University. His research interests include software defined networking, inter-domain routing, and distributed computing.



**Yang Jace Liu** received the bachelor's degree in engineering from the Department of Computer Science and Engineering, Tongji University, in 2017. He is currently pursuing the degree with the Computer Science Department, University of Calgary, Canada. His research interests include software defined networking, large-scale data analytics systems, and high-performance computing.

**Chin Guok** received the B.S. degree in computer science from the University of Pacific in 1991 and the M.S. degree in computer science from The University of Arizona in 1997. He joined ESnet in 1997 as a Network Engineer, where he is focusing primarily on network statistics. He was a Core Engineer in the testing and production deployment of MPLS and QoS (Scavenger Service) within ESnet. He is the Technical Lead of the ESnet On-demand Secure Circuits and Advanced Reservation System (OSCARS) Project which enables end-users to provision guaranteed bandwidth virtual circuits within ESnet. He also serves as a Co-Chair for the Open Grid Forum On-Demand Infrastructure Service Provisioning Working Group. His research interests include high-performance networking and network protocols, dynamic network resource provisioning, network tuning issues, and hybrid network traffic engineering.

**Franck Le** received the Diplome d'Ingenieur from the Ecole Nationale Superieure des Telecommunications de Bretagne in 2000 and the Ph.D. degree from Carnegie Mellon University in 2010. He is currently a Research Staff Member with the IBM Thomas J. Watson Research Center. His current research interests lie at the intersection of Internet of Things, artificial intelligence, and distributed systems & networks.

**John MacAuley** is currently a Chief Software Architect with Energy Sciences Network. His main research interests include high-speed computer networks and systems, resource discovery, and orchestration in science networks.

**Harvey Newman** received the Sc.D. degree from MIT in 1974. From 1973 to 1974, he co-led the team that discovered fourth quark flavor known as charm. He co-led the MARK J Collaboration that discovered the gluon, the carrier of the strong force in 1979. Since 1982, he has been a Faculty Member with the California Institute of Technology (Caltech), where he is currently the Marvin L. Goldberger Professor of physics. He has been leading a role in originating, developing, and operating state of the art international networks and collaborative systems serving the high energy and nuclear physics communities since 1982. He served on the IETF and the Technical Advisory Group that led to the NSFNet from 1985 to 1986, originated the worldwide LHC Computing Model in 1996, and has been leading the science and network engineering teams defining the state of the art in long distance data transfers since 2002. Since 1994, he has been a member of CMS that discovered the Higgs boson at LHC in 2012.

**Y. Richard Yang** received the B.E. degree in computer science and technology from Tsinghua University in 1993, and the M.S. and Ph.D. degrees in computer science from The University of Texas at Austin in 1998 and 2001, respectively. He is currently a Professor of computer science and electrical engineering with Yale University. His research is supported by both U.S. government funding agencies and leading industrial corporations, and spans areas including computer networks, mobile computing, wireless networking, and network security. His work has been implemented/adopted in products/systems of major companies (e.g., AT&T, Alcatel-Lucent, Cisco, Google, Microsoft, and Youku), and featured in mainstream media including Economist, Forbes, Guardian, Chronicle of Higher Education, Information Week, MIT Technology Review, Science Daily, USA Today, Washington Post, and Wired, among others. His awards include a CAREER Award from the National Science Foundation and a Google Faculty Research Award.

# Optimizing in the Dark: Learning an Optimal Solution through a Simple Request Interface

**Qiao Xiang,**[1,2*] **Haitao Yu,**[1] **James Aspnes,**[2] **Franck Le,**[3] **Linghe Kong,**[4] **Y. Richard Yang**[1,2]
[1]Tongji University, [2]Yale University, [3]IBM T.J. Watson Research Center, [4]Shanghai Jiao Tong University
qiao.xiang@cs.yale.edu, haitao.yu@tongji.edu.cn, james.aspnes@yale.edu,
fle@us.ibm.com, linghe.kong@sjtu.edu.cn, yry@cs.yale.edu

## Abstract

Network resource reservation systems are being developed and deployed, driven by the demand and substantial benefits of providing performance predictability for modern distributed applications. However, existing systems suffer limitations: They either are inefficient in finding the optimal resource reservation, or cause private information (*e.g.*, from the network infrastructure) to be exposed (*e.g.*, to the user). In this paper, we design BoxOpt, a novel system that leverages efficient oracle construction techniques in optimization and learning theory to automatically, and swiftly learn the optimal resource reservations without exchanging any private information between the network and the user. We implement a prototype of BoxOpt and demonstrate its efficiency and efficacy via extensive experiments using real network topology and trace. Results show that (1) BoxOpt has a 100% correctness ratio, and (2) for 95% of requests, BoxOpt learns the optimal resource reservation within 13 seconds.

## 1 Introduction

*When facing a genie that only tells you whether it can grant a wish or not, how can you find the best wish it can grant?*

Although the question may sound like one from fairy tales, people deal with such question in many real world scenarios. For example, modern distributed applications (*e.g.*, (Zaharia et al. 2012; White 2012)) construct complex data flows between end hosts, *e.g.*, in data center networks. The key to supporting these applications is the ability to provide guaranteed network resources (*i.e.*, bandwidth) for performance predictability (Mogul and Popa 2012). As such, many network resource reservation systems are developed and deployed (Campanella et al. 2006; Guok and Robertson 2006; Johnston, Guok, and Chaniotakis 2011; Riddle 2005; Zheng et al. 2005; Sobieski, Lehman, and Jabbari 2004). However, because of the underlying networks' concern of revealing sensitive information, existing reservation systems do not provide applications with an interface to access information of the underlying network infrastructure (*e.g.*, topology, links' available bandwidth). Instead, networks only offer a *simple reservation interface* for applications to submit requests for reserving a specific amount of bandwidths for a set of flows: $\mathrm{request}(\mathrm{flow\_set}, \mathrm{bw\_values})$), and returns either success or failure. A major concern of this design is its inefficiency for the applications/users to find the optimal amount of network resources to reserve. To further illustrate the issues, consider the example in Figure 1, where a user (*e.g.*, application) wants to determine and reserve the maximum achievable bandwidth for two flows from $S_1$ to $D_1$, and $S_2$ to $D_2$, respectively. Using existing solutions that offer only a *simple reservation interface*, finding the constraint that both flows can collectively get only 100 Mbps of bandwidth is already an instance of the NP-hard membership-query based constraint acquisition problem (Bessiere et al. 2017), letting alone finding the optimal reservation for both flows (*e.g.*, 50 Mbps for each flow).

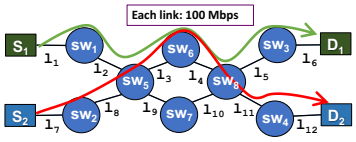



Figure 1: An example network topology: the routes of two flows share bottleneck links, *i.e.*, $l_3$ and $l_4$, hence they can only collectively get a 100 Mbps bandwidth.

To address this problem, researchers have proposed several solutions, but all of them suffer limitations, and violate privacy requirements. For example, to determine optimal bandwidth reservations, recent proposals depart from the simple reservation interface, and require either networks to reveal sensitive information to users (Soulé et al. 2014; Subramanian, D'Antoni, and Akella ; Heorhiadi, Reiter, and Sekar 2016; Lee et al. ), or vice versa (Gao et al. 2016; Gao et al. 2017; Xiang et al. 2018). These solutions are therefore limited to settings where the level of trust between the applications and the underlying network is high. These solutions cannot be deployed in general settings as malicious parties may use the exposed network information to identify vulnerable links and launch attacks (*e.g.*, DDoS).

In this paper, we explore the feasibility and benefits of learning the optimal network resource reservation for the user without exposing the private information of the network (*i.e.*, bandwidth capacity region) and the user (*e.g.*, resource orchestration policy) to each other. In particular, we tackle the following question: *How can a user learn the optimal*

*The corresponding authors are Q.Xiang and Y. R. Yang.

*network resource reservation using only the simple reservation interface?* This task is non-trivial due to the extremely limited feedback (*i.e.*, success/failure) provided by simple reservation interface.

Our solution to this problem is BoxOpt, a novel learning system that automatically, and efficiently learns the optimal resource reservations for the user through the simple reservation interface, without exchanging any private information between the network and the user (*e.g.*, bandwidth feasible region of the network and the resource orchestration policy of the user). Specifically, BoxOpt allows users to include their resource reservation objectives as concave utility functions of the requested resources (*e.g.*, bandwidths) in the reservation requests. Upon receiving a reservation request, BoxOpt models the simple reservation interface of network resource reservation systems as a membership oracle over a polytope. It then expands oracle construction techniques (Lovasz, Grotschel, and Schrijver 1993; Lee, Sidford, and Vempala 2017) from optimization and learning theory to construct a separation oracle through invoking the membership oracle in near $O(n)$ iterations ($n$ being the number of flows), which when called upon will accurately infer a search space in which the optimal reservation vector lies. With such a separation oracle, BoxOpt then constructs an optimization oracle based on ellipsoid method, which can learn the optimal reservation vector through $O(n^2)$ calls on the separation oracle.[1] In this way, BoxOpt not only can learn the optimal resource reservation efficiently, but also is privacy-preserving in that no private information is exchanged between the user and the network.

The **main contributions** of this paper are as follows:

• We study the important problem of learning the optimal network resource reservation through the simple reservation interface of network resource reservation systems. In particular, we design BoxOpt, a novel, fast, automatic, privacy-preserving learning system. To the best of our knowledge, BoxOpt is the first working system that solves this problem, and can be extended to other optimization problems.

• We model the simple reservation interface as a membership oracle over a polytope, and expand oracle construction techniques from optimization and learning theory to develop an efficient optimization oracle in BoxOpt, which learns the optimal resource reservation in near $O(n^3)$ of calls on the membership oracle.

• We implement a prototype of BoxOpt and demonstrate both its efficiency and efficacy through extensive experiments using real topologies and traces. Results show that (1) BoxOpt has a 100% correctness ratio, and (2) for 95% cases, it can learn the optimal reservation within 13 seconds.

The remaining of this paper is organized as follows. We present an overview of BoxOpt in Section 2. We give details on how BoxOpt efficiently learns the optimal network resource reservation only using the simple reservation interface in Section 3. We present the evaluation results of BoxOpt in Section 4. We discuss related work in Section 5 and conclude the paper in Section 6.

[1] We choose the ellipsoid method because it is a classic cutting plane method. However, the design of BoxOpt is modular and other cutting plane methods, *e.g.*, analytic center method (Atkinson and Vaidya 1995) and random walk (Bertsimas and Vempala ), can also be used to construct an optimization oracle from a separation oracle.

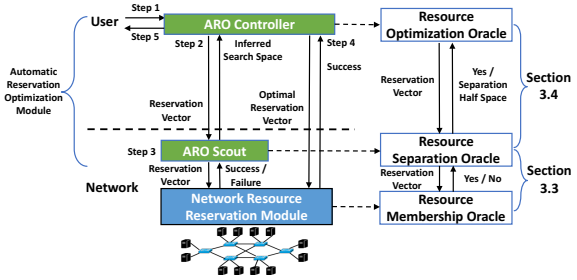



Figure 2: The architecture and workflow of BoxOpt.

## 2 Overview of BoxOpt

In this section, we first present the architecture and the workflow of BoxOpt. We then give a formal, mathematical formulation of the key technical challenge in BoxOpt: how to find the optimal network resource reservation through the simple reservation interface.

### 2.1 Architecture

BoxOpt is composed of two components: an automatic reservation optimization (ARO) module for the user, and a network resource reservation (NRR) module for the network (Figure 2). The two components interact with each other through the simple reservation interface commonly used in traditional network resource reservation systems.

**Automatic reservation optimization module**: The ARO module is a private component belonging to the user, and is composed of two sub-components: an ARO controller, and an ARO scout.

The ARO controller is the main interface for the user to submit the resource reservation requests. A request consists of a set of $n$ flows, $F = \{f_1, f_2, \ldots, f_n\}$, to reserve the resources for, and a concave utility function $util(\mathbf{x})$ to maximize, with $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ and each $x_i$ representing the available bandwidth that can be reserved for flow $f_i \in F$. Example utility functions include total throughput and priority-based total throughput. Given a user resource reservation request, the objective of the ARO controller is to infer the optimal resource reservation to maximize $util(\mathbf{x})$. The ARO controller achieves it with the assistance of the ARO scout: Specifically, the ARO controller iteratively selects a vector $\check{\mathbf{x}}$ of bandwidth values for $F$ (called *reservation vector*) and sends it to the ARO scout. For each reservation vector, the ARO scout returns a search space where the optimal reservation vector lies in. With the inferred search spaces returned by the ARO scout, the ARO controller gradually converges to the optimal reservation vector that maximizes $util(\mathbf{x})$.

The ARO scout is the main user entity interacting with the NNR. For each $\check{\mathbf{x}}$ from the ARO controller, the ARO scout infers a search space where the optimal reservation

vector lies in, and returns the inferred search space back to the ARO controller. To infer the search space where the the optimal reservation vector lies in, the ARO scout sends a sequence of reservation vectors to the NRR through the simple reservation interface. As further described in Section 3 and Section 4, for each reservation vector submitted from the ARO controller, the ARO scout might submit tens or hundreds of reservation vectors to the NNR to get an accurately-inferred search space, potentially, leading to a high overhead. As such, to reduce the total latency to find the optimal reservation vector, the ARO scout is placed with the network instead of the user. This design decision reduces the user-network communication latency by 20x as demonstrated in the evaluation section. More importantly, this design does not expose the private information of the user (*i.e.*, $util(\mathbf{x})$) to the network, as the ARO controller does not send such information to the scout.

**Network Resource Reservation Module**: The NRR module is a private component belonging to the network. Its primary role is to verify whether the reservation vectors submitted by the ARO scout can be satisfied. Upon receiving a reservation vector from the ARO scout, the NRR extracts the relevant constraints from the network. The constraints include both physical network constraints (*e.g.*, if two flows share a same link, their allocated bandwidths cannot exceed the link's available bandwidth), and network policies (*e.g.*, rate limiting, etc.) The constraints are captured as an abstraction of linear inequalities (Gao et al. 2016; Xiang et al. 2018). For example, to capture the physical network constraints, the NNR first retrieves the routes (i.e., sequence of traversed links) for each flow. Then, for each link $l$ in the network, the NNR generates the following linear inequality to ensure that the allocated bandwidths to the flows do not exceed the link's available bandwidth:

$$\sum x_i \leq w_l, \forall f_i \text{ that uses } l \text{ in this route},$$

where $w_l$ is the available bandwidth on link $l$. Considering the example in Figure 1, the NRR module generates the following linear inequalities:

$$\begin{array}{ll} x_1 \leq 100 & \forall l_u \in \{l_1, l_2, l_5, l_6\}, \\ x_2 \leq 100 & \forall l_u \in \{l_7, l_8, l_{11}, l_{12}\}, \\ x_1 + x_2 \leq 100 & \forall l_u \in \{l_3, l_4\}, \\ x_1, x_2, x_3 \geq 0. & \end{array} \quad (1)$$

Then, the NRR generates additional linear inequalities to represent the network's internal traffic engineering policies, such as load-balancing and bandwidth limiting. For example, suppose the network wants to limit the total bandwidth of flows $f_1$ and $f_2$ to be no more than 80 Mbps even if there is no common link in their routes. Then a linear inequality $x_1 + x_2 + x_3 \leq 80$ is generated to represent this policy. Geometrically, the abstraction of linear inequalities represents the *bandwidth feasible region* of the network for providing bandwidths to a set of flows.

Finally, for each generated linear inequality, the NRR checks if it is satisfied by the bandwidth values specified in the reservation vector. If any inequality is violated, it returns a FAILURE signal. Otherwise, it returns SUCCESS.

## 2.2 Workflow

Having presented the basic components of BoxOpt, we now briefly present its workflow to automatically compute and reserve the optimal network resources for a set of flows as follows (Figure 2):

• Step 1: The user submits a resource reservation request for a set of flows $F$ to the ARO controller. The request also includes a concave utility function $util(\mathbf{x})$ of the bandwidths of $F$.

• Step 2: In an outer loop, the ARO controller iteratively selects reservation vectors to send to the ARO scout. The selection of the reservation vectors is described in Section 3.4, Algorithm 3. In return, for each reservation vector, the ARO scout determines and replies with an inferred search space.

• Step 3: In an inner loop, upon receiving a reservation vector from the controller, the ARO scout interacts with the NRR, according to Algorithm 1 from Section 3.3, to infer the next search space and send it back to the ARO controller.

The nested iteration of Step 2 and 3 stops when the ARO controller converges to the optimal reservation vector maximizing $util(\mathbf{x})$.

• Step 4: The ARO controller sends the optimal reservation vector to the NRR module to reserve the optimal resources for the user.

• Step 5: The ARO controller confirms with the user that the optimal network resource reservation has been successful.

## 2.3 Key Challenge

Through the introduction of its architecture and workflow, we show that BoxOpt is *privacy-preserving* by design: neither the user nor the network exposes the private information (*i.e.*, internal optimization objective of the user and the bandwidth capacity region of the network) to the other party. As such, the remaining key challenge for BoxOpt lies in Step 2 and 3: *how can the ARO module interact with the NRR module through the simple reservation interface to compute the optimal network resource reservation?*. To address this challenge, we first give a formal, mathematical formulation.

Specifically, we first model the NRR module as a *resource membership oracle*. Without loss of generality, we use $\mathbf{Ax} \leq \mathbf{b}$ to denote the set of linear inequalities generated by the NRR module, and use $K : \{\mathbf{x} | \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ to represent the bandwidth feasible region for a set of flows $F$. In this way, we give the definition of resource membership oracle:

**Definition 1.** *[Reservation Membership Oracle (ReMEM)] Given a reservation vector $\check{\mathbf{x}}$, return YES if $\check{\mathbf{x}} \in K$, and return NO otherwise.*

$ReMEM(\check{\mathbf{x}})$ accurately captures the interaction between the ARO scout and the NRR module. Next, we formally define the problem of network resource reservation optimization via simple reservation interface.

**Problem 1** (Optimization via Membership Oracle)**.** *Find the optimal solution to the following optimization problem*

$$\textit{maximize} \quad util(\mathbf{x}), \quad (2)$$

*subject to,*

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}, \tag{3}$$

$$\mathbf{x} \geq \mathbf{0}, \tag{4}$$

*without the knowledge of* $\mathbf{A}$ *and* $\mathbf{b}$*, but only using ReMEM defined in Definition 1.*

Maximizing $util(\mathbf{x})$ subject to $K : \{\mathbf{x}|\mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is a classic convex optimization problem. There has been a rich body of literature on how to efficiently solve such problems (Boyd and Vandenberghe 2004). However, most of the existing algorithms require the knowledge of the feasible region (in our case $\mathbf{A}\mathbf{x} \leq \mathbf{b}$). One may think of a strawman to learn $K$ through the ReMEM oracle, and apply the standard optimization techniques to find the optimal $\mathbf{x}$. However, finding the feasible region through a membership oracle is NP-hard (Bessiere et al. 2017), making this strawman impractical. In contrast, as we will present next, BoxOpt resorts to efficient oracle transformation techniques in optimization and learning theory (Lee, Sidford, and Vempala 2017; Lovasz, Grotschel, and Schrijver 1993) to solve this problem (*i.e.*, efficiently learn the optimal resource reservation via membership oracle).

# 3 Optimizing Network Resource Reservation via Simple Reservation Interface

Having formally defined the key challenge for BoxOpt as a problem of optimization via membership oracle, this section discusses how we solve this problem. For presentation clarity, this section starts by reviewing some concepts in optimization theory. Then, it presents the basic idea of our solution, followed by its details.

## 3.1 Notations

Unless explicitly noted, we use $v$ to denote a scalar and $\mathbf{v}$ to denote a vector of $n$ dimensions, where $n$ is the number of flows the user wants to reserve bandwidth for (see Section 2.1). We use $\|\mathbf{v}\|_2 = \sqrt{\sum v_i^2}$ to denote the Euclidean norm of $\mathbf{v}$, and use $\|\mathbf{v}\|_\infty = \max|v_i|$ to denote the maximum norm of $\mathbf{v}$. We use $B_2^+(\mathbf{m}, \eta) = \{\mathbf{x}|\|\mathbf{x} - \mathbf{m}\|_2 \leq \eta, \mathbf{x} \geq \mathbf{0}\}$ to denote the set of all positive vectors whose Euclidean distance to $\mathbf{m}$ is at most $\eta$, and use $B_\infty^+(\mathbf{m}, \eta) = \{\mathbf{x}|\|\mathbf{x} - \mathbf{m}\|_\infty \leq \eta, \mathbf{x} \geq \mathbf{0}\}$ to denote the set of all positive vectors whose maximum norm distance to $\mathbf{m}$ is at most $\eta$.

## 3.2 Basic Idea

Our approach to solve Problem 1 utilizes the equivalence and polar relationships between different oracles in optimization theory (Lovasz, Grotschel, and Schrijver 1993). In particular, we focus on the relationships between ReMEM with the following two oracles:

**Definition 2.** *[Resource Separation Oracle (ReSEP)] Given a reservation vector* $\check{\mathbf{x}}$*, return YES if* $\check{\mathbf{x}} \in K$*, and otherwise return a half space* $\{\mathbf{y}|\mathbf{p}^T(\mathbf{y} - \check{\mathbf{x}}) \leq \delta\}$ *that contains* $K$ *but not* $\check{\mathbf{x}}$*.*

**Definition 3.** *[Resource Optimization Oracle (ReOPT)] Given a reservation request for a set of flows* $F$ *and the utility function* $util(\mathbf{x})$*, find* $\mathbf{x}^* \in K$ *that maximizes* $util(\mathbf{x})$*.*

**Algorithm 1:** Resource Separation Oracle $ReSEP(\check{\mathbf{x}})$

```
Select ε ∈ (0, r], ρ ∈ (0, 1);
κ ← R/r;
if ReMEM returns YES for x̌ then
    return x̌ ∈ K ;
else if x̌ ∉ B_2(0, R) then
    return the half space {y|x̌^T(y − x̌) ≤ 0};
else
    r_1 ← rR^{−1/3} ε^{1/3};
    g̃ ← Subgradient(0, r_1, 4ε, 3κ);
    return the half space
      {y|g̃^T(y − x̌) ≤ (40n + 1)ρ^{−1} R^{2/3} κε^{1/3}};
```

Given that there exist efficient algorithms (*e.g.*, ellipsoid method) that can construct an optimization oracle through invoking a separation oracle with a polynomial number of iterations, if we can construct a separation oracle through a polynomial number of calls to a membership oracle, we will be able to solve Problem 1.

One may think classic half space learning techniques can achieve such a construction of separation oracle via membership oracle. However, the problems are different. In half space learning, the goal is to compute a hyperplane to separate a set of given samples (in our case, the reservation vectors) from two predefined classes. In contrast, the goal of ReMEM to ReSEP construction is to compute a hyperplane separating $K$ and a reservation vector not belonging to $K$ by strategically choosing a minimal number of reservation vectors to send to ReMEM.

Specifically, we develop our solution to Problem 1 in two phases. First, we leverage recent progress on geometric algorithms (Lee, Sidford, and Vempala 2017) to develop an efficient algorithm that constructs ReSEP through invoking ReMEM for a polynomial number of times. Specifically, our algorithm expands the weak membership/separation oracle construction in (Lee, Sidford, and Vempala 2017) to strong membership/separation oracle construction. Second, we develop an ellipsoid-method-based algorithm that constructs ReOPT through local feasibility checks and invoking ReSEP for a polynomial number of times. Mapping these two phases to Step 2 and 3 in the workflow of BoxOpt, we see that the ARO scout is essentially the separation oracle ReSEP, and the ARO controller is the optimization oracle ReOPT (Figure 2). Next, we give the details of each phase.

## 3.3 From Resource Membership Oracle to Resource Separation Oracle

To construct ReSEP from ReMEM, we first define two auxiliary functions on vector $\mathbf{d} \in K$ given a reservation vector $\check{\mathbf{x}}$:

$$\alpha_{\check{\mathbf{x}}}(\mathbf{d}) \leftarrow \max_{\mathbf{d}+\alpha\check{\mathbf{x}} \in K} \alpha, \tag{5}$$

$$h_{\check{\mathbf{x}}}(\mathbf{d}) \leftarrow -\alpha_{\check{\mathbf{x}}}(\mathbf{d})\|\check{\mathbf{x}}\|_2 \tag{6}$$

We see that given a vector $\mathbf{d} \in K$, $\mathbf{d} + \alpha_{\check{\mathbf{x}}}(\mathbf{d})$ is the last vector on the line from $\mathbf{d}$ to $\mathbf{d} + \check{\mathbf{x}}$ that is in $K$, and that $-h_{\check{\mathbf{x}}}(\mathbf{d})$

**Algorithm 2:** Computing the subgradient of $h_{\check{\mathbf{x}}}(\mathbf{d})$ $Subgradient(\mathbf{d}, r_1, \tau, L)$

1 $r_2 \leftarrow \sqrt{\frac{\tau r_1}{\sqrt{n}L}}$;
2 Randomly select $\mathbf{y}$ from $B_\infty(\mathbf{d}, r_1)$ following a uniform distribution;
3 Randomly select $\mathbf{z}$ from $B_\infty(\mathbf{y}, r_2)$ following a uniform distribution;
4 **for** $i \leftarrow 1, \ldots, n$ **do**
5 Define line segment $B_\infty(\mathbf{y}, r_2) \cap \mathbf{z} + s\mathbf{e_i}$, where $s \in R$ and $\mathbf{e_i}$ is a vector whose elements are all zeros except the $i$th one;
6 Denote the endpoints of this line segment as $\mathbf{s_i}$ and $\mathbf{t_i}$, respectively;
7 Evaluate $h_{\check{\mathbf{x}}}(\mathbf{t_i})$ and $h_{\check{\mathbf{x}}}(\mathbf{s_i})$ using binary search and ReMEM;
8 $\tilde{g}_i = \frac{h_{\check{\mathbf{x}}}(\mathbf{t_i}) - h_{\check{\mathbf{x}}}(\mathbf{s_i})}{2r_2}$;
9 **return** $\tilde{\mathbf{g}}$;

is the Euclidean distance from $d$ to this point. Without loss of generality, we assume that $B_2^+(\mathbf{0}, r) \subset K \subset B_2^+(\mathbf{0}, R)$ for some positive numbers $r, R$ and such an assumption can be trivially satisfied in practice. Extending the proof technique in (Lee, Sidford, and Vempala 2017) for an $n$-dimensional ball to only a partial ball on the first orthant, we get

**Lemma 1.** *Given $\check{\mathbf{x}}$, $h_{\check{\mathbf{x}}}(\mathbf{d})$ is convex on $K$, and is $\frac{R+\theta}{R-\theta}$ Lipschitz in $B_2^+(\mathbf{0}, \theta)$ for $0 < \theta < r$.*

With these auxiliary functions and a theorem that for any Lipschitz function, it is linear on a small ball (Bubeck and Eldan 2016), we can construct ReSEP by computing the subgradient of $h_{\check{\mathbf{x}}}(\mathbf{d})$ at $\mathbf{d} = \mathbf{0}$, which can be computed by binary search and invoking ReMEM. The constructed separation oracle is presented in Algorithm 1, and the computation of the subgradient of $h_{\check{\mathbf{x}}}(\mathbf{d})$ is presented in Algorithm 2.

A key insight in Algorithm 1 is that it expands the applicability of similar construction process from weak membership/separation oracles to strong membership/separation oracles (*i.e.*, ReMEM and ReSEP). In particular, we have

**Lemma 2.** *There is a random variable $\phi$ with expectation $E(\phi) \leq 2n\sqrt{\frac{\tau L}{r_1}}$ such that $\forall \mathbf{q} \in K$,*

$$h_{\check{\mathbf{x}}}(\mathbf{q}) \geq h_{\check{\mathbf{x}}}(\mathbf{d}) + \tilde{\mathbf{g}}^T(\mathbf{q} - \mathbf{d}) - \phi\|\mathbf{q} - \mathbf{d}\|_\infty - 4nr_1L.$$

With this lemma, we show the correctness of Algorithm 1 in the following theorem.

**Theorem 1.** *If $\check{\mathbf{x}} \notin K$, Algorithm 1 yields a half space containing $K$ but not $\check{\mathbf{x}}$ with probability $1 - \rho$.*

*Proof.* When $\check{\mathbf{x}} \notin B_2^+(\mathbf{0}, R)$, it is easy to see that the returned half space $\{\mathbf{y}|\check{\mathbf{x}}(\mathbf{y}) - \check{\mathbf{x}} \leq \mathbf{0}\}$ (Line 6 of Algorithm 1) contains $K$ but not $\check{\mathbf{x}}$. When $\check{\mathbf{x}} \notin K$ but $\check{\mathbf{x}} \in B_2^+(\mathbf{0}, R)$, from Lemma 1, we know that $h_{\check{\mathbf{x}}}(\mathbf{d})$ has a Lipschitz constant of $3\kappa$ on $B_2^+(\mathbf{0}, \frac{r}{2})$. By selecting $\epsilon \in (0, r]$ and setting $r_1 = rR^{-\frac{1}{3}}\epsilon^{\frac{1}{3}}$ (Line 1 and 8 of Algorithm 1, respectively), we can get $B_\infty^+(\mathbf{0}, 2r_1) \subset B_2^+(\mathbf{0}, \frac{r}{2})$. As such, we can apply Lemma 2 and get that $\forall \mathbf{q} \in K$

$$h_{\check{\mathbf{x}}}(\mathbf{q}) \geq h_{\check{\mathbf{x}}}(\mathbf{0}) + \tilde{\mathbf{g}}^T \cdot \mathbf{q} - \phi\|\mathbf{q}\|_\infty - 12nr_1\kappa. \quad (7)$$

Next, because $\check{\mathbf{x}} \in B_2^+(\mathbf{0}, R)$, we have $-\frac{1}{\kappa}\check{\mathbf{x}} \in K$ and $h_{\check{\mathbf{x}}}(-\frac{1}{\kappa}\check{\mathbf{x}}) = h_{\check{\mathbf{x}}}(\mathbf{0}) - \frac{\|\check{\mathbf{x}}\|_2}{\kappa}$. Then we use Lemma 2 to get

$$h_{\check{\mathbf{x}}}(-\frac{1}{\kappa}\check{\mathbf{x}}) \geq h_{\check{\mathbf{x}}}(\mathbf{0}) + \tilde{\mathbf{g}}^T \cdot -\frac{1}{\kappa}\check{\mathbf{x}} - \frac{\phi}{\kappa}\|\check{\mathbf{x}}\|_\infty - 12nr_1\kappa, \quad (8)$$

As such, we then get

$$\tilde{\mathbf{g}}^T \cdot \check{\mathbf{x}} \geq \|\check{\mathbf{x}}\|_2 - \phi\|\check{\mathbf{x}}\|_\infty - 12nr_1\kappa^2. \quad (9)$$

Next, because $\epsilon \in (0, r]$, $\check{\mathbf{x}} \notin K$ and $B_2^+(\mathbf{0}, r) \subset K$, we have $(1 - \frac{\epsilon}{r})K \subset K$. By definition of $h_{\check{\mathbf{x}}}(\mathbf{d})$, we have

$$h_{\check{\mathbf{x}}}(\mathbf{0}) \geq -(1 - \frac{\epsilon}{r})\|\check{\mathbf{x}}\|_2 \geq -\|\check{\mathbf{x}}\|_2 + \epsilon\kappa. \quad (10)$$

Adding Equations (9) and (10) and then subtracting $2\epsilon\kappa$ on the right hand side, we get

$$h_{\check{\mathbf{x}}}(\mathbf{0}) + \tilde{\mathbf{g}}^T \cdot \check{\mathbf{x}} \geq -\phi\|\check{\mathbf{x}}\|_\infty - 12nr_1\kappa^2 - \epsilon\kappa. \quad (11)$$

Next, we add Equation (11) to Equation (7) and get that $\forall \mathbf{q} \in K$,

$$\begin{aligned} h_{\check{\mathbf{x}}}(\mathbf{q}) &\geq \tilde{\mathbf{g}}^T(\mathbf{q} - \check{\mathbf{x}}) - \phi\|\mathbf{q}\|_\infty - \phi\|\check{\mathbf{x}}\|_\infty \\ &\quad -12nr_1\kappa - 12nr_1\kappa^2 - \epsilon\kappa \\ &\geq \tilde{\mathbf{g}}^T(\mathbf{q} - \check{\mathbf{x}}) - 2\phi R - 24nr_1\kappa^2 - \epsilon\kappa. \end{aligned} \quad (12)$$

$\forall \mathbf{q} \in K$, we have $h_{\check{\mathbf{x}}}(\mathbf{q}) \leq 0$, and then we can have $\tilde{\phi} \geq \tilde{\mathbf{g}}(\mathbf{q} - \check{\mathbf{x}})$, where $\tilde{\phi}$ is a random scalar independent of $\mathbf{q}$ that satisfies

$$E(\tilde{\phi}) \leq 4\sqrt{\frac{12\epsilon\kappa}{r_1}}nR + 24nr_1\kappa^2 - \epsilon\kappa. \quad (13)$$

Putting $r_1 = rR^{-\frac{1}{3}}\epsilon^{\frac{1}{3}}$ into Equation (13), we get

$$E(\tilde{\phi}) \leq 40n\epsilon^{\frac{1}{3}}R^{\frac{2}{3}}\kappa + \epsilon\kappa. \quad (14)$$

Further leveraging $\epsilon \leq r \leq R$, we get

$$E(\tilde{\phi}) \leq (40n + 1)\epsilon^{\frac{1}{3}}R^{\frac{2}{3}}\kappa. \quad (15)$$

Then we can finish the proof using Equation (15) and Markov inequality. □

In addition, observing Algorithm 1 and Algorithm 2, we see that the bottleneck to construct ReSEP is to compute $h_{\check{\mathbf{x}}}$ using binary search and ReMEM (Line 4-8 in Algorithm 2). As such, we give the following theorem on the complexity of Algorithm 1.

**Theorem 2.** *Algorithm 1 constructs the reservation separation oracle (ReSEP) through an $O(n \log R)$ calls on the reservation membership oracle (ReMEM).*

### 3.4 From Resource Separation Oracle to Resource Optimization Oracle

Having constructed ReSEP from ReMEM, we next develop an ellipsoid-method algorithm to construct ReOPT from ReSEP, which is summarized in Algorithm 3.

This algorithm adopts a binary search strategy to find the largest $util(\mathbf{x})$ that is feasible on $K$. In each main iteration (Line 3-24), it constructs a feasible problem

$$\begin{aligned} util(\mathbf{x}) &\geq u_{next}, \\ K : \mathbf{A}\mathbf{x} &\leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \end{aligned} \quad (16)$$

and uses ellipsoid method to test if this problem is feasible. One difference from the classic ellipsoid method is: when evaluating if the center $\mathbf{p}$ of an ellipsoid $E_i$ is a feasible solution, we first evaluate if $\mathbf{p}$ is feasible for $util(\mathbf{x}) \geq u_{next}$, and only invoke ReSEP if $util(\mathbf{p}) \geq u_{next}$. This is because $util(\mathbf{x})$ is kept at the ARO controller, where ReOPT runs, but not shared to ARO scout, where ReSEP resides. This would not affect the correctness of the ellipsoid method for verifying the feasibility of Equation (16) because a half space containing $util(\mathbf{x}) \geq u_{next}$ will also contain the feasible region defined in Equation (16). In the meantime, the privacy of user is also preserved.

**Algorithm 3:** Reservation Optimization Oracle $ReOPT(util(\mathbf{x}))$.

```
Compute the maximum and minimum of util(x) subject to
  x_i ∈ [0, R] where i = 1, ..., n and denote the value as
  u_max and u_min;
u_l ← u_min, u_r ← u_max;
while u_l < u_r do
    u_next ← (u_l + u_r)/2;
    Build an ellipsoid E_0 bounding B_2^+(0, R);
    V_l ← Vol(B_2^+(0, r)), i ← 0;
    feasible ← false;
    while Vol(E_i) ≥ V_l do
        p ← center of E_i;
        if util(p) < u_next then
            feasible ← false;
            H ← {y|(∇util(p))^T (y − p) >= 0};
        else if ReSEP(p) returns a half space H then
            feasible ← false;
        else
            feasible ← true;
            x* ← p;
            break;
        E_{i+1} ← the minimum-volume ellipsoid containing
          E_i ∩ H;
        i ← i + 1;
    if feasible == false then
        u_r ← u_next;
    else
        u_l ← u_next;
return x*;
```

We present the following theorem on the optimality and efficiency of Algorithm 3.

**Theorem 3.** *Algorithm 3 finds $\mathbf{x}^*$ that maximizes $util(\mathbf{x})$ subject to $K$ through an $O(n^2)$ calls on the reservation separation oracle (ReMEM).*

*Proof.* The complexity result in this theorem follows the classic ellipsoid method. For the optimality claim, we observe that even if $util(\mathbf{x})$ is concave, $K$ is still a polytope. As such, $\mathbf{x}^*$ will be a vertex of this polytope. In this way, the optimality of Algorithm 3 can be proved in the same way as the ellipsoid method optimally solves linear programming. □

Putting Theorems 2 and 3 together, we get the following theorem on the optimality and efficiency of BoxOpt.

**Theorem 4.** *BoxOpt finds $\mathbf{x}^*$ that maximizes $util(\mathbf{x})$ subject to $K$ through an $O(n^3 \log R)$ calls on the reservation membership oracle (ReMEM).*

# 4 Performance Evaluation

We implement a prototype of BoxOpt and evaluate its performance on an operational federation network supporting large-scale distributed science collaborations, and using real traffic traces from recent science experiments. We first describe our setup, followed by the detailed results.

## 4.1 Methodology

We evaluate the performance of BoxOpt on the topology from LHC Open Network Environment (LHCONE), a global science network consisting of 62 institutes (Martelli and Stancu 2015). We randomly select a topology for each institute from the Topology Zoo (Knight et al. 2011), and then assemble the connections and topologies from previous steps into a unified large network. We replay the actual trace from the CMS experiment (cms-dashb ), a main source of traffic in LHCONE. We focus on a 48-hour trace starting from December 14, 2017, consisting of 716 resource reservation requests. The number of flows in each request varies between 1 and 7. Because the CMS experiment is one of the largest ongoing distributed scientific experiments with complex, distributed analytics across tens of geographically distributed locations, we believe the trace is representative of complex data flow of modern distributed applications.

## 4.2 Results

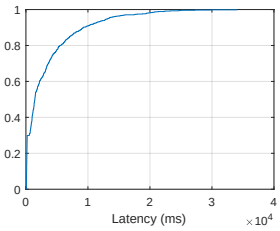
In our experiments, we set $R$ and $r$ in Algorithm 1 to be the maximum and minimum of link bandwidth in the network topology. We run extensive experiments by choosing different values of $\epsilon$ and $\rho$ and different utility functions. In what follows, we present the results of one setting: maximizing total throughput when $\rho = 0.001$ and $\epsilon = \frac{1}{5}r$. Results of other settings are highly similar as this setting, hence are omitted due to page limit.

**Correctness of BoxOpt**: For each reservation request, we compare the optimal resource reservation computed by BoxOpt with the optimal solution to the problem $util(\mathbf{x})$ subject to $K$ computed by a state-of-the-art optimization solver (*e.g.*, CPLEX (CPLEX 2018)). We find that in all 716 requests, BoxOpt outputs the same optimal solution as the solver does, *i.e.*, BoxOpt has a 100% correctness ratio.

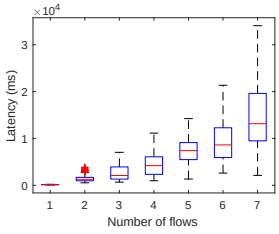
**Efficiency of BoxOpt**: As illustrated in Figure 2 (Section 2), the main bottleneck of BoxOpt is the communication latency between the optimization oracle at the ARO controller invoking the separation oracle at the ARO scout as the computation latency is ignorable. As such, we use the total communication latency to find the optimal resource reservation for each request to represent the efficiency of BoxOpt. Specifically, we assume the user is located at New York and the network is in Los Angeles. For each invocation of ReSEP at ARO scout, we assign it a round trip time (RTT) randomly chosen from the statistic RTT data collected in (global-ping

). Then the communication latency to find the optimal resource reservation for a given request is the sum of all RTTs incurred by corresponding ReSEP invocations.

Figure 3a plots the CDF of communication latency of all requests in the experiment. We observe that for 95% of the requests, BoxOpt is able to learn the optimal resource reservation within 13 seconds. This demonstrates the efficiency of BoxOpt to swiftly learn the optimal resource reservation via the simple reservation interface. Figure 3b plots the statistics of the communication latency for different sizes of reservation requests. The nonlinear increase of the latency is consistent with conclusion in Theorem 3. However, compared with the lasting time of network resource reservation (*e.g.*, hours and days) and the amount of data being transmitted (*e.g.*, TBs), BoxOpt is highly efficient.



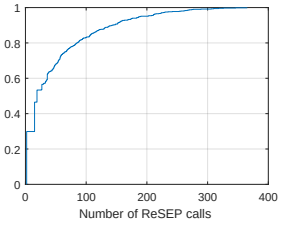

(a) CDF of the communication latency of BoxOpt.



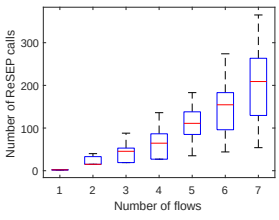

(b) Communication latency of BoxOpt with different sizes of request.

Figure 3: Efficiency of BoxOpt: total communication latency to compute the optimal resource reservation.

**Efficiency of ReOPT**: We next study the efficiency of the ReOPT oracle to learn the optimal resource reservation. To this end, we count the number of ReSEP invocations (*i.e.*, the bottleneck operation of the ReOPT oracle) for each request. Figure 4a gives the CDF of the number of ReSEP invocations. We observe that for 95% requests, the ReOPT learns the optimal reservation within 200 ReSEP calls. Figure 4b further breaks down the statistics based on the size of requests. We observe that the nonlinear increase of ReSEP invocations is consistent with Theorem 3 and Figure 3b.



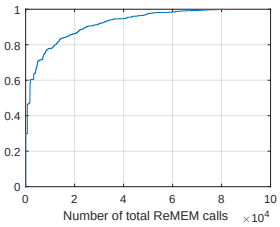

(a) CDF of the number of ReSEP invocations.



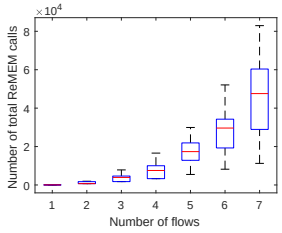

(b) Number of ReSEP invocations with different sizes of request.

Figure 4: Efficiency of ReOPT: number of ReSEP invocations to learn the optimal resource reservation.

**Efficiency of ReSEP**: In the end, we study the efficiency of the ReSEP oracle to infer the search space for ReOPT. To this end, we count the number of ReMEM invocations (*i.e.*, the bottleneck operation of the ReSEP oracle) for each request. Figure 5a gives the CDF of the number of ReMEM invocations. We observe that for 95% requests, the total ReMEM invocations required is within 30000. This large number demonstrates the necessity and benefits of putting ReSEP (*i.e.*, the ARO scout) with the network. Integrating this observation from Figure 4a and Figure 3a, we can conclude that this design improves the efficiency of BoxOpt (*i.e.*, the communication latency) by 20 times. Figure 5b further breaks down the statistics based on the size of requests. We observe that the almost linear increase of ReMEM invocations is consistent with Theorem 2.




(a) CDF of the number of ReMEM invocations.




(b) Number of ReMEM invocations with different sizes of request.

Figure 5: Efficiency of ReSEP: number of ReMEM invocations to learn the optimal resource reservation.

## 5 Related Work

Many network resource reservation systems have been developed and deployed (Campanella et al. 2006; Guok and Robertson 2006; Johnston, Guok, and Chaniotakis 2011; Riddle 2005; Zheng et al. 2005; Sobieski, Lehman, and Jabbari 2004). However, existing systems either are inefficient, or cause private information to be exposed. In contrast, BoxOpt adopts a novel approach to efficiently learn the optimal resource reservation through the limited feedback from the simple interface provided by reservation systems.

One area closely related to our problem is *constraint learning* (De Raedt, Passerini, and Teso 2018; Bessiere et al. 2017; Ruggieri 2012; Bessiere et al. 2004; Ruggieri 2013). We refer readers to (De Raedt, Passerini, and Teso 2018) for a comprehensive survey. Instead of learning all linear inequalities that compose the bandwidth feasible region, in BoxOpt, we show that the optimal solution to an optimization problem can be learnt efficiently and accurately without knowing any constraints. One future direction is to integrate constraint learning into BoxOpt to further accelerate the learning of the optimal resource reservation.

BoxOpt leverages several powerful tools from optimization theory (Lovasz, Grotschel, and Schrijver 1993; Boyd and Vandenberghe 2004; Lee, Sidford, and Vempala 2017). We expand the recent theoretical progress on efficient oracle constructions to a broader scenario. To the best of our knowledge, BoxOpt is the first working system that demonstrates the feasibility and benefits of learning the optimal solution of an optimization problem with only membership oracle. In addition to network resource reservation, it also sheds light for other areas such as multi-domain traffic engineering and collaborative data analytics.

## 6 Conclusion

We design BoxOpt, a novel, automatic learning system to efficiently learn the optimal resource reservations through the simple reservation interface of network resource reservation systems, without exposing the private information of network or user. We demonstrate its efficiency and efficacy through extensive evaluation using real network topology and trace.

## Acknowledgment

The authors thank Christian Bessiere, Haizhou Du, Kai Gao, Chin Guok, Max Del Giudice, Chris Harshaw, Amin Karbasi, Yin Tat Lee, Geng Li, Yang Liu, John MacAuley, Harvey Newman, Lam M. Nguyen, Salvatore Ruggieri, Mudhakar Srivatsa, Xin Wang, Jingxuan Zhang and Rui Zhang for their help during the preparation of this paper. The authors also thank the anonymous reviewers for their valuable comments. This research is supported in part by NSFC grants 61702373, 61672385, 61701347, and 61672349; China Postdoctoral Science Foundation 2017-M611618; NSF awards CCF-1637385, CCF-1650596, OAC-1440745; Google Research Award; and the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001.

## References

Atkinson, D. S., and Vaidya, P. M. 1995. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming* 69(1-3):1–43.

Bertsimas, D., and Vempala, S. Solving convex programs by random walks. In *Proceedings of the 34th ACM STOC*.

Bessiere, C.; Hebrard, E.; Hnich, B.; and Walsh, T. 2004. Disjoint, partition and intersection constraints for set and multiset variables. In *International Conference on Principles and Practice of Constraint Programming*, 138–152. Springer.

Bessiere, C.; Koriche, F.; Lazaar, N.; and O'Sullivan, B. 2017. Constraint acquisition. *Artificial Intelligence* 244:315 – 342.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Bubeck, S., and Eldan, R. 2016. Multi-scale exploration of convex functions and bandit convex optimization. In *Conference on Learning Theory*, 583–589.

Campanella, M.; Krzywania, R.; Reijs, V.; Wilson, D.; Sevasti, A.; Stamos, K.; and Tziouvaras, C. 2006. Bandwidth on demand services for european research and education networks. In *IEEE Bandwidth on Demand 06*, 65–72. IEEE.

CMS Task Monitoring. http://dashb-cms-job.cern.ch/.

2018. Ilog cplex.

De Raedt, L.; Passerini, A.; and Teso, S. 2018. Learning constraints from examples. In *Proceedings in Thirty-Second AAAI Conference on Artificial Intelligence, AAAI, New Orleans, USA*, 02–07.

Gao, K.; Gu, C.; Xiang, Q.; Wang, X.; Yang, Y. R.; and Bi, J. 2016. ORSAP: abstracting routing state on demand. In *24th IEEE International Conference on Network Protocols, ICNP 2016, Singapore, November 8-11, 2016*, 1–2.

Gao, K.; Xiang, Q.; Wang, X.; Yang, Y. R.; and Bi, J. 2017. Nova: Towards on-demand equivalent network view abstraction for network optimization. In *IWQoS'17*, 1–10.

Global Ping Statistics - WonderNetwork, 2018. https://wondernetwork.com/pings/.

Guok, C., and Robertson, D. 2006. Esnet on-demand secure circuits and advance reservation system (oscars). *Internet2 Joint* 92.

Heorhiadi, V.; Reiter, M. K.; and Sekar, V. 2016. Simplifying software-defined network optimization using sol. In *NSDI*, 223–237.

Johnston, W.; Guok, C.; and Chaniotakis, E. 2011. Motivation, design, deployment and evolution of a guaranteed bandwidth network service. In *Proceedings of the TERENA Networking Conference*.

Knight, S.; Nguyen, H. X.; Falkner, N.; Bowden, R.; and Roughan, M. 2011. The internet topology zoo. 29(9):1765–1775.

Lee, J.; Turner, Y.; Lee, M.; Popa, L.; Banerjee, S.; Kang, J.-M.; and Sharma, P. Application-driven bandwidth guarantees in datacenters. In *SIGCOMM'14*.

Lee, Y. T.; Sidford, A.; and Vempala, S. S. 2017. Efficient convex optimization with membership oracles. *arXiv preprint arXiv:1706.07357*.

Lovasz, L.; Grotschel, M.; and Schrijver, A. 1993. *Geometric algorithms and combinatorial optimization - 2nd corrected edition*. Springer.

Martelli, E., and Stancu, S. 2015. Lhcopn and lhcone: status and future evolution. In *Journal of Physics: Conference Series*, volume 664, 052025. IOP Publishing.

Mogul, J. C., and Popa, L. 2012. What we talk about when we talk about cloud network performance. *SIGCOMM CCR 12* 42(5):44–48.

Riddle, B. 2005. Bruw: A bandwidth reservation system to support end-user work. In *TERENA Networking Conference, Poznan, Poland*.

Ruggieri, S. 2012. Deciding membership in a class of polyhedra. In *ECAI*, 702–707.

Ruggieri, S. 2013. Learning from polyhedral sets. In *23rd Int. Joint Conference on Artificial Intelligence (IJCAI 2013)*, 1069–1075. AAAI Press.

Sobieski, J.; Lehman, T.; and Jabbari, B. 2004. Dragon: Dynamic resource allocation via gmpls optical networks. In *MCNC Optical Control Planes Workshop, Chicago, Illinois*.

Soulé, R.; Basu, S.; Marandi, P. J.; Pedone, F.; Kleinberg, R.; Sirer, E. G.; and Foster, N. 2014. Merlin: A language for provisioning network resources. In *CoNEXT'14*, 213–226. ACM.

Subramanian, K.; D'Antoni, L.; and Akella, A. Genesis: Synthesizing forwarding tables in multi-tenant networks. *ACM POPL'17*.

White, T. 2012. *Hadoop: The definitive guide*. O'Reilly Media, Inc.

Xiang, Q.; Zhang, J. J.; Wang, X. T.; Liu, Y. J.; Guok, C.; Le, F.; MacAuley, J.; Newman, H.; and Yang, Y. R. 2018. Fine-grained, multi-domain network resource abstraction as a fundamental primitive to enable high-performance, collaborative data sciences. In *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*, 27–29. ACM.

Zaharia, M.; Chowdhury, M.; Das, T.; Dave, A.; Ma, J.; McCauley, M.; Franklin, M. J.; Shenker, S.; and Stoica, I. 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *NSDI'12*, 2–2. USENIX Association.

Zheng, X.; Veeraraghavan, M.; Rao, N. S.; Wu, Q.; and Zhu, M. 2005. Cheetah: Circuit-switched high-speed end-to-end transport architecture testbed. *IEEE Communications Magazine* 43(8):S11–S17.

# Toward Optimal Software-Defined Interdomain Routing

Qiao Xiang*‡, Jingxuan Zhang‡*, Kai Gao°,
Yeon-sup Lim◇, Franck Le◇, Geng Li‡*, Y. Richard Yang†‡,
*Tongji University, ‡Yale University, †Peng Cheng Laboratory,
°Sichuan University, ◇IBM T.J. Watson Research Center,

***Abstract*—End-to-end route control spanning a set of networks can provide opportunities to both end users to optimize interdomain control and network service providers to increase business offering. BGP, the de facto interdomain routing protocol, provides no programmable control. Recent proposals for interdomain control, such as MIRO, ARROW and SDX, provide more mechanisms and interfaces, but they are only either point or incremental solutions. In this paper, we provide the first, systematic formulation of the *software-defined internetworking (SDI)* model, in which a network exposes a programmable interface to allow clients to define the interdomain routes of the network, just as a traditional SDN switch exposes Openflow or another programmable interface to allow clients to define its next hops, extending SDN from intra-domain control to generic interdomain control. Different from intradomain SDN, which allows complete client control, SDI should also maximize network autonomy, such as by allowing a network to maintain the control of its interdomain export policies, to avoid fundamental violations such as valley routing. We define the *optimal end-to-end SDI routing problem* and conduct rigorous analysis to show that the problem is NP-hard. We develop a blackbox optimization algorithm, which leverages Bayesian optimization theory and important properties of interdomain routing algebra, to sample end-to-end routes sequentially and find a near-optimal policy-compliant end-to-end route with a small number of sample routes. We implement a prototype of our optimization algorithm and validate its effectiveness via extensive experiments using real interdomain network topology. Results show that in an interdomain network with over 60000 ASes and over 320000 AS-level links, in 80% experiment cases, the blackbox optimization algorithm can find a near-optimal policy-compliant end-to-end route by sampling less than 33 routes.**

## I. Introduction

Although flexible, end-to-end route control may provide substantial benefits to both networks and end users (*e.g.*, conduct traffic engineering, or prevent DDoS attacks), it is extremely complex and difficult to achieve, if not impossible, in the current Internet. This is due to the design of the Border Gateway Protocol (BGP) [1], the *de facto* interdomain routing protocol. Specifically, with BGP, each autonomous systems (AS) can make and execute its own policy to select routes on a destination prefix basis, and export the selected routes, in terms of path vectors (*i.e.*, AS path), to its neighbor ASes. Although this design is simple and widely adopted, it provides very limited mechanisms for network operators and end users to achieve flexible, end-to-end route control. AS paths that span from a source AS to a destination AS, and that the path does not traverse a certain AS.

To appreciate the limitation of BGP, consider AS $S$ shown in Fig. 1, who has a strong performance requirement to select a shortest AS path from it to a destination prefix $p$ located at AS $T$. The ranking used by AS $A$ and $D$ to select routes is also shown in the figure. Assume that the export policies of each AS is to always announce the selected route to its neighbor ASes. When the network converges, AS $A$ selects the AS-path $[A, B, D, F, H, T]$ and announces to AS $S$. As such, the only AS-path to $T$ learnt by $S$ is $[S, A, B, D, F, H, T]$. One can observe that there are indeed AS paths shorter than $[S, A, B, D, F, H, T]$ in the network, such as $[S, A, B, D, E, T]$. To use such a path, AS $D$ must select the route $[D, E, T]$. Unfortunately, in BGP, AS $S$ cannot control $D$ to select $[D, E, T]$. As such, $S$ can only use $[S, A, B, D, F, H, T]$, not satisfying its requirement. At the same time, AS $D$ misses a potential business opportunity.

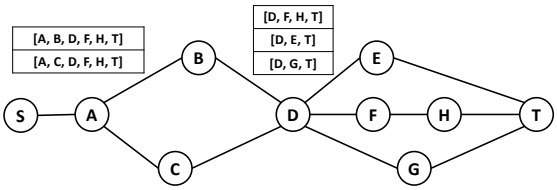



**Fig. 1:** A motivating example to illustrate the limitations of BGP and other interdomain routing systems for supporting flexible, end-to-end route control.

To provide more flexible, end-to-end route control, several interdomain routing systems have been designed and deployed [2]–[10]. For example, in MIRO [9] and ARROW [10], an AS allows a client (*i.e.*, an operator of another AS or an end user) to control its interdomain routing, by mapping the client's traffic into tunnels to achieve more flexible interdomain traffic control. In SDX [5], an Internet exchange point, which can also be considered as an AS, allows a direct upstream AS of the exchange to control how traffic is distributed among its downstream, and a direct downstream AS to control how traffic gets into the downstream, using flexible matching conditions (*e.g.*, match on TCP/IP 5-tuple). Although these systems provide clients more mechanisms than BGP to control interdomain routing, they either are point solutions (*e.g.*, SDX) or may have datapath overhead such as tunneling processing on each data packet.

In this paper, we investigate a novel, systematic, low-overhead interdomain route control model which we call the software-defined internetworking (SDI) model. Motivated by the success of intradomain SDN models such as Openflow or P4 but extending to interdomain, SDI defines an interdomain

programmable interface so that a network exposes to a client its available interdomain routes (*i.e.*, its interdomain routing information base) to a destination, and the client can then choose one of them, just as an intradomain SDN client can select a port as the next hop among a set of available output ports of an SDN switch; Different from intradomain SDN, however, SDI maximizes network autonomy, by allowing a network to maintain the control of its interdomain export policies, to avoid fundamental violations such as valley routing.

Allowing client control and at the same time maximizing network autonomy by allowing a network to fully control its export policies exposes challenging problems not investigated before, although the problems also exist in aforementioned early work. Specifically, consider the same interdomain network in Fig. 1 and assume that the ASes have the following autonomous, *private* export policies: (1) AS $D$ will not announce any AS path containing AS $E$ to AS $B$; (2) AS $C$ will not announce any AS path containing AS $E$ to AS $A$; (3) AS $B$ will not announce any AS path containing AS $G$ to AS $A$. Still consider AS $S$, whose goal is to send traffic toward $p$ along a shortest AS path. Conducting SDI control, AS $S$ may ask AS $D$ and AS $A$ to select route $[D, E, T]$ and route $[A, B, D, E, T]$, respectively, hoping that it can then use the route $[S, A, B, D, E, T]$. However, because $D$ will not announce route $[D, E, T]$ to $B$ due to its export policy, the route $[S, A, B, D, E, T]$ cannot be used by $S$. In the worst case, only after enumerating many paths can $S$ find that $[S, A, C, D, T]$ is the only *export-policy-compliant* shortest AS path. Such an interdomain SDN with autonomy problem has not been studied before.

We formulate the optimal interdomain SDI route control with policy compliance problem, which we refer to as the optimal end-to-end SDI routing problem. We conduct rigorous analysis on the computational complexity of this problem and prove its strong NP-hardness. The fundamental reason behind the computational intractability of this problem is that each AS can autonomously make and execute its own export policy. Previous studies [11], [12] show that when ASes all follow the Gao-Rexford route selection and export policies based on the customer/peer/provider business relationship [13], the shortest policy-compliant route can be computed in polynomial time. However, a recent survey [14] on many network operators shows that a high percentage of ASes are actually breaking the Gao-Rexford condition in their export policies.

We develop a blackbox optimization algorithm to sample end-to-end routes sequentially and find a near-optimal policy-compliant end-to-end route with a small number of samples. Our algorithm, built on the Bayesian optimization framework, is highly efficient for two reasons. First, it iteratively leverages the prior belief about the problem to help direct the sampling, and to trade exploration and exploitation of the search space [15], [16]. Second, it leverages important properties from interdomain routing algebra [17], [18] to derive an accurate estimation on the expected improvement of an end-to-end route, which avoids the efficiency loss brought by rounding continuous sample points into discrete sample routes [19].

| Symbol | Meaning |
|---|---|
| $v$ | An AS |
| $(u, v)$ | An edge from AS $u$ to AS $v$ in the AS Graph |
| $\mathbf{r}$ | A route (AS Path) |
| $p$ | A destination IP prefix |
| $e_v(p, \mathbf{r})$ | The export policy of AS $v$ for prefix $p$ and route $\mathbf{r}$ |
| $c_v(\mathbf{r})$ | The pricing policy of AS $v$ for route $\mathbf{r}$ |
| $f$ | The objective function |

**TABLE I:** Symbols

The **main** contributions of this paper are as follows:

- We provide the first, systematic formulation of the software-defined internetworking(SDI) model, extending intradomain SDN to generic interdomain SDN;
- We systematically study the optimal end-to-end SDI routing problem, and show that the problem is strongly NP-hard;
- We develop a blackbox optimization algorithm, which integrates the Bayesian optimization theory and important properties in interdomain routing algebra, to efficient find near-optimal end-to-end routes with a small number of route sampling;
- We implement a prototype of our algorithm and evaluate its performance via extensive experiment using real-world topology. Results show that in an interdomain network with over 60000 ASes and over 320000 AS-level links, in 80% experiment cases, the blackbox optimization algorithm can find a near-optimal policy-compliant end-to-end route by sampling less than 33 routes.

## II. SDI Model and Problem Formulation

In this section, we first give the systematic formulation of the SDI model. Then we define the problem of finding a route that is compliant to local policies and achieves global optimal objectives, namely the optimal end-to-end SDI routing problem.

### A. Basic Definitions

**Global network.** We use a graph $G = (V, E)$ to denote an interdomain AS network, where ASes are interconnected by BGP[1]. A vertex $v \in V$ represents an AS in the network, and a bidirectional edge $(u, v) \in E$ represents a BGP session between AS $u$ and AS $v$.

**Interdomain route.** An interdomain route $\mathbf{r}$ is represented as a sequences of ASes $[v_1, v_2, \ldots, v_n]$ (also called an AS path), where $v_i \neq v_j$ for any two integers $i \neq j \in [1, n]$. Given a route $\mathbf{r}$, we call $v_1$ the source AS and $v_n$ the destination AS. Given two routes $\mathbf{r}_1 = [v_1, \ldots, v_n]$ and $\mathbf{r}_2 = [u_1, \ldots, u_m]$, if $(v_n, u_1) \in E$ and $\forall i \in [1, n], \forall j \in [1, m], v_i \neq u_j$, the concatenation of $\mathbf{r}_1$ and $\mathbf{r}_2$, represented as $\mathbf{r}_1 \oplus \mathbf{r}_2 = [v_1, \ldots, v_n, u_1, \ldots, u_m]$, is also a route.

**Network configuration.** Each AS node $v$ is configured with two key attributes: an export policy, and a pricing policy. In

[1]For simplicity of presentation, we stick to the one-big-switch abstraction widely used in BGP studies (*e.g.*, [1], [13], [17], [20]).

this paper, we assume the ASes are propagating the routing information from destination to source. Thus, the export policy also only applies to the routes from the current AS to the destination. The export policy is modeled as a function $e_v(p, \mathbf{r}, u)$. It takes a destination IP prefix $p$ and a route $\mathbf{r}$, and returns either 1 if AS $v$ will announce route $[v] \oplus \mathbf{r}$ to its neighbor $u$ if $\mathbf{r}$ is available, or 0 otherwise. A pricing model is important when an AS allows a client to override the network's default interdomain route selection policy. In this paper, we denote the pricing model as a function $c_v(\mathbf{r})$. It takes a route $\mathbf{r}$ and returns a real number, which stands for the price to set up $\mathbf{r}$ for a match in AS $v$.

***Definition** 1 (Policy-Compliant Route):* A route $\mathbf{r} = [v_1, v_2, \ldots, v_n]$ toward a destination IP prefix $p$ is export-policy-compliant or simply policy-compliant if and only if the following condition holds:

$$e(p, \mathbf{r}) \triangleq \bigwedge_{i=2}^{n} e_{v_i}(p, [v_{i+1}, \ldots, v_n], v_{i-1}) = 1. \quad (1)$$

With the aforementioned model, one is able to verify if an AS path is policy compliant and also to calculate the total cost of establishing a route $\mathbf{r} = [v_1, \ldots, v_n]$ as follows:

$$c(\mathbf{r}) \triangleq \sum_{i=1}^{n} c_{v_i}([v_i, v_{i+1}, \ldots, v_n]). \quad (2)$$

**Client's objective function.** An objective function $f(\mathbf{r})$ takes route $\mathbf{r}$ and returns a real number, which represents the objective for the route. Without loss of generality, we focus on functions $f$ that satisfy the following two properties:

- *monotonicity*: Given a route $\mathbf{r}$, $f([u, v] \oplus \mathbf{r}) \geq f(\mathbf{r})$;
- *isotonicity*: Given two routes $\mathbf{r}_1$ and $\mathbf{r}_2$, if $f(\mathbf{r}_1) \geq f(\mathbf{r}_2)$, then $f([u, v] \oplus \mathbf{r}_1) \geq f([u, v] \oplus \mathbf{r}_2)$.

A typical example of the monotone and isotonic objective function is the shortest AS path, where $f(\mathbf{r})$ is the inverse of the AS path length. Other monotone and isotonic objective functions include weighted shortest AS path[2], widest shortest AS path, reachability, etc. A more complete list of monotone and isotonic objective functions can be found in [21].

### B. SDI Programmable Network

**SDI programming service at an AS.** In addition to an actual BGP speaker, each AS $v \in G$ also runs a virtual BGP speaker, which has the same route selection and export policies as the actual BGP speaker of $v$, and establishes BGP sessions with the virtual BGP speakers of the neighboring ASes of $v$ in $G$.

Each AS $v$ exposes the following information to clients. First, given a destination IP prefix $p$ specified by a client, it exposes the routing information base (RIB), *i.e.*, all available routes $v$ has to reach $p$. Second, given a destination IP prefix $p$ and a route $\mathbf{r}$ from $v$ to $p$ specified by a client, it exposes the price for the client to use $\mathbf{r}$. Exposing such information will not expose the private policies (*i.e.*, the route selection / export / pricing policies) of ASes, because inferring these policies based on the exposed information is a constraint acquisition problem, which is computationally intractable [22].

Each AS $v$ provides three programming interfaces to clients. Specifically, it uses a two-phase commit design pattern to allow a client to test the policy-compliance of an end-to-end interdomain route $\mathbf{r}$ before actually using and paying for it. This design has two advantages: (1) it avoids the disruptions and churns in the interdomain network caused by the client using a non-policy-compliant route; (2) it avoids the waste of monetary expense of the client paying for a non-policy-compliant route. The programming interfaces are as follows.

- $selectRoute(match, \mathbf{r})$: This interface is to select a route $\mathbf{r}$ to forward the class of traffic specified by the packet header matching condition $match$ in the virtual BGP speaker. Upon receiving this request from a client, AS $v$ first checks if $\mathbf{r}$ is in its RIB. If so, it selects this route in its virtual BGP speaker, and lets the virtual speaker follow its export policy to announce $\mathbf{r}$ to neighboring virtual BGP speakers.
- $commitRoute(match, \mathbf{r})$: This interface is to install a route $\mathbf{r}$ in the forwarding information base (FIB) of AS $v$, to forward the class of traffic specified by the packet header matching condition $match$. AS $v$ first checks if $\mathbf{r}$ is in its RIB. If so, it will configure its real BGP speaker to install this route in the FIB of the router, and follow its export policy to announce this route to neighboring BGP speakers.
- $deleteRoute(match, \mathbf{r})$: This interface is to stop using the route $r$ for the class of traffic specified by $match$. AS $v$ will remove the $\mathbf{r}$ from its FIB.

**Interaction between client and ASes.** A client can use the exposed information and programming interfaces of ASes to achieve flexible, end-to-end interdomain route control.

First, given an end-to-end route $\mathbf{r} = [v_1, v_2, \ldots, v_n]$, a client checks if it is policy-compliant. Specifically, a client can iteratively interact with the ASes along $\mathbf{r}$ in backward order: for any $i = n, \ldots, 2$, the client asks AS $v_i$ to select route $[v_i, \ldots, v_n]$ in its virtual BGP speaker using the $selectRoute$ interface and observes the RIB of AS $v_{i-1}$. If route $[v_i, \ldots, v_n]$ is observed in the RIB of $v_{i-1}$ for all $i = n, \ldots, 2$, the route $\mathbf{r}$ is policy-compliant.

Second, a client also interacts with the ASes along $\mathbf{r}$ to evaluate the total cost of using $\mathbf{r}$. This can be achieved by asking the price of using route $[v_i, v_{i+1}, \ldots, v_n]$ for each AS $v_i$ along $\mathbf{r}$.

Third, after the client finds a policy-compliant end-to-end route $\mathbf{r} = [v_1, v_2, \ldots, v_n]$ he wants to use for forwarding certain traffic. He can interact with the ASes along $\mathbf{r}$ in a backward order (*i.e.*, from $v_n$ to $v_1$) to setup the this route. Specifically, for each AS $v_i$, the client uses the $commiteRoute$ interface to request $v_i$ to install a route $[v_i, v_{i+1}, \ldots, v_n]$ in its forwarding information base to forward the traffic specified by the packet header matching condition $match$.

[2]With all weights being positive

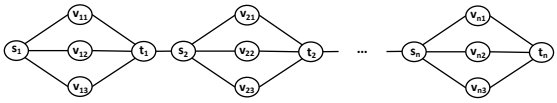


**Fig. 2:** An illustration of constructing $G$ from an instance of 3-SAT problem.

### C. Optimal End-to-End SDI Routing Problem

Based on the aforementioned models, we now formally define the optimal end-to-end SDI routing problem.

***Problem** 1 (Optimal End-to-End SDI Routing Problem):* For an interdomain network $G = (V, E)$, where the export policy and the pricing policy of AS $v$ are denoted as $e_v(p, \mathbf{r}, u)$ and $c_v(\mathbf{r})$ respectively, and $f(\mathbf{r})$ denote the global objective function while $B$ denote the cost budget, the optimal policy-compliant route $\mathbf{r}^*$ for a match $m$ from a source AS $s$ to a destination AS $t$ is the solution of the following optimization problem:

$$\text{maximize} \quad f(\mathbf{r})$$

subject to,

$$v_1 = s, v_n = t, e(p, \mathbf{r}) = 1, c(\mathbf{r}) \leq B$$

## III. Complexity Analysis

In this section, we analyze the computational complexity of Problem 1. Our main finding is summarized in the following theorem:

***Theorem** 1:* The optimal end-to-end SDI routing problem (Problem 1) is strong NP-hard.

To prove the NP-hardness of Problem 1, we first consider the following problem:

***Problem** 2 (Shortest Policy Compliant AS-Path Problem):* Assume an interdomain network $G = (V, E)$, where ASes not only provide the SDI service described in Section II, but also expose their export policies to the client. Given a source AS $v_s$, a destination AS $v_d$, a positive integer $K$, and a positive real number $B$, is it possible to find a route $\mathbf{r}$ from $v_s$ to $v_d$, such that (1) the AS path length of $\mathbf{r}$ is less than or equal to $K$, (2) $\mathbf{r}$ is policy-compliant, and (3) the total cost of using $\mathbf{r}$ is less than or equal to $B$.

It is easy to see that Problem 2 is a simplified variant of the optimal end-to-end SDI routing problem. And we propose and prove the following lemma:

***Proposition** 1:* Problem 2 is strong NP-hard.

*Proof:* We prove the NP-hardness of Problem 2 via a reduction from the 3-SAT problem. Specifically, given an instance of a 3-SAT problem with $n$ clauses $\{C_1, C_2, \ldots, C_n\}$. We use $x_{ij}$ to denote the $j$-th literal in the $i$-th clause $C_i$. For each clause $C_i$, we construct a graph $G_i = (V_i, E_i)$, where $V_i = \{s_i, t_i, v_{i1}, v_{i2}, v_{i3}\}$, and $E_i = \{(s_i, v_{ij}), (v_{ij}, t_i)|$ where $j = 1, 2, 3.\}$.

We then can construct an instance of Problem 2 as follows: First, we construct the graph $G$ by connecting $G_i$ and $G_{i+1}$ with an edge $(t_i, s_{i+1})$ for $i = 1, 2, \ldots, n-1$ (Fig. 2). For each AS $v_{ij}$, we define its export policy as follows: given an AS path $\mathbf{r}$, if $\mathbf{r}$ contains an AS $v_{i'j'}$, whose corresponding literal $x_{i'j'} = \neg x_{ij}$, AS $v_{ij}$ will not announce $\mathbf{r}$ to AS $t_i$. For each AS $s_i$, we define its export policy as: announcing every route to every neighbor $v_{ij}$. For each AS $t_i$, we define its export policy as: announcing every route to neighbor $s_{i+1}$. We set $K = 3n - 1$. For each AS in the constructed graph $G$, we set its pricing policy to be always charging the client $B/3n$ for any route the client want so use. We suppose the client wants to find an end-to-end route $\mathbf{r}$ from $s_1$ to $t_n$ such that (1) $\mathbf{r}$ policy-compliant, (2) $\mathbf{r}$ has an AS path length less than or equal to $K$, and (3) the cost of using $\mathbf{r}$ is less than or equal to $B$. We can see that this process of constructing an instance of Problem 2 from an instance of the 3-SAT problem is polynomial.

After the construction, if the 3-SAT instance is satisfiable, assume the in clause $C_i$, the $j_i$-th literal $x_{ij_i}$ is true, then the route $[s_1, v_{1j_1}, t_1, s_2, v_{2j_2}, t_2, \ldots, t_n]$ is a policy-compliant route in the constructed instance of Problem 2. On the other hand, if there exists a policy-compliant route $[s_1, v_{1j_1}, t_1, s_2, v_{2j_2}, t_2, \ldots, t_n]$ in the constructed instance of Problem 2, a satisfiable truth assignment can be found for the original 3-SAT instance by setting literal $x_{ij_i}$ to be true. As such, we complete the proof. ■

With the proof of Proposition 1, the correctness of Theorem 1 is an immediate result.

## IV. Blackbox Optimization Algorithm

In this section, we first present a strawman solution based on the Yen's $k$-shortest-path algorithm [23], and show that its poor performance is caused by 1) the high latency of testing the policy-compliance of routes and 2) the worst case of exponential enumeration complexity. To resolve these problems, we devise an efficient blackbox optimization algorithm, which samples end-to-end routes sequentially and finds a near-optimal policy-compliant end-to-end route with a small number of samples.

### A. Naive Route Enumeration Algorithm

A naive solution is to iteratively check the $k$-th optimal route from $v_s$ to $v_t$ computed by the Yen's algorithm [23] until finding the first policy-compliant route of which cost does not exceed $B$. Its pseudocode is presented in Algorithm 1.

**Algorithm 1:** Naive Route Enumeration Algorithm.

```
foreach k = 1, 2, ..., do
    Compute the k-th optimal AS path using the Yen's
      algorithm and denote as r_k;
    Test the policy-compliance of r_k;
    Compute the cost of using r_k;
    if r_k is policy-compliant and c(r_k) ≤ B then
        return r_k;
return null;
```

Specifically, in this solution, Yen's algorithm employs a generalized Dijkstra's algorithm proposed in [21]. In each iteration $k$, the Yen's algorithm computes a set of deviation paths from the $k-1$-th optimal path $\mathbf{r}_{k-1}$, which shares a

subroute with $\mathbf{r}_{k-1}$. Among the newly computed deviation paths and the computed but unselected deviation paths in previous iterations, the algorithm selects the one with the largest utility $f()$ as the $k$-th optimal path. As such, this naive solution can always find the optimal end-to-end route that is policy-compliant and does not exceed the client's budget.

However, there are two issues with this solution. First, in each iteration, a client needs to test the policy-compliance of the $k$-th shortest AS path $\mathbf{r}$. This process incurs a long latency because the client needs to interact with ASes along $\mathbf{r}$ sequentially. Second, in the worst case, the algorithm needs to enumerate all possible routes between $v_s$ and $v_t$, of which complexity is $O(V!)$.

### B. Blackbox Optimization Algorithm

Given the hardness of finding the optimal end-to-end route, we instead design a blackbox optimization algorithm that samples end-to-end routes sequentially and finds a near-optimal policy-compliant end-to-end route with a small number of sampled routes. The key insights behind our algorithms are: First, it leverages the prior belief about the problem to help direct the sampling, and to trade exploration and exploitation of the search space [15], [16]. Second, it leverages important properties from interdomain routing algebra [17], [18] to derive an accurate estimation on the expected improvement of an end-to-end route, which avoids the efficiency loss brought by rounding continuous sample points into discrete sample routes [19].

**Reformulation as blackbox optimization problem.** We first transfer the formulation of Problem 1 into a blackbox optimization problem. For simplicity, we omit prefix $p$ from function $e$ and use $e(\mathbf{r})$ instead. We move $e(\mathbf{r})$ from the constraint to the objective, and get the blackbox optimization formulation as follows:

$$\text{maximize} \quad u(\mathbf{r}) = e(\mathbf{r})f(\mathbf{r})$$

subject to,

$$v_1 = s, v_n = t, c(\mathbf{r}) \leq B$$

Based on the fact that both $e(\mathbf{r})$ and $c(\mathbf{r})$ are unknown to the client, but can be observed through the interaction between client and ASes, our algorithm utilizes the framework of Bayesian Optimization (BO), a powerful framework for solving optimization problems whose objective functions and constraints are unknown beforehand, but can be observed through experiments [15], [16].

**Sampling routes using Bayesian optimization.** BO is a sequential model-based approach. Within this framework, we can define a prior belief over the possible objective functions $u(\mathbf{r}) = e(\mathbf{r})f(\mathbf{r})$ and constraints $c(\mathbf{r})$, and then sequentially refines this model as end-to-end routes are sampled via Bayesian posterior updating [16]. To sample efficiently, BO uses an *acquisition function* to determine the next route to sample. As such, BO has the nice property that it typically only needs to sample a small number of routes to find a near-optimal end-to-end route.

Different types of acquisition functions can be used in the BO framework to guide the sequential sampling process. We adopt the Expected Improvement (EI) acquisition function as it has been verified to perform better than other acquisition functions in most cases [24], [25].

Given a route $\mathbf{r}$ that $u(\mathbf{r})$ has not been evaluated yet, we define its improvement function as follows:

$$I(\mathbf{r}) = max\{u(\mathbf{r}) - u(\mathbf{r}^+), 0\}, \tag{3}$$

where $\mathbf{r}^+$ is the route that provides the maximum objective value observed so far. In other words, $I(\mathbf{r})$ is positive when the prediction of $u(\mathbf{r})$ is higher than the best value $u(\mathbf{r}^+)$ so far. Otherwise, $I(\mathbf{r})$ is set to zero.

Next route $\mathbf{r}$ to evaluate is the one that maximizes the expected improvement:

$$\mathbf{r} = \underset{\mathbf{r}}{\operatorname{argmax}}\, E[I(\mathbf{r})|D_t],$$

where $D$ is a set of routes whose value of $u()$ is known.

If we assume $u(\mathbf{r})$ in Equation IV-B is a Gaussian process, we can adopt a covariance function (*i.e.*, the kernel function) to measure the similarity between to different routes (*e.g.*, the $Matern5/2$ kernel [25]), and plug in the kernel function in to the close-form expression of EI [26]. However, in the optimal end-to-end route problem, the search space is discrete.

A strawman approach to tackle the discrete search space is to still use the close-form expression of EI derived in [26] to maximize EI, and then rounding the result to get the next route $\mathbf{r}$. However, it has been reported in recent studies that this rounding approach performs poorly in the BO framework [19]. As such, we need to explore other approaches to compute EI for discrete decision variables (*i.e.*, routes).

**Discrete EI maximization leveraging properties from interdomain routing algebra.** Our solution to the discrete EI maximization problem is to leverage the following two properties derived from interdomain routing algebra [17], [18] to derive an accurate estimation of $EI(\mathbf{r})$.

***Proposition** 2 (Subroute Policy-Compliance):* If a route $\mathbf{r} = [v_1, v_2, \ldots, v_n]$ is policy-compliant, any route $[v_i, \ldots, v_n]$, where $i = 1, 2, \ldots, n$ is policy-compliant.

***Proposition** 3 (Non-policy-compliance by inclusion):* If a route $\mathbf{r} = [v_1, v_2, \ldots, v_n]$ is not policy-compliant, any route $\mathbf{r}'$ that is constructed by prepending a segment in graph $G$ before $\mathbf{r}$ is not policy-compliant.

These two properties help the client to better predict the probability of a route $\mathbf{r}$ being policy-compliant or not. Specifically, assuming the export policies of ASes are independent from each other, given a route $\mathbf{r} = [v_1, v_2, \ldots, v_n]$, we use $P_{i,i-1}([v_i, \ldots, v_n]|D)$ to represent the probability that AS $v_i$ will announce route $[v_i, \ldots, v_n]$ to AS $v_{i-1}$, given the current set of sampled routes $D$.

Then the probability that $\mathbf{r}$ is policy-compliant, represented by $P_{\mathbf{r}}$, is computed as

$$P_{\mathbf{r}} = \prod_{i=n,\ldots,2} P_{i,i-1}([v_i, \ldots, v_n]|D). \tag{4}$$

Then we have

$$E[e(\mathbf{r})|D] = 1 \cdot P_{\mathbf{r}} + 0 \cdot (1 - P_{\mathbf{r}}) = \prod_{i=n,\ldots,2} P_{i,i-1}([v_i, \ldots, v_n]|D). \tag{5}$$

With Equation 5, we can do the following derivation on the expression of expected improvement of a route:

$$\begin{aligned} E[I(\mathbf{r})|D] =& E[\max\{u(\mathbf{r}) - u(\mathbf{r}^+), 0\}|D] \\ =& \max\{E[u(\mathbf{r})|D] - u(\mathbf{r}^+), 0\} \\ =& \max\{\frac{E[e(\mathbf{r})|D]}{f(\mathbf{r})} - u(\mathbf{r}^+), 0\} \\ =& \max\{f(\mathbf{r}) \prod_{i=n,\ldots,2} P_{i,i-1}([v_i, \ldots, v_n]|D) - u(\mathbf{r}^+), 0\} \end{aligned} \tag{6}$$

In addition, as suggested by [27], we also add another term to the right-hand side of Equation 7, changing it to:

$$\begin{aligned} E(I(\mathbf{r})|D) =& P(c(\mathbf{r}) \leq B) \cdot \\ & max\{f(\mathbf{r}) \prod_{i=n,\ldots,2} P_{i,i-1}([v_i, \ldots, v_n]|D) \\ & - u(\mathbf{r}^+), 0\}, \end{aligned} \tag{7}$$

to make the choice of next route to test bias towards one that is likely to satisfy the cost constraint.

Next we derive an estimator on $P_{i,i-1}([v_i, \ldots, v_n]|D)$. In particular, during the sampling process, we keep the record on the policy-compliance test results for previous sampled routes in $D$. For any two ASes $v_i$, $v_{i-1}$ in $\mathbf{r}$, we calculate two values:

- $w_{i,i-1}(\mathbf{r})$: to count the number of routes in $D$ such that (1) AS $v_i$ announces its subroute to AS $v_{i-1}$, and (2) the announced subroute intersects with the subroute $[v_i, \ldots, v_n]$.
- $b_{i,i-1}(\mathbf{r})$: the number of routes in $D$ such that (1) AS $v_i$ does not announce its subroute to AS $v_{i-1}$, and (2) the unannounced subroute intresetcts with the subroute $[v_i, \ldots, v_n]$.

Then we can approximate $P_{i,i-1}([v_i, \ldots, v_n]|D)$ as:

$$P_{i,i-1}([v_i, \ldots, v_n]|D) = \frac{exp(w_{i,i-1}(\mathbf{r}))}{exp(w_{i,i-1}(\mathbf{r})) + exp(b_{i,i-1}(\mathbf{r}))}. \tag{8}$$

Plugging Equation 8 into Equation 7, we get a close-form estimation on $E(I(\mathbf{r})|D)$, the expected improvement of route $\mathbf{r}$ over sampled set of routes $D$.

**Two-level sequential sampling to quickly find near-optimal end-to-end route.** We now present our complete blackbox optimization algorithm for the optimal end-to-end route problem (Algorithm 2). Note that the algorithm also contains several subroutines. Due to the page limit, we only present two most important ones for reference (*i.e.*, Algorithm 4 and Algorithm 3).

In the initialization phase, the algorithm defines 4 empty sets: $R_F$ is the set of policy-noncompliant subroutes found during the test of route policy-compliance; $D$ is the set of sampled routes that have been evaluated for policy-compliance and cost-compliance; $R_c$ is the set of sampled routes that have not been evaluated for policy-compliance or cost-compliance; $R_q$ is the set of routes that are found to be both policy- and cost-compliant during the search process (Line 1).

**Algorithm 2:** Blackbox Optimization Algorithm.

1 $R_F = \emptyset, D = \emptyset, R_c = \emptyset, R_q = \emptyset$;
2 $\mathbf{r} = genericDijkstraAlg(G, v_s, v_t)$;
3 $e(\mathbf{r}) = PCTest(\mathbf{r}, R_F)$;
4 **if** $e(\mathbf{r}) == 1$ *and* $getCost(\mathbf{r} \leq B)$ **then**
5  **return** $\mathbf{r}$;
6 **else**
7  $\mathbf{r}_{last} = \mathbf{r}$;
8  $u(\mathbf{r}_{last}) = e(\mathbf{r}) \times f(\mathbf{r})$;
9  $D = D \cup (\mathbf{r}_{last}, u(\mathbf{r}_{last}))$;
10  $t_{local} = t_{global} = 0$;
11  **while** $t_{global} >= T_{global}$ *and* $R_q \neq \emptyset$ **do**
12  **if** $t_{local} >= T_{local}$ *and* $R_q == \emptyset$ **then**
13  $t_{local} = 0$;
14  $R = getGlobalCandidates(G, R_F, v_s, v_t)$;
15  **else**
16  $R = getLocalCandidates(G, \mathbf{r}_{last})$;
17  **foreach** $\mathbf{r} \in R$ **do**
18  **if** $isForbidden(\mathbf{r}, R_F)$ **then**
19  $D = D \cup (\mathbf{r}, 0)$;
20  $R = R - \{\mathbf{r}\}$;
21  $R_c = R_c \cup R$;
22  $\mathbf{r}_t = \text{argmax}_{\mathbf{r} \in R_c} E(I(\mathbf{r})|D)$, where $E(I(\mathbf{r})$ is computed in Equation 7;
23  $R_c = R_c - \{\mathbf{r}_t\}$;
24  $e(\mathbf{r}) = PCTest(\mathbf{r}, R_F)$;
25  $D = D \cup (\mathbf{r}_t, u(\mathbf{r}_t^g))$;
26  **if** $e(\mathbf{r}) == 1$ *and* $getCost(\mathbf{r} \leq B)$ **then**
27  $R_q = R_q \cup \{\mathbf{r}\}$;
28  $t_{local} = t_{local} + 1, t_{global} = t_{global+1}$;
29  $\mathbf{r}_{last} = \mathbf{r}_t$;
30  **return** $\text{argmax}_{(r) \in R_q} f(\mathbf{r})$;

As a starting point, the algorithm chooses an optimal end-to-end route computed by the general Dijkstra algorithm [21], and tests its policy-compliance and cost (Line 2-4, and Algorithm 4). If this route is indeed policy-compliant and does not exceed the client's budget, the algorithm simply stops since the optimal solution has been found (Line 5). Otherwise, the algorithm keeps the record of this route in set $D$ (Line 7-9), and enters the sequential sampling process (Line 11).

In each iteration of the sampling process, the algorithm first tries to do a "local exploration", *i.e.*, it samples a set of candidate routes $R$, each of which shares a subroute with $\mathbf{r}_{last}$, the last route that is tested for policy-compliance and cost-compliance (Line 16 and Algorithm 3). If the algorithm has already sampled locally for a total of $T_{local}$ iterations and still cannot find any route that is both policy-compliant and cost-compliant, it will do a "global exploration" by sampling a set of random candidate routes from $v_s$ to $v_t$ that differs from

the found policy-non-compliant subroutes in $R_F$ as much as possible, and resets the number of local exploration $t_{local}$ (Line 13-14).

**Algorithm 3:** Local exploration of candidate routes: $getLocalCandidates(G, \mathbf{r})$.

1 $R_c = \emptyset$;
2 $\mathbf{r} = [v_1, v_2, \ldots, v_n]$;
3 **foreach** $i = 2, \ldots, n-1$ **do**
4 $\quad E_G^i = E_G - \cup_{j=1,\ldots,i}\{(v_j, v_{j+1})\}$;
5 $\quad V_G^i = V_G$;
6 $\quad \mathbf{r}_i = [v_1, \ldots, v_i] \oplus genericDijkstra((V_G^i, E_G^i), v_i, v_n)$;
7 $\quad R_c = R_c \cup \{\mathbf{r}_i\}$;
8 **return** $R_c$;

After exploring candidate routes, the algorithm leverages Proposition 3 to only keep candidate routes that do not contain any subroute in $R_F$ (Line 17-20), merges them with candidate routes left before in $R_c$ (Line 21), and selects $\mathbf{r}$ from the remaining sampled routes whose expected improvement is the largest among all the sampled, yet unevaluated routes $R_c$i (Line 22). After that, the algorithm evaluates the policy-compliance and cost-compliance of $\mathbf{r}$, and added it to $R_q$ if it is both policy-compliant and cost-compliant (Line 24-27).

The sequential sampling process terminates after a total number of $T_{global}$ iterations or $R_q$ becomes non-empty. Then the route which has the largest utility value $f()$ is the resulting near-optimal, end-to-end route (Line 30).

**Algorithm 4:** Policy-compliance test of $\mathbf{r}$: $PCTest(\mathbf{r}, R_F)$.

1 Input: $\mathbf{r} = [v_1, v_2, \ldots, v_n]$;
2 **foreach** $i = n, \ldots, 2$ **do**
3 $\quad$ Call $selectRoute(match, [v_i, \ldots, v_n])$ at AS $v_i$;
4 $\quad$ Get $RIB_{i-1}$ at AS $v_{i-1}$;
5 $\quad$ **if** $[v_i, \ldots, v_n] \notin RIB_{i-1}$ **then**
6 $\quad\quad R_F = R_F \cup [v_{i-1}, v_i, \ldots, v_n]$;
7 $\quad\quad$ **return** 0;
8 **return** 1;

**Discussion.** Due to the combinatory nature of Problem 1, it is very challenging to derive an explicit convergence bound of our blackbox optimization algorithm. In order to guarantee that at least one policy-compliant and cost-compliant route is found, we explicitly specify it as part of the stopping condition. As such, in the worst case, our blackbox optimization algorithm still needs to enumerate a large number of routes. However, in practice, this worst case would rarely happen. This is because, with the help of the expected improvement acquisition function, our algorithm can smartly decide the next route to test for policy and cost compliance, substantially reducing the numbers of sample routes needed. In addition, by leveraging Proposition 3, our algorithm substantially reduces the ratio of sampled routes that require policy-compliance test between the client and ASes, which is the most expensive part of enumeration. We validate our argument on the efficiency and efficacy of our algorithm using real-world data. As we will show in the next section, in an interdomain network with over 60000 ASes and over 320000 AS-level links, in 80% experiment cases, the blackbox optimization algorithm can find a near-optimal policy-compliant end-to-end route by sampling less than 33 routes.

## V. Evaluation

In this section, we conduct extensive evaluations aiming to answer the following questions: 1. How fast can the Bayesian Optimization framework find the (near-)optimal end-to-end route? 2. How likely can the Bayesian Optimization framework find a (near-)optimal end-to-end route in a reasonable time?

### A. Experiment Setting

**Network Topology.** To evaluate our approach, we use the AS-level Internet topology derived from the CAIDA dataset [28] which includes 63361 ASes and 320978 AS-level links.

**Initial RIB.** Each AS follows the derived AS business relationship to set its local preference for BGP route selection by default, which provides the initial RIB information.

**Export Policies.** Each AS configures the following types of route export policies to determine whether announcing a received route to a neighbor AS: 1. *Neighboring Business Relationship*: An AS never exports a non-customer route to a non-customer neighbor; 2. *Blackhole*: For some security reason, *e.g.*, BGP hijacking prevention, an AS may not export any routes across a set of suspected ASes; 3. *Forbiden Segment*: For some security or business reason, an AS may not export any routes including some specific segments.

**Searching Algorithms.** We implement the following end-to-end route search algorithms: 1. *Shortest Path Enumeration* (SPE): Use YEN's algorithm to iterate K-shortest path in order until the first policy-compliant route is found; 2. *Software-Defined Internetworking Bayesian Optimization* (SDIBO): Use the Bayesian Optimization framework with the discrete EI acquisition function and the two-level sequential sampling approach to finding the near-optimal policy-compliant route.

**Intents.** We generate global end-to-end intents based on Internet traffic dataset [28]: we choose the top 10 ASes which receive the most traffic volume as destinations. For each selected destination AS, we choose top 200 source ASes sending the most traffic volume to this destination AS. We generate 2000 end-to-end intents in total and check the connectivity[3] in the derived AS-level Internet topology. Finally, we get 1620 effective end-to-end intents.

**Experiment Workflow.** We implement a BGP simulator to simulate the RIB computation and route announcement, and a prototype SDI client to find the optimal end-to-end route. We design the experiment as follows: First, we run our customized BGP simulator for this AS-level Internet topology until it converges, and collect the RIB of each AS as the initial prior knowledge. Then we feed the converged topology with

[3]We suspect the reason of disconnectivity is because the inferred AS relationship is incomplete.

the BGP RIBs to our prototype SDI client. After that, the prototype SDI client proposes a global end-to-end intent and tries to find the optimal policy-compliant route within the minimal number of policy-compliance tests by using one of end-to-end route search algorithms above.

**Policy-compliance Tests.** There are two methods to test whether a path is policy-compliant as a whole or at a certain AS as follows: 1. *Active Test*: The client actively sends a test request of route $\mathbf{r} = [v_1, \ldots, v_n]$ to the remote virtual BGP speaker at $v$ and observes whether $[v] \oplus \mathbf{r}$ appears in a given neighbor $u$. 2. *Local Test*: If the client has ever actively tested the visibility of a route for a node, it will cache the result. Next time when the client has to check it again, the client only need to locally check it using the cache.

### B. Searching Efficiency

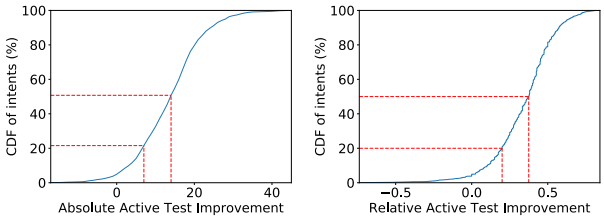
In this section, we evaluate the searching efficiency of different search algorithms, *i.e.*, how fast they can find the (near-)optimal end-to-end route.

**Metric.** We use the following metrics to measure the searching efficiency of an algorithm: 1. *Number of Active Tests*: It represents the number of active test requests sent before finding the solution. 2. *Number of Local Tests*: It represents the number of local tests (*i.e.*, cache lookups). 3. *Average Number of Active/Local Tests*: It represents the average number of active/local tests on a path for an intent. 4. *Absolute Active/Local Test Improvement*: It is calculated by subtracting the number of active/local tests for SPE by the number of active/local tests for SDIBO. 5. *Relative Active Test Improvement*: It is calculated by dividing the absolute active test improvement by the number of active tests for SPE.

**Early Termination Thresholds.** The route optimization problem is NP-hard. So in the worst case, whatever the algorithm is adopted, the client has to enumerate all simple paths from the source to the destination. It is not realistic in practice so we set two thresholds to terminate the searching: 1. *Maximum Number of Active Tests*: In practice, an active test action usually takes a long time. So we set a threshold (60) to limit the maximum number of such actions for each intent. 2. *Maximum Number of Local Tests*: Local test action is usually quick. However, in the worst case, one will enumerate all routes. Considering the scale of the Internet, it still may take intolerant time. So we also set a threshold (2500) to limit the maximum number of local tests for each intent.

**Results.** We first demonstrate the performance gain of SDIBO for each intent. Fig. 3 demonstrates the CDF curve of the absolute active test improvement and the relative active test improvement. As we can see, **SDIBO improves the active test in most cases and is likely to get substantial improvement**: for 80% of the requests, SDIBO can reduce more than 7 active tests and more than 20% of active tests made by SPE; for 50% requests, SDIBO can reduce more than 15 active tests and by 40% to 80% of active tests made by SPE, which yields an improvement of 1.6x to 5x on the execution time.

To further understand the performance of SDIBO, we draw a scatter figure where the x-coordinate represents the absolute active test improvement and y-coordinate represents the absolute local test improvement. As we can see in Fig. 4, most dots fall into the lower right region ($x > 0$, $y < 0$, red color), where SDIBO takes less active tests but more local tests. This indicates that **SDIBO searches routes more effectively and fully leverages the local information**. We also see that many dots even fall into the upper right region ($x > 0$, $y > 0$, green color), where SDIBO uses less active tests and fewer local tests, which further demonstrates the benefits of SDIBO's intelligent searching method.




**(a)** Absolute Improvement **(b)** Relative Improvement

**Fig. 3:** CDF of Active Test Improvement

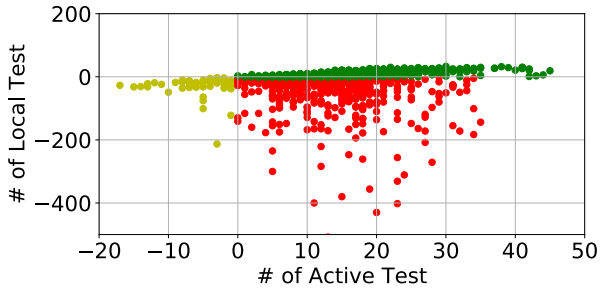



**Fig. 4:** Absolute Active/Local Test Improvement

### C. Searching Effectiveness

In this section, we demonstrate the search effectiveness of different search algorithms, *i.e.*, whether they can find a (near-)optimal end-to-end route in a reasonable time and how optimal the route is.

**Metrics.** We use the following metrics: 1. *Proportion of Effective Search*: It represents the proportion of intents where at least one policy-compliant end-to-end route is found before early termination. 2. *Proportion of Optimal Search*: It represents the number of intents for which the SDIBO finds the optimal policy-compliant end-to-end route, *i.e.*, the route has the same length as the one found by SPE if any.

**Early Termination Thresholds.** We use the same early termination thresholds as in the last section.

**Results.** Fig. 5 demonstrates the proportion of successful search of SPE and SDIBO with the aforementioned setting. We can see that within a given time scale (bounded by the thresholds), SDIBO can substantially improve the chance that a policy-compliant end-to-end route is found: 61% (SDIBO) to 24% (SPE), which yields a 2.5x improvement.

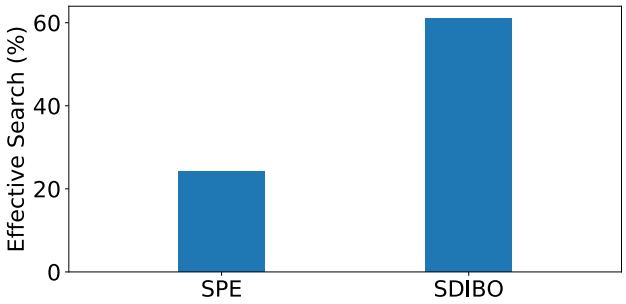



**Fig. 5:** Proportion of Effective Search.

We have also calculated the proportion of optimal search. Surprisingly, the proportion is 100%, *i.e.*, SDIBO is able to find a policy-compliant route with the same optimal length as SPE if any. While the result may be related to the selected objective function, this still demonstrates that **SDIBO has a very high probability to find an optimal policy-compliant end-to-end route within a reasonable time.**

## VI. Related Work

**BGP mechanisms for end-to-end route control.** The main mechanisms in BGP to support end-to-end route control are selective announcement [29], AS path prepending [30], BGP FlowSpec [2] and BGP communities [31]. However, they are limited due to the destination-based forwarding nature and the peering model of BGP. Selective announcement, AS path prepending and BGP FlowSpec are mainly used for an AS to control its inbound traffic. And BGP communities only enable the interaction between two peering ASes. As illustrated in the motivating example in Section I, BGP does not support a source AS or end user to affect the route selection of remote ASes.

**Recent interdomain routing protocols and systems.** Many interdomain routing protocols [32]–[38] and systems [3]–[7], [9], [10], [39], [40] have been proposed to provide more mechanisms and interfaces for end-to-end route control. Their design can be categorized into three classes. The first one is third-party composition [3]–[7], [41]. An important representative system of this category is SDX [5] and its variants [6], [7], [41]. A major limitation of SDX is that a client's routing control actions are only used within the exchange point without being announced to other ASes. Hence, enforcing clients' routing policies at different SDXes can cause correctness issues such as persistent forwarding loops. To resolve this, forwarding loop detection approaches (*e.g.*, [7], [41]) have been proposed, but they still have a problem with a large number of false-positive alerts.

The second category is tunnel-based overlay [9], [10], [42]–[44]. MIRO [9], ARROW [10] and RCS [43] are the most recent systems in this category. The basic idea is to let a stub AS interact with a remote AS to select routes different from the BGP route, and then build a tunnel between stub and remote ASes to utilize the negotiated routes. As such, these systems have datapath overhead such as tunneling processing on each data packet. In addition, to ensure the convergence of interdomain routing, tunnels built and used in these systems are not announced to other ASes. This use-announcement inconsistency is usually not preferred by network operators for security reasons, *e.g.*, violation of data traffic regulation may happen without being detected.

Third, Google and Facebook [39], [40] also develop flexible peering systems to take route selection back to edge by overriding the BGP. However, these systems can only control the next-hop AS of traffic forwarding.

In contrast, in this paper, we propose SDI, a novel, systematic, low datapath overhead route control model, in which a network exposes a programmable interface to allow clients to define the interdomian routes of the network.

**Bayesian optimization.** Our blackbox optimization algorithm is built on the Bayesian optimization framework [15], [16], [19]. Bayesian optimization is a powerful framework for optimizing objective functions that are expensive to evaluate. However, when the decision variables are discrete, it is shown that the commonly rounding approach has a negative impact on the efficiency of BO [19]. A recent work proposes to use the graph to encode the discrete search space and applies spectral graph theory to solve the corresponding expected improvement maximization problem [45]. However, this design has severe scalability issues. In contrast, we leverage important properties from interdomain routing algebra to derive an accurate estimation of the expected improvement of an end-to-end route. As demonstrated in Section V, our EI estimation is accurate and fast even in large networks.

**Routing algebra.** Routing algebra has been used to analyze the stability, correctness and optimality of routing protocols [17], [18], [21], [34]. To the best of our knowledge, we are the first to leverage properties derived from interdomain routing algebra to improve the efficiency of blackbox end-to-end route optimization.

## VII. Conclusion

In this paper, we provide the first, systematic formulation of the software-defined internetworking model, in which a network exposes a programmable interface to allow clients to define the interdomain routes of the network, but still maintains the control of its export policies. We define the optimal end-to-end SDI routing problem, prove its NP-hardness, and develop a blackbox optimization algorithm to efficiently find the near-optimal, policy-compliant end-to-end route with a small number of sample routes. We evaluate our algorithm extensively using real interdomain network topologies to demonstrate its efficiency and efficacy.

## Acknowledgment

The authors thank Shenshen Chen, Haizhou Du, Linghe Kong, Alan Liu, Ennan Zhai and Yan Zhu for their help during the preparation of this paper. The authors also thank the anonymous reviewers for their valuable comments. This research is supported in part by NSFC grants #61702373, #61672385, #61701347, and #61902266; NSF award #1440745; Facebook Research Award, Google Research Award, and the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001.

## REFERENCES


[1] Y. Rekhter, S. Hares, and D. T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Jan. 2006. [Online]. Available: https://rfc-editor.org/rfc/rfc4271.txt

[2] P. R. Marques, J. Mauch, N. Sheth, B. Greene, R. Raszuk, and D. R. McPherson, "Dissemination of Flow Specification Rules," RFC 5575, Aug. 2009. [Online]. Available: https://rfc-editor.org/rfc/rfc5575.txt

[3] V. Kotronis, X. Dimitropoulos, and B. Ager, "Outsourcing the routing control logic: better internet routing based on sdn principles," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*. ACM, 2012, pp. 55–60.

[4] K. Lakshminarayanan, I. Stoica, S. Shenker, and J. Rexford, *Routing as a Service*. Computer Science Division, University of California Berkeley, 2004.

[5] A. Gupta, L. Vanbever, M. Shahbaz, S. P. D. B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A Software Defined Internet Exchange," in *Proceedings og SIGCOMM 2014*. IEEE, August 2014, pp. 233–239.

[6] A. Gupta, R. MacDavid, R. Birkner, M. Canini, N. Feamster, J. Rexford, and L. Vanbever, "An industrial-scale software defined internet exchange point." in *NSDI*, vol. 16, 2016, pp. 1–14.

[7] R. Birkner, A. Gupta, N. Feamster, and L. Vanbever, "Sdx-based flexibility or internet correctness?: Pick two!" in *Proceedings of the Symposium on SDN Research*. ACM, 2017, pp. 1–7.

[8] G. Asharov, D. Demmler, M. Schapira, T. Schneider, G. Segev, S. Shenker, and M. Zohner, "Privacy-preserving interdomain routing at internet scale," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 3, pp. 147–167, 2017.

[9] W. Xu and J. Rexford, "Multi-path interdomain routing," in *In SIGCOMM*. Citeseer, 2006.

[10] S. Peter, U. Javed, Q. Zhang, D. Woos, T. Anderson, and A. Krishnamurthy, "One tunnel is (often) enough," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 99–110.

[11] D. N. Bauer, D. Dechouniotis, C.-X. Dimitropoulos, and A. Kind, "Valley-free shortest path method," Mar. 15 2011, uS Patent 7,907,596.

[12] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang, "On as-level path inference," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1. ACM, 2005, pp. 339–349.

[13] L. Gao and J. Rexford, "Stable internet routing without global coordination," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 6, pp. 681–692, 2001.

[14] P. Gill, M. Schapira, and S. Goldberg, "A survey of interdomain routing policies." *Computer Communication Review*, vol. 44, no. 1, pp. 28–34, 2014.

[15] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[16] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[17] T. G. Griffin and J. L. Sobrinho, "Metarouting," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 1–12.

[18] J. L. Sobrinho, "Network routing with path vector protocols: Theory and applications," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 49–60.

[19] E. C. Garrido-Merchán and D. Hernández-Lobato, "Dealing with integer-valued variables in bayesian optimization with gaussian processes," *arXiv preprint arXiv:1706.03673*, 2017.

[20] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking (ToN)*, vol. 10, no. 2, pp. 232–243, 2002.

[21] J. L. Sobrinho, "Algebra and algorithms for qos path computation and hop-by-hop routing in the internet," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, vol. 2. IEEE, 2001, pp. 727–735.

[22] C. Bessiere, F. Koriche, N. Lazaar, and B. O'Sullivan, "Constraint acquisition," *Artificial Intelligence*, vol. 244, pp. 315 – 342, 2017, combining Constraint Solving with Mining and Learning. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370215001162

[23] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[24] O. Alipourfard, H. H. Liu, J. Chen, S. Venkataraman, M. Yu, and M. Zhang, "Cherrypick: Adaptively unearthing the best cloud configurations for big data analytics," in *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 2017, pp. 469–482.

[25] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[26] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[27] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints." in *ICML*, 2014, pp. 937–945.

[28] CAIDA, "As relationships and internet traffic dataset," http://www.caida.org/data/as-relationships/, 2016.

[29] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain traffic engineering with bgp," *IEEE Communications magazine*, vol. 41, no. 5, pp. 122–128, 2003.

[30] R. K. Chang and M. Lo, "Inbound traffic engineering for multihomed ass using as path prepending," *IEEE network*, vol. 19, no. 2, pp. 18–25, 2005.

[31] B. Donnet and O. Bonaventure, "On bgp communities," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 55–59, 2008.

[32] R. R. Sambasivan, D. Tran-Lam, A. Akella, and P. Steenkiste, "Bootstrapping evolvability for inter-domain routing with d-bgp," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 474–487.

[33] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs, "R-bgp: Staying connected in a connected world." USENIX, 2007.

[34] J. L. Sobrinho, D. Fialho, and P. Mateus, "Stabilizing bgp through distributed elimination of recurrent routing loops," in *Network Protocols (ICNP), 2017 IEEE 25th International Conference on*. IEEE, 2017, pp. 1–10.

[35] T. Holterbach, S. Vissicchio, A. Dainotti, and L. Vanbever, "Swift: Predictive fast reroute," in *Proceedings of the 2017 conference on ACM SIGCOMM 2017 Conference*. ACM, 2017, pp. 460–473.

[36] R. Mahajan, D. Wetherall, and T. E. Anderson, "Mutually controlled routing with independent isps." in *NSDI*, 2007.

[37] P. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet routing," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 111–122, 2009.

[38] R. Chandra, Y. Rekhter, T. J. Bates, and D. Katz, "Multiprotocol Extensions for BGP-4," RFC 4760, Jan. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc4760.txt

[39] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng, "Engineering egress with edge fabric: steering oceans of content to the world," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 418–431.

[40] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain *et al.*, "Taking the edge off with espresso: Scale, reliability and programmability for global internet peering," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 432–445.

[41] A. Dethise, M. Chiesa, and M. Canini, "Prelude: Ensuring interdomain loop-freedom in˜ sdn-enabled networks," *arXiv preprint arXiv:1806.09566*, 2018.

[42] X. Yang, D. Clark, and A. W. Berger, "Nira: a new inter-domain routing architecture," *IEEE/ACM Transactions on Networking (ToN)*, vol. 15, no. 4, pp. 775–788, 2007.

[43] Y. Wang, J. Bi, and K. Zhang, "A sdn-based framework for fine-grained inter-domain routing diversity," *Mobile Networks and Applications*, vol. 22, no. 5, pp. 906–917, 2017.

[44] H. Wang, Y. R. Yang, P. H. Liu, J. Wang, A. Gerber, and A. Greenberg, "Reliability as an interdomain service," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 229–240.

[45] C. Oh, J. M. Tomczak, E. Gavves, and M. Welling, "Combinatorial bayesian optimization using graph representations," *arXiv preprint arXiv:1902.00448*, 2019.