Dynamic Resource Allocation and User IP Routing in Cloud COmputing

Abstract

This Reserach explores Dynamic Resource Allocation in cloud computing to achieve cost optimization and enhanced performance. Leveraging Machine Learning algorithms, our approach adapts resource allocation in real time, demonstrating improved efficiency and cost-effectiveness.

PREPARED FOR

MegaMinds IT

PREPARED BY

Tanish Seth

EXECUTIVE SUMMARY

As modern networks evolve to meet the demands of increasing user traffic and diverse applications, the need for efficient resource allocation and intelligent IP routing becomes paramount. This research project delves into the intricate dynamics of dynamic resource allocation and user IP routing, aiming to enhance network performance, scalability, and user experience.

Objectives and Contributions:

The primary objective of this research is to develop a comprehensive system that dynamically allocates resources within a network and optimizes IP routing for end-users. Our contributions focus on achieving a seamless balance between resource utilization efficiency and user-centric routing, ensuring optimal network performance.

Key Concepts and Methodology:

We propose an innovative approach to dynamic resource allocation that considers the real-time demands of network applications. Simultaneously, our research addresses user IP routing, emphasizing the importance of personalized routing paths for improved service quality. Machine learning algorithms are incorporated to adaptively allocate resources and optimize routing decisions based on historical and real-time data.

1. Project Overview

Objectives:

The primary objective of this research project is to investigate and develop innovative solutions for the dynamic allocation of resources in a network environment while simultaneously optimizing IP routing for end-users. The project

aims to achieve a seamless balance between resource efficiency and user-centric routing, ultimately enhancing network performance and user experience.

Scope:

The project will encompass the following key areas:

Dynamic Resource Allocation:

- Explore adaptive algorithms for real-time resource allocation based on changing network workloads.
- Address challenges associated with heterogeneous application needs and diverse user demands.

User IP Routing Optimization:

- Investigate methodologies for personalized IP routing paths to improve service quality.
- Integrate machine learning algorithms for adaptive routing decisions based on historical and real-time user data.

Methodology:

The research will employ a multifaceted methodology, including:

- **Literature Review:** Thoroughly review existing research on dynamic resource allocation and IP routing to identify gaps and challenges.
- **Algorithm Development:** Propose and develop adaptive algorithms for resource allocation and user IP routing, integrating machine learning components.
- **Simulation and Experimentation:** Validate proposed algorithms through simulations and experiments to assess their effectiveness in diverse network scenarios.

Deliverables:

The project will deliver the following key outcomes:

Research Paper: A comprehensive research paper outlining the findings, methodologies, and contributions in the field of dynamic resource allocation and user IP routing.
Algorithms: Adaptive algorithms for dynamic resource allocation and user IP routing, accompanied by documentation.

2. Obstacles

1.Algorithm Complexity:

- **Description:** The development of adaptive algorithms for dynamic resource allocation and user IP routing may encounter challenges related to complexity, impacting implementation timelines.
- **Management Strategy:** Regular code reviews, collaboration with experienced team members, and leveraging existing frameworks will be implemented to manage and simplify algorithmic complexities.

2. Data Availability:

- **Description:** The success of the project heavily relies on the availability of diverse and representative data for simulations and experiments.
- **Management Strategy:** Establish collaborations with industry partners, leverage publicly available datasets, and implement data augmentation techniques to ensure a robust dataset for testing and validation.

3. Resource Constraints:

- **Description:** Limited access to computational resources or unforeseen budget constraints may affect the scale and efficiency of simulations and experiments.
- **Management Strategy:** Prioritize resource utilization, explore cloud computing options, and collaborate with academic or industry partners to access additional computational resources if required.

4. Dynamic Network Conditions:

- **Description:** Real-world network conditions are highly dynamic, making it challenging to simulate and predict all possible scenarios accurately.
- **Management Strategy:** Implement continuous monitoring and adaptation mechanisms in algorithms to respond effectively to dynamic changes. Regularly update simulations based on real-world data to enhance the model's adaptability.

5. Stakeholder Expectations:

- **Description:** Misalignment of stakeholder expectations regarding project outcomes and timelines may arise.

- **Management Strategy:** Establish clear communication channels, conduct regular progress meetings, and provide transparent updates to stakeholders. Adjust project timelines and expectations based on feedback and evolving requirements.

6. Algorithm Evaluation Metrics:

- **Description:** Determining suitable metrics for evaluating the effectiveness of algorithms in dynamic resource allocation and user IP routing may be challenging.
- **Management Strategy**: Conduct an extensive literature review to identify commonly used metrics, collaborate with experts in the field, and establish a robust evaluation framework that aligns with project goals.

3. Technical Obstacles

1. Integration Challenges:

- **Obstacle:** Ensuring seamless integration between the developed adaptive algorithms and existing network infrastructure.
- **Mitigation:** Conduct thorough compatibility testing, collaborate with network administrators, and implement well-documented integration guidelines.

2. Data Security Concerns:

- **Obstacle:** Managing sensitive user data for algorithm training while ensuring data security and privacy.
- **Mitigation:** Implement robust encryption methods, adhere to data protection regulations, and anonymize user data during algorithm development.

3. Scalability Issues:

- **Obstacle:** Adapting algorithms to handle the scale of larger networks without compromising performance.
- **Mitigation:** Employ scalable algorithms, leverage cloud computing resources, and conduct performance testing under varying network sizes.

4. Real-time Adaptability:

- **Obstacle:** Ensuring real-time adaptability of algorithms to dynamic changes in network conditions.
- **Mitigation:** Implement continuous monitoring, leverage machine learning models with low latency, and optimize algorithms for real-time responsiveness.

5. Algorithm Robustness:

- **Obstacle:** Developing algorithms robust enough to handle diverse applications and unforeseen network scenarios.
- **Mitigation:** Conduct rigorous testing under various conditions, employ machine learning model interpretability techniques, and iteratively refine algorithms based on simulation results.

6. Protocol Compatibility:

- **Obstacle:** Ensuring compatibility with different network protocols and communication standards.
- **Mitigation:** Conduct comprehensive protocol compatibility testing, collaborate with networking experts, and design algorithms with protocol-agnostic features.

7. Machine Learning Integration:

- **Obstacle:** Integrating machine learning components seamlessly into the resource allocation and IP routing algorithms.
- **Mitigation:** Leverage established machine learning frameworks, collaborate with experts in the field, and ensure clear documentation for integration steps.

6. Hardware

The proposed software for dynamic resource allocation and user IP routing is designed to be compatible with a range of hardware configurations. The hardware compatibility includes, but is not limited to:

Servers and Data Center Infrastructure:

- The software is compatible with standard server architectures commonly used in data centers, including x86-based servers and hardware from various vendors.

Network Devices:

- Compatibility extends to network devices such as routers, switches, and load balancers commonly found in enterprise network environments.

Cloud Computing Platforms:

- The software is designed to seamlessly integrate with popular cloud computing platforms, including but not limited to Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

Virtualization Technologies:

- Compatibility is ensured with virtualization technologies such as VMware, KVM, and Hyper-V, allowing the software to operate in virtualized environments.

General-Purpose Hardware:

- The software is adaptable to general-purpose hardware commonly used in both enterprise and small to medium-sized business (SMB) environments.

7. Software

The development of the proposed software for dynamic resource allocation and user IP routing will involve the use of various software technologies. The software stack includes, but is not limited to:

Programming Languages:

- **Python:** Used for algorithm development, data processing, and scripting.

Machine Learning Libraries:

- **TensorFlow:** Utilized for implementing machine learning components for adaptive resource allocation.
- **Scikit-learn:** Employed for developing machine learning models and algorithms.

Networking Libraries and Protocols:

- **OpenFlow:** Implemented for programmable and software-defined networking.
- **TCP/IP and UDP:** Utilized for communication between network devices and software components.

Database Management:

- **MySQL/PostgreSQL:** Employed for data storage and retrieval related to historical network conditions and resource usage.

Simulation Tools:

- **ns-3 (Network Simulator 3):** Used for simulating network conditions and evaluating algorithm performance in controlled environments.

Virtualization Technologies:

- **Docker:** Employed for containerization to enhance portability and ease of deployment.
- **VirtualBox:** Utilized for local development and testing of virtualized environments.

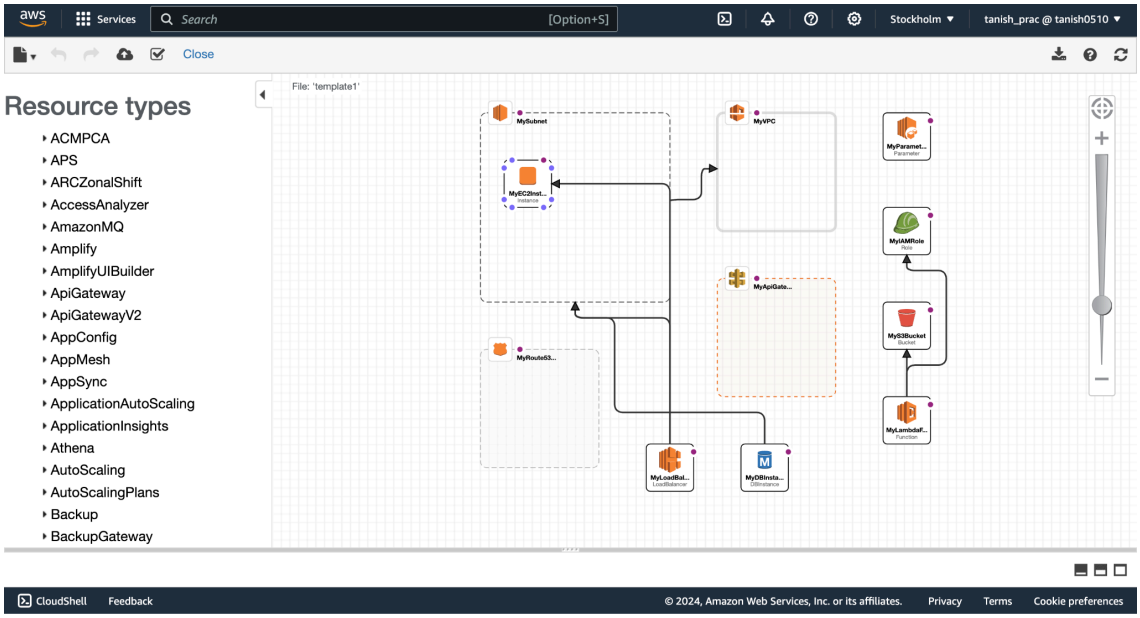
Version Control:

- **Git:** Implemented for version control and collaborative development.

Development Frameworks:

- **Flask (Python):** Employed for developing lightweight and efficient web-based interfaces.

Architecture Diagram:



Parameters:

MyVPC (AWS::EC2::VPC):

- **CidrBlock:** The IP address range for the VPC (default: "10.0.0.0/16").
- **EnableDnsSupport and EnableDnsHostnames:** Enabling DNS support and hostnames.

MySubnet (AWS::EC2::Subnet):

- **VpcId:** Reference to the VPC created (!Ref MyVPC).
- **CidrBlock:** The IP address range for the subnet (default: "10.0.0.0/24").

MyDBInstance (AWS::RDS::DBInstance):

- Configuration for a MySQL RDS instance.
- DBInstanceIdentifier, AllocatedStorage, DBInstanceClass, etc.

MyEC2Instance (AWS::EC2::Instance):

- Configuration for an EC2 instance.
- SubnetId, InstanceType, ImageId, KeyName, etc.

MyS3Bucket (AWS::S3::Bucket):

- Basic configuration for an S3 bucket.

MyLambdaFunction (AWS::Lambda::Function):

- Configuration for a Lambda function.
- Handler, Role, FunctionName, Runtime, etc.

MyApiGateway (AWS::ApiGateway::RestApi):

- Configuration for an API Gateway.
- Name, FailOnWarnings, etc.

MyRoute53HostedZone (AWS::Route53::HostedZone):

- Configuration for a Route 53 hosted zone.
- Name: "example.com."

MyLoadBalancer (AWS::ElasticLoadBalancingV2::LoadBalancer):

- Configuration for an Elastic Load Balancer.
- Subnets, SecurityGroups, etc.

MyIAMRole (AWS::IAM::Role):

- Configuration for an IAM role used by Lambda.
- AssumeRolePolicyDocument with a trust policy for Lambda.

MyParameter (AWS::SSM::Parameter):

- Configuration for an SSM parameter storing configuration.

Outputs:

MyVPCOutput: Export of the VPC ID.

MyDBInstanceOutput: Export of the RDS instance endpoint address.
MyEC2InstanceOutput: Export of the EC2 instance ID.
MyS3BucketOutput: Export of the S3 bucket name.
MyLambdaFunctionOutput: Export of the Lambda function name.
MyApiGatewayOutput: Export of the API Gateway ID.
MyRoute53HostedZoneOutput: Export of the Route 53 hosted zone ID.
MyLoadBalancerOutput: Export of the Load Balancer ID.

Code:

```yaml
AWSTemplateFormatVersion: "2010-09-09"

Resources:
 MyVPC:
   Type: "AWS::EC2::VPC"
   Properties:
     CidrBlock: "10.0.0.0/16"
     EnableDnsSupport: true
     EnableDnsHostnames: true

 MySubnet:
   Type: "AWS::EC2::Subnet"
   Properties:
     VpcId: !Ref MyVPC
     CidrBlock: "10.0.0.0/24"

 MyDBInstance:
   Type: "AWS::RDS::DBInstance"
   Properties:
     DBInstanceIdentifier: "MyDBInstance"
     AllocatedStorage: 20
     DBInstanceClass: "db.t2.micro"
     Engine: "mysql"
     EngineVersion: "5.7"
     MasterUsername: "admin"
     MasterUserPassword: "adminpassword"
     DBSubnetGroupName: !Ref MySubnet

 MyEC2Instance:
   Type: "AWS::EC2::Instance"
   Properties:
     SubnetId: !Ref MySubnet
     InstanceType: "t2.micro"
     ImageId: "ami-0c55b159cbfafe1f0"
     KeyName: "my-key-pair"

 MyS3Bucket:
   Type: "AWS::S3::Bucket"

 MyLambdaFunction:
   Type: "AWS::Lambda::Function"
   Properties:
     Handler: "index.handler"
     Role: !GetAtt MyIAMRole.Arn
     FunctionName: "MyLambdaFunction"
     Runtime: "nodejs14.x"
     Code:
       S3Bucket: !Ref MyS3Bucket
       S3Key: "lambda-code.zip"

 MyApiGateway:
```

```yaml
  Type: "AWS::ApiGateway::RestApi"
  Properties:
    Name: "MyApiGateway"
    FailOnWarnings: "true"

MyRoute53HostedZone:
  Type: "AWS::Route53::HostedZone"
  Properties:
    Name: "example.com."

MyLoadBalancer:
  Type: "AWS::ElasticLoadBalancingV2::LoadBalancer"
  Properties:
    Subnets: [!Ref MySubnet]
    SecurityGroups: [!GetAtt MyEC2Instance.SecurityGroups.0]

MyIAMRole:
  Type: "AWS::IAM::Role"
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Principal:
            Service: "lambda.amazonaws.com"
          Action: "sts:AssumeRole"

MyParameter:
  Type: "AWS::SSM::Parameter"
  Properties:
    Type: "String"
    Description: "Parameter for storing configuration"
    Name: "/MyProject/Config/Parameter"
    Value: "parameter-value"

Outputs:
MyVPCOutput:
  Value: !Ref MyVPC
  Export:
    Name: "MyVPC"

MyDBInstanceOutput:
  Value: !GetAtt MyDBInstance.Endpoint.Address
  Export:
    Name: "MyDBInstanceEndpoint"

MyEC2InstanceOutput:
  Value: !Ref MyEC2Instance
  Export:
    Name: "MyEC2Instance"

MyS3BucketOutput:
  Value: !Ref MyS3Bucket
  Export:
    Name: "MyS3Bucket"

MyLambdaFunctionOutput:
  Value: !Ref MyLambdaFunction
  Export:
    Name: "MyLambdaFunction"

MyApiGatewayOutput:
  Value: !Ref MyApiGateway
  Export:
    Name: "MyApiGateway"

MyRoute53HostedZoneOutput:
  Value: !Ref MyRoute53HostedZone
  Export:
    Name: "MyRoute53HostedZone"
```

```
MyLoadBalancerOutput:
  Value: !Ref MyLoadBalancer
  Export:
    Name: "MyLoadBalancer"
```

Ethical Considerations:

Privacy and Data Security:

Ensuring the privacy and security of user data is of paramount importance in this research. All data collected and utilized in simulations and experiments will be anonymized and handled in compliance with relevant data protection regulations. Rigorous encryption methods will be implemented to safeguard sensitive user information during algorithm development and training.

Informed Consent:

In scenarios where user data or feedback is directly involved, informed consent protocols will be established. Participants will be fully informed about the nature of their involvement, the purpose of data collection, and how their information will be used. Their consent will be obtained explicitly, and they will have the option to withdraw from the study at any time.

Transparent Communication:

Transparent communication will be maintained throughout the project, ensuring that stakeholders, users, and collaborators are kept informed about the research objectives, methodologies, and outcomes. Regular updates will be provided, and any changes to the project plan will be communicated promptly.

Risk Management:

Risk Identification:

Algorithmic Complexity:

- **Mitigation:** Regular code reviews, collaboration with experienced team members, and leveraging existing frameworks will be implemented to manage and simplify algorithmic complexities.

Data Availability:

- **Mitigation:** Establish collaborations with industry partners, leverage publicly available datasets, and implement data augmentation techniques to ensure a robust dataset for testing and validation.

Resource Constraints:

- **Mitigation**: Prioritize resource utilization, explore cloud computing options, and collaborate with academic or industry partners to access additional computational resources if required.

Dynamic Network Conditions:

- **Mitigation:** Implement continuous monitoring and adaptation mechanisms in algorithms to respond effectively to dynamic changes. Regularly update simulations based on real-world data to enhance the model's adaptability.

Stakeholder Expectations:

- **Mitigation:** Establish clear communication channels, conduct regular progress meetings, and provide transparent updates to stakeholders. Adjust project timelines and expectations based on feedback and evolving requirements.

Risk Assessment:

A formal risk assessment will be conducted at regular intervals to evaluate the impact and likelihood of identified risks. Risks will be categorized based on severity, and mitigation strategies will be refined as needed.

Risk Monitoring and Adaptation:

Continuous monitoring of potential risks will be implemented throughout the project lifecycle. Any emerging risks will be promptly assessed, and mitigation strategies will be adapted to address evolving circumstances. Regular reviews and updates to the risk management plan will ensure its effectiveness.

Dynamic Service Provisioning in Elastic Optical Networks With Hybrid Single-/Multi-Path Routing

Zuqing Zhu, *Senior Member, IEEE*, Wei Lu, Liang Zhang, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—Empowered by the optical orthogonal frequency-division multiplexing (O-OFDM) technology, flexible online service provisioning can be realized with dynamic routing, modulation, and spectrum assignment (RMSA). In this paper, we propose several online service provisioning algorithms that incorporate dynamic RMSA with a hybrid single-/multi-path routing (HSMR) scheme. We investigate two types of HSMR schemes, namely HSMR using online path computation (HSMR-OPC) and HSMR using fixed path sets (HSMR-FPS). Moreover, for HSMR-FPS, we analyze several path selection policies to optimize the design. We evaluate the proposed algorithms with numerical simulations using a Poisson traffic model and two mesh network topologies. The simulation results have demonstrated that the proposed HSMR schemes can effectively reduce the bandwidth blocking probability (BBP) of dynamic RMSA, as compared to two benchmark algorithms that use single-path routing and split spectrum. Our simulation results suggest that HSMR-OPC can achieve the lowest BBP among all HSMR schemes. This is attributed to the fact that HSMR-OPC optimizes routing paths for each request on the fly with considerations of both bandwidth utilizations and lengths of links. Our simulation results also indicate that the HSMR-FPS scheme that use the largest slots-over-square-of-hops first path-selection policy obtains the lowest BBP among all HSMR-FPS schemes. We then investigate the proposed algorithms' impacts on other network performance metrics, including network throughput and network bandwidth fragmentation ratio. To the best of our knowledge, this is the first attempt to consider dynamic RMSA based on both online path computation and offline path computation with various path selection policies for multipath provisioning in O-OFDM networks.**

***Index Terms*—Bandwidth blocking probability (BBP), bandwidth fragmentation ratio, dynamic routing, elastic optical networks, hybrid single-/multi-path routing (HSMR), modulation and spectrum assignment (RSA).**

I. Introduction

OVER the past decade, Internet traffic has been growing at an annual rate of more than 30%, and the consequent bandwidth (BW) demands stimulated research and development for highly flexible and scalable networking technologies. Recent research advance has experimentally demonstrated transmission of 20 Tb/s signals on a single fiber with the dense wavelength division multiplexing (DWDM) technology [1]. However, due to the coarse granularity of DWDM channels (typically at 50 or 100 GHz), wavelength-routed DWDM network infrastructure [2] has been considered rigid with limited elasticity and flexibility in the optical layer. When the support of highly dynamic traffic becomes necessary, repeated optical-to-electrical-to-optical (O/E/O) conversions are required to forward the data to electrical routers for packet switching. These O/E/O conversions usually incur additional capital expenditures (CAPEX) and operational expenditures (OPEX) owing to relatively high equipment cost and power consumption [3]. To this end, it is highly desirable to develop networking technology that provides subwavelength granularity in the optical layer.

A. Optical Orthogonal Frequency-Division Multiplexing (O-OFDM)-Based Elastic Optical Networks

The O-OFDM technology [4], [5] packs subcarrier frequency slots overlapping with each other in the optical spectrum. Since the subcarriers are orthogonal in the frequency domain, data modulation on them can be recovered without interference at the receiver [5]. Hence, O-OFDM can achieve subwavelength granularity, by using elastic BW allocation that manipulates the subcarrier slots. Specifically, a BW-variable O-OFDM transponder can assign an appropriate number of subcarrier slots to serve a lightpath request using just-enough BW [6]. Moreover, the modulation level of the subcarrier slots can be adaptive to accommodate various quality of transmission [7], [8]. The elastic nature of O-OFDM imposes sophisticated network planning and provisioning procedures for efficient and robust operations. To address these, we need to develop routing, modulation-level, and spectrum assignment (RMSA) algorithms for network control and management. If modulation level is not adaptive in the networks, RMSA reduces to routing and spectrum assignment (RSA).

Planning and provisioning of elastic O-OFDM networks have started to attract research interests just recently [9]–[15]. When the lightpath requests are known *a prior*, offline planning of O-OFDM networks with RSA/RMSA under the spectrum-continuity constraints is known as nonpolynomial complete [9]. An RSA heuristic that combined shortest path routing and first-fit spectrum assignment was discussed in [10]. In [9], several integer linear programming (ILP) models were formulated and solved for offline RMSA, and a heuristic based on shortest path routing and simulated annealing optimization was proposed to reduce the computation complexity. Jinno *et al.* [11] proposed a BW-efficient RMSA, which examined K-shortest routing paths

Manuscript received August 01, 2012; revised October 22, 2012; accepted November 09, 2012. Date of publication November 15, 2012; date of current version December 14, 2012. This work was supported in part by the Program for New Century Excellent Talents in University under Project NCET-11-0884, and the Natural Science Foundation of Anhui Province under Project 1208085MF88.

Z. Zhu, W. Lu, and L. Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: zqzhu@ieee.org; luwei11@mail.ustc.edu.cn; mnizh@mail.ustc.edu.cn).

N. Ansari is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: nirwan.ansari@njit.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/JLT.2012.2227683

for each request and then chose the one with the lowest available contiguous slots. Wang *et al.* [12] formulated an ILP model for offline RSA and designed two heuristics, K-shortest path routing and balanced-load spectrum assignments and shortest path routing and maximum spectrum reuse assignments. Online provisioning of O-OFDM networks considers how to serve time-variant lightpath requests with dynamic RSA/RMSA. By leveraging the generalized multiprotocol label switching signaling mechanism, a distributed dynamic RMSA was proposed in [13], which chose the least congested routing path and performed first-fit spectrum assignments. Sone *et al.* [14] developed a dynamic RSA that used a metric to quantify the consecutiveness of available slots among relevant fibers. The investigation in [15] considered spectrum defragmentation during online provisioning with dynamic RSA.

B. Service Provisioning With Multipath Routing

From the aforementioned discussion, we can see that most of the previous works on O-OFDM networks were based on single-path routing. However, for online provisioning, we may have difficulty to serve certain large-BW requests with single-path routing due to the BW limitation, thus resulting in high request blocking probability [16]. It is known that multipath routing provides increased throughput and utilizes the network resources more efficiently [2], [16]. Researchers have previously considered to include multipath routing support in SONET/SDH transport systems [17]–[19]. Multipath routing is also explicitly supported by several standardized routing protocols, such as the open shortest path first [20] and the routing information protocol [21].

What is more promising is that with the elastic nature of O-OFDM, network nodes can easily split data traffic over multiple routing paths and support multipath provisioning. Recently, Dahlfort *et al.* proposed a split-spectrum approach [22], which could be considered as a multipath approach as a request might be divided into several subflows for transmitting on noncontiguous optical spectra. However, since this approach still restricted all subflows of a request to be routed over the same path, it may not fully explore the benefits of multipath provisioning. In order to support multipath routing and traffic-splitting in O-OFDM networks, each switching node requires a wavelength-selective switch (WSS) that can add/drop subcarrier channels using relatively low BW granularity. Thanks to the technology advances in liquid crystal-on-silicon (LCOS) WSS, switching granularity at 12.5 GHz can be realized [23]. Barros *et al.* proposed a colorless LCOS WSS node architecture in which each add/drop port had both narrow-band and wide-band modes [24]. Hence, BW-flexible switching could be achieved with low loss. Such WSS provides an important enabling technology for supporting multipath routing in O-OFDM networks.

C. Our Contributions

In this paper, we propose several dynamic service provisioning algorithms that incorporate a hybrid single-/multi-path routing (HSMR) scheme. To the best of our knowledge, this is the first attempt to consider dynamic RMSA based on both online path computation and offline path computation with various path selection policies for multipath provisioning in O-OFDM networks. We evaluate the proposed algorithms with numerical simulations using a Poisson traffic model. The simulation results have demonstrated that the proposed HSMR schemes can effectively reduce the bandwidth blocking probability (BBP) of dynamic RMSA, as compared to two benchmark algorithms that use single-path routing or split spectrum. We also evaluate our proposed algorithms in terms of other performance metrics, such as network throughput and network BW fragmentation ratio. Notice that for multipath provisioning, the differential delay between the routing paths can lead to the requirement for additional buffers on the end nodes [25]. How to address the differential delay during multipath provisioning is out of the scope of this paper. We expect that the issue can be resolved with either the split-spectrum approach that restricts all subflows of a request to be routed over the same path [22] or a multipath provisioning approach that considers the differential delay constraint.

The rest of this paper is organized as follows. Section II formulates the problem of service provisioning using dynamic RMSA with HSMR. The dynamic RMSA algorithm that incorporates HSMR with online path computation is discussed in Section III. Section IV explains the dynamic RMSA with HSMR using fixed path sets. The numerical simulation setup and results for performance evaluation are discussed in Section V. Finally, Section VI summarizes the paper.

II. Service Provisioning Using Dynamic RMSA With HSMR

In this section, we formulate the problem of service provisioning using dynamic RMSA with HSMR, explain operation constraints, and define design metrics.

Consider the physical network topology $G(V, E, B, D)$, where V is the node set, E is the fiber link set, each fiber link can accommodate B frequency slots at most, and D represents the lengths of $e \in E$. We assume that the BW of each subcarrier slot is unique as $\mathrm{BW}_{\mathrm{slot}}$ GHz. The capacity of a slot is $M \cdot C_{\mathrm{slot}}$, where M is the modulation level in terms of bits per symbol, and C_{slot} denotes the capacity of a slot when the modulation is BPSK ($M = 1$) and is a function of $\mathrm{BW}_{\mathrm{slot}}$ [9]. In this study, we assume that M can be 1, 2, 3, and 4 for BPSK, QPSK, eight quadrature-amplitude modulation (8-QAM), and 16-QAM, respectively. For a lightpath request $\mathrm{LR}(s, d, C)$ from node s destined to d for a capacity of C, the provisioning algorithm using dynamic RMSA with the HSMR scheme needs to determine a set of routing paths $\{R_{s,d,i}\}$ to serve the request, where i is the index of each routing path. Note that for different i, the routing paths $R_{s,d,i}$ can be identical, but since their spectrum allocations are not contiguous, more than one sets of O-OFDM transceivers are required and this scheme is considered as a multipath one. In this study, we propose two algorithms to determine $\{R_{s,d,i}\}$ for each request, i.e., one with online path computation and the other with fixed path sets. The details of the algorithms will be discussed in Sections III and IV.

We denote the length of a link $e \in E$ as d_e, $d_e \in D$. When the transmission distance of the ith routing path $R_{s,d,i}$ is known, we derive the modulation level M_i as

$$M_i = mlvl\left(\sum_e d_e\right), e \in R_{s,d,i} \tag{1}$$

where $mlvl(\cdot)$ returns the highest modulation level that a transmission distance can support. Specifically, we assume that each modulation M can support a maximum transmission distance based on the receiver sensitivities [7], and when the distance of $R_{s,d,i}$ permits, we always assign the highest modulation level to guarantee high spectral efficiency.

Then, we figure out the load distribution $\{C_i\}$ on each routing path based on the network status, which should satisfy

$$C = \sum_i C_i. \tag{2}$$

The number of contiguous slots N_i we need to assign on each path is

$$N_i = \left\lceil \frac{C_i}{M_i \cdot C_{\text{slot}}} \right\rceil + N_{\text{GB}} \tag{3}$$

where N_{GB} is the number of slots for the guard band. Note that when splitting the traffic, we have to take the cost that more slots will be used for the guard band. In the context of this study, we assume that $N_{\text{GB}} = 1$ and this guard band is inserted as the highest indexed slot in the spectrum assignment of each connection. Therefore, in the following sections, we do not mention the guard band explicitly, but when we refer to the size of a block of contiguous available slots, we actually mean the available size after deducting the guard band.

The last step of dynamic RMSA is the spectrum assignment to finalize the allocations of contiguous slots along the fiber links on $R_{s,d,i}$. We assume that there is not any spectrum converter in the network. For each fiber link $e \in E$, we define a bit-mask b_e consisting of B bits. When the jth slot on e is taken, $b_e[j] = 1$; otherwise, $b_e[j] = 0$. When assigning the frequency slots, we define a bit-mask a_i for each path, which also contains B bits. Then, the spectrum assignment on $R_{s,d,i}$ becomes the problem of finding N_i contiguous bits in a_i to turn on based on all current $b_e, e \in R_{s,d,i}$. Finally, the RMSA with HSMR for $LR(s,d,C)$ is $\{\{R_{s,d,i}, M_i, a_i\}, i = 1, \ldots\}$. We say LR is blocked, if we cannot find a feasible $\{\{R_{s,d,i}, M_i, a_i\}, i = 1, \ldots\}$ for it.

The dynamic RMSA has to satisfy the spectrum nonoverlapping and spectrum contiguousness.

Spectrum Nonoverlapping Constraint:

$$\text{sum}(a_i \cap b_e) = 0, \quad \forall e \in R_{s,d,i}. \tag{4}$$

Spectrum Contiguousness Constraint:

$$\text{sum}(a_i \cap ROR(a_i, 1)) = \begin{cases} N_i - 1, & N_i < B \\ B, & N_i = B \end{cases} \tag{5}$$

where $\text{sum}(\cdot)$ is the function to add all bits in a bit mask together, $\cap$ is the bitwise AND operator, and $ROR(\cdot)$ is the circular bit-right-shift operator. In this study, the objective of the service provisioning is to minimize requests' BBP.

Definition 1 (Slot block):: We define a slot block as a block of contiguous subcarrier slots in the optical spectrum.

Definition 2 (BW allocation granularity):: To avoid a request LR from being split over too many paths, we define a BW allocation granularity as g slots. Specifically, when LR is provisioned over more than one routing paths, the minimum size of the slot blocks we can allocate on each path is g. Note that increasing

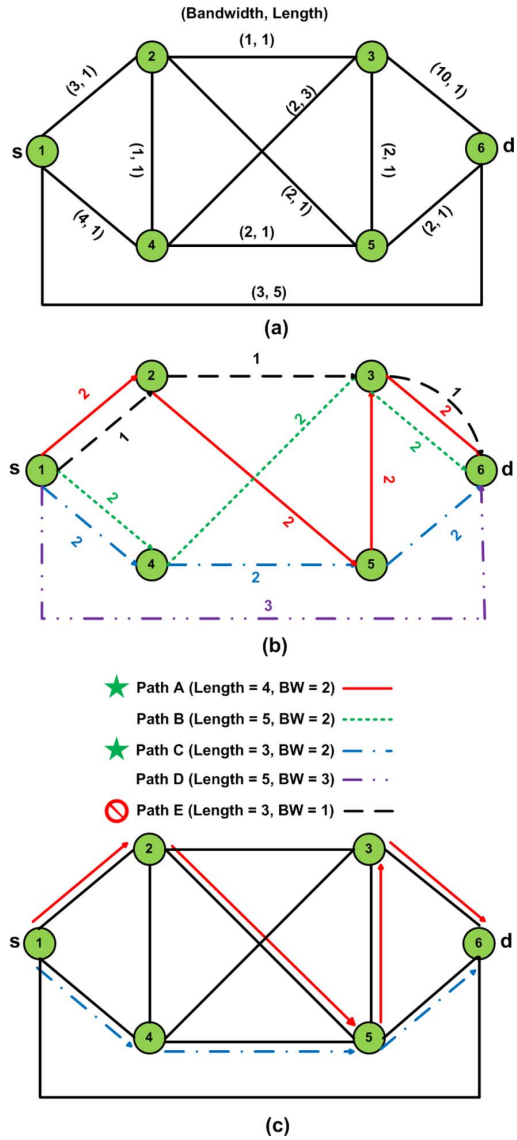


Fig. 1. Example of service provisioning using multipath routing with a BW allocation granularity of $g = 2$. (a) Network topology G(V,E,B,D). (b) Path computation results. (c) Elastic multipath provisioning scheme for a request with $\text{BW} = 4$ and $g = 2$.

g discourages multipath provisioning schemes, and will eventually lead to a single-path-only scenario when g is comparable to the largest size of the requests. From the viewpoint of a BW-flexible WSS [24], g can be the smallest switching granularity that it can handle. From the viewpoint of network management, g can correspond to the smallest switching granularity that the network operator is willing to offer.

Fig. 1 illustrates an intuitive example of the usage of BW allocation granularity g in service provisioning with HSMR. Fig. 1(a) shows a network topology with six nodes, and we label each link with (BW and length), i.e., its available BW in terms of the number of slots and its link length. For simplicity, we assume that each link only has one slot block available. With this $G(V,E,B,D)$, we will not be able to serve a request from node 1 to 6 for a BW of four contiguous slots with a single routing path. Hence, we calculate multiple routing paths and label them with the sizes of available slot blocks, as shown in Fig. 1(b). It

is clear that Path E: 1-2-3-6 is not a qualified path for a BW allocation granularity $g = 2$ because it only has a slot block of one slot available. We select Paths A and C for less lengths and provision the request for a BW of four slots from node 1 to 6 successfully [as shown in Fig. 1(c)].

Definition 3 (Bandwidth blocking probability):: BBP is defined as the ratio of blocked connection BW versus total request BW. BBP is a commonly used metric for assessing the performance of service provisioning algorithms.

Definition 4 (BW fragmentation ratio): BW fragmentation is another interesting factor to investigate in dynamic RMSA [15]. BW fragmentation, which is similar to the file system fragmentation in computer storage, usually refers to the existing of non-aligned, isolated and small-sized slot blocks in the spectrum of elastic optical networks. Since these slot blocks are neither contiguous in the spectral domain nor aligned along fiber links, it is hard for the network operator to get them utilized for future connection requests, especially for those going across multiple hops and/or requesting for large BW. Inspired by the fragmentation ratio definition for computer storage [26], we define the BW fragmentation ratio of a link e as

$$\eta_e = \begin{cases} 1 - \frac{\mathrm{Max\,Block}(b_e)}{B - \mathrm{sum}(b_e)}, & \mathrm{sum}(b_e) < B \\ 0, & \mathrm{sum}(b_e) = B \end{cases} \tag{6}$$

where $\mathrm{Max\,Block}(\cdot)$ returns the maximum size of available slot blocks in b_e. In light of previous works on network BW fragmentation in elastic optical networks [27], [28], the fragmentation ratio F_η of the network G is defined as the average of link fragmentation ratio

$$F_\eta = \frac{\sum_e \eta_e}{|E|}. \tag{7}$$

III. Dynamic RMSA With HSMR Using Online Path Computation

We first investigate a dynamic RMSA-HSMR algorithm that considers link spectrum usage on the fly with an online path computation. Specifically, we convert $G(V, E, B, D)$ to a virtual topology $G'(V, E, D')$ based on link spectrum usage, where V and E are the same as those in G, but each link weight d'_e in set D' is recalculated as

$$d'_e = \begin{cases} +\infty, & \mathrm{Max\,Block}(b_e) < g \\ w_e \cdot \frac{sum(b_e)+g}{B}, & \mathrm{Max\,Block}(b_e) \geq g \end{cases} \tag{8}$$

where g is the BW allocation granularity, $\mathrm{sum}(b_e)$ returns the current spectrum usage of link e, and w_e is calculated from d_e with

$$w_e = M_{\max} - mlvl(d_e) + 1 \tag{9}$$

where $M_{\max}$ is the highest modulation level that can be supported in the network, and $mlvl(\cdot)$ is defined in (1) to return the highest modulation level that a transmission distance can support. Since a higher modulation means a less number of slots to allocate and better utilization of network spectrum resource, we quantify d_e with $mlvl(d_e)$ and map it to w_e to assist routing path calculation in the virtual topology. A link e is omitted from the online path computation, if it does not have a block of available contiguous slots with the size $\geq g$. Otherwise, the link weight d'_e is proportional to the product of w_e and the number of used slots $\mathrm{sum}(b_e)$. *Algorithm 1* shows the detailed procedure in implementing the proposed algorithm, and we calculate the routing path set for the path selection of each request using network status on the fly.

Algorithm 1 Dynamic RMSA With HSMR Using Online Path Computation

collect link status of $G(V, E, B, D)$;
while the network is operational **do**
  restore network resources used by expired requests;
  update link weights $\{d'_e\}$ based on the current network status, using (8)–(9);
  construct virtual topology $G'(V, E, D')$ with $\{d'_e\}$;
  get parameters of an incoming request $LR(s, d, C)$;
  calculate K-shortest routing paths from s to d in $G'(V, E, D')$;
  sort the paths based on the weighted total distances $\sum_e d'_e$;
  for all paths in the ascending order **do**
    determine the highest modulation level M_i for the path with its real distance $\sum_e d_e$ using (1);
    for all available slot blocks with sizes $\geq g$ **do**
      allocate capacity C_i to slot blocks with (3);
      if $\sum_i C_i = C$ **then**
        break inner and outer for-loops;
      end if
    end for
  end for
  if $\sum_i C_i < C$ **then**
    revert all the spectrum allocations;
    mark the request as blocked;
  end if
end while

IV. Dynamic RMSA With HSMR Using Fixed Path Sets

The major drawback of online path computation is the high computation complexity, as we need to reconstruct the virtual topology G' for each request and to perform path computation on the fly. Dynamic RMSA with HSMR can also be realized using fixed path sets, where the path-set containing K-shortest routing paths for each s--d pair in G are precomputed before operating the network. Hence, the overhead from path computation can be effectively reduced. *Algorithm 2* shows the detailed procedure in implementing the proposed algorithm. In provisioning a lightpath request $\mathrm{LR}(s, d, C)$, we sort the paths in the path set of s--d based on a path-selection policy and then process the paths one by one. We will elaborate on the details of the path-selection policies in the following. In performing spectrum allocations for C, we prefer a single routing path in a best effort scenario. Specifically, the largest slot block in the top-ranked path is selected first, and only if the largest slot block in the top-ranked path cannot support C in full, a multipath scheme is applied.

We evaluate the following path-selection policies:

a) *Shortest path first (SPF):* We select the routing path candidates in the ascending order based on the total transmission distance of the routing path, $\sum_e d_e, e \in R_{s,d,i}$.

b) *Most slots first (MSF):* We select the paths in the descending order based on the total available slots on each of them. The number of available slots on a path is

$$\mathrm{bw}(R_{s,d,i}) = B - \mathrm{sum}\left(\bigcup_{e \in R_{s,d,i}} b_e\right). \tag{10}$$

c) *Largest slots-over-hops first (LSoHF):* We select the paths in the descending order based on the metric

$$\mathrm{soh}(R_{s,d,i}) = \frac{\mathrm{bw}(R_{s,d,i})}{\mathrm{hop}(R_{s,d,i})} \tag{11}$$

where $\mathrm{hop}(R_{s,d,i})$ returns the number of hops of $R_{s,d,i}$.

d) *Largest slots-over-square-of-hops first (LSoSHF):* We order the paths in the descending order based on the metric

$$\mathrm{sosh}(R_{s,d,i}) = \frac{\mathrm{bw}(R_{s,d,i})}{\sqrt{\mathrm{hop}(R_{s,d,i})}}. \tag{12}$$

e) *Most left slots first (MLSF):* We order the paths in the descending order based on the metric

$$ls(R_{s,d,i}) = \mathrm{bw}(R_{s,d,i}) - mlvl(R_{s,d,i}) \tag{13}$$

where $mlvl(R_{s,d,i})$ returns the number of contiguous slot a capacity C uses on $R_{s,d,i}$ with the highest possible modulation-level M_i according to (3). Note that $ls(R_{s,d,i})$ can return a negative value.

Algorithm 2 Dynamic RMSA With HSMR Using Fixed Path Sets

Phase 1: Routing path precomputation by a K-shortest path algorithm

```
collect link status of G(V, E, B, D);
for all s-d pairs in G, s, d ∈ V do
    calculate K-shortest routing paths;
    record the paths;
end for
```

Phase 2: Dynamic RMSA provisioning with HSMR

```
while the network is operational do
    restore network resources used by expired requests;
    get parameters of an incoming request LR(s, d, C);
    load the pre-computed routing paths from s to d;
    sort the paths based on a path-selection policy;
    for all paths in the sorted order do
        determine the highest modulation level M_i for the path with its distance using (1);
        for all available slot-blocks with sizes ≥ g do
            allocate capacity C_i to slot-blocks with (3);
            if Σ_i C_i = C then
                break inner and outer for-loops;
            end if
        end for
    end for
    if Σ_i C_i < C then
        revert all the spectrum allocations;
        mark the request as blocked;
    end if
end while
```

V. Performance Evaluation

In this section, we discuss simulation results and evaluate the performance of the proposed algorithms for RMSA with HSMR. Fig. 2 shows the network topologies, NSFNET, and US Backbone, which we used in simulations for performance evaluation of the proposed service provisioning algorithms. The lightpath requests, $\mathrm{LR}(s, d, C, \Delta t)$, arrive one by one, following a Poisson process with an average arrival rate of λ requests per time-unit, and the lifetime Δt of each request follows the negative exponential distribution with an average of $1/\mu$ time units. Hence, the traffic load can be quantified with λ/μ in Erlangs. The s--d pair of each request $\mathrm{LR}(s, d, C, \Delta t)$ is randomly selected from the nodes in the simulation topology. The BW capacity C is also randomly selected according to a uniform distribution within 12.5–200 Gb/s. The transmission reaches of BPSK, QPSK, 8-QAM, and 16-QAM signals are determined based on the experimental results reported in [7]. Table I summarizes the simulation parameters.

We first perform simulations with $g = 1$ to compare the proposed HSMR algorithms to two benchmark algorithms. Between them, one benchmark uses single-path routing, which is the exhaustive path-search RMSA (EPS-RMSA), and the other uses the split-spectrum approach [22]. The EPS-RMSA is a greedy algorithm designed by ourselves, in which we compute all feasible routing paths for the s--d pair of a request and try to serve it with an exhaustive path search. Note that we use the first-fit spectrum assignment for the proposed HSMR algorithms in the simulations.

Figs. 3 and 4 show the simulation results on BBP in the NSFNET and US Backbone topologies, respectively. We observe that the BBP curves in both figures follow the same trend. When comparing the results from the HSMR schemes with those from the benchmark algorithms, we observe that the HSMR schemes achieve significantly lower BBP for all traffic loads in both topologies.

The results also suggest that the dynamic RMSA with HSMR using online path computation (HSMR-OPC) achieves the lowest BBP among all HSMR schemes. This is attributed to the fact that HSMR-OPC optimizes routing paths for each request on the fly with considerations of the BW utilizations and lengths of links. Among the HSMR schemes that use fixed path sets, the scheme that employs the shortest path first path-selection policy (HSMR-FPS-SPF) has the highest BBP because that selecting the shortest paths to serve requests can make the network load distribution unbalanced. The HSMR-FPS schemes that employ load-balancing path-selection policies, such as the most slots first (HSMR-FPS-MSF) and the most left slots first (HSMR-FPS-MLSF), serve the requests in a more load-balanced way and outperforms HSMR-FPS-SPF. However, HSMR-FPS-MSF and HSMR-FPS-MLSF have the same drawback that they tend to use routing paths that are less congested regardless of how many hops they have. For RMSA, serving a request with a path that has more hops means that the actual usage of subcarrier slots in the network is larger, as more

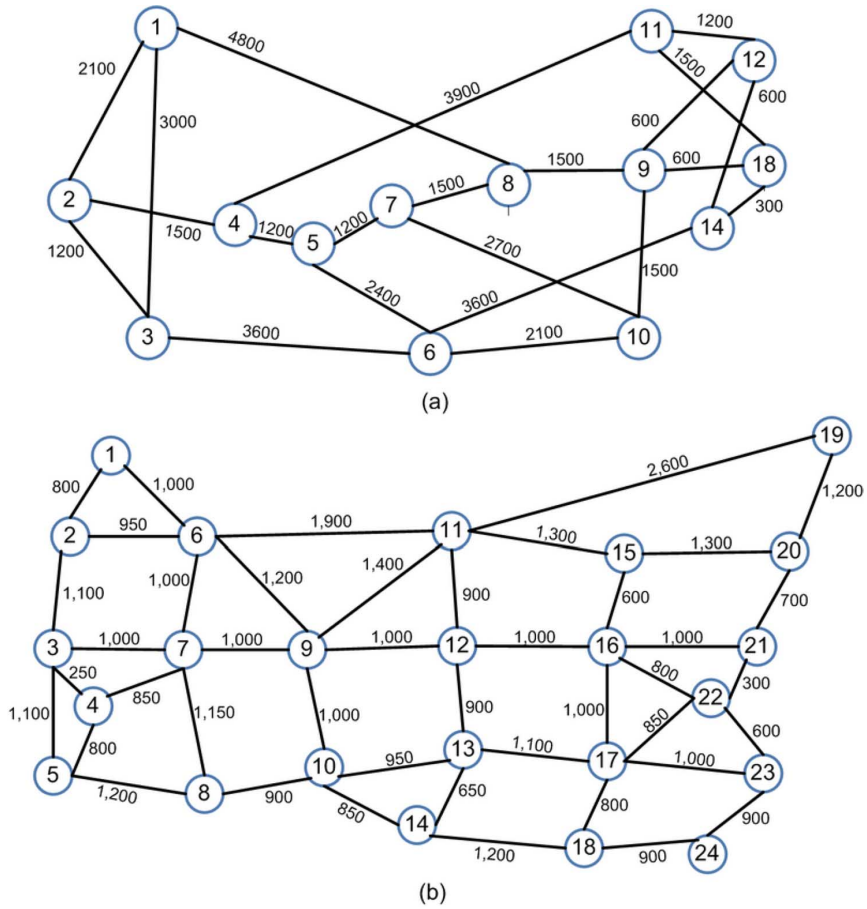


Fig. 2. Topologies used in simulations with fiber length in kilometers marked on links. (a) NSFNET topology (14 nodes). (b) US Backbone topology (24 nodes).

TABLE I
SIMULATION PARAMETERS

B, number of frequency slots per link	300
BW_{slot}, bandwidth of a frequency slot	12.5 GHz
C_{slot}, capacity of a frequency slot with $M = 1$	12.5 Gb/s
N_{GB}, number of slots for guard-band per connection	1
g, bandwidth allocation granularity	1 - 5
Transmission reach of BPSK ($M = 1$)	9,600 km
Transmission reach of QPSK ($M = 2$)	4,800 km
Transmission reach of 8-QAM ($M = 3$)	2,400 km
Transmission reach of 16-QAM ($M = 4$)	1,200 km
K, number of path candidates for a s-d pair	5
Range of requested capacity (C)	12.5 - 200 Gb/s

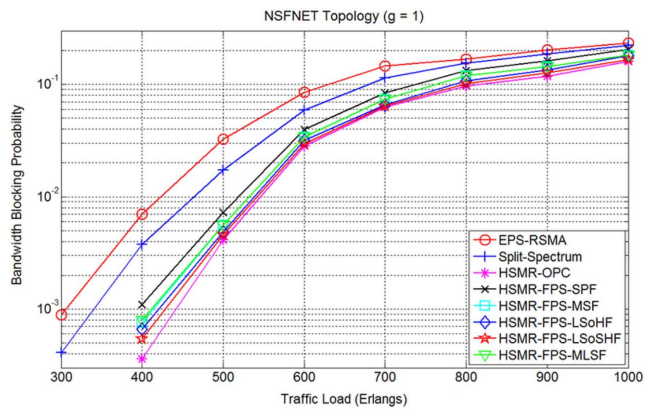


Fig. 3. Simulation results on BBP versus traffic load in NSFNET using $g = 1$ for HSMR schemes.

slots have to be allocated on additional hops. This is similar to the routing and wavelength assignment in fixed grid DWDM networks [29]. The HSMR-FPS schemes that use the largest slots-over-hops first and the largest slots-over-square-of-hops first (HSMR-FPS-LSoSHF) path-selection policies consider the balance between path length and link utilization, and hence achieve better BBP performance. The HSMR-FPS-LSoSHF obtains the best BBP performance among all HSMR-FPS schemes and its BBP performance is just slightly worse than that of HSMR-OPC.

We then investigate the BBP performance of HSMR schemes by changing g from 1 to 5. Figs. 5 and 6 show the results for the HSMR-OPC scheme in the two topologies. The BBP results of the other HSMR schemes follow the same trend. The results suggest that the BBP performance of HSMR schemes gets worse with a larger BW allocation granularity g. The reason behind this trend is that increasing g reduces the number of path splitting (i.e., splitting the traffic of a request over multiple paths) in the HSMR schemes. Therefore, g can be a convenient control parameter for the network operator to balance the tradeoff between request blocking and network management complexity.

The simulation results on average network throughput are plot in Fig. 7 for using HSMR-OPC ($g = 1-5$) in the US Backbone topology. We observe that when $g \leq 3$, the HSMR-OPC

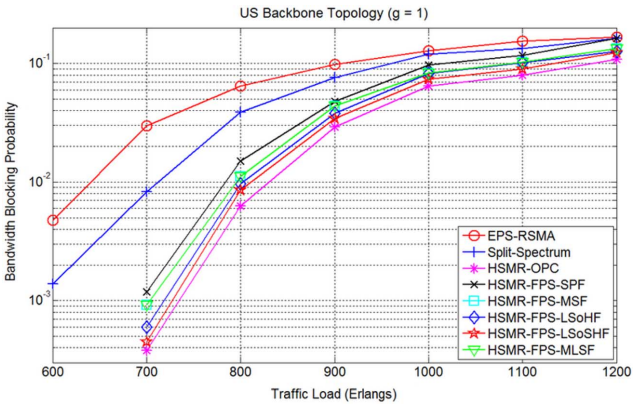


Fig. 4. Simulation results on BBP versus traffic load in US Backbone using $g = 1$ for HSMR schemes.

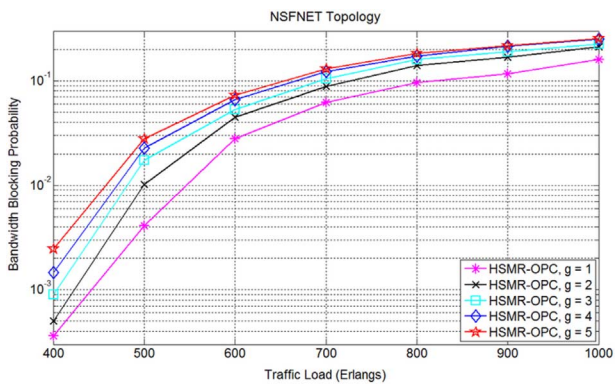


Fig. 5. Simulation results on BBP versus traffic load in NSFNET for HSMR-OPC scheme using $g = 1$–5.

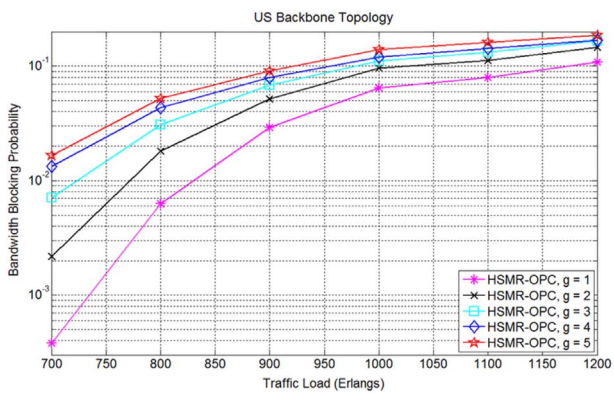


Fig. 6. Simulation results on BBP versus traffic load in US Backbone for HSMR-OPC scheme using $g = 1$–5.

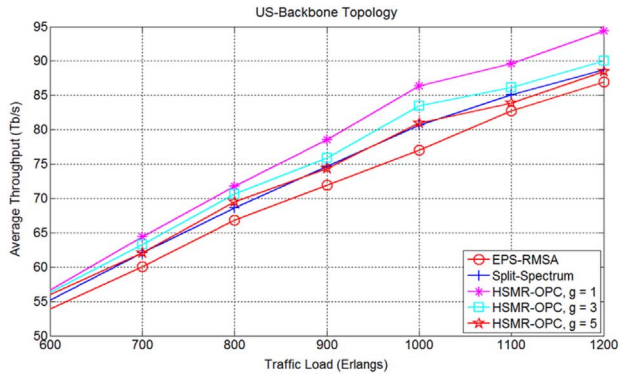


Fig. 7. Simulation results on average network throughput versus traffic load in US Backbone for HSMR-OPC scheme using $g = 1$–5.

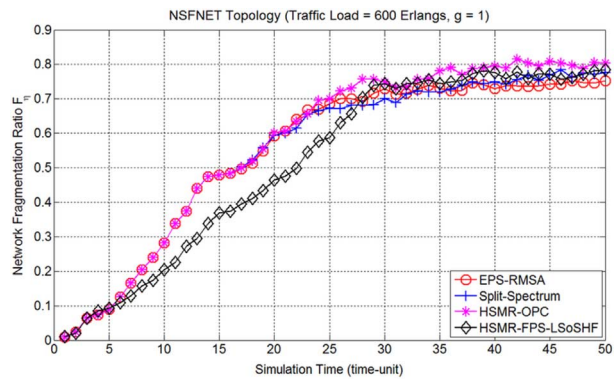


Fig. 8. Simulation results on network fragmentation ratio in NSFNET for HSMR schemes using traffic load at 600 Erlangs and $g = 1$.

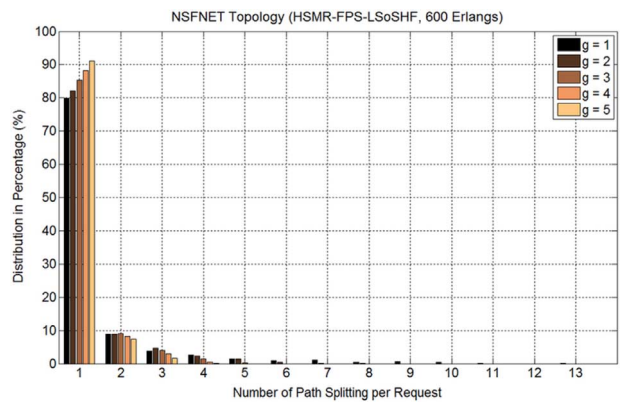


Fig. 9. Simulation results on the distribution of path splitting per request in NSFNET using the HSMR-FPS-LSoSHF scheme.

scheme achieves larger network throughput as compared to the benchmarks. The network throughput achieved by HSMR-OPC with $g = 5$ is comparable with that by the split-spectrum approach. We also study the proposed algorithms' impacts on BW fragmentation in the network. When we fix the traffic load at 600 Erlangs and set $g = 1$, Fig. 8 plots the network fragmentation ratio [defined in (6) and (7)] versus simulation time. We observe that the network fragmentation ratio from the HSMR-FPS-LSoSHF scheme increases slower than those from the two benchmark algorithms. Due to the fact that HSMR-OPC calculates routing paths on the fly, the network fragmentation ratio from it increases faster than those from the two benchmarks.

Finally, we investigate the distribution of the number of path splitting per request for the HSMR schemes. Our simulation results indicate that the distributions for different HSMR schemes are similar, and so we choose the HSMR-FPS-LSoSHF to illustrate the trend. Fig. 9 shows the distributions from the simulations using the NSFNET topology, when the traffic load is fixed at 600 Erlangs and $g = 1$–5. We can see that even for the worst case with $g = 1$, 79.80% of the requests are still served by a single routing path and the largest number of path splitting per

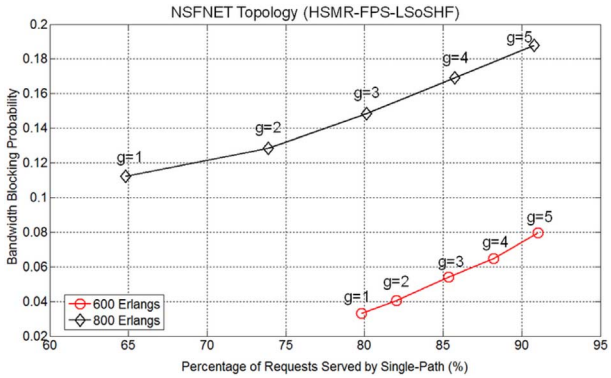


Fig. 10. BBP versus percentage of requests served by single-path in NSFNET using the HSMR-FPS-LSoSHF scheme.

request is 13. As expected, g has a clear effect on the distribution of path splitting per request. Specifically, choosing a larger g can make more requests be served by a single path and hence reduce the complexity of network management. However, as shown in Figs. 5 and 6, a larger g also results in worse BBP. Therefore, there is a tradeoff between BBP and network management complexity for our proposed HSMR schemes. Fig. 10 investigates this tradeoff for traffic loads at 600 and 800 Erlangs. The results indicate that g can be a convenient control parameter for a network operator to balance the tradeoff mentioned previously.

VI. Conclusion

In this paper, we have proposed several online service provisioning algorithms that incorporated dynamic RMSA with a HSMR scheme. Two types of HSMR schemes have been investigated: 1) HSMR-OPC, and 2) HSMR-FPS. Moreover, for HSMR-FPS, we analyzed several path selection policies to optimize the design. The proposed algorithms were evaluated with numerical simulations using a Poisson traffic model and two mesh network topologies. The simulation results verified that the proposed HSMR schemes could effectively reduce the BBP of dynamic RMSA, as compared to two benchmark algorithms that used single-path routing and split spectrum. Among all HSMR schemes, HSMR-OPC achieved the lowest BBP, while the HSMR-FPS scheme that used the HSMR-FPS-LSoSHF path-selection policy obtained the lowest BBP among all HSMR-FPS schemes. We also investigated the proposed algorithms' impacts on other network performance metrics, including network throughput and network BW fragmentation ratio. The study on the distribution of the number of path splitting per request showed that the majority of the requests were still served over a single routing path with the proposed HSMR schemes.

References

[1] J. Cai, "20 Tbit/s transmission over 6860 km with sub-Nyquist channel spacing," *J. Lightw. Technol.*, vol. 30, no. 4, pp. 651–657, Feb. 2012.
[2] B. Mukherjee, *Optical WDM Networks*. New York: Springer-Verlag, 2006.
[3] S. J. B. Yoo, "Energy efficiency in the future internet: The role of optical packet switching and optical label switching," *IEEE J. Sel. Topics Quantum Electron.*, vol. 17, no. 2, pp. 381–393, Mar. 2011.
[4] W. Shieh, X. Yi, and Y. Tang, "Transmission experiment of multi-gigabit coherent optical OFDM systems over 1000 km SSMF fibre," *IEEE Electron. Lett.*, vol. 43, no. 3, pp. 183–185, Feb. 2007.
[5] J. Armstrong, "OFDM for optical communications," *J. Lightw. Technol.*, vol. 27, no. 3, pp. 189–204, Feb. 2009.
[6] H. Takara, T. Goh, K. Shibahara, K. Yonenaga, S. Kawai, and M. Jinno, "Experimental demonstration of 400 Gb/s multi-flow, multi-rate, multi-reach optical transmitter for efficient elastic spectral routing," in *Proc. 37th Eur. Conf. Opt. Commun.*, Sep. 2011, pp. 1–3.
[7] A. Bocoi, M. Schuster, F. Rambach, M. Kiese, C.-A. Bunge, and B. Spinnler, "Reach-dependent capacity in optical networks enabled by OFDM," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2009, pp. 1–3.
[8] B. Kozicki, H. Takara, Y. Sone, A. Watanabe, and M. Jinno, "Distance-adaptive spectrum allocation in elastic optical path network (slice) with bit per symbol adjustment," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2010, pp. 1–3.
[9] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *J. Lightw. Technol.*, vol. 29, no. 9, pp. 1354–1366, May 2011.
[10] W. Zheng, Y. Jin, W. Sun, and W. Hu, "On the spectrum-efficiency of bandwidth-variable optical OFDM transport networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2010, pp. 1–3.
[11] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," *IEEE Commun. Mag.*, vol. 48, no. 8, pp. 138–145, Aug. 2010.
[12] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1503–1511.
[13] N. Sambo, F. Cugini, G. Bottari, P. Iovanna, and P. Castoldi, "Distributed setup in optical networks with flexible grid," in *Proc. 37th Eur. Conf. Opt. Commun.*, Sep. 2011, pp. 1–3.
[14] Y. Sone, A. Hirano, A. Kadohata, M. Jinno, and O. Ishida, "Routing and spectrum assignment algorithm maximizes spectrum utilization in optical networks," in *Proc. 37th Eur. Conf. Opt. Commun.*, Sep. 2011, pp. 1–3.
[15] K. Wen, Y. Yin, D. J. Geisler, S. Chang, and S. J. B. Yoo, "Dynamic on-demand lightpath provisioning using spectral defragmentation in flexible bandwidth networks," in *Proc. Eur. Conf. Opt. Commun.*, Sep. 2011, pp. 1–3.
[16] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[17] D. Cavendish, K. Murakami, S.-H. Yun, O. Matsuda, and M. Nishihara, "New transport services for next-generation SONET/SDH systems," *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 80–87, May 2002.
[18] K. Zhu, H. Zang, and B. Mukherjee, "Exploiting the benefit of virtual concatenation technique to the optical transport networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2003, pp. 363–364.
[19] S. Huang, C. Martel, and B. Mukherjee, "Survivable multipath provisioning with differential delay constraint in telecom mesh networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 657–669, Jun. 2011.
[20] J. Moy, 1998, OSPF version 2. Internet RFC2328.
[21] D. Thaler and C. Hopps, Multipath issues in unicast and multicast nexthop selection 2000, Internet RFC2991.
[22] S. Dahlfort, M. Xia, R. Proietti, and S. Yoo, "Split spectrum approach to elastic optical networking," in *Proc. Eur. Conf. Opt. Commun.*, Sep. 2012, pp. 1–3.
[23] P. Colbourne and B. Collings, "ROADM switching technologies," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2011, pp. 1–3.
[24] D. Barros, J. Kahn, J. Wilde, and T. Zeid, "Bandwidth-scalable long-haul transmission using synchronized colorless transceivers and efficient wavelength-selective switches," *J. Lightw. Technol.*, vol. 30, no. 16, pp. 2646–2660, Aug. 2012.
[25] A. Srivastava, "Flow aware differential delay routing for next-generation Ethernet over SONET/SDH," in *Proc. Int. Conf. Commun.*, Jun. 2006, pp. 140–145.
[26] [Online]. Available: http://en.wikipedia.org/wiki/Fragmentation(computing)
[27] C. Politi, V. Anagnostopoulos, C. Matrakidis, and A. Stavdas, "Dynamic flexi-grid OFDM optical networks," in *Proc. Eur. Conf. Opt. Commun.*, Sep. 2012, pp. 1–3.
[28] S. Kosaka, H. Hasegawa, K.-I. Sato, T. Tanaka, A. Hirano, and M. Jinno, "Shared protected elastic optical path network design that applies iterative re-optimization based on resource utilization efficiency measures," in *Proc. Eur. Conf. Opt. Commun.*, Sep. 2012, pp. 1–3.
[29] X. Chu, B. Li, and Z. Zhang, "A dynamic RWA algorithm in a wavelength-routed all-optical network with wavelength converters," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 1795–1804.

Author biographies not included by author request due to space constraints.

Dynamic Constrained Multipath Routing for MPLS Networks

Yongho Seok, Youngseok Lee, Yanghee Choi
Seoul National University
Seoul, Korea
{yhseok, yslee, yhchoi}@mmlab.snu.ac.kr

Changhoon Kim
Electronic Telecommunication Research Institute
Taejon, Korea
kimch@etri.re.kr

Abstract— **Multipath routing employs multiple parallel paths between a traffic source and destination in order to relax the most heavily congested link in Internet backbone. A large bandwidth path can be easily set up too. Although multipath routing is useful, the total network resources, i.e. sum of link bandwidths consumed, could be wasted when the acquired path is larger(in terms of number of hops) than the conventional shortest path. In addition, even though we can accommodate more traffic by establishing more paths between the same node pair, it is advised to limit the number of paths for practical reason such as manageability. This paper presents a heuristic algorithm for hop-count and path-count constrained dynamic multipath routing. The objective we adopted in this paper is to minimize the maximum of link utilization. We also obtain the traffic split ratio among the paths, for routers based on traffic partitioning by hashing at flow level. The extensive simulation results show that the proposed algorithm always minimizes the maximum of link utilization and reduces the number of blocked requests.**

I. Introduction

The dynamic traffic engineering problem in Internet is how to set up paths between edge routers in a network to meet the traffic demand of a request while achieving low congestion and optimizing the utilization of network resources. In practice, the key objective of traffic engineering is usually to minimize the utilization of the most heavily used link in the network, or the maximum of link utilization. Since the queueing delay increases rapidly as link utilization becomes high, it is important to minimize the link utilization throughout the network so that no bottleneck link exists. It has been known that this problem of minimizing the maximum link utilization could be solved by the multicommodity network flow formulation, that leads to splitting traffic over multiple paths between source-destination pairs.

Multipath routing provides increased bandwidth, and the network resources are more efficiently used than the single shortest path algorithm. Multipath routing has been incorporated in recently developed or proposed routing protocols. The easiest extension to multipath routing is to use the equal-cost multiple shortest paths when calculating the shortest one, which is known as Equal-Cost Multi-Path(ECMP) routing. This is explicitly supported by several routing protocols such as Open Shortest Path First(OSPF)[3] and Intermediate System to Intermediate System(IS-IS)[4]. Some router implementations allow equal-cost multipath with Routing Information Protocol(RIP) and with some other routing protocols. In Multi-Protocol Label Switching (MPLS) networks[2] where IP packets are switched through the pre-established Label Switched Paths (LSPs) by signaling protocols, multiple paths can be used to forward packets belonging to the same "forwarding equivalent class (FEC)" by explicit routing.

However, multiple paths may require more total network bandwidth resources, i.e. sum of assigned bandwidth at each link of the paths, than the single shortest path. Therefore, the maximum hop-count constraint should be incorporated into multipath routing scheme in order not to waste bandwidth. In addition, as the number of paths will be restricted between a source-destination pair in the real network topology, the maximum path-count constraint should be considered in multipath routing.

This paper proposes a practical heuristic algorithm that finds hop-count and path-count constrained multiple paths that minimize the maximum of link utilization, while satisfying the requested traffic demand. A traffic demand represents the average traffic volume between edge routers, in bps. For Virtual Private Network (VPN) application, the traffic demand may be the requested amount of bandwidth reservation. Even though the traffic demand varies largely at nodes near end users, it becomes quite stable for the backbone network with aggregated traffic.

Traffic split ratios for the calculated paths are also obtained from the proposed algorithm. The split ratio is fed to the routers for dividing the traffic of the same source-destination pair to multiple paths. Partitioning a traffic demand will be done by adjusting the output range of the hashing function[5]. In multipath routing, routers should also provide a flow-level forwarding mechanism not to cause the out-of-order packet delivery problem which will degrade end-to-end performance.

Through splitting a traffic demand, it is expected that the maximal revenue can be achieved by increasing the probability that more traffic demand requests will be accepted in the future. Also, the utilization of the total network resource will be maximized, while guaranteeing requested bandwidth by reservation. For the sake of users, it would be better that the queueing delay is minimized, especially for expedited forwarding (EF) in differentiated services (Diffserv)[13]. However, when minimizing the queue-

This work was supported in part by the Brain Korea 21 project of Ministry of Education, in part by the National Research Laboratory project of Ministry of Science and Technology, and in part by Electronic Telecommunication Research Institute, 2001, Korea.

ing delay for a newly arrived request, it is necessary that all the established paths for the previous requests need to be re-optimized whenever a new request arrives or the traffic characteristics change. This is not suitable in the real-time environment, even causes a lot of path disruptions. Therefore, instead of minimizing the sum of the queueing delay on a link, we use the objective of minimizing the maximum of link utilization which makes little difference in routing performance[1].

The remainder of this paper is organized as follows. The related works are introduced in section II. The proposed algorithm is explained in section III. The results of the performance evaluation by simulation are discussed in section IV, and section V concludes this paper.

II. Related Work

In connection-oriented networks, [6] analyzed the performance of multipath routing algorithms and showed that the connection establishment time for a reservation is significantly lowered in the multipath case. However, they did not really consider the path computation problem. [7] proposed a dynamic multipath routing algorithm in connection-oriented networks, where the shortest path is used under light traffic condition and multiple paths are utilized as the shortest path becomes congested. In their work, only connection or call-level, not flow-level routing and forwarding are considered. In [8], Quality-of-Service(QoS) routing via multiple paths under time constraint is proposed when the bandwidth can be reserved, assuming all the reordered packets are recovered by optimal buffering at the receiver. This scheme has much overhead for the dynamic buffer adjustment at the receiver. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed in [9], and it is shown that congestion can be greatly reduced. In [10], it is shown that the quality of services can be enhanced by dividing the transport-level flows into UDP and TCP flows. These works did not consider path calculation problem.

For the MPLS network, a traffic engineering method using multiple multipoint-to-point LSPs is proposed in [11], where backup routes are used against failures. Hence, the alternate paths are used only when primary routes do not work. In [12], the traffic bifurcation linear programming (LP) problem is formulated and heuristics for the non-bifurcating problem are proposed. Although [12] minimizes the maximum of link utilization, it does not consider the total network resources and constraints. Wang and et al. showed that the traffic bifurcation LP problem can be transformed to the shortest path problem by adjusting link weights in [15].

In [14], the dynamic routing algorithm for MPLS networks is proposed where the path for each request is selected to prevent the interference among paths for the future demands. It considers only single path routing for simplicity and does not include the constraint.

[16] proposes an adaptive traffic assignment method to multiple paths with measurement information for load balancing. For differentiated services, finding the traffic split ratios to minimize the end-to-end delay and loss rates is proposed in [13]. However, how to find the appropriate multiple paths is not covered.

III. Hop-count and Path-count Constrained Multipath Routing

A. Problem Definition

In this section, we define the hop-count and path-count constrained routing problem in mixed integer programming (MIP) formulation.

The network is modeled as a directed graph, $G = (V, E)$, where V is the set of nodes and E is the set of links. The capacity of a directed link (i, j) is c_{ij}. Each traffic demand ($k \in K$) is given for a node pair between an ingress router (s_k) and an egress router (t_k). For each traffic demand, there are the maximum number of hop counts constraint, H_k[1] and the maximum number of multiple paths constraint, P. The variable $X_{ij}^k(h,p)$ represents the fraction of the traffic demand k assigned to link (i, j) where j is h hops far from s_k and the path p, $1 \le p \le P_k$, includes (i, j) link. The integer variable $Y_{ij}^k(h,p)$ tells whether link (i, j) is used or not for the path p, where j is within h hops from s_k, for the traffic demand k. Let d_k be a scaling factor to normalize the total traffic demand from the source to become 1. The mixed integer programming (MIP) problem is formulated as follows.

$$Minimize\ \alpha$$
subject to

$$\sum_{p=1}^{P_k} \sum_{j:(i,j)\in E} X_{ij}^k(h,p) = \begin{cases} 1, & k \in K, i = s_k, h = 1 \\ 0, & k \in K, i \neq s_k, h = 1 \end{cases} \quad (1)$$

$$\sum_{j:(i,j)\in E} X_{ij}^k(h+1,p) - \sum_{j:(j,i)\in E} X_{ji}^k(h,p) = 0, \quad (2)$$
$$k \in K, i \neq s_k, t_k, 1 \le p \le P_k, 1 \le h < H_k$$

$$\sum_{h=1}^{H_k} \sum_{j:(j,i)\in E} X_{ji}^k(h,p) = 1, k \in K, i = t_k, \forall p \quad (3)$$

$$\sum_{p=1}^{P_k} \sum_{h=1}^{H_k} \sum_{k\in K} d_k X_{ij}^k(h,p) \le c_{ij}\alpha, \forall (i,j) \in E \quad (4)$$

$$\sum_{j:(i,j)\in E} Y_{ij}^k(h,p) = 1, k \in K, i \neq t_k, \forall h, p \quad (5)$$

$$X_{ij}^k(h,p) \le Y_{ij}^k(h,p), \forall k, h, p, i, j \quad (6)$$

$$where,\ \ 0 \le X_{ij}^k(h,p) \le 1, 0 \le \alpha,$$
$$Y_{ij}^k(h,p) \in \{0,1\}, h, p \in Z.$$

The objective is to minimize the maximum of link utilization, α. Constraint (1) says that the sum of total outgoing traffic of each path over the first hop from the source is 1,

[1] $H_k = H + H_{MH_k}$, H_{MH_k} is the minimum number of hop counts from s_k to t_k for traffic demand k. H is additional hop-count that is added to H_{MH_k}.

and the all node over the first hop from the source never receive the traffic without the source node. Constraint (2) is the hop-level flow constraint which means that for all nodes except source and destination, the amount of total incoming traffic to a node is the same as that of outgoing traffic from the node. Constraint (3) means that the amount of total traffic comes into the destination is 1, which is the same as that of goes out of the source. Constraint (4) means that the maximum link utilization among all paths for traffic demand k is α. Constraint (5),(6) shows that for a path, there is only one outgoing edge from a node. This problem is $NP-hard$ because it includes the constrained integer variable.

B. Proposed Heuristic

We propose a heuristic algorithm to find multiple paths and their split ratios for each traffic demand request on the ingress and egress pair. The proposed algorithm consists of three parts: 1) modifying the original graph to the hop-count constrained one, 2) finding path-count constrained multiple paths, and 3) calculating the load split ratios for multiple paths.

Step 1 : **Hop-count constrained graph conversion**
The given network, $G = (N, E)$, is converted to H_k hop-count constrained graph, $G' = (N', E')$, where N' and E' are transformed as follows,

$$N' = \cup_{0 \le m \le H_k} N'_m,$$

$$N'_0 = \{s_k\},$$

$$N'_m = \{j_m | (i,j) \in A, i_{m-1} \in N'_{m-1}\}.$$

$$E' = \cup_{1 \le m \le H_k} E'_m,$$

$$E'_1 = \{(s_k, i) | (s_k, i) \in E\},$$

$$E'_m = \{(i_m, j_m) | i_m \in N'_{m-1}, j_m \in N'_m, (i,j) \in E\}.$$

An example of graph conversion is given in Fig. 1. Fig. 1 (a) represents the original network topology. When a traffic demand request from node 1 to node 4 which requires bandwidth of 3 Mbps with the hop-count constraint of one additional hop and the path-count constraint of two arrives, the graph in Fig. 1 (b) is derived after adding redundant nodes and links. It is easily seen that any path traversed from node 1 to node 4 in Fig. 1 (b) does not exceed three hop counts.

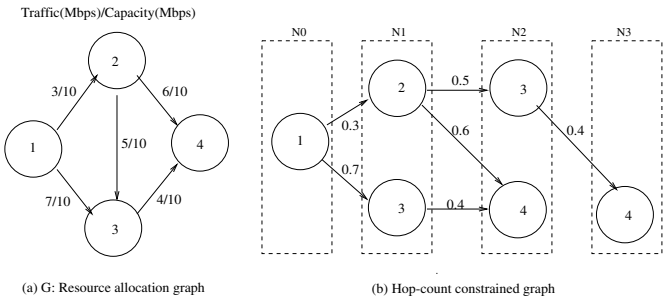


Fig. 1. Topology conversion example

Step 2 : **Finding multiple paths**
On the modified graph G', the link metric(c_{ij}) is given with the current utilization ratio (allocated bandwidth / link capacity). We propose two ways of choosing M multiple paths on the modified graph, $G' = (N', E')$. M multiple paths are found by the well-known M shortest path algorithm[18]. When finding multiple paths, paths are selected 1) to minimize the sum of the link utilization (M shortest paths), or 2) to minimize the maximum of link utilization (M widest paths).

- M shortest paths

The M shortest paths are obtained by selecting the adjacent node with the minimum cost. The cost of a node i reached from source is denoted as $dist(i)$,

$$dist(i) = min_{j \in S}(dist(i), dist(j) + c_{ji}).$$

, where S is the set of nodes whose shortest path from source is already determined.

- M widest paths

The M widest paths are selected in order to minimize the usage of the bottle neck link, the link with the maximum utility. In this case, $dist(i)$, the cost of a node i, denotes the maximum link utility from source to the node.

$$dist(i) = min_{j \in S}(dist(i), max(dist(j), c_{ji})).$$

Step 3 : **Calculating load split ratios**
After finding M multiple paths through the previous step, the amount of a traffic demand, d_k, is divided to M paths. If the maximum of link utilization on M paths (α_M) is less than α of the current bottleneck link, splitting the traffic demand, d_k, is performed to minimize the amount of used total resources(i.e., the number of links and routers). Then, the path with the smallest hop counts is selected among the M paths, and the traffic demand is assigned to the path until the maximum of link utilization of the path does not exceed α. This step is repeated until there remains no traffic demand or there exists no available path. Yet, if the traffic demand is not fully assigned, it is allocated on M multiple paths in proportion to the sum of the link utilization of each path.

The detailed algorithm is explained in Fig. 2. Therefore, the proposed heuristic is either the shortest path(SP) based algorithm or the widest path (WP) based one. Fig. 3 explains the results of the multiple paths and their load split ratios on the graph in Fig. 1 (a). It is seen that the paths found by the single shortest path algorithm(SSP) wastes more network resources, while the proposed algorithm can find the optimal solution.

C. Complexity Analysis

Proposed algorithm consists of two parts, finding multiple paths, calculating load split ratios. For each part, time complexity is bound as follow. First, for the M shortest simple paths problem, the best known bound is O(Mn(m+nlogn)) in a directed graph, where M is the number of paths, m is the number of edges and n is number of vertices in the graph. Algorithm for splitting traffic

demands into M paths is bound by O(MlogM+2M), because sorting M paths according to hop counts is bound by O(MlogM) and calculating splitting ratio for each path is bound by O(2M). Hence, the time complexity of the proposed algorithm is bound by O(Mn(m+nlogn) + MlogM+2M).

Heuristic : *Find constrained multiple paths and their split ratios*

- Set α to be current maximum link utilization;
- Set d_k to be the normalized traffic demand k, $0 < d_k < 1$;
- Modify G to G' satisfying H_k hops;
- Find M shortest(widest) paths from s_k to t_k;
- Set P to be a set of candidate multiple paths previously found;
- Set α_M to be the maximum link utilization of P;

if $(\alpha_M < \alpha)$

while$(d_k > 0$and P is not empty)

- Set p to be the minimum hop-count path of P_k;
- Set $d_k(p)$ to be $min(\alpha - \alpha_M, d_k)$;
- Assign $d_k(p)$ to each link along the path p;
- Set d_k to be $d_k - d_k(p)$;
- Delete p from P;

endwhile

endif

while$(d_k > 0)$

- Assign remaining d_k to M paths in proportion to the available link capacity;

endwhile

Fig. 2. The flow of proposed Heuristic.

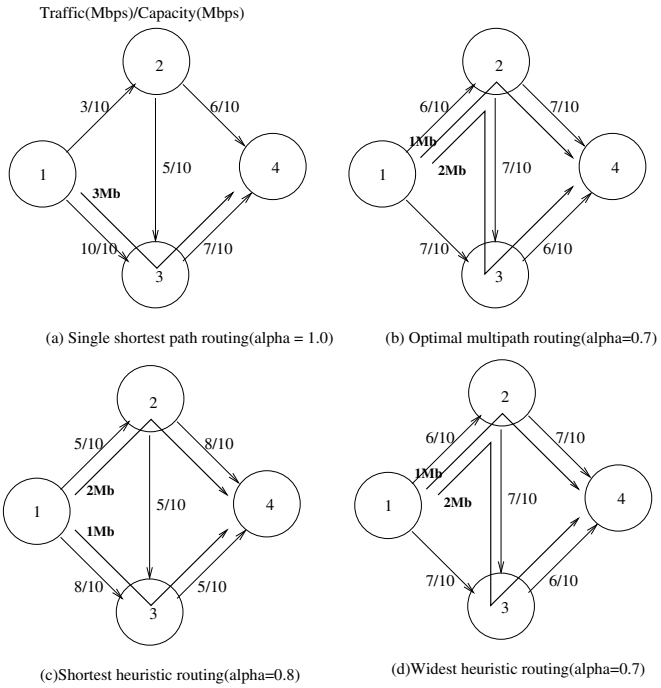


Fig. 3. The result of several multiple path calculation methods

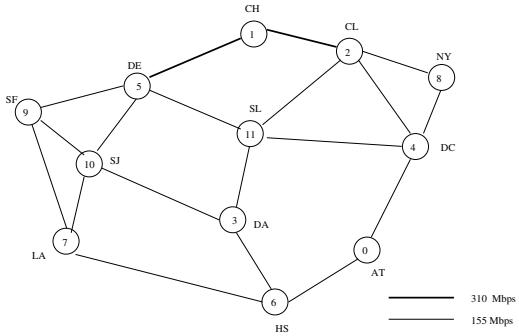


Fig. 4. Abstract US Network

IV. Performance Evaluation

A. Simulation Environment

The network topology shown in Fig. 4 represents the abstract US backbone topology[17]. On this topology we assume that the background traffic demands are given as in [17].

In this network condition, we generate ten random requests of traffic demands between two nodes selected randomly. Therefore, 1,320 requests are tested in total. The duration of each traffic demand is exponentially distributed (ten seconds), and the inter-arrival time is randomly distributed between zero and one hundred seconds. The average rate of each traffic demand is set to 2 Mbps.

The proposed heuristics are compared with the simple shortest path algorithm and the optimal MIP solution. The maximum hop-count constraint(H_k) is given as zero or more additional to that of the minimum hop-count(H_{MH_k}) between an ingress and an egress router. As the number of multiple paths(P_k) will be usually restricted, the path-count constraint is set to be one of 1, 3, and 5.

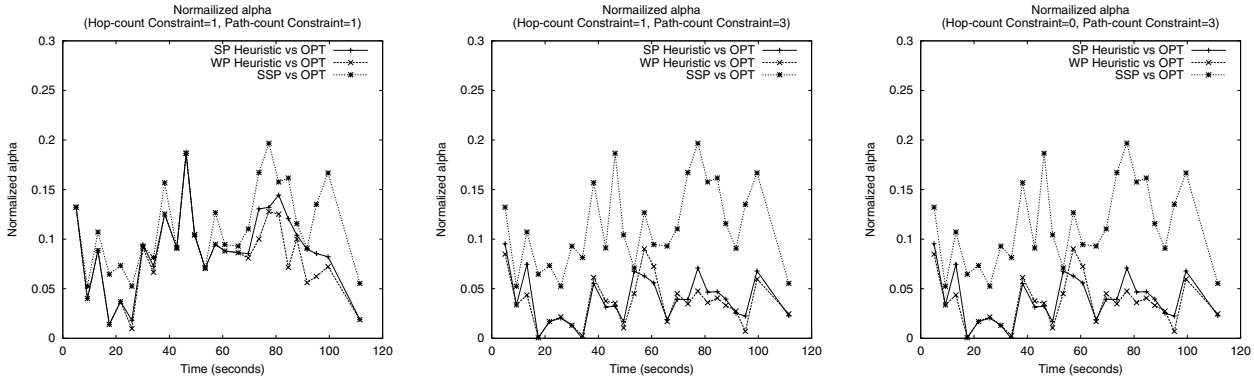
As shown in Table I with the average of α, the shortest path(widest path) based heuristic increases only 4.12(3.01) % when compared to the optimal solution by MIP formulation. Yet, the maximum link utilization by the single shortest path(SSP) algorithm increases by 11.34 %.

Fig. 5 shows the *normalized* α which was obtained by dividing α of our algorithm in Fig.2 by α of optimal solution of the MIP. In Fig. 5 (a), it is seen that although only one additional hop is constrained on the single path, the proposed heuristic performs better than the shortest path. As the path-count constraint is increased to three (Fig. 5 (b)), the maximum of link utilization is greatly reduced. Also, even when the hop-count constraint is zero (i.e., the equal cost multiple paths) (Fig. 5 (c)), multiple paths are well utilized, giving the similar α to the case with $(H, P_k) = (1, 3)$

However, the performance of the proposed algorithm may not be more enhanced although the number of hop-count or path-count constraint increases, because many multiple paths are overlapped. With the path-count constraint of five(Fig. 6 (a)) or with H of two (Fig. 6 (b)), the proposed algorithm shows the similar result under

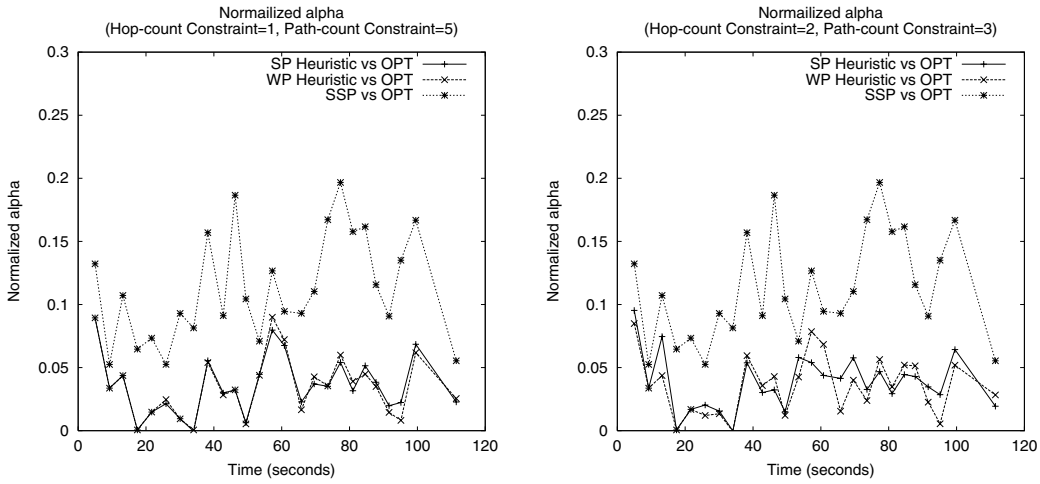
TABLE I
AVERAGE OF MAXIMUM OF LINK UTILIZATION (α)

	SSP	SP Heuristic	WP Heuristic	OPT
α	1.08	1.01	1.00	0.97



(a) $(H, P_k) = (1,1)$ (b) $(H, P_k) = (1,3)$ (c) $(H, P_k) = (0,3)$

Fig. 5. Maximum of link utilization (α) with the hop-count constraint and the path-count constraint



(a) $(H, P_k) = (1,5)$ (b) $(H, P_k) = (2,3)$

Fig. 6. Maximum of link utilization (α) with the hop-count constraint and the path-count constraint: when α is not more enhanced

$(H, P_k) = (1, 3)$.

In general, when α is greater than 1, the request will be blocked because of scarce network bandwidth assuming that over-booking is not permitted. We plotted the ratio of blocked requests to the total requests in every five seconds which cause α to be greater than 1 in Fig. 7. It is shown that the number of rejected requests is reduced by the proposed heuristic.

V. CONCLUSION

In this paper, we propose dynamic multipath traffic engineering schemes for MPLS networks that minimize the maximum of link utilization, α by finding multiple paths with the hop-count and path-count constraints. The proposed heuristic approximates the traffic bifurcation MIP problem that is $NP-hard$, by calculating constrained multiple paths and their split ratios efficiently in polynomial time. The simulation results show that the proposed algorithm solves nearly the same α of the optimal solution even with only one additional hop and three paths because many new candidate paths are derived. In addition, the number of blocked requests is reduced. Therefore, the proposed traffic engineering scheme is practical and will be useful for reducing the probability of congestion by minimizing the utilization of the most heavily used link in the network.

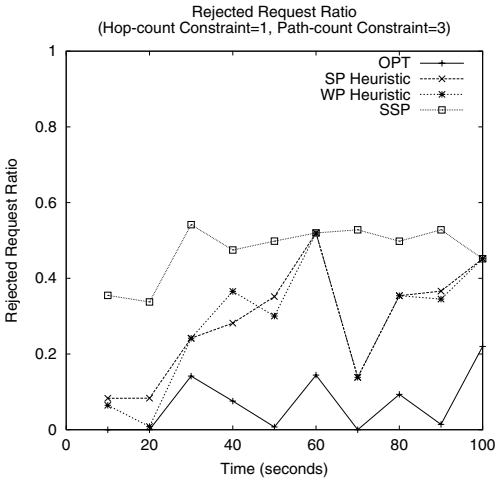


Fig. 7. Rejected Traffic Demand Request Ratio

References

[1] D. Bertsekas, and R. Gallager, Data Networks, Prentice Hall, 1992

[2] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," Internet RFC3031, 2001

[3] J. Moy, "OSPF Version 2," Internet RFC2328, 1998

[4] R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments," Internet RFC1195, 1990

[5] Z. Cao, Z. Wang and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," INFOCOM'2000

[6] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of Multi-Path Routing," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 885 - 896, Dec. 1999

[7] S. Bahk, and M. Zarki, "Dynamic Multi-path Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide Area Networks," Computer Communications Review, vol. 22, no. 4, Oct. 1992

[8] N. S. V. Rao, and S. G. Batsell, "QoS Routing Via Multiple Paths Using Bandwidth Reservation," INFOCOM'98

[9] A. Shaikh, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," SIGCOMM'99

[10] P. Bhaniramka, W. Sun, and R. Jain, "Quality of Service using Traffic Engineering over MPLS: An Analysis," LCN'2000

[11] H. Saito, Y. Miyao, and M. Yoshida, "Traffic Engineering using Multiple Multipoint-to-Point LSPs," INFOCOM'2000

[12] Y. Wang, and Z. Wang, "Explicit Routing Algorithms for Internet Traffic Engineering," ICCCN'99

[13] E. Dinan, D. O. Awduche, and B. Jabbari, "Analytical Framework for Dynamic Traffic Partitioning in MPLS Networks," ICC'2000

[14] M. Kodialam, and T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," INFOCOM'2000

[15] Z. Wang, Y. Wang, and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," INFOCOM'2001

[16] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering," INFOCOM'2001

[17] Optimized Multipath, http://www.fictitious.org/omp

[18] E. L. Lawler, Combinatorial Optimization: Newtorks and Matroids, Holt, Rinehart and Winston, 1976

Dynamic Resource Allocation in Cognitive Radio Networks: A Convex Optimization Perspective

Rui Zhang, Ying-Chang Liang, and Shuguang Cui

***Abstract*—This article provides an overview of the state-of-art results on communication resource allocation over space, time, and frequency for emerging cognitive radio (CR) wireless networks. Focusing on the interference-power/interference-temperature (IT) constraint approach for CRs to protect primary radio transmissions, many new and challenging problems regarding the design of CR systems are formulated, and some of the corresponding solutions are shown to be obtainable by restructuring some classic results known for traditional (non-CR) wireless networks. It is demonstrated that convex optimization plays an essential role in solving these problems, in a both rigorous and efficient way. Promising research directions on interference management for CR and other related multiuser communication systems are discussed.**

I. Introduction

In recent years, *cognitive radio* (CR) networks, where CRs or the so-called secondary users (SUs) communicate over certain bandwidth originally allocated to a primary network, have drawn great research interests in the academic, industrial, and regulation communities. Accordingly, there is now a rapidly growing awareness that CR technology will play an essential role in enabling *dynamic spectrum access* for the next generation wireless communications, which could hopefully resolve the spectrum scarcity vs. under-utilization dilemma caused by the current static spectrum management polices. Specifically, the users in the primary network, or the so-called primary users (PUs), could be licensed users, who have the absolute right to access their spectrum bands, and yet would be willing to share the spectrum with the unlicensed SUs. Alternatively, both the PUs and SUs could equally coexist in an unlicensed band, where the PUs are regarded as existing active communication links while the SUs are new links to be added. A unique feature of CRs is that they are able to identify and acquire useful environmental information (cognition) across the primary and secondary networks, and thereby adapt their transmit strategies to achieve the best performance while maintaining a required quality of service (QoS) for each coexisting active primary link. Depending on the type of cognitive knowledge collected (e.g., on/off statuses of primary links, PU messages, interference power levels at PU receivers, or primary link performance margins) and the primary/secondary network models of interests (e.g., infrastructure-based vs. ad hoc), many new and challenging problems on the design of CR networks can be formulated, as will be reviewed in this article.

To date, quite a few operation models have been proposed for CRs; however, there is no consensus yet on the terminology used for the associated definitions [1], [2], [3]. Generally speaking, there are two basic operation models for CRs: Opportunistic Spectrum Access (OSA) vs. Spectrum Sharing (SS). In the OSA model, the SUs are allowed to transmit over the band of interest when all the PUs are not transmitting at this band. One essential enabling technique for OSA-based CRs is *spectrum sensing*, where the CRs individually or collaboratively detect active PU transmissions over the band, and decide to transmit if the sensing results indicate that all the PU transmitters are inactive at this band with a high probability. Spectrum sensing is now a very active area for research; the interested readers may refer to, e.g., [4], [5], [6], [7] for an overview of the state-of-art results in this area. As a counterpart, the SS model allows the SUs to transmit simultaneously with PUs at the same band even if they are active, provided that the SUs know how to control their resultant interference at the PU receivers such that the performance degradation of each active primary link is within a tolerable margin. Thus, OSA and SS can be regarded as the *primary-transmitter-centric* and *primary-receiver-centric* dynamic spectrum access techniques, respectively. Consequently, there will be an inevitable debate on which operation model, OSA or SS, is better to deploy CRs in practical systems; however, a rigorous comparative study for these two models, in terms of spectrum efficiency and implementation complexity tradeoffs, is still open. Generally speaking, SS utilizes the spectrum more efficiently than OSA, since the former supports concurrent PU and SU transmissions over the same band while the latter only allows orthogonal transmissions between them. Moreover, the receiver-centric approach for SS is more effective for CRs to manage the interference to the PU links than the transmitter-centric approach for OSA.

Hence, the SS model for CRs will be focused in this article. It is worth noting that the optimal design approach for SS-based CR networks should treat all coexisting PU and SU links as a giant interference network and jointly optimize their transmissions to maximize the SU network throughput with a prescribed PU network throughput guarantee. From this viewpoint, recent advances in network information theory [8] have provided promising guidelines to approach the fundamental limits of such networks. However, from a practical viewpoint, the centralized design approach for PU and SU networks is not desirable, since PU and SU systems usually belong to different operators and thus it is difficult, if not infeasible, for them to cooperate. Consequently, a *decentralized* design approach is more favorable, where the PU network is designed without the awareness of the existence of the SU network, while the SU network is designed with only partial knowledge (cognition) of the PU network.

Following this (simplified) decentralized approach, there are furthermore two design paradigms proposed for SS-based CRs.

One is based on the "cognitive relay" concept [9], where the SU transmitter allocates only part of its power to deliver the SU messages, and uses the remaining power to relay the PU messages so as to compensate for the additional SU interference experienced at the PU receiver. However, this technique requires non-causal knowledge of the PU messages at the SU transmitter, which may be difficult to realize in practice. In contrast, a more feasible SS design for the SU to protect the PU is to impose a constraint on the maximum SU interference power at the PU receiver, also known as the "interference temperature (IT)" constraint [10], by assuming that the SU-to-PU channels are either perfectly known at the SU transmitters, or can be practically estimated.

In this article, we will focus our study on the IT-based SS model for CRs, namely the IT-SS, as it is a more feasible approach compared with other existing ones. In a wireless communication environment, channels are usually subject to space-time-frequency variation due to multipath propagation, mobility, and location-dependent shadowing. Thus, *dynamic resource allocation* (DRA) becomes an essential technique for CRs to optimally deploy their transmit strategies to maximize the secondary network throughput, where the transmit power, bit-rate, bandwidth, and antenna beam should be dynamically allocated based upon the available channel state information (CSI) of the primary and secondary networks. In particular, this article will focus on DRA problem formulations unique to CR systems under the IT-SS model, and the associated solutions that are non-obvious in comparison with existing results [11] known for the traditional (non-CR) wireless networks. More importantly, we will emphasize the key role of various *convex optimization* techniques in solving these problems.

The remainder of this article is organized as follows. Section II presents different models of the SU network coexisting with the PU network, and various forms of transmit power and interference power (IT) constraints over the SU transmissions. Section III is devoted to the spatial-domain transmit optimization at the SUs for different SU networks subject to transmit and interference power constraints. Section IV extends the results to the more general case of joint space-time-frequency transmit optimization of the SUs, and addresses the important issue on how to optimally set the IT thresholds in CR systems to achieve the best spectrum sharing performance. Finally, conclusions are drawn and future research directions are discussed in Section V.

Notation: Lower-case and upper-case bold letters denote vectors and matrices, respectively. $\mathrm{Rank}(\cdot)$, $\mathrm{Tr}(\cdot)$, $|\cdot|$, $(\cdot)^{-1}$, $(\cdot)^H$, and $(\cdot)^{1/2}$ denote the rank of a matrix, trace, determinant, inverse, Hermitian transpose, and square-root, respectively. $\mathbf{I}$ and $\mathbf{0}$ denote an identity matrix and an all-zero matrix, respectively. $\mathrm{Diag}(\mathbf{a})$ denotes a diagonal matrix with diagonal elements given in $\mathbf{a}$. $\mathrm{E}(\cdot)$ denotes the statistical expectation. A circularly symmetric complex Gaussian (CSCG) distributed random vector with zero mean and covariance matrix $\mathbf{S}$ is denoted by $\mathcal{CN}(\mathbf{0}, \mathbf{S})$. $\mathbb{C}^{m\times n}$ denotes the space of $m \times n$ complex matrices. $\|\cdot\|$ denotes the 2-norm of a complex vector. $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ denote the real and imaginary parts of a complex number, respectively. The base of the logarithm function $\log(\cdot)$ is 2 by default.

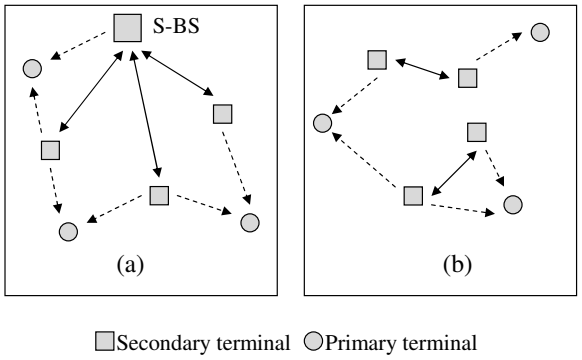


Fig. 1. CR networks: (a) infrastructure-based; (b) ad hoc.

II. CR Network Models

We consider two general types of CR networks, which are of both theoretical and practical interests: One is *infrastructure-based*, as shown in Fig. 1(a), where multiple secondary terminals communicate with a common secondary node referred to as the secondary base station (S-BS); the other is *ad hoc*, as shown in Fig. 1(b), which consists of multiple distributed secondary links. In both types of CR networks, there are coexisting primary terminals operating in the same spectrum band. For the IT-SS model of CRs, the exact operation model of the primary network is not important to our study, provided that all the secondary terminals satisfy the prescribed IT constraints to protect the primary terminals. Without loss of generality, we assume that there are K secondary links and J primary terminals in each type of the CR networks.

Consider first the infrastructure-based secondary/CR network with the S-BS coordinating all the CR transmissions, which usually corresponds to one particular cell in a CR cellular network. The uplink transmissions from the SUs to the S-BS are usually modeled by a multiple-access channel (MAC), while the downlink transmissions from the S-BS to different SUs are modeled by a broadcast channel (BC). For the MAC, the equivalent baseband transmission can be represented as

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{H}_k \mathbf{x}_k + \mathbf{z} \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{M\times 1}$ denotes the received signal at the S-BS, with M denoting the number of antennas at S-BS; $\mathbf{H}_k \in \mathbb{C}^{M\times N_k}$ denotes the channel from the kth SU to S-BS, $k = 1, \cdots, K$, with N_k denoting the number of antennas at the kth SU; $\mathbf{x}_k \in \mathbb{C}^{N_k\times 1}$ denotes the transmitted signal of the kth SU; and $\mathbf{z} \in \mathbb{C}^{M\times 1}$ denotes the noise received at S-BS. We assume that $\mathbf{x}_k$'s are independent over k.

Similarly, the BC can be represented as

$$\mathbf{y}_k = \mathbf{H}_k^H \mathbf{x} + \mathbf{z}_k, \quad k = 1, \cdots, K \tag{2}$$

where $\mathbf{y}_k \in \mathbb{C}^{N_k\times 1}$ denotes the received signal at the kth SU; for convenience, we have used the Hermitian transposed uplink channel matrix for the corresponding downlink channel matrix, i.e., $\mathbf{H}_k^H$ denotes the channel from the S-BS to the kth SU; $\mathbf{x} \in \mathbb{C}^{M\times 1}$ denotes the transmitted signal from S-BS; and $\mathbf{z}_k \in \mathbb{C}^{N_k\times 1}$ denotes the receiver noise of the kth SU.

In the case that $\mathbf{x}$ carries information common to all SUs, the associated downlink transmission is usually called *multicast*, while if $\mathbf{x}$ carries independent information for different SUs, it is called *unicast*.

Next, consider the ad hoc secondary/CR network, which is usually modeled as an interference channel (IC). For convenience, we assume that for the kth secondary link, $k = 1, \cdots, K$, the transmitter is denoted as SU-TX$_k$ and the receiver is denoted as SU-RX$_k$, although in general a secondary terminal can be both a transmitter and a receiver. The baseband transmission of the IC can be represented as

$$\tilde{\mathbf{y}}_k = \mathbf{H}_{kk}\tilde{\mathbf{x}}_k + \sum_{i=1, i\neq k}^{K} \mathbf{H}_{ik}\tilde{\mathbf{x}}_i + \tilde{\mathbf{z}}_k, \quad k = 1, \cdots, K \tag{3}$$

where $\tilde{\mathbf{y}}_k \in \mathbb{C}^{B_k \times 1}$ denotes the received signal at SU-RX$_k$, with B_k denoting the number of receiving antennas; $\tilde{\mathbf{x}}_k \in \mathbb{C}^{A_k \times 1}$ denotes the transmitted signal of SU-TX$_k$, with A_k denoting the number of transmitting antennas; $\mathbf{H}_{kk} \in \mathbb{C}^{B_k \times A_k}$ denotes the direct-link channel from SU-TX$_k$ to SU-RX$_k$, while $\mathbf{H}_{ik} \in \mathbb{C}^{B_k \times A_i}$ denotes the cross-link channel from SU-TX$_i$ to SU-RX$_k$, $i \neq k$; and $\tilde{\mathbf{z}}_k \in \mathbb{C}^{B_k \times 1}$ denotes the noise at SU-RX$_k$. It is assumed that $\tilde{\mathbf{x}}_k$'s are independent over k.

Furthermore, we assume that the jth PU, $j = 1, \cdots, J$, in each type of the CR networks is equipped with D_j antennas, $D_j \geq 1$. We then use $\mathbf{G}_{kj} \in \mathbb{C}^{D_j \times N_k}$ to denote the channel from the kth SU to the jth PU in the CR MAC, $\mathbf{F}_j \in \mathbb{C}^{D_j \times M}$ to denote the channel from S-BS to the jth PU in the CR BC, and $\mathbf{E}_{kj} \in \mathbb{C}^{D_j \times A_k}$ to denote the channel from SU-TX$_k$ to the jth PU in the CR IC. Moreover, the receiving terminals in the secondary networks may experience interference from active primary transmitters. For simplicity, we assume that such interference is treated as additional noise at the secondary receivers, and the total noise at each secondary receiving terminal is distributed as a CSCG random vector with zero mean and the identity covariance matrix.

Note that the (spatial) channels defined in the above CR network models are assumed constant for a fixed transmit dimension such as one time-block in a time-division-multiple-access (TDMA) system or one frequency-bin in an orthogonal-frequency-division-multiplexing (OFDM) system. In a wireless environment, these channels usually change over time and/or frequency dimensions as governed by an underlying joint stochastic process. As such, DRA becomes relevant to schedule SUs into different transmit dimensions based on their CSI. In general, the secondary transmitting terminals need to satisfy two types of power constraints for DRA: One is due to their own transmit power budgets; the other is to limit their resulting interference level at each PU to be below a prescribed threshold. These constraints can be applied over each fixed dimension as *peak* power constraints, or over multiple dimensions as *average* power constraints. Without loss of generality, we consider DRA for the secondary network over L transmit dimensions with different channel realizations, with $L \geq 1$. In total, four different types of power constraints can be defined for the secondary network. By taking the CR MAC as an example (similarly as for the CR BC/IC), we have

- **Peak transmit power constraint (PTPC)**:

$$\mathrm{Tr}(\mathbf{S}_k[l]) \leq P_k \tag{4}$$

  where $\mathbf{S}_k[l]$ denotes the transmit covariance matrix for the lth transmit dimension of the kth SU, $l \in \{1, \cdots, L\}, k \in \{1, \cdots, K\}$; P_k denotes the kth SU's peak power constraint that applies to each of the L transmit dimensions.
- **Average transmit power constraint (ATPC)**:

$$\frac{1}{L}\sum_{l=1}^{L} \mathrm{Tr}(\mathbf{S}_k[l]) \leq \bar{P}_k \tag{5}$$

  where $\bar{P}_k$ denotes the kth SU's average transmit power constraint over the L transmit dimensions.
- **Peak interference power constraint (PIPC)**:

$$\sum_{k=1}^{K} \mathrm{Tr}\left(\mathbf{G}_{kj}[l]\mathbf{S}_k[l]\mathbf{G}_{kj}^H[l]\right) \leq \Gamma_j \tag{6}$$

  where $\mathbf{G}_{kj}[l]$ denotes the realization of channel $\mathbf{G}_{kj}$ for a given l; and Γ_j denotes the peak interference power constraint for protecting the jth PU, $j \in \{1, \cdots, J\}$, which limits the total interference power caused by all the K SUs across all the receiving antennas of the jth PU, for each of the L transmit dimensions.
- **Average interference power constraint (AIPC)**:

$$\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K} \mathrm{Tr}\left(\mathbf{G}_{kj}[l]\mathbf{S}_k[l]\mathbf{G}_{kj}^H[l]\right) \leq \bar{\Gamma}_j \tag{7}$$

  where $\bar{\Gamma}_j$ denotes the average interference power constraint for the jth PU to limit the total interference power from the K SUs, which is averaged over the L transmit dimensions.

Note that DRA for traditional (non-CR) wireless networks under PTPC and/or ATPC has been thoroughly studied in the literature [12], while the study of DRA subject to PIPC and/or AIPC as well as their various combinations with PTPC/ATPC is unique to CR networks and is relatively new. In order to gain more insights into the optimal DRA designs for CR networks, we will first study the case of a single transmit dimension ($L = 1$) with PTPCs and PIPCs by focusing on the spatial-domain transmit optimization for multi-antenna CRs in Section III, and then study the general case of $L > 1$ for joint space-time-frequency DRA in CR networks under ATPCs and AIPCs in Section IV.

III. Cognitive Beamforming Optimization

In this section, we consider the case of $L = 1$, where the DRA for CR networks reduces to the spatial-domain transmit optimization under PTPCs and PIPCs to maximize the CR network throughput. We term this practice as *cognitive beamforming*. In order to investigate the fundamental performance limits of cognitive beamforming, we study the optimal designs with the availability of perfect knowledge on all the channels in the SU networks, and those from all the secondary transmit terminals to PUs. For convenience, we drop the dimension index l for the rest of this section given $L = 1$.

First, it is worth noting that the PIPC given in (6) can be unified with the PTPC given in (4) into a form of *generalized linear transmit covariance constraint* (GLTCC):

$$\sum_{i=1}^{K} \mathrm{Tr}(\mathbf{W}_i \mathbf{S}_i) \leq w \tag{8}$$

where $\mathbf{W}_i$'s and w are constants. For example, with each PIPC given in (6), $\mathbf{W}_i = \mathbf{G}_{ij}^H \mathbf{G}_{ij}, \forall i$, and $w = \Gamma_j$, while for each PTPC given in (4), $\mathbf{W}_i = \mathbf{I}$ if $i = k$ and $\mathbf{0}$ otherwise, with $w = P_k$. Previous studies on transmit optimization for multi-antenna or multiple-input multiple-output (MIMO) systems have mostly adopted some special forms of GLTCC such as the user individual power constraints and sum-power constraint. However, it remains unclear whether such existing solutions are applicable to the general form of GLTCC, which is crucial to the problem of CR MIMO transmit optimization with the newly added PIPCs. In the following, we provide an overview of the state-of-art solutions for this problem under different CR network models, while the developed solutions also apply to the case with the general form of GLTCCs as in (8). From a convex optimization perspective, we next divide our discussions into two parts, which deal with the cases of convex and non-convex problem formulations, respectively.

A. Convex Problem Formulation

First, consider the case where the associated optimization problem in a traditional MIMO system without PIPC is *convex*. In such cases, since the extra PIPCs are linear over the SU transmit covariance matrices, the resulting transmit covariance optimization problem for CR systems remains convex; and thus, it can be efficiently solved by standard convex optimization techniques.

CR Point-to-Point MIMO Channel: We elaborate this case by first considering the CR point-to-point MIMO channel, which can be treated as the special case with only one active SU link in the MAC, BC, or IC based CR network. Without loss of generality, we will use the notations developed for the CR MAC with $K = 1$ in the following discussions. Specifically, the optimal transmit covariance to achieve the CR point-to-point MIMO channel capacity under both the PTPC and PIPCs can be obtained from the following problem [13]:

$$\begin{aligned} \underset{\mathbf{S}}{\text{Max.}} \quad & \log\left|\mathbf{I} + \mathbf{H}\mathbf{S}\mathbf{H}^H\right| \qquad \text{(P1)} \\ \text{s. t.} \quad & \mathrm{Tr}(\mathbf{S}) \leq P \\ & \mathrm{Tr}\left(\mathbf{G}_j \mathbf{S} \mathbf{G}_j^H\right) \leq \Gamma_j, \ j = 1, \cdots, J \\ & \mathbf{S} \succeq \mathbf{0} \end{aligned}$$

where for conciseness we have removed the SU index k in the symbol notations since $K = 1$, while $\mathbf{S} \succeq \mathbf{0}$ means that $\mathbf{S}$ is a positive semi-definite matrix [14].

We see that (P1) is a convex optimization problem since its objective function is concave over $\mathbf{S}$ and its constraints define a convex set over $\mathbf{S}$. Thus, (P1) can be efficiently solved by, e.g., the interior point method [14]. In the special case of CR multiple-input single-output (MISO) channel, i.e., $\mathbf{H}$ degrades to a row-vector denoted by $\mathbf{h} \in \mathbb{C}^{1\times N}$, it can be shown by exploiting the Karush-Kuhn-Tucker (KKT) optimality conditions of (P1) that transmit beamforming is capacity optimal, i.e., $\mathrm{Rank}(\mathbf{S}) = 1$ [13]. Thus, without loss of generality we could write $\mathbf{S} = \mathbf{v}\mathbf{v}^H$, where $\mathbf{v} \in \mathbb{C}^{N\times 1}$ denotes the precoding vector. Accordingly, (P1) for the special case of MISO CR channel is simplified as (P1-S) [13]:

$$\begin{aligned} \underset{\mathbf{v}}{\text{Max.}} \quad & \|\mathbf{h}\mathbf{v}\| \\ \text{s. t.} \quad & \|\mathbf{v}\|^2 \leq P \\ & \|\mathbf{G}_j \mathbf{v}\|^2 \leq \Gamma_j, \ j = 1, \cdots, J, \end{aligned}$$

which is non-convex due to the non-concavity of its objective function. However, by observing the fact that if $\mathbf{v}$ is the solution of (P1-S), so is $e^{j\theta}\mathbf{v}$ for any arbitrary θ, we thus assume without loss of generality that $\mathbf{h}\mathbf{v}$ is a real number and modify (P1-S) by rewriting its objective function as $\mathrm{Re}(\mathbf{h}\mathbf{v})$ and adding an additional linear constraint $\mathrm{Im}(\mathbf{h}\mathbf{v}) = 0$. Thereby, (P1-S) can be converted into a second-order cone programming (SOCP) [14] problem, which is convex and thus can be efficiently solved by available convex optimization softwares [15]. Alternatively, (P1-S) can be shown equivalent to its Lagrange dual problem [13], which is a convex semi-definite programming (SDP) [14] problem and is thus efficiently solvable [15]. For (P1-S) in the case of one single-antenna PU, a closed-form solution for the optimal precoding vector $\mathbf{v}$ was derived in [13] via a geometric approach.

In order to reveal the structure of the optimal $\mathbf{S}$ for (P1), we consider its Lagrange dual problem defined as (P1-D):

$$\underset{\boldsymbol{\eta}\succeq\mathbf{0}}{\text{Min.}} \ d(\boldsymbol{\eta})$$

where $\boldsymbol{\eta} = [\eta_0, \eta_1, \cdots, \eta_J]$ denotes a vector of dual variables for (P1) with η_0 associated with the PTPC, and η_j associated with the jth PIPC, $j = 1, \cdots, J$, while we have the dual function defined as

$$\begin{aligned} d(\boldsymbol{\eta}) \triangleq \max_{\mathbf{S}\succeq\mathbf{0}} \ & \log\left|\mathbf{I} + \mathbf{H}\mathbf{S}\mathbf{H}^H\right| - \eta_0(\mathrm{Tr}(\mathbf{S}) - P) \\ & - \sum_{j=1}^{J} \eta_j (\mathrm{Tr}\left(\mathbf{G}_j \mathbf{S} \mathbf{G}_j^H\right) - \Gamma_j). \end{aligned} \tag{9}$$

Since (P1) is convex with Slater's condition satisfied [14], the duality gap between the optimal values of (P1) and (P1-D) is zero, i.e., (P1) can be solved equivalently as (P1-D). Accordingly, an iterative algorithm can be developed to solve (P1-D) by alternating between solving $d(\boldsymbol{\eta})$ for a given $\boldsymbol{\eta}$ and updating $\boldsymbol{\eta}$ to minimize $d(\boldsymbol{\eta})$. At each iteration, $\boldsymbol{\eta}$ can be updated by a subgradient-based method such as the ellipsoid method [16], according to the subgradients of $d(\boldsymbol{\eta})$, which can be shown equal to $P - \mathrm{Tr}(\mathbf{S}^\star)$ and $\Gamma_j - \mathrm{Tr}\left(\mathbf{G}_j \mathbf{S}^\star \mathbf{G}_j^H\right)$ for η_0 and $\eta_j, j \neq 0$, respectively, where $\mathbf{S}^\star$ denotes the optimal $\mathbf{S}$ to obtain $d(\boldsymbol{\eta})$ for a given $\boldsymbol{\eta}$. From (9), it follows that $\mathbf{S}^\star$ is the solution of the following equivalent problem (by discarding irrelevant constant terms):

$$\max_{\mathbf{S}\succeq\mathbf{0}} \ \log\left|\mathbf{I} + \mathbf{H}\mathbf{S}\mathbf{H}^H\right| - \mathrm{Tr}(\mathbf{T}\mathbf{S}) \tag{10}$$

where $\mathbf{T} = \eta_0 \mathbf{I} + \sum_{j=1}^{J} \eta_j (\mathbf{G}_j^H \mathbf{G}_j)$ is a constant matrix for a given $\boldsymbol{\eta}$. In order to solve Problem (10), we introduce an

auxiliary variable: $\hat{\mathbf{S}} = \mathbf{T}^{1/2}\mathbf{S}\mathbf{T}^{1/2}$. Problem (10) is then re-expressed in terms of $\hat{\mathbf{S}}$ as

$$\max_{\hat{\mathbf{S}}\succeq\mathbf{0}} \ \log\left|\mathbf{I}+\mathbf{H}\mathbf{T}^{-1/2}\hat{\mathbf{S}}\mathbf{T}^{-1/2}\mathbf{H}^H\right| - \mathrm{Tr}(\hat{\mathbf{S}}). \tag{11}$$

The above problem can be shown equivalent to the standard point-to-point MIMO channel capacity optimization problem subject to a single sum-power constraint [17], and its solution can be expressed as $\hat{\mathbf{S}}^\star = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^H$, where $\mathbf{V}$ is obtained from the singular-value decomposition (SVD) given as follows: $\mathbf{H}\mathbf{T}^{-1/2} = \mathbf{U}\mathbf{\Theta}\mathbf{V}^H$, with $\mathbf{\Theta} = \mathrm{Diag}([\theta_1,\ldots,\theta_T])$ and $T = \min(M,N)$, while $\mathbf{\Sigma} = \mathrm{Diag}([\sigma_1,\ldots,\sigma_T])$ follows the standard water-filling solution [17]: $\sigma_i = (1/\ln 2 - 1/\theta_i^2)^+, i = 1,\ldots,T$, with $(\cdot)^+ \triangleq \max(0,\cdot)$. Thus, the solution of Problem (10) for a given $\boldsymbol{\eta}$ can be expressed as $\mathbf{S}^\star = \mathbf{T}^{-1/2}\mathbf{V}\mathbf{\Sigma}\mathbf{V}^H\mathbf{T}^{-1/2}$.

Next, we present a heuristic method for solving (P1), which leads to a suboptimal solution in general and could serve as a benchmark to evaluate the effectiveness of the above two approaches based on convex optimization. To gain some intuitions for this method, we first take a look at two special cases of (P1). For the first case, supposing that all the PIPCs are inactive (e.g., by setting $\Gamma_j = \infty, \forall j$) and thus can be removed, (P1) reduces to the standard MIMO channel capacity optimization problem under the PTPC only, for which the optimal solution of $\mathbf{S}$ is known to be derivable from the SVD of $\mathbf{H}$ [17]. For the second case, assuming that $\Gamma_j = 0, \forall j$, the solution for (P1) is then obtained by the "zero-forcing (ZF)" algorithm [18], which first projects $\mathbf{H}$ into the space orthogonal to all $\mathbf{G}_j$'s, and then designs the optimal $\mathbf{S}$ based on the SVD of the projected channel. Note that the (non-trivial) ZF-based solution exists only when $N > \sum_{j=1}^J D_j$. From the above two special cases, we observe that as Γ_j's decrease, the optimal $\mathbf{S}$ should evolve along with a sequence of subspaces of $\mathbf{H}$ with decreasing dimensions as a result of keeping certain orthogonality to $\mathbf{G}_j$'s, which motivates a new design method for cognitive beamforming, named as *partial channel projection* [13]. Specifically, let $\bar{\mathbf{G}}_j = \mathbf{G}_j/\Gamma_j, \forall j$. Then, define $\bar{\mathbf{G}} \triangleq [\bar{\mathbf{G}}_1^T,\cdots,\bar{\mathbf{G}}_J^T]^T$. Denote the SVD of $\bar{\mathbf{G}}$ as $\bar{\mathbf{G}} = \mathbf{U}_G\mathbf{\Lambda}_G\mathbf{V}_G^H$. Without loss of generality, assume that the singular values in $\mathbf{\Lambda}_G$ are arranged in a decreasing order. Then, we propose a generalized channel projection operation:

$$\mathbf{H}_\perp = \mathbf{H}\left(\mathbf{I} - \mathbf{V}_G^{(b)}\left(\mathbf{V}_G^{(b)}\right)^H\right) \tag{12}$$

where $\mathbf{V}_G^{(b)}$ consists of the first b columns of $\mathbf{V}_G$ corresponding to the b largest singular values in $\mathbf{\Lambda}_G$, $1 \le b \le \min(N-1, \sum_{j=1}^J D_j)$. Note that b could also take a zero value for which $\mathbf{V}_G^{(0)} \triangleq \mathbf{0}$. Now, we are ready to present the transmit covariance matrix for the partial projection method in the form of its eigenvalue decomposition (EVD) as $\mathbf{S} = \mathbf{V}_\perp\mathbf{\Sigma}_\perp\mathbf{V}_\perp^H$, where $\mathbf{V}_\perp$ is obtained from the SVD of the projected channel $\mathbf{H}_\perp$, i.e., $\mathbf{H}_\perp = \mathbf{U}_\perp\mathbf{\Lambda}_\perp\mathbf{V}_\perp^H$. By substituting this new form of $\mathbf{S}$ into (P1), it can be shown that the problem reduces to maximizing the sum-rate of a set of parallel channels (with channel gains given by $\mathbf{\Lambda}_\perp$) over their power allocation $\mathbf{\Sigma}_\perp$ subject to $(J+1)$ linear power constraints, for which the optimal power allocation can be obtained by a generalized "water-filling" algorithm [13]. Note that the partial channel projection works for any values of N and D_j's.

In Fig. 2, we plot the achievable rate of a CR MIMO channel under the PTPC and PIPCs with the optimal transmit covariance solution for (P1) via the convex optimization approach, against those with suboptimal covariance solutions via the partial channel projection method with different values of b. The system parameters are given as follows: $M = N = 4$, $J = 2$, $D_1 = D_2 = 1$, and $\Gamma_1 = \Gamma_2 = 0.1$. The SU achievable rate is plotted vs. the SU PTPC, P. It is observed that the optimal covariance solution obtained via the convex optimization approach yields notable rate gains over suboptimal solutions via the heuristic method, for which the optimal value of b (the number of SU-to-PU channel dimensions to be nulled) to maximize the SU achievable rate increases with the SU PTPC.

CR MIMO-MAC: The solutions proposed for the CR point-to-point MIMO channel shed insights to transmit optimization for the CR MIMO-MAC defined in (1) with $K > 1$. Assume that in the CR MIMO-MAC, the optimal multiuser detection is deployed at the S-BS to successively decode different SU messages from the received sum-signal. We then consider the problem for jointly optimizing SU transmit covariance matrices to maximize their weighted sum-rate subject to individual PTPCs and joint PIPCs. This problem is referred to as weighted sum-rate maximization (WSRMax). Without loss of generality, we assume that the given user rate weights satisfy that $\mu_1 \ge \mu_2 \ge \cdots \ge \mu_K \ge 0$; thus, the optimal decoding order of users at the S-BS to maximize the weighted sum-rate is in accordance with the reverse user index [19]. Accordingly, the WSRMax for the CR MIMO-MAC can be expressed as

$$\begin{aligned}\underset{\mathbf{S}_1,\cdots,\mathbf{S}_K}{\text{Max.}} \quad & \sum_{k=1}^K \mu_k \log\frac{\left|\mathbf{I}+\sum_{i=1}^k \mathbf{H}_i\mathbf{S}_i\mathbf{H}_i^H\right|}{\left|\mathbf{I}+\sum_{i=1}^{k-1} \mathbf{H}_i\mathbf{S}_i\mathbf{H}_i^H\right|} \\ \text{s. t.} \quad & \mathrm{Tr}(\mathbf{S}_k) \le P_k, \ k = 1,\ldots,K \\ & \sum_{k=1}^K \mathrm{Tr}\left(\mathbf{G}_{kj}\mathbf{S}_k\mathbf{G}_{kj}^H\right) \le \Gamma_j, \ j = 1,\cdots,J \\ & \mathbf{S}_k \succeq \mathbf{0}, \ k = 1,\cdots,K.\end{aligned} \tag{P2}$$

Reordering terms in the objective function of (P2) yields

$$\begin{aligned}&\sum_{k=1}^{K-1}(\mu_k - \mu_{k+1})\log\left|\mathbf{I}+\sum_{i=1}^k \mathbf{H}_i\mathbf{S}_i\mathbf{H}_i^H\right| \\ &+ \mu_K \log\left|\mathbf{I}+\sum_{i=1}^K \mathbf{H}_i\mathbf{S}_i\mathbf{H}_i^H\right|\end{aligned} \tag{13}$$

From the above new form of the objective function, it can be verified that (P2) is a convex optimization problem over $\mathbf{S}_k$'s. Thus, similarly as for (P1), (P2) can be solved by an interior-point-method based algorithm or an iterative algorithm via solving the equivalent Lagrange dual problem, for which the details are omitted here for brevity.

It is noted that (P2) is for the case with the optimal non-linear multiuser decoder at the S-BS, while in practice the low-complexity linear decoder is usually more preferable. The

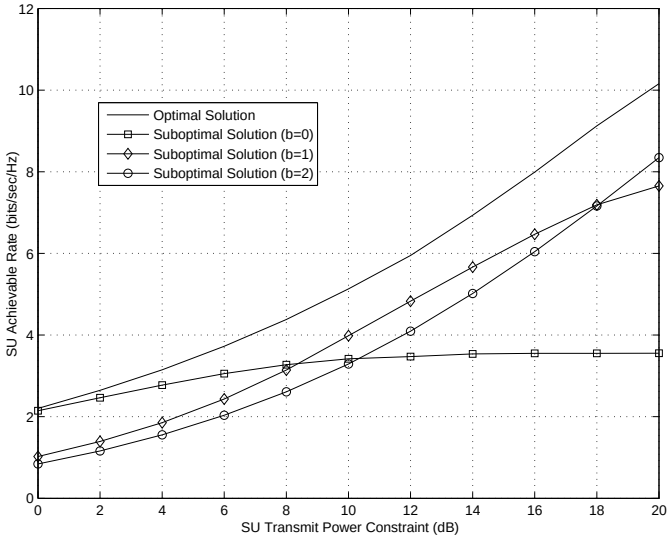


Fig. 2. Comparison of the achievable rates for the CR MIMO channel under the PTPC and PIPCs with the optimal transmit covariance solution for (P1) via the the convex optimization approach vs. suboptimal covariance solutions via the partial channel projection method with different values of b.

use of linear instead of non-linear decoder at the receiver will change the user achievable rates for the CR MIMO-MAC, thus resulting in new problem formulations for transmit optimization. For example, in [20], the authors have considered the CR SIMO-MAC (single-antenna for each SU transmitter) with a linear decoder at the receiver, where the power allocation across the SUs is optimized to maximize their signal-to-interference-plus-noise ratios (SINRs) at the receiver subject to both transmit and interference power constraints.

B. Non-Convex Problem Formulation

Next, we consider the case where the optimization problems in the associated traditional (non-CR) MIMO systems are *non-convex*. It thus becomes more challenging whether these non-convex problems with the addition of convex PIPCs in the corresponding CR MIMO systems can be efficiently solvable. In the following, we present some promising approaches to solve these problems for the CR MIMO-BC and MIMO-IC.

CR MIMO-BC: First, consider the CR MIMO-BC defined in (2) under both the PTPC at the S-BS and J PIPCs each for one of the J PUs, which can be similarly defined as for the MAC case in (4) and (6), respectively. We focus on the unicast downlink transmission for the CR BC, while for the case of multicast, the interested readers may refer to [21]. For the purpose of exposition, we consider two commonly adopted design criteria for the traditional multi-antenna Gaussian BC in the literature: One is for the MIMO-BC deploying the non-linear "dirty-paper-coding (DPC)" at the transmitter [22], which maximizes the weighted sum-rate of all the users (i.e., the WSRMax problem); the other is for the MISO-BC (single-antenna for each SU receiver) deploying only linear encoding at the transmitter, which maximizes the minimum SINR among all the users, referred to as "SINR balancing".

Specifically, the WSRMax problem for the CR MIMO-BC can be formulated as:

$$\begin{aligned}\underset{\mathbf{Q}_1,\cdots,\mathbf{Q}_K}{\text{Max.}} \quad & \sum_{k=1}^{K}\mu_k \log \frac{\left|\mathbf{I}+\mathbf{H}_k^H\left(\sum_{i=k}^{K}\mathbf{Q}_i\right)\mathbf{H}_k\right|}{\left|\mathbf{I}+\mathbf{H}_k^H\left(\sum_{i=k+1}^{K}\mathbf{Q}_i\right)\mathbf{H}_k\right|} \qquad \text{(P3)}\\ \text{s. t.} \quad & \sum_{k=1}^{K}\text{Tr}(\mathbf{Q}_k)\le P\\ & \text{Tr}\left(\mathbf{F}_j\left(\sum_{k=1}^{K}\mathbf{Q}_k\right)\mathbf{F}_j^H\right)\le\Gamma_j,\ j=1,\cdots,J\\ & \mathbf{Q}_k\succeq\mathbf{0},\ k=1,\cdots,K\end{aligned}$$

where $\mathbf{Q}_k \in \mathbb{C}^{M\times M}$ denotes the covariance matrix for the transmitted signal of S-BS intended for the kth SU, $k=1,\cdots,K$; μ_k's are the given user rate weights; and P denotes the transmit power constraint for the S-BS. Without loss of generality, we assume that $\mu_1\ge\mu_2\ge\cdots\ge\mu_K\ge 0$; thus, in (P3) the optimal encoding order of users for DPC to maximize the weighted sum-rate is in accordance with the user index [22]. Note that (P3) is non-convex with or without the PIPCs due to the fact that the objective function is non-concave over $\mathbf{Q}_k$'s for $K\ge 2$. As a result, unlike (P1) for the point-to-point CR channel, the standard Lagrange duality method cannot be applied for this problem. For (P3) in the case without the PIPCs, a so-called "BC-MAC duality" relationship was proposed in [23] to transform the non-convex MIMO-BC problem into an equivalent convex MIMO-MAC problem, which is solvable by efficient convex optimization techniques such as the interior point method. In [24], another form of BC-MAC duality, the so-called "mini-max duality" was explored to solve the MIMO-BC problem under a special case of GLTCC: the per-antenna transmit power constraint. However, these existing forms of BC-MAC duality are yet unable to handle the case with arbitrary numbers of GLTCCs, which is the case for (P3) with both the PTPC and PIPCs.

In [25], a general method was proposed to solve various MIMO-BC optimization problems under multiple GLTCCs, thus including the CR MIMO-BC WSRMax problem given in (P3). For this method, the first step is to combine all $(J+1)$ power constraints in (P3) into a single GLTCC as shown in the following optimization problem:

$$\begin{aligned}\underset{\mathbf{Q}_1,\cdots,\mathbf{Q}_K}{\text{Max.}} \quad & \sum_{k=1}^{K}\mu_k \log \frac{\left|\mathbf{I}+\mathbf{H}_k^H\left(\sum_{i=k}^{K}\mathbf{Q}_i\right)\mathbf{H}_k\right|}{\left|\mathbf{I}+\mathbf{H}_k^H\left(\sum_{i=k+1}^{K}\mathbf{Q}_i\right)\mathbf{H}_k\right|}\\ \text{s. t.} \quad & \text{Tr}\left(\mathbf{A}\sum_{k=1}^{K}\mathbf{Q}_k\right)\le Q\\ & \mathbf{Q}_k\succeq\mathbf{0},\ k=1,\cdots,K \qquad (14)\end{aligned}$$

where $\mathbf{A}=\lambda_0\mathbf{I}+\sum_{j=1}^{J}\lambda_j\mathbf{F}_j^H\mathbf{F}_j$, and $Q=\lambda_0 P+\sum_{j=1}^{J}\lambda_j\Gamma_j$ with $\lambda_0,\lambda_1,\cdots,\lambda_J$ being non-negative constants. For a given set of λ_i's, $i=0,\cdots,J$, let the optimal value of the above problem be denoted by $F(\lambda_0,\lambda_1,\cdots,\lambda_J)$. Clearly, $F(\lambda_0,\lambda_1,\cdots,\lambda_J)$ is an upper bound on the optimal value of (P3) since any feasible solutions for (P3) must satisfy the constraints of Problem (14) for a given set of λ_i's. Interestingly, it can be shown that the optimal value of (P3)

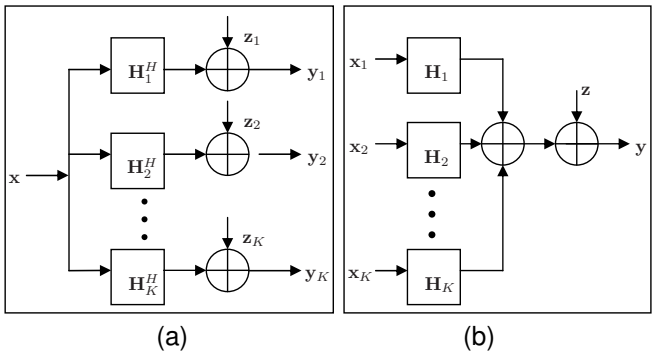


Fig. 3. Generalized MIMO MAC-BC Duality: (a) Primal MIMO-BC channel with downlink channels $\mathbf{H}_k^H$ and receiver noise vectors $\mathbf{z}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$, $k = 1, \ldots, K$, and a GLTCC: $\mathrm{Tr}(\mathbf{A}\sum_{k=1}^K \mathbf{Q}_k) \leq Q$; (b) Dual MIMO-MAC with uplink channels $\mathbf{H}_k, k = 1, \ldots, K$ and receiver noise vector $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, \mathbf{A})$, and a sum-power constraint: $\sum_{k=1}^K \mathrm{Tr}(\mathbf{S}_k) \leq Q$. The MIMO-BC and dual MIMO-MAC have the same achievable rate region [25].

is equal to the minimum value of function $F(\lambda_0, \lambda_1, \cdots, \lambda_J)$ over all non-negative λ_i's [25]. Therefore, (P3) can be resolved by iteratively solving Problem (14) for a given set of λ_i's and updating λ_i's towards their optimal values to minimize function $F(\lambda_0, \lambda_1, \cdots, \lambda_J)$. Specifically, λ_i's can be updated via the ellipsoid method according to the subgradients of $F(\lambda_0, \lambda_1, \cdots, \lambda_J)$, which can be shown [25] equal to $P - \sum_{k=1}^K \mathrm{Tr}(\mathbf{Q}_k^\star)$ and $\Gamma_j - \mathrm{Tr}(\mathbf{F}_j(\sum_{k=1}^K \mathbf{Q}_k^\star)\mathbf{F}_j^H)$ for λ_0 and λ_j $(j \neq 0)$, respectively, where $\mathbf{Q}_k^\star$'s are the solution of Problem (14) for the given λ_k's.

Furthermore, Problem (14) with a given set of λ_k's can be solved by applying the *generalized BC-MAC duality* proposed in [25], which extends the existing forms of BC-MAC duality [23], [24] to transform the MIMO-BC problem subject to a single GLTCC as in Problem (14) to an auxiliary (dual) MIMO-MAC problem subject to a corresponding sum-power constraint. Specifically, it is shown in [25] that the MIMO-BC as in Problem (14) and the dual MIMO-MAC, as depicted in Fig. 3, have the same achievable rate region. Accordingly, the optimal objective value (weighted sum-rate) of Problem (14) for the primal MIMO-BC can be obtained as that of the following equivalent problem for the dual MIMO-MAC:

$$\begin{aligned} \underset{\mathbf{S}_1,\cdots,\mathbf{S}_K}{\text{Max.}} \quad & \sum_{k=1}^{K-1}(\mu_k - \mu_{k+1}) \log \left| \mathbf{A} + \sum_{i=1}^{k} \mathbf{H}_i \mathbf{S}_i \mathbf{H}_i^H \right| \\ & + \mu_K \log \left| \mathbf{A} + \sum_{i=1}^{K} \mathbf{H}_i \mathbf{S}_i \mathbf{H}_i^H \right| \\ \text{s. t.} \quad & \sum_{k=1}^{K} \mathrm{Tr}\left(\mathbf{S}_k\right) \leq Q \\ & \mathbf{S}_k \succeq \mathbf{0}, \ k = 1, \cdots, K. \end{aligned} \tag{15}$$

Similar to (P2), the above problem is a WSRMax problem for the MIMO-MAC subject to a single sum-power constraint, which is convex and thus can be efficiently solvable by, e.g., the interior point method. After solving Problem (15), the optimal user transmit covariance solutions for the MIMO-MAC, $\mathbf{S}_k^\star$'s, can be transformed to the corresponding ones for the original MIMO-BC, $\mathbf{Q}_k^\star$'s, via a MAC-BC covariance transformation algorithm given in [25]. Furthermore, it is worth noting that with $K = 1$, the above method can be shown equivalent to that developed for (P1) in the CR point-to-point MIMO channel case based on the Lagrange duality.

Consider next the SINR balancing problem for the CR MISO-BC, which can be expressed as:

$$\begin{aligned} \underset{\alpha, \mathbf{v}_1, \cdots, \mathbf{v}_K}{\text{Max.}} \quad & \alpha \\ \text{s. t.} \quad & \frac{\|\mathbf{h}_k^H \mathbf{v}_k\|^2}{1 + \sum_{i \neq k} \|\mathbf{h}_k^H \mathbf{v}_i\|^2} \geq \alpha, \ k = 1, \cdots, K \\ & \sum_{k=1}^{K} \|\mathbf{v}_k\|^2 \leq P \\ & \sum_{k=1}^{K} \|\mathbf{F}_j \mathbf{v}_k\|^2 \leq \Gamma_j, \ j = 1, \cdots, J \end{aligned} \tag{P4}$$

where α denotes an achievable SINR for all the SUs; $\mathbf{v}_k \in \mathbb{C}^{M \times 1}$ denotes the precoding vector for the transmitted signal of S-BS intended for the kth SU; and $\mathbf{h}_k$ represents $\mathbf{H}_k$ for the MISO-BC case. Similarly as for (P1-S), by treating $\mathbf{h}_k^H \mathbf{v}_k$ on the left-hand side (LHS) of each SINR constraint in (P4) as a positive real number [26], it can be shown that (P4) for a given α is equivalent to a SOCP feasibility problem and thus efficiently solvable [15]. For a given α, if the associated SOCP problem is feasible, we know that the optimal solution of (P4) for α, denoted by $\alpha^\star$, must satisfy $\alpha^\star \geq \alpha$; otherwise, $\alpha^\star < \alpha$. Based on this fact, $\alpha^\star$ can be found by a simple bisection search [14]; with $\alpha^\star$, the corresponding optimal solution for $\mathbf{v}_k$'s in (P4) can also be obtained. The above technique has also been applied in [27] for (P4) without the PIPCs.

The SINR balancing problem for the conventional MISO-BC without the PIPCs has also been studied in [28], where an algorithm was proposed using the virtual uplink formulation and a fixed-point iteration. However, this algorithm cannot be extended directly to deal with multiple PIPCs for the case of CR MISO-BC. Similarly as for the previous discussions on the WSRMax problem for the CR MIMO-BC where a generalized MIMO MAC-BC duality holds, a counterpart beamforming duality also holds for the MISO-BC and SIMO-MAC [25]. With this duality result, the SINR balancing problem (P4) for the CR MISO-BC can be converted into an equivalent problem for the dual SIMO-MAC, where the efficient iterative algorithm in [28] can be directly applied. The interested readers may refer to [25] for the details of this method.

CR MIMO-IC: Second, consider the CR MIMO-IC given in (3), subject to both the PTPCs for the K SU-TXs and the PIPCs for the J PUs, which can be similarly defined as for the MAC case in (4) and (6), respectively. From an information-theoretic perspective, the capacity region for the Gaussian IC under PTPCs, which consists of all the simultaneously achievable rates of all the users, still remains unknown in general even for the case of $K = 2$ and $A_k = B_k = 1, k = 1, 2$ [29]. A pragmatic approach that leads to suboptimal achievable rates in the Gaussian IC is to restrict the system to operate in a decentralized manner, i.e., allowing only single-user encoding and decoding by treating the co-channel interferences from the other users as additional Gaussian noises. For this approach, transmit optimization for the CR MIMO-IC reduces to finding

a set of optimal transmit covariance matrices for the K SU links, denoted by $\mathbf{R}_k \in \mathbb{C}^{A_k \times A_k}, k = 1, \cdots, K$, to maximize the secondary network throughput under both the PTPCs and PIPCs. More specifically, the WSRMax problem for the CR MIMO-IC can be expressed as:

$$\underset{\mathbf{R}_1,\cdots,\mathbf{R}_K}{\text{Max.}} \sum_{k=1}^{K} \mu_k \log \left| \mathbf{I} + (\mathbf{I} + \sum_{i \neq k} \mathbf{H}_{ik}\mathbf{R}_i\mathbf{H}_{ik}^H)^{-1}\mathbf{H}_{kk}\mathbf{R}_k\mathbf{H}_{kk}^H \right| \tag{P5}$$

$$\text{s. t.} \quad \text{Tr}(\mathbf{R}_k) \leq P_k, \ k = 1, \cdots, K$$
$$\sum_{k=1}^{K} \text{Tr}\left(\mathbf{E}_{kj}\mathbf{R}_k\mathbf{E}_{kj}^H\right) \leq \Gamma_j, \ j = 1, \cdots, J$$
$$\mathbf{R}_k \succeq \mathbf{0}, \ k = 1, \cdots, K$$

where μ_k's are the given non-negative user rate weights. We see that (P5) is non-convex with or without the PIPCs due to the fact that the objective function is non-concave over $\mathbf{R}_k$'s for $K > 1$. As a result, there are no efficient algorithms yet to obtain the globally optimal solution for this problem. For the same problem setup, there have been recent progresses on characterizing the maximum achievable "degrees of freedom (DoF)" for the user sum-rate (i.e., $\mu_k = 1, \forall k$) [30].

Next, we discuss some feasible solutions for (P5). First, it is worth noting that for (P5) in the case without the PIPCs, a commonly adopted suboptimal approach is to iteratively optimize each user's transmit covariance subject to its individual PTPC with the transmit covariances of all the other users fixed. This decentralized approach has been first proposed in [31], [32] to obtain some local optimal points for (P5) with the PTPCs only, where they differ in that the one in [31] maximizes the user individual rate at each iteration, while the one in [32] maximizes the user weighted sum-rate. It is also noted that a parallel line of works with similar iterative user optimizations has been pursued in the single-antenna but multi-carrier based interference channels such as the wired discrete-multi-tone (DMT) based digital subscriber line (DSL) network [33], and the wireless OFDM based ad hoc network [34]. One important question to answer for such iterative algorithms is under what conditions the algorithm will guarantee to converge to a local optimal point. This problem has been addressed in the contexts of both multi-carrier and multi-antenna based interference channels in, e.g., [35], [36], via game-theoretic approaches.

However, the above iterative approach cannot be applied directly to solve (P5) with both the PIPCs and PTPCs, since each PIPC involves all the user transmit covariances and is thus not separable over the SUs. Thus, a feasible approach for (P5) is to decompose each of the J PIPCs into a set of interference-power constraints over the K SU-TXs, i.e., for the jth PIPC, $j \in \{1, \cdots, J\}$,

$$\text{Tr}\left(\mathbf{E}_{kj}\mathbf{R}_k\mathbf{E}_{kj}^H\right) \leq \Gamma_j^{(k)}, \ k = 1, \cdots, K \tag{16}$$

where $\Gamma_j^{(k)}$ is a constant, and all $\Gamma_j^{(k)}$'s, $k = 1, \cdots, K$, satisfy $\sum_k \Gamma_j^{(k)} \leq \Gamma_j$ such that the jth PIPC is guaranteed. Then, the iterative algorithm works here, where each SU link independently optimizes $\mathbf{R}_k$ to maximize its achievable rate under its PTPC and J interference-power constraints given by (16), with all other $\mathbf{R}_i$'s, $i \neq k$, fixed. It is observed that the resulting problem is in the same form of our previously studied (P1) for the CR point-point MIMO channel; thus, similar techniques developed for (P1) can be applied. Note that a suboptimal method for this problem in the same spirit of the partial channel projection method to reduce the design complexity for each SU transmit covariance matrix has also been proposed in [37]. Moreover, it is noted that $\Gamma_j^{(k)}$'s, $j = 1, \cdots, J$, $k = 1, \cdots, K$, can be searched over the SUs to further improve their weighted sum-rate.

Alternatively, assuming that a centralized optimization is feasible with the global knowledge of all the channels in the SU network, as well as those from different SU-TXs to all PUs, another heuristic algorithm for (P5) was proposed in [38]. By rewriting the SU transmit covariance matrices into their equivalent precoding vectors and power allocation vectors, this algorithm iteratively updates the SU transmit precoding vectors (based on the "network duality" [39]) or the power allocation vectors (by solving geometric programming (GP) problems [40]), with the others being fixed.

It is worth pointing out that there are other problem formulations different from (P5) to address the transmit optimization for the CR MIMO-IC. In [41], a new criterion was proposed to design the SU link transmission in a CR MISO-IC via an alternative decentralized approach, where each SU-TX independently designs its transmit precoding vector to maximize the ratio between the received signal power at the desired SU-RX and the resulted total interference power at all the PUs, in order to regulate the interference powers at PUs. Moreover, the above discussions are all based on the assumption that each SU-RX treats the interferences from all the other SU links as additional noises, which is of practical interest since it simplifies the receiver design for each SU link. However, due to independent cross-link channels between SU terminals, it may be possible that a SU-RX could occasionally observe "strong" interference signals from some co-existing SU-TXs and thus be able to decode their messages via multiuser detection techniques and then cancel the associated interferences. With such "opportunistic" multiuser detection at each SU-RX, the achievable rate of each SU link becomes a function of not only its own transmit covariance, but also those of the other SUs as well as their instantaneous transmit rates. Thus, the corresponding transmit optimization for the CR MIMO-IC leads to new and more challenging problem formulations than (P5); the interested readers may refer to [42], [43].

IV. Joint Space-Time-Frequency DRA Optimization

In the previous section, we have studied DRA for different CR networks at a single transmit dimension in time/frequency, by focusing on spatial-domain transmit optimization under the *peak* transmit and interference power constraints (PTPC and PIPC). In this section, we bring the additional time and/or frequency dimensions into the DRA problem formulations, by applying the *average* transmit and interference power constraints (ATPC and AIPC) in CR networks. Consider the DRA over L time/freqeuncy dimensions, for which all the required

channel knowledge is assumed to be known. Taking the CR MAC as an example (similar arguments can be developed for the CR BC/IC), under both the ATPCs and AIPCs given in (5) and (7), respectively, a generic problem formulation for DRA optimization can be formulated as:

$$\begin{aligned} \underset{\mathbf{S}_k[l]\succeq\mathbf{0},\forall k,l}{\text{Max.}} \quad & C(\{\mathbf{S}_k[l]\}) \qquad\qquad (\text{P6}) \\ \text{s. t.} \quad & (5), (7) \end{aligned}$$

where $\{\mathbf{S}_k[l]\}$ denotes the set of $\mathbf{S}_k[l]$'s, $k = 1, \cdots, K$, and $l = 1, \cdots, L$, while $C(\cdot)$ is an arbitrary utility function to measure the CR network performance. We assume that $C(\cdot)$ is separable over l's, i.e., $C(\{\mathbf{S}_k[l]\}) = \frac{1}{L}\sum_{l=1}^{L} U_l(\mathbf{S}_1[l], \cdots, \mathbf{S}_K[l])$ with $U_l(\cdot)$'s denoting individual utility functions. Since both the ATPC and AIPC involve L transmit covariance matrices, the *Lagrange dual decomposition* (see, e.g., a tutorial paper [44]) is a general method to deal with this type of average constraints for optimization over a number of parallel dimensions, which is explained as follows. By introducing a set of dual variables, ν_k's, each for one of the K ATPCs, and δ_k's, each for one of the J AIPCs, the Lagrange dual problem of (P6) can be written as (P6-D):

$$\underset{\boldsymbol{\nu}\succeq\mathbf{0},\boldsymbol{\delta}\succeq\mathbf{0}}{\text{Min.}} \quad d(\boldsymbol{\nu},\boldsymbol{\delta})$$

with $\boldsymbol{\nu} = [\nu_1, \cdots, \nu_K], \boldsymbol{\delta} = [\delta_1, \cdots, \delta_J]$, and the dual function

$$\begin{aligned} d(\boldsymbol{\nu},\boldsymbol{\delta}) \triangleq \max_{\mathbf{S}_k[l]\succeq\mathbf{0},\forall k,l} & C(\{\mathbf{S}_k[l]\}) - \sum_{k=1}^{K}\nu_k\Big(\frac{1}{L}\sum_{l=1}^{L}\text{Tr}(\mathbf{S}_k[l]) \\ & -\bar{P}_k\Big) - \sum_{j=1}^{J}\delta_j\Big(\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K}\text{Tr}\left(\mathbf{G}_{kj}[l]\mathbf{S}_k[l]\mathbf{G}_{kj}^H[l]\right) - \bar{\Gamma}_j\Big). \end{aligned} \tag{17}$$

Since the dual problem (P6-D) is convex regardless of the convexity of the primal problem (P6) [14], (P6-D) can be efficiently solved by the ellipsoid method according to the subgradients of the dual function $d(\boldsymbol{\nu},\boldsymbol{\delta})$, similarly as in our previous discussions, provided that the maximization problem in (17) is solvable for any given set of $\boldsymbol{\nu}$ and $\boldsymbol{\delta}$. It is interesting to observe that this maximization problem can be decomposed into L parallel subproblems each for one of the L dimensions, and all of these subproblems have the same structure and are thus solvable by the same algorithm, a practice known as "dual decomposition". Without loss of generality, we drop the dimension index l and express each subproblem as

$$\max_{\mathbf{S}_k\succeq\mathbf{0},\forall k} U(\mathbf{S}_1, \cdots, \mathbf{S}_K) - \sum_{k=1}^{K}\text{Tr}\left(\mathbf{B}_k(\nu_k,\boldsymbol{\delta})\mathbf{S}_k\right) \tag{18}$$

where $\mathbf{B}_k(\nu_k,\boldsymbol{\delta}) = \nu_k\mathbf{I} + \sum_{j=1}^{J}(\delta_j\mathbf{G}_{kj}^H\mathbf{G}_{kj})$ is a constant matrix for the given ν_k and $\boldsymbol{\delta}$, $k = 1 \cdots, K$.

We then discuss the following two cases. For the first case, consider that $U_l(\cdot)$ is a concave function over $\mathbf{S}_k[l]$'s, $\forall l$ (e.g., the point-to-point CR channel capacity in (P1), or the weighted sum-rate for the CR MIMO-MAC in (P2)). Then, (P6) is convex and thus the duality gap between the optimal values of (P6) and (P6-D) is zero, i.e., (P6) and (P6-D) are equivalent problems. Furthermore, each subproblem in (18) is also convex. Thus, the dual decomposition method solves (P6) via its dual problem (P6-D), which is decomposable into L convex subproblems. For the second case, as a counterpart, consider that $U_l(\cdot)$ is non-concave over $\mathbf{S}_k[l]$'s (e.g., the weighted sum-rate for the CR MIMO-BC/MIMO-IC in (P3)/(P5)). As a result, (P6) is non-convex and the duality gap between (P6) and (P6-D) may not be zero. Furthermore, the subproblem (18) is also non-convex. For this case, even when the optimal solutions of the L subproblems are obtainable, the optimal value of (P6-D) in general only serves as an upper bound on that of (P6). However, in [45] it is pointed out that if a set of so-called "time-sharing" conditions are satisfied by a non-convex optimization problem, the duality gap for this problem and its dual problem is indeed zero. Furthermore, for the class of DRA problems in the form of (P6), the associated time-sharing conditions are usually satisfied asymptotically as $L \to \infty$ under some cautious considerations on the continuity of channel distributions [46]. Therefore, the dual decomposition method could still be applied to solve (P6) in the non-convex case for sufficiently large values of L, provided that the optimal solutions for the subproblems in (18) are obtainable (e.g., a variation of (P3) for the CR MIMO-BC). However, with finite values of L, how to efficiently solve (P6) in the case of non-concave objective functions is still open.

With the above discussions on the general approaches to design joint space-time-frequency DRA for CR networks, we next present some examples of unique interests to CR systems.

A. TDMA/FDMA Constrained DRA: When Is It Optimal?

Time-/frequency-division multiple-access (TDMA/FDMA), which schedules only one user for transmission at each time/frequency dimension, is usually preferable in practice due to their implementation ease. For the TDMA/FDMA based CR MAC (similar arguments hold for the CR BC/IC), the optimal DRA over L transmit dimensions to maximize the sum-capacity of the SUs can be formulated as (P6) with properly chosen functions for $U_l(\cdot)$'s, where for any given l, $U_l(\cdot)$ is expressed as (l is dropped for conciseness)

$$U(\mathbf{S}_1, \cdots, \mathbf{S}_K) = \begin{cases} \log\left|\mathbf{I} + \mathbf{H}_k\mathbf{S}_k\mathbf{H}_k^H\right| & \mathbf{S}_i = \mathbf{0}, \forall i \neq k \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Note that $U(\cdot)$ defined above implies the TDMA/FDMA constraint, i.e., only scheduling one user for transmission at a given dimension with a positive contribution to the sum-capacity. However, it can be shown that $U(\cdot)$ is non-concave over $\mathbf{S}_k$'s in this case and as a result, the corresponding (P6) is non-convex. Nevertheless, according to our previous discussions, since the time-sharing conditions hold approximately when $L \to \infty$, the dual decomposition method can be applied to solve (P6) for this case with very large values of L, where the optimal solution of the associated subproblem at each dimension given in (18) can be obtained by finding the SU (selected for transmission) with the largest objective value of the following problem (which is of the same form as Problem

([10]) and thus solvable in a similar way):

$$\max_{\mathbf{S}_k \succeq \mathbf{0}} \ \log\left|\mathbf{I}+\mathbf{H}_k\mathbf{S}_k\mathbf{H}_k^H\right| - \mathrm{Tr}(\mathbf{B}_k(\nu_k,\boldsymbol{\delta})\mathbf{S}_k). \tag{20}$$

An important question to investigate for TDMA/FDMA based DRA is how much the performance is degraded as compared with the optimal DRA that allows more than one users to transmit at a given dimension. From an information-theoretic viewpoint, it is thus pertinent to investigate the conditions for the optimality of TDMA/FDMA, i.e., when they are optimal to achieve the system sum-capacity. For the traditional single-antenna fading MAC under the user ATPCs over time, it has been shown in [47] that TDMA is optimal for achieving the ergodic/long-term sum-capacity. This result has been shown to hold for the fading CR MAC and CR BC under both the ATPCs and AIPCs in [48], where by exploiting the KKT optimality conditions of the associated optimization problems, the optimality conditions for TDMA in other cases of combined peak/average transmit/interference power constraints have been characterized. For the traditional single-antenna IC with interference treated as noise, the optimality of TDMA/FDMA for the sum capacity has been investigated under the ATPCs in [49], [50]. It would be interesting to extend these results to the case of CR IC under the additional PIPCs and/or AIPCs.

B. Peak vs. Average Interference Power Constraints: A New Interference Diversity

From a SU's perspective, it is obvious that the ATPC/AIPC is more flexible than the PTPC/PIPC for DRA under the same power threshold and thus results in a larger SU link capacity. However, from a PU's perspective, it remains unknown whether the AIPC or PIPC causes more PU link performance degradation. Intuitively speaking, the PIPC should be more favorable than the AIPC since the former limits the interference power at the PU to be below certain threshold at each time/frequency dimension, while the latter results in variations of interference power levels over different dimensions although their average level is kept below the same threshold as that for the PIPC.

Somehow surprisingly, in [51] it is shown that for the single-antenna PU fading channel subject to the interference from a SU transmitter, the AIPC is in fact better than its PIPC counterpart under the same average power threshold in terms of minimizing the PU capacity losses, which holds for the cases of both ergodic and outage capacities of the PU channel, with/without power control. To illustrate this result, we consider for simplicity the case without the PU link power control, i.e., the PU transmits with a constant power, Q, over all the fading states. Suppose that the PU link channel power gain is denoted by h_p, and that from the SU transmitter to the PU receiver denoted by h_{sp}. Next, consider the following two cases, where the interference power from the SU transmitter at the PU receiver, denoted by $I_{sp}=h_{sp}p_s$, with p_s denoting the SU transmit power, is fixed over all the fading states in Case I (corresponding to the case of PIPC), and is allowed to be variable in Case II (corresponding to the case of AIPC). For both cases, a constant interference power threshold Γ is set and is assumed to hold with equality, i.e., for Case I, $I_{sp}^{(\mathrm{I})}=\Gamma$, for all the fading states, while for Case II, $\mathrm{E}(I_{sp}^{(\mathrm{II})})=\Gamma$. Taking the PU ergodic capacity as an example, which can be expressed as (assuming unit-power receiver Gaussian noise):

$$C_p = \mathrm{E}\left(\log\left(1+\frac{h_pQ}{1+I_{sp}}\right)\right). \tag{21}$$

Let $C_p^{(\mathrm{I})}$ and $C_p^{(\mathrm{II})}$ denote the values of C_p in Cases I and II, respectively. The following equalites/inequalities then hold

$$\begin{aligned} C_p^{(\mathrm{I})} &= \mathrm{E}_{h_p}\left(\log\left(1+\frac{h_pQ}{1+\Gamma}\right)\right) \\ &= \mathrm{E}_{h_p}\left(\log\left(1+\frac{h_pQ}{1+\mathrm{E}(I_{sp}^{(\mathrm{II})})}\right)\right) \\ &\le \mathrm{E}_{h_p}\left(\mathrm{E}_{I_{sp}}\left(\log\left(1+\frac{h_pQ}{1+I_{sp}^{(\mathrm{II})}}\right)\right)\right) \qquad (22) \\ &= C_p^{(\mathrm{II})} \end{aligned}$$

where (22) is due to the Jensen's inequality (see, e.g., [17]) and the convexity of the function $f(x)=\log\left(1+\frac{\kappa}{1+x}\right)$ where κ is any positive constant and $x\ge 0$. Thus, it follows that given the same average power of the interference, Γ, it is desirable for the PU to have the instantaneous interference power I_{sp} fluctuate over fading states (Case II) rather than stay constant (Case I), to achieve a larger ergodic capacity.

In general, the results in [51] reveal a new *interference diversity* phenomenon for SS-based CR networks, i.e., the randomized interference powers from the secondary network can be more advantageous over deterministic ones across different transmit dimensions over space, time, or frequency for minimizing the resulted primary network capacity losses. Further investigations are required on interference diversity driven DRA for CR or other spectrum sharing systems.

C. Beyond Interference Temperature: Exploiting Primary Link Performance Margins

So far, we have studied DRA for CR networks based on the IT constraints for protecting the PU transmissions. Given that the IT constraints in general conservatively lead to an upper bound on the PU capacity loss due to the interference from the SUs [13], [52], it would be possible to improve the spectrum sharing capacities for both the SUs and PUs over the IT-based methods if additional cognition on the PU transmissions is available at the CR transmitters. For example, by exploiting CSI of the PU links, the CRs could allocate transmit/interference powers more flexibly over the dimensions where the PU channels exhibit poor conditions, without degrading too much the PU link performances. These PU "null" dimensions could come up in time, frequency, or space. Thus, the IT constraints could be replaced by the more relevant *primary link performance margin constraints* [52], [53] for the design of DRA in CR networks, in order to optimally exploit the available primary link performance margins to accommodate the interference from the SUs. Following this new paradigm, many new and challenging DRA problems can be formulated for CR networks. As an example, consider the

same setup with a pair of single-antenna PU and SU links over fading channels as in the previous subsection. Instead of applying the conventional AIPC: $\mathrm{E}(h_{sp}p_s) \leq \Gamma$, over the SU power allocation, we may apply the following PU ergodic capacity constraint [52]

$$\mathrm{E}\left(1+\frac{h_pQ}{1+h_{sp}p_s}\right) \geq \bar{C}_p \tag{23}$$

where $\bar{C}_p$ is a given threshold for the minimum PU ergodic capacity. Note that the new constraint in (23) is more directly related to the PU transmission than the conventional AIPC. However, it can be verified that the constraint in (23) is non-convex over p_s in general, thus resulting in more challenging SU power allocation problems than that with the convex AIPC. The optimal power allocation rules for the SU link subject to the AIPC vs. the newly introduced PU ergodic capacity constraint given by (23) are compared in [52], where it is shown that the new constraint achieves notable rate improvements for both the PU and SU links over the conventional AIPC.

V. Conclusions and Directions for Future Work

Dynamic resource allocation (DRA) has become an essential building block in CR networks to exploit various cognitions over both the primary and secondary networks for CR transmit optimization subject to certain required primary protection. In this article, we have presented an extensive list of new, challenging, and unique problems for designing the optimal DRA in CR networks, and demonstrated the key role of various convex optimization techniques in solving the associated design problems. In addition to those open issues as highlighted in our previous discussions, other promising areas of practical and theoretical interests are discussed as follows, which open an avenue for future work.

Robust Cognitive Beamforming: In our previous discussions on cognitive beamforming, we have observed that the knowledge of channels from each secondary transmitting terminal to all PUs is essential to the design optimization. However, since the primary and secondary networks usually belong to different operators, it is difficult for the PUs to feed back the required CSI to the CRs. As a result, the SU usually needs to rely on its own observations over the received signals from the primary terminals to extract the required CSI [54]. Nevertheless, the estimated CSI on the SU-to-PU channels may contain errors, which should be taken into account for the design of practical CR systems. This motivates a new and challenging research direction on robust designs for cognitive beamforming to cope with imperfect CSI [55], [56]. More investigations on the robust cognitive beamforming designs for more general CR networks and CSI uncertainty models are appealing.

Active Interference-Temperature Control: In this article, we have focused on the design of CR networks subject to the given interference-power constraints for protecting the PUs. We have also discussed some promising rules on how to optimally set the IT constraints in the CR network to achieve the best spectrum sharing throughput. These results lead to a new and universal design paradigm for interference management in CR or other related multiuser communication systems [57], [58], via appropriately setting the IT levels across the coexisting links. The active IT control approach to interference management for multiuser communication systems is relatively new, and more research endeavors are required along this direction.

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13-18, Aug. 1999.

[2] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Sig. Proces. Mag.*, vol. 24, no. 3, pp. 79-89, May 2007.

[3] A. Goldsmith, S. A. Jafar, I. Marić, and S. Srinivasay, "Breaking spectrum gridlock with cognitive radios: an information theoretic perspective," *Proc. IEEE*, vol. 97, no. 5, pp. 894-914, May, 2009.

[4] Z. Quan, S. Cui, and A. Sayed, "Optimal linear cooperation for spectrum sensing in cognitive radio networks," *IEEE J. Sel. Topics Sig. Proces.*, vol. 2, no. 1, pp. 28-40, Feb. 2008.

[5] Y.-C. Liang, Y. Zeng, E. C. Y. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1326-1337, Apr. 2008.

[6] B. H. Juang, Y. Li, and J. Ma, "Signal processing in cognitive radio," *Proc. IEEE*, vol. 97, no. 5, pp. 805-823, May 2009.

[7] Y. H. Zeng, Y.-C. Liang, A. T. Hoang, and R. Zhang, "A review on spectrum sensing for cognitive radio: challenges and solutions," *EURASIP J. Advances in Sig. Proces.*, Article ID 381465, 2010.

[8] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 388-404, Mar. 2000.

[9] N. Devroye, P. Mitran, and V. Tarokh, "Achievable rates in cognitive radio channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 1813-1827, May 2006.

[10] M. Gastpar, "On capacity under receive and spatial spectrum-sharing constraints," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 471-487, Feb. 2007.

[11] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1426-1438, Aug. 2006.

[12] E. Biglieri, J. Proakis, and S. Shamai (Shitz), "Fading channels: information-theoretic and communications aspects," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2619-2692, Oct. 1998.

[13] R. Zhang and Y.-C. Liang, "Exploiting multi-antennas for opportunistic spectrum sharing in cognitive radio networks," *IEEE J. S. Topics Sig. Proces.*, vol. 2, no. 1, pp. 88-102, Feb. 2008.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[15] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," available [online] at http://stanford.edu/∼boyd/cvx.

[16] R. G. Bland, D. Goldfarb, and M. J. Todd, "The ellipsoid method: a survey," *Operations Research*, vol. 29, no. 6, pp. 1039-1091, 1981.

[17] T. Cover and J. Thomas, *Elements of information theory*, New York: Wiley, 1991.

[18] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Trans. Sig. Proces.*, vol. 52, no. 2, pp. 461-471, Feb. 2004.

[19] D. Tse and S. Hanly,"Multi-access fading channels-Part I: polymatroid structure, optimal resource allocation and throughput capacities," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 2796-2815, Nov. 1998.

[20] L. Zhang, Y.-C. Liang, and Y. Xin, "Joint beamforming and power control for multiple access channels in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol.26, no.1, pp.38-51, Jan. 2008.

[21] K. Phan, S. Vorobyov, N. Sidiropoulos, and C. Tellambura, "Spectrum sharing in wireless networks via QoS-aware secondary multicast beamforming," *IEEE Trans. Sig. Proces.*, vol. 57, no. 6, pp. 2323-2335, June 2009.

[22] H. Weingarten, Y. Steinberg, and S. Shamai (Shitz), "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3936-3964, Sep. 2006.

[23] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, achievable rates, and sum-rate capacity of Gaussian MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2658-2668, Oct. 2003.

[24] W. Yu and T. Lan, "Transmitter optimization for the multi-antenna downlink with per-antenna power constraints," *IEEE Trans. Sig. Proces.*, vol. 55, no. 6, pp. 2646-2660, June 2007.

[25] L. Zhang, R. Zhang, Y.-C. Liang, Y. Xin, and H. V. Poor, "On the Gaussian MIMO BC-MAC duality with multiple transmit covariance constraints," submitted to *IEEE Trans. Inf. Theory*. Available [online] at arXiv:0809.4101.

[26] M. Bengtsson and B. Ottersten, "Optimal and suboptimal transmit beamforming," in *Handbook of Antennas in Wireless Communications*, CRC Press, 2001.
[27] A. Wiesel, Y. C. Eldar, and S. Shamai (Shitz), "Linear precoding via conic optimization for fixed MIMO receivers," *IEEE Trans. Sig. Process.*, vol. 54, no. 1, pp. 161-176, Jan. 2006.
[28] M. Schubert and H. Boche, "Solution of the multiuser downlink beamforming problem with individual SINR constraints," *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 18-28, Jan. 2004.
[29] T. S. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Trans. Inf. Theory*, vol. 27, no. 1, pp. 49-60, Jan. 1981.
[30] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom for the K user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425-3441, Aug. 2008.
[31] M. F. Demirkol and M. A. Ingram, "Power-controlled capacity for interfering MIMO links," in *Proc. IEEE Vehicular Tech. Conf. (VTC)*, vol. 1 pp. 187-191, 2001.
[32] S. Ye and R. S. Blum, "Optimized signaling for MIMO interference systems with feedback," *IEEE Trans. Sig. Process.*, vol. 51, pp. 2839-2848, Nov. 2003.
[33] W. Yu, G. Ginis, and J. Cioffi "Distributed multiuser power control for digital subscriber lines," *IEEE J. Sel. Areas Commun.*, vol. 20, no.5, pp. 1105-1115, June 2002.
[34] J. Huang, R. Berry, and M. L. Honig, "Distributed interference compensation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1074-1084, May 2006.
[35] Z.-Q. Luo and J.-S. Pang, "Analysis of iterative waterfilling algorithm for multiuser power control in digital subscriber line," *EURASIP J. Appl. Sig. Process.*, Article ID 24012, 2006.
[36] G. Scutari, D. P. Palomar, and S. Barbarossa, "Competitive design of multiuser MIMO systems based on game theory: a unified view," *IEEE J. Sel. Areas Commun.* vol. 25, no. 7, pp. 1089-1103, Sep. 2008.
[37] G. Scutari, D. P. Palomar, and S. Barbarossa, "Cognitive MIMO radio," *IEEE Sig. Proces. Mag.*, vol. 25, no. 6, Nov. 2008.
[38] S. J. Kim and G. B. Giannakis, "Optimal resource allocation for MIMO ad hoc cognitive radio networks," in *Proc. Annual Allerton Conf. Commun. Control Comput.*, pp. 39-45, Sep. 2008.
[39] B. Song, R. Cruz, and B. Rao, "Network duality for multiuser mimo beamforming networks and applications," *IEEE Trans. Commun.*, vol. 55, no. 3, pp. 618-630, Mar. 2007.
[40] M. Chiang, C. W. Tan, D. P. Palomar, D. Neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2640-2651, July 2007.
[41] S. Yiu, M. Vu, and V. Tarokh, "Interference reduction by beamforming in cognitive networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pp. 1-6, Dec. 2008.
[42] A. Tajer, N. Prasad, and X. Wang, "Beamforming and rate allocation in MISO cognitive radio networks," *IEEE Trans. Sig. Proces.*, vol. 58, no. 1, pp. 362-377, Jan. 2010.
[43] R. Zhang and J. Cioffi, "Iterative spectrum shaping with opportunistic multiuser detection," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2009.
[44] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439-1451, Aug. 2006.
[45] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Trans. Commun.*, vol. 54. no. 7. pp. 1310-1322, July 2006.
[46] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: complexity and duality," *IEEE J. S. Topics Sig. Proces.*, vol. 2, no. 1, pp. 57-73, Feb. 2008.
[47] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multi-user communications," in *Proc. IEEE Int. Conf. Comm. (ICC)*, pp. 331-335, 1995.
[48] R. Zhang, S. Cui, and Y.-C. Liang, "On ergodic sum capacity of fading cognitive multiple-access and broadcast channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5161-5178, Nov. 2009.
[49] R. Etkin, A. Parekh, and D. Tse, "Spectrum sharing for unlicensed bands," *IEEE J. Sel. Areas in Commun.*, vol. 25, no. 3, pp. 517- 528, Apr. 2007.
[50] S. Hayashi and Z.-Q. Luo, "Spectrum management for interference-limited multiuser communication systems," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1153-1175, Mar. 2009.
[51] R. Zhang, "On peak versus average interference power constraints for protecting primary users in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 2112-2120, Apr. 2009.
[52] R. Zhang, "Optimal power control over fading cognitive radio channels by exploiting primary user CSI," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Nov. 2008.
[53] X. Kang, R. Zhang, Y.-C. Liang, and H. K. Garg, "Optimal power allocation for cognitive radio under primary user outage capacity constraint," in *Proc. IEEE Int. Conf. Commun. (ICC)*, June 2009.
[54] R. Zhang, F. Gao, and Y.-C. Liang, "Cognitive beamforming made practical: effective interference channel and learning-throughput tradeoff," to appear in *IEEE Trans. Commun.*.
[55] L. Zhang, Y.-C. Liang, Y. Xin, and H. V. Poor, "Robust cognitive beamforming with partial channel state information," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4143-4153, Aug. 2009.
[56] G. Zheng, K.-K. Wong, and B. Ottersten, "Robust cognitive beamforming with bounded channel uncertainties," to appear in *IEEE Trans. Sig. Proces.*.
[57] L. Zhang, R. Zhang, Y.-C. Liang, Y. Xin, and S. Cui, "On the relationship between the multi-antenna secrecy communications and cognitive radio communications," in *Proc. Annual Allerton Conf. Commun. Control and Comput.*, 2009.
[58] R. Zhang and S. Cui, "Cooperative interference management in multi-cell downlink beamforming," submitted to *IEEE Trans. Sig. Proces.*. Available [online] at arXiv: 0910.2771.

AUTHORS:

Rui Zhang (rzhang@i2r.a-star.edu.sg) received the B.Eng. and M.Eng. degrees in electrical and computer engineering from National University of Singapore in 2000 and 2001, respectively, and the Ph.D. degree in electrical engineering from Stanford University, California, USA, in 2007. He is now a Senior Research Fellow with the Institute for Infocomm Research (I2R), Singapore. He also holds an Assistant Professorship position in electrical and computer engineering with National University of Singapore. He has authored/co-authored more than 100 refereed international journal and conference papers. His current research interests include cognitive radios, cooperative communications, and multiuser MIMO systems.

Ying-Chang Liang (ycliang@i2r.a-star.edu.sg) is presently a Senior Scientist in the Institute for Infocomm Research (I2R), Singapore. He also holds adjunct associate professorship positions in Nanyang Technological University (NTU) and National University of Singapore (NUS). He has been teaching graduate courses in NUS since 2004 and was a visiting scholar with the Department of Electrical Engineering, Stanford University, CA, USA, from Dec 2002 to Dec 2003. His research interest includes cognitive radio, dynamic spectrum access, reconfigurable signal processing for broadband communications, space-time wireless communications, wireless networking, information theory and statistical signal processing.

Shuguang Cui (cui@tamu.edu) received Ph.D in Electrical Engineering from Stanford University in 2005, M.Eng in Electrical Engineering from McMaster University, Canada, in 2000, and B.Eng. in Radio Engineering with the highest distinction from Beijing University of Posts and Telecommunications, China, in 1997. He is now working as an assistant professor in Electrical and Computer Engineering at the Texas A&M University, College Station, TX. His current research interests include cross-layer optimization for resource-constrained networks and network information theory. He has been serving as the associate editors for the IEEE Communication Letters and IEEE Transactions on Vehicular Technology, and the elected member for IEEE Signal Processing Society SPCOM Technical Committee.

IP Router Architectures: An Overview

by

James Aweya

{Email: aweyaj@nortelnetworks.com; Tel: 1-613-763-6491; Fax: 1-613-763-5692}

Nortel Networks

Ottawa, Canada, K1Y 4H7

Abstract

In the emerging environment of high performance IP networks, it is expected that local and campus area backbones, enterprise networks, and Internet Service Providers (ISPs) will use multigigabit and terabit networking technologies where IP routers will be used not only to interconnect backbone segments but also to act as points of attachments to high performance wide area links. Special attention must be given to new powerful architectures for routers in order to play that demanding role. In this paper, we identify important trends in router design and outline some design issues facing the next generation of routers. It is also observed that the achievement of high throughput IP routers is possible if the critical tasks are identified and special purpose modules are properly tailored to perform them.

1. Introduction

The popularity of the Internet has caused the traffic on the Internet to grow drastically every year for the last several years. It has also spurred the emergence of many Internet Service Providers (ISPs). To sustain growth, ISPs need to provide new differentiated services, e.g., tiered service, support for multimedia applications, etc. The routers in the ISPs' networks play a critical role in providing these services. Internet Protocol (IP) traffic on private enterprise networks has also been growing rapidly for some time. These

networks face significant bandwidth challenges as new application types, especially desktop applications uniting voice, video, and data traffic need to be delivered on the network infrastructure. This growth in IP traffic is beginning to stress the traditional processor-based design of current-day routers and as a result has created new challenges for router design.

Routers have traditionally been implemented purely in software. Because of the software implementation, the performance of a router was limited by the performance of the processor executing the protocol code. To achieve wire-speed routing, high-performance processors together with large memories were required. This translated into higher cost. Thus, while software-based wire-speed routing was possible at low-speeds, for example, with 10 megabits per second (Mbps) ports, or with a relatively smaller number of 100 Mbps ports, the processing costs and architectural implications make it difficult to achieve wire-speed routing at higher speeds using software-based processing.

Fortunately, many changes in technology (both networking and silicon) have changed the landscape for implementing high-speed routers. Silicon capability has improved to the point where highly complex systems can be built on a single integrated circuit (IC). The use of 0.35 μm and smaller silicon geometries enables application specific integrated circuit (ASIC) implementations of millions gate-equivalents. Embedded memory (SRAM, DRAM) and microprocessors are available in addition to high-density logic. This makes it possible to build single-chip, low-cost routing solutions that incorporate both hardware and software as needed for best overall performance.

In this paper we investigate the evolution of IP router designs and highlight the major performance issues affecting IP routers. The need to build fast IP routers is being addressed in a variety of ways. We discuss these in various sections of the paper. We discuss in detail the various router mechanisms needed for high-speed operation. In particular, we examine the architectural constraints imposed by the various router design alternatives. The scope of the discussion presented here does not cover more recent *label switching* routing techniques such as IP Switching [1], the Cell Switching Router (CSR) architecture [2], Tag Switching [3], and Multiprotocol Label Switching (MPLS), which is

a standardization effort underway at the Internet Engineering Task Force (IETF). The discussion is limited to routing techniques as described in RFC 1812 [4].

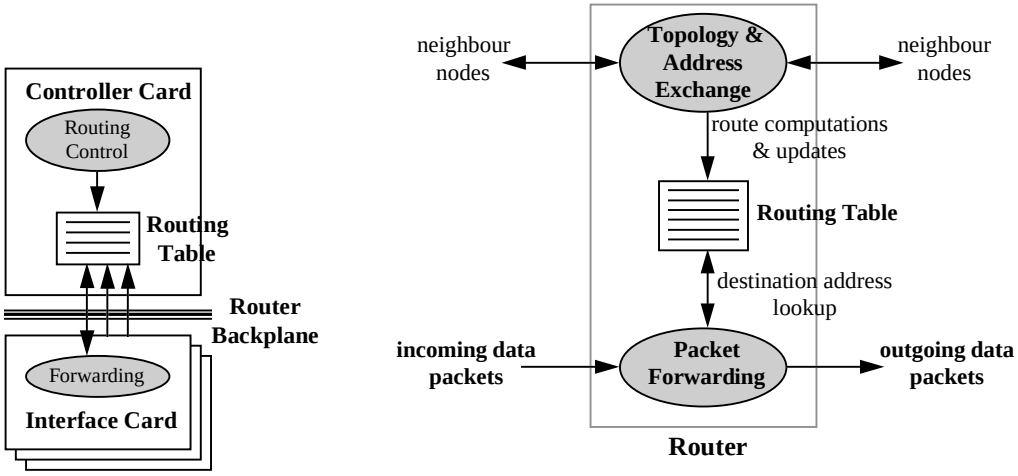
In Section 2, we briefly review the basic functionalities in IP routers. The IETF's *Requirements for IP Version 4 Routers* [4] describes in great detail the set of protocol standards that Internet Protocol version 4 (IPv4) routers need to conform to. Section 3 presents the design issues and trends that arise in IP routers. The most common switch fabric technologies in use today are buses, shared memories, and crossbars. Section 4 presents an overview of these switch fabric technologies. The concluding remarks are given in Section 5.

2. Basic IP Router Functionalities

Generally, routers consist of the following basic components: several network interfaces to the attached networks, processing module(s), buffering module(s), and an internal interconnection unit (or switch fabric). Typically, packets are received at an inbound network interface, processed by the processing module and, possibly, stored in the buffering module. Then, they are forwarded through the internal interconnection unit to the outbound interface that transmits them on the next hop on the journey to their final destination. The aggregate packet rate of all attached network interfaces needs to be processed, buffered and relayed. Therefore, the processing and memory modules may be replicated either fully or partially on the network interfaces to allow for concurrent operations.

A generic architecture of an IP router is given in Figure 1. Figure 1a shows the basic architecture of a typical router: the controller card (which holds the CPU), the router backplane, and interface cards. The CPU in the router typically performs such functions as path computations, routing table maintenance, and reachability propagation. It runs which ever routing protocols is needed in the router. The interface cards consists of adapters that perform inbound and outbound packet forwarding (and may even cache routing table entries or have extensive packet processing capabilities). The router backplane is responsible for transferring packets between the cards. The basic functionalities in an IP

router can be categorized as: route processing, packet forwarding, and router special services. The two key functionalities are route processing (i.e., path computation, routing table maintenance, and reachability propagation) and packet forwarding (see Figure 1b). We discuss the three functionalities in more detail below.



a). Basic architecture. b). Routing components.

Figure 1. Generic architecture of a router.

- *Route Processing*:

  This includes routing table construction and maintenance using routing protocols (such as RIP or OSPF) to learn about and create a view of the network's topology [5][6][7]. Updates to the routing table can also be done through management action where routes are added and deleted manually.

- *Packet Forwarding*:

  Typically, IP packet forwarding requires the following:

  - **IP Packet Validation:** The router must check that the received packet is properly formed for the protocol before it proceeds with protocol processing. This involves checking the version number, checking the header length field (also needed to determine whether any options are present in the packet), and calculating the header checksum.

- **Destination IP Address Parsing and Table Lookup:** The router performs a table lookup to determine the output port onto which to direct the packet and the next hop to which to send the packet along this route. This is based on the destination IP address in the received packet and the subnet mask(s) of the associated table entries. The result of this lookup could imply:

    - A local delivery (that is, the destination address is one of the router's local addresses and the packet is locally delivered).
    - A unicast delivery to a single output port, either to a next-hop router or to the ultimate destination station (in the case of a direct connection to the destination network).
    - A multicast delivery to a set of output ports that depends on the router's knowledge of multicast group membership.

  The router must also determine the mapping of the destination network address to the data link address for the output port (address resolution or ARP). This can be done either as a separate step or integrated as part of the routing lookup.

- **Packet Lifetime Control:** The router adjusts the time-to-live (TTL) field in the packet used to prevent packets from circulating endlessly throughout the internetwork. A packet being delivered to a local address within the router is acceptable if it has any positive value of TTL. A packet being routed to output ports has its TTL value decremented as appropriate and then is rechecked to determine if it has any life before it is actually forwarded. A packet whose lifetime is exceeded is discarded by the router (and may cause an error message to be generated to the original sender).

- **Checksum Calculation:** The IP header checksum must be recalculated due to the change in the TTL field. Fortunately, the checksum algorithm employed (a 16-bit one's complement addition of the header fields) is both commutative and associative, thereby allowing simple, differential recomputation. RFC 1071 [8]

contains implementation techniques for computing the IP checksum. Since a router often changes only the TTL field (decrementing it by 1), a router can incrementally update the checksum when it forwards a received packet, instead of calculating the checksum over the entire IP header again. RFC 1141 [9] describes an efficient way to do this.

IP packets might also have to be fragmented to fit within the Maximum Transmission Unit (MTU) specified for the outgoing network interface. Fragmentation, however, can affect performance adversely [10] but now that IP MTU discovery is prevalent [11], fragmentation should be rare.

- *Special Services*:

  Anything beyond core routing functions falls into this category: packet translation, encapsulation, traffic prioritization, authentication, and access services such as packet filtering for security/firewall purposes. In addition, routers possess network management components (e.g., SNMP, Management Information Base (MIB), etc.).

2.1 Route Table Lookup

For a long time, the major performance bottleneck in IP routers has been the time it takes to look up a route in the routing table. The problem is defined as that of searching through a database (routing table) of destination prefixes and locating the longest prefix that matches the destination address of a given packet. Longest prefix matching was introduced as a consequence of the requirement for increasing the number of networks addressed through Classless Inter-Domain Routing (CIDR) [12]. The CIDR technique is used to summarize a block of class C addresses into a single routing table entry. This consolidation results in a reduction in the number of separate routing table entries.

Given a packet, the router performs a routing table lookup, using the packet's IP destination address as key. This lookup returns the best-matching routing table entry, which tells the router which interface to forward the packet out of and the IP address of the packet's next hop

The first approaches for longest prefix matching used radix trees or modified Patricia trees [13][14] combined with hash tables, (Patricia stands for "Practical Algorithm to Retrieve Information Coded in Alphanumeric"). These trees are binary trees, whereby the tree traversal depends on a sequence of single-bit comparisons in the key, the destination IP address. These lookup algorithms have complexity proportional to the number of address bits which, for IPv4 is only 32. In the worst case it takes time proportional to the length of the destination address to find the longest prefix match. Worse, the commonly used Patricia algorithm may need to backtrack to find the longest match, leading to poor worst-case performance. The performance of Patricia is somewhat data dependent. With a particularly unfortunate collection of prefixes in the routing table, the lookup of certain addresses can take as many as 32 bit comparisons, one for each bit of the destination IP address.

Early implementations of routers, however, could not afford such expensive computations. Thus, one way to speed up the routing table lookup is to try to avoid it entirely. The routing table lookup provides the next hop for a given IP destination. Some routers cache this IP destination-to-next-hop association in a separate database that is consulted (as the front end to the routing table) before the routing table lookup. Finding a particular destination in this database is easier because an exact match is done instead of the more expensive best-match operation of the routing table. So, most routers relied on route caches [15][16]. The route caching techniques rely on there being enough locality in the traffic so that the cache hit rate is sufficiently high and the cost of a routing lookup is amortized over several packets.

This front-end database might be organized as a *hash table* [17]. After the router has forwarded several packets, if the router sees any of these destinations again (a *cache hit*), their lookups will be very quick. Packets to new destinations will be slower, however, because the cost of a failed hash lookup will have to be added to the normal routing table lookup. Front-end caches to the routing table can work well at the edge of the Internet or within organizations. However, cache schemes do not seem to work well in the Internet's core. The large number of packet destinations seen by the core routers can cause caches to

overflow or for their lookup to become slower than the routing table lookup itself. Cache schemes are not really effective when the hash bucket size (the number of destinations that hash to the same value) starts getting large. Also, the frequent routing changes seen in the core routers can force them to invalidate their caches frequently, leading to a small number of cache hits.

Typically, two types of packets arrive at a router: packets to be forwarded to another network or packets destined to the router itself. Whether a packet to a router causes a routing table reference depends on the router implementation. Some implementations may speed up routing table lookups. One possibility is for the router to explicitly check each incoming packet against a table of all of the router's addresses to see if there is a match. This explicit check means that the routing table is never consulted about packets destined to the router. Another possibility is to use the routing table for all packets. Before the packet is sent, the router checks if the packet is to its own address on the appropriate network. If the packet is for the router, then it is never transmitted. The explicit check after the routing table lookup requires checking a smaller number of router addresses at the increased cost of a routing table lookup. New routing table lookup algorithms are still being developed in attempts to build even faster routers. Recent examples are found in the references [18][19][20][21][22][23].

The basic idea of one of the recent algorithms [18] is to create a small and compressed data structure that represents large lookup tables using a small amount of memory. The technique exploits the sparseness of actual entries in the space of all possible routing table entries. This results in a lower number of memory accesses (to a fast memory) and hence faster lookups. The proposal reduces the routing table to very efficient representation of a binary tree such that the majority of the table can reside in the primary cache of the processor, allowing route lookups at gigabit speeds. In addition, the algorithm does not have to calculate expensive perfect hash functions, although updates to the routing table are still not easy. Reference [19] proposes another approach for implementing the compression and minimizing the complexity of updates.

The recent work of Waldvogel et al. [20] presents an alternative approach which reduces the number of memory references rather than compact the routing table. The main idea is to first create a perfect hash table of prefixes for each prefix length. A binary search among all prefix lengths, using the hash tables for searches amongst prefixes of a particular length, can find the longest prefix match for an N bit address in $O(\log N)$ steps. Although the algorithm has very fast execution times, calculating perfect hashes can be slow and can slow down updates.

Hardware based techniques for route lookup are also actively being investigated both in research and commercial designs (e.g., [24][25][26][27]). Other designs of forwarding engine have concentrated on IP packet header processing in hardware, to remove the dependence upon caching, and to avoid the cost of high-speed processor. Designs based upon content-addressable memory have been investigated [24], but such memory is too far expensive to be applied to a large routing table. Hardware route lookup and forwarding is also under active investigation in both research and commercial designs [25][26]. The argument for a software-based implementation stresses flexibility. Hardware implementations can generally achieve a higher performance at lower cost but are less flexible.

3. IP Router Architectures

In the next sections, we examine important trends in IP router design and outline some design issues facing the next generation of routers.

3.1 Bus-based Router Architectures with Single Processor

The first generation of IP router were based on software implementations on a single general-purpose central processing unit (CPU). These routers consist of a general-purpose processor and multiple interface cards interconnected through a shared bus as depicted in Figure 2.

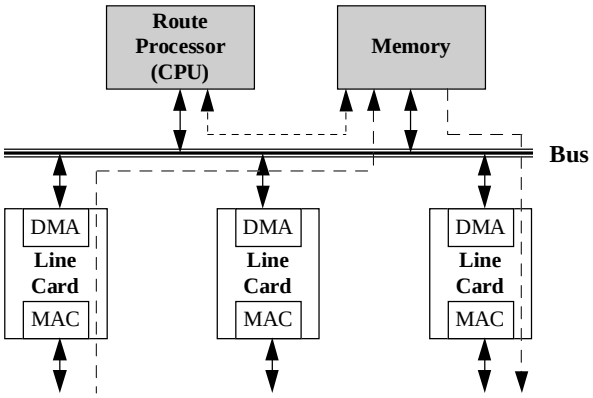


Figure 2. Traditional bus-based router architecture.

Packets arriving at the interfaces are forwarded to the CPU which determines the next hop address and sends them back to the appropriate outgoing interface(s). Data is usually buffered in a centralized data memory [28][29], which leads to the disadvantage of having the data cross the bus twice, making it the major system bottleneck. Packet processing and node management software (including routing protocol operation, routing table maintenance, routing table lookups, and other control and management protocols such as ICMP, SNMP) are also implemented on the central processor. Unfortunately, this simple architecture yields low performance for the following reasons:

- The central processor has to process all packets flowing through the router (as well as those destined to it). This represents a serious processing bottleneck.

- Some major packet processing tasks in a router involve memory intensive operations (e.g., table lookups) which limits the effectiveness of processor power upgrades in boosting the router packet processing throughput. Routing table lookups and data movements are the major consumer of processing cycles. The processing time of these tasks does not decrease linearly if faster processors are used. This is because of the sometimes dominating effect of memory access rate.

- Moving data from one interface to the other (either through main memory or not) is a time consuming operation that often exceeds the packet header processing time. In

many cases, the computer input/output (I/O) bus quickly becomes a severe limiting factor to overall router throughput.

Since routing table lookup is a time-consuming process of packet forwarding, some traditional software-based routers cache the IP destination-to-next-hop association in a separate database that is consulted as the front end to the routing table before the routing table lookup [15]. Still, the performance of the traditional bus-based router depends heavily on the throughput of the shared bus and on the forwarding speed of the central processor. This architecture cannot scale to meet the increasing throughput requirements of multigigabit network interface cards.

3.2 Bus-based Router Architectures with Multiple Processors

3.2.1 Architectures with Route Caching

For the second generation IP routers, improvement in the shared-bus router architecture was introduced by distributing the packet forwarding operations. Distributing fast processors and route caches, in addition to receive and transmit buffers, over the network interface cards reduces the load on the system bus. Packets are therefore transmitted only once over the shared bus. This reduces the number of bus copies and speeds up packet forwarding by using a route cache of frequently seen addresses in the network interface as shown in Figure 3. This architecture allows the network interface cards to process packets locally some of the time.

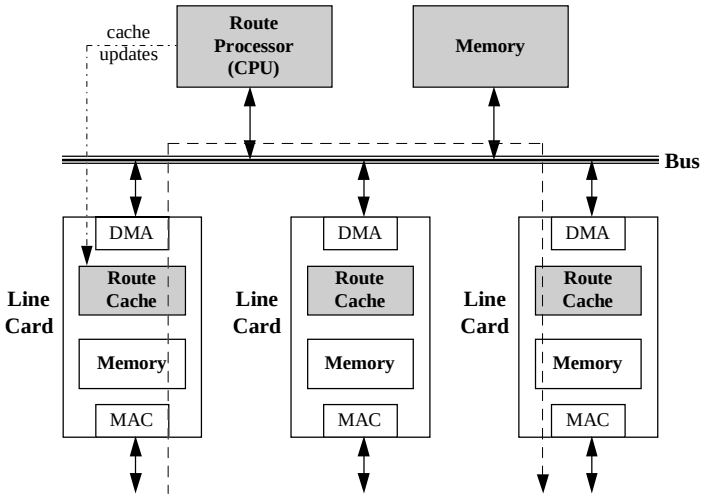


Figure 3. Reducing the number of bus copies using a route cache in the network interface.

In this architecture, a router keeps a central master routing table and the satellite processors in the network interfaces each keep only a modest cache of recently used routes. If a route is not in a network interface processor's cache, it would request the relevant route from the central table. The route cache entries are traffic-driven in that the first packet to a new destination is routed by the main CPU (or route processor) via the central routing table information and as part of that forwarding operation, a route cache entry for that destination is then added in the network interface. This allows subsequent packet flows to the same destination network to be switched based on an efficient route cache match. These entries are periodically aged out to keep the route cache current and can be immediately invalidated if the network topology changes. At high-speeds, the central routing table can easily become a bottleneck because the cost of retrieving a route from the central table is many times more expensive than actually processing the packet local in the network interface.

A major limitation of this architecture is that it has a traffic dependent throughput and also the shared bus is still a bottleneck. The performance of this architecture can be improved by enhancing each of the distributed network interface cards with larger memories and complete forwarding tables. The decreasing cost of high bandwidth memories makes this possible. However, the shared bus and the general purpose CPU in the slow data path can neither scale to high capacity links nor provide traffic pattern independent throughput.

3.2.2 Architectures with Multiple Parallel Forwarding Engines

Another bus-based multiple processor router architecture is described in [30]. Multiple forwarding engines are connected in parallel to achieve high packet processing rates as shown in Figure 4. The network interface modules transmit and receive data from the links at the required rates. As a packet comes in, the IP header is stripped by the control circuitry, augmented with an identifying tag, and sent to a forwarding engine for validation and routing. While the forwarding engine is performing the routing function, the remainder of the packet is deposited in an input buffer in parallel. The forwarding engine determines which outgoing link the packet should be transmitted on, and sends the updated header fields to the appropriate destination interface module along with the tag information. The packet is then moved from the buffer in the source interface module to a buffer in the destination interface module and eventually transmitted on the outgoing link.

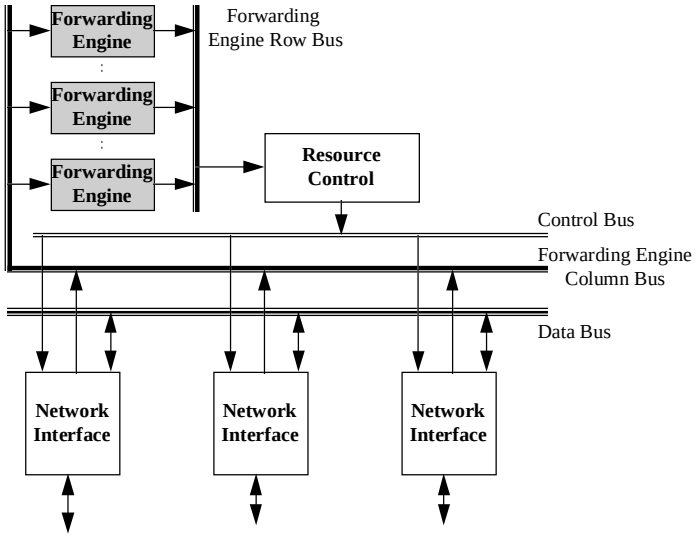


Figure 4. Bus-based router architecture with multiple parallel forwarding engines.

The forwarding engines can each work on different headers in parallel. The circuitry in the interface modules peels the header off of each packet and assigns the headers to the forwarding engines in a round-robin fashion. Since in some (real time) applications packet order maintenance is an issue, the output control circuitry also goes round-robin, guaranteeing that packets will then be sent out in the same order as they were received. Better load-balancing may be achieved by having a more intelligent input interface which

assigns each header to the lightest loaded forwarding engine [30]. The output control circuitry would then have to select the next forwarding engine to obtain a processed header from by following the demultiplexing order followed at the input, so that order preservation of packets is ensured. The forwarding engine returns a new header (or multiple headers, if the packet is to be fragmented), along with routing information (i.e., the immediate destination of the packet). The route processor (controller) runs the routing protocols and creates a forwarding table that is used by the forwarding engines.

The choice of this architecture was premised on the observation that it is highly unlikely that all interfaces will be bottlenecked at the same time. Hence sharing of the forwarding engines can increase the port density of the router. The forwarding engines are only responsible for resolving next-hop addresses. Forwarding only IP headers to the forwarding engines eliminates an unnecessary packet payload transfer over the bus. Packet payloads are always directly transferred between the interface modules and they never go to either the forwarding engines or the route processor unless they are specifically destined to them.

3.3 Switch-based Router Architectures with Multiple Processors

To alleviate the bottlenecks of the second generation of IP routers, the third generation of routers were designed with the shared bus replaced by a switch fabric. This provides sufficient bandwidth for transmitting packets between interface cards and allows throughput to be increased by several orders of magnitude. With the interconnection unit between interface cards not the bottleneck, the new bottleneck is packet processing.

The multigigabit router (MGR) is an example of this architecture [31]. The design has dedicated IP packet forwarding engines with route caches in them. The MGR consists of multiple line cards (each supporting one or more network interfaces) and forwarding engine cards, all connected to a high-speed (crossbar) switch as shown in Figure 5. The design places forwarding engines on boards distinct from line cards. When a packet arrives at a line card, its header is removed and passed through the switch to a forwarding engine. The remainder of the packet remains on the inbound line card. The forwarding engine

reads the header to determine how to forward the packet and then updates the header and sends the updated header and its forwarding instructions back to the inbound line card. The inbound line card integrates the new header with the rest of the packet and sends the entire packet to the outbound line card for transmission. The MGR, like most routers, also has a control (and route) processor that provides basic management functions such as generation of routing tables for the forwarding engines and link (up/down) management. Each forwarding engine has a set of the forwarding tables (which are a summary of the routing table data).

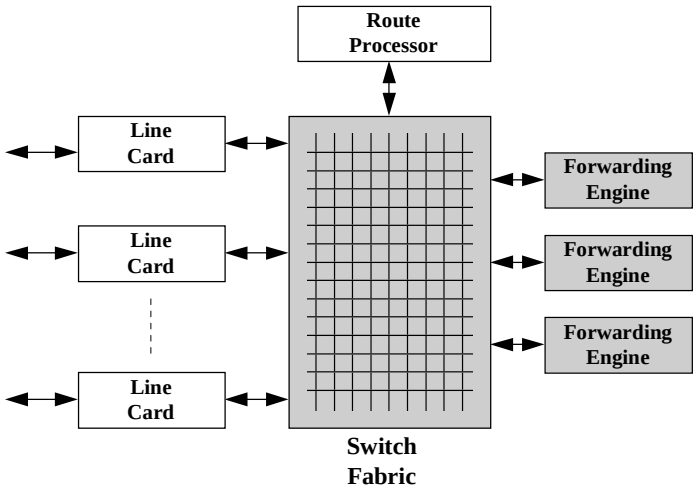


Figure 5. Switch-based router architecture with multiple forwarding engines.

In the MGR, once headers reach the forwarding engine, they are placed in a request first-in first-out (FIFO) queue for processing by the forwarding processor. The forwarding process can be roughly described by the following three stages [31].

1. The first stage includes the following which are done in parallel:

    - The forwarding engine does basic error checking to confirm that the header is indeed from an IPv4 datagram;
    - confirms that the packet and header lengths are reasonable;
    - confirms that the IPv4 header has no options;

- computes the hash offset into the route cache and loads the route; and
- starts loading the next header.

2. In the second stage, the forwarding engine checks to see if the cached route matches the destination of the datagram (a cache hit). If not, the forwarding engine carries out an extended lookup of the forwarding table associated with it. In this case, the processor searches the routing table for the correct route, and generates a version of the route for the route cache. Since the forwarding table contains prefix routes and the route cache is a cache of routes for particular destination, the processor has to convert the forwarding table entry into an appropriate destination-specific cache entry. Then, the forwarding engine checks the IP time-to-live (TTL) field and computes the updated TTL and IP checksum, and determines if the datagram is for the router itself.
3. In the third stage the updated TTL and checksum are put in the IP header. The necessary routing information is extracted from the forwarding table entry and the updated IP header is written out along with link-layer information from the forwarding table.

3.4 Limitation of IP Packet Forwarding based on Route Caching

Regardless of the type of interconnection unit used (bus, shared memory, crossbar, etc.), a route cache can be used in conjunction with a (centralized) processing unit for IP packet forwarding [15]. In this section, we examine the limitations of route caching techniques.

The route cache model creates the potential for cache misses which occur with “demand-caching” schemes as described above. That is, if a route is not found in the forwarding cache, the first packet(s) then looks to the routing table maintained by the CPU to determine the outbound interface and then a cache entry is added for that destination. This means when addresses are not found in the cache, the packet forwarding defaults to a classical software-based route lookup (sometimes described as a “slow-path”). Since the cache information is derived from the routing table, routing changes cause existing cache entries to be invalidated and reestablished to reflect any topology changes. In networking

environments which frequently experience significant routing activity (such as in the Internet) this can cause traffic to be forwarded via the main CPU (the slow path), as opposed to via the route cache (the fast path).

In enterprise backbones or public networks, the combination of highly random traffic patterns and frequent topology changes tends to eliminate any benefits from the route cache, and performance is bounded by the speed of the software slow path which can be many orders of magnitude lower than the caching fast path.

This demand-caching scheme, maintaining a very fast access subnet of the routing topology information, is optimized for scenarios whereby the majority of the traffic flows are associated with a subnet of destinations. However, given that traffic profiles at the core of the Internet (and potentially within some large enterprise networks) do not follow closely this model, a new forwarding paradigm is required that would eliminate the increasing cache maintenance resulting from growing numbers of topologically dispersed destinations and dynamic network changes.

The performance of a product using the route cache technique is influenced by the following factors:

- how big the cache is,
- how the cache is maintained (the three most popular cache maintenance strategies are random replacement, first-in-first-out (FIFO), and least recent use (LRU)), and
- what the performance of the slow path is, since at least some percentage of the traffic will take the slow path in any application.

The main argument in favor of cache-based schemes is that a cache hit is at least less expensive than a full route lookup (so a cache is valuable provided it achieves a modest hit rate). Even with an increasing number of flows, it appears that packet bursts and temporal correlation in the packet arrivals will continue to ensure that there is a strong chance that two datagrams arriving close together will be headed for the same destination.

In current backbone routers, the number of flows that are active at a given interface can be extremely high. Studies have shown that an OC-3 interface might have an average of 256,000 flows active concurrently [32]. It is observed in [33] that, for this many flows, use of hardware caches is extremely difficult, especially if we consider the fact that a fully-associative hardware cache is required. So caches of such size are most likely to be implemented as hash tables since only hash tables can be scaled to these sizes. However, the $O(1)$ lookup time of a hash table is an average case result, and the worst-case performance of a hash table can be poor since multiple headers might hash into the same location. Due to the large number of flows that are simultaneously active in a router and due to the fact that hash tables generally cannot guarantee good hashing under all arrival patterns, the performance of cache based schemes is heavily traffic dependent. If a large number of new flows arrive at the same time, the slow path of the router can be overloaded, and it is possible that packet loss can occur due to the (slow path) processing overload and not due to output link congestion.

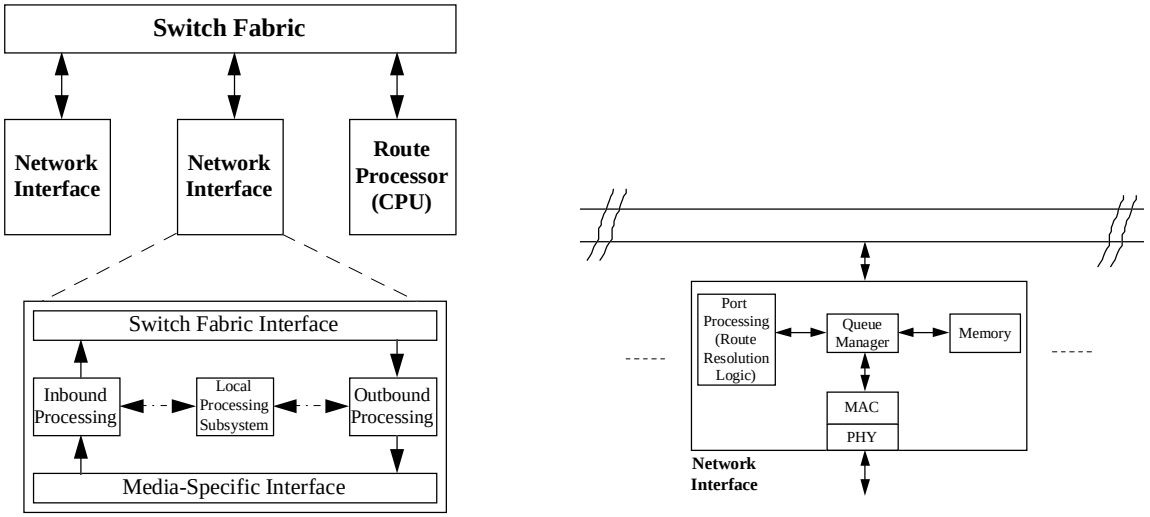
Some architectures have been proposed that avoid the potential overload of continuous cache churn (which results in a performance bottleneck) by instead using a forwarding database in each network interface which mirrors the entire content of the IP routing table maintained by the CPU (route processor), i.e., there is a one-to-one correspondence between the forwarding database entries and routing table prefixes; therefore no need to maintain a route cache [34][35][36]. By eliminating the route cache, the architecture fully eliminates the slow path. This offers significant benefits in terms of performance, scalability, network resilience and functionality, particularly in large complex networks with dynamic flows. These architectures can best accommodate the changing network dynamics and traffic characteristics resulting from increasing numbers of short flows typically associated with Web-based applications and interactive type sessions.

3.5 Switch-based Router Architectures with Fully Distributed Processors

From the discussion in the preceeding sections, we find that the three main bottlenecks in a router are: processing power, memory bandwidth, and internal bus bandwidth. These three bottlenecks can be avoided by using a distributed switch based architecture with

properly designed network interfaces [34][35][36]. Since routers are mostly dedicated systems not running any specific application tasks, off-loading processing to the network interfaces reflects a proper approach to increase the overall router performance. A successful step towards building high performance routers is to add some processing power to each network interface in order to reduce the processing and memory bottlenecks. General-purpose processors and/or dedicated VLSI components can be applied. The third bottleneck (internal bus bandwidth) can be solved by using special mechanisms where the internal bus is in effect a switch (e.g., shared memory, crossbar, etc.) thus allowing simultaneous packet transfers between different pairs of network interfaces. This arrangement must also allow for efficient multicast capabilities.

We investigate in this section, decentralized router architectures where each network interface is equipped with appropriate processing power and buffer space. A generic modular switch-based router architecture is shown in Figure 6.



a). Functional diagram [34][35][36].

b). Generic architecture.

Figure 6. A generic switch-based distributed router architecture.

Each network interface provides the processing power and the buffer space needed for packet processing tasks related to all the packets flowing through it. Functional

components (inbound, outbound, and local processing elements) process the inbound, outbound traffic and time-critical port processing tasks. They perform the processing of all protocol functions (in addition to quality of service (QoS) processing functions) that lie in the critical path of data flow. In order to provide QoS guarantees, a port may need to classify packets into predefined service classes. Depending on router implementation, a port may also need to run data-link level protocols or network-level protocols. The exact features of the processing components depend on the functional partitioning and implementation details. Concurrent operation among these components can be provided. The network interfaces are interconnected via a high performance switch that enables them to exchange data and control messages. In addition, a CPU is used to perform some centralized tasks. As a result, the overall processing and buffering capacity is distributed over the available interfaces and the CPU.

The Media-Specific Interface (MSI) performs all the functions of the physical layer and the Media Access Control (MAC) sublayer (in the case of the IEEE 802 protocol model). The Switch Fabric Interface (SFI) is responsible for preparing the packet for its journey across the switch fabric. The SFI may prepend an internal routing tag containing port of exit, the QoS priority, and drop priority, onto the packet.

To analyze the processing capabilities and to determine potential performance bottlenecks, the functions and components of a router and, especially, of all its processing subsystems have to be identified. Therefore, all protocols related to the task of a router need to be considered. In an IP router, the IP protocol itself as well as additional protocols, such as ICMP, ARP, RARP, BGP, etc. are required.

First, a distinction can be made between the processing tasks directly related to packets being forwarded through the router and those related to packets destined to the router, such as maintenance, management or error protocol data. Best performance can be achieved when packets are handled by multiple heterogeneous processing elements, where each element specializes in a specific operation. In such a configuration, special purpose modules perform the time critical tasks in order to achieve high throughput and low latency. Time critical tasks are the ones related to the regular data flow. The non-time

critical tasks are performed in general purpose processors (CPU). A number of commercial routers follow this design approach (e.g., [33][37][38][39][40][41][42][43][44][45[46]).

3.5.1 Critical Data Path Processing (Fast Path)

The *time critical processing tasks* forms the *critical path* (sometimes called the *fast path*) through a router and need to be highly optimized in order to achieve multigigabit rates. These processing tasks comprise all protocols involved in the critical path (LLC, SNAP, and IP) as well as ARP which can be processed in the network interface because it needs direct access to the network, even though it is not time critical. The time critical tasks mainly consist of header checking, and forwarding (and may include segmentation) functions. These protocols directly affect the performance of an IP router in terms of the number of packets that can be processed per second.

The router architecture should be optimized for those fast path functions that must be performed in real time. Most high-speed routers implement this fast path in hardware. Generally, the fast path of IP routing requires the following functions: IP packet validation, destination address parsing and table lookup, packet lifetime control (TTL update), and checksum calculation. The fast path may also be responsible for making packet classifications for QoS control and access filtering. Flows can be identified based on source IP address, destination IP address, TCP/UDP port numbers as well as IP Type of Service (TOS) field. Classification can even be based on higher layer packet attributes.

3.5.2 Non-critical Data Path Processing (Slow Path)

Packets destined to a router, such as maintenance, management or error protocol data are usually not time critical. However, they have to be integrated in an efficient way that does not interfere with the packet processing and, thus, does not slow down the time-critical path. Typical examples of these *non-time critical processing* tasks are error protocols (e.g., ICMP), routing protocols (e.g., RIP, OSPF, BGP), and network management protocols (e.g., SNMP). These processing tasks need to be centralized in a router node and typically reside above the network or transport protocols.

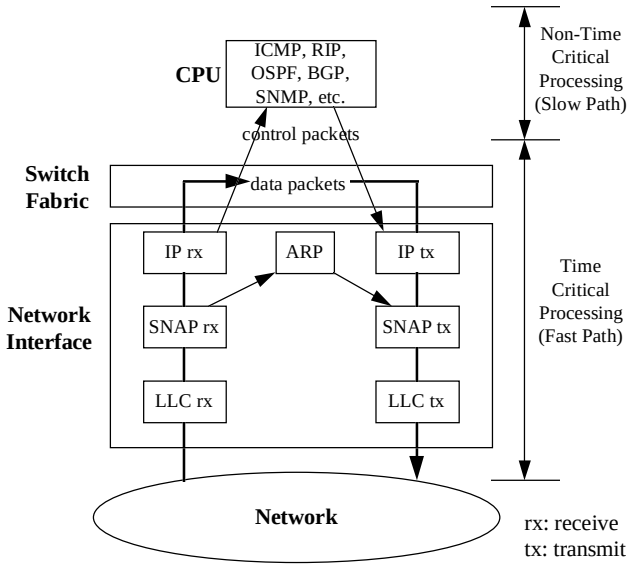


Figure 7. Example IEEE 802 protocol entities in an IP router [Adapted from 34].

As shown in Figure 7, only protocols in the forwarding path of a packet through the IP router are implemented on the network interface itself. Other protocols such as routing and network management protocols are implemented on the CPU. This way, the CPU does not adversely affect performance because it is located out of the data path, where it maintains route tables and determines the policies and resources used by the network interfaces. As an example, the CPU subsystem can be attached to the switch fabric in the same way as a regular network interface. In this case, the CPU subsystem is viewed by the switch fabric as a regular network interface. It has, however, a completely different internal architecture and function. This subsystem receives all non-critical protocol data units and requests to process certain related protocol entities (ICMP, SNMP, TCP, UDP, and the routing protocol entities RIP, OSPF, BGP, etc.). Any protocol data unit that needs to be sent on any network by these protocol entities is sent to the proper network interface, as if it was just another IP datagram relayed from another network.

The CPU subsystem can communicate with all other network interfaces through the exchange of coded messages across the switch fabric (or on a separate control bus [30][33]). IP datagrams generated by the CPU protocol entities are also coded in the same format. They carry the IP address of the next hop. For that, the CPU needs to access its

individual routing table. This routing table can be the master table of the entire router. All other routing tables in the network interfaces will be exact replicas (or summaries in compressed table format) of the master table. Routing table updates in the network interfaces can then be done by broadcasting (if the switch fabric is capable of that) or any other suitable data push technique. Any update information (e.g., QoS policies, access control policies, packet drop policies, etc.) originated in the CPU has to be broadcast to all network interfaces. Such special data segments are received by the network interfaces which takes care of the actual write operation in their forwarding tables. Updates to the routing table in the CPU are done either by the various routing protocol entities or by management action. This centralization is reasonable since routing changes are assumed to happen infrequently and not particularly time critical. The CPU can also be configured to handle any packet whose destination address cannot be found in the forwarding table in the network interface card.

3.5.3 Fast Path or Slow Path?

It is not always obvious which router functions are to be implemented in the fast path or slow path. Some router designers may choose to include the ARP processing in the fast path instead of in the slow path of a router for performance reasons, and because ARP needs direct access to the physical network. Other may argue for ARP implementation in the slow path instead of the fast path. For example, in the ARP used for Ethernet, if a router gets a datagram to an IP address whose Ethernet address it does not know, it is supposed to send an ARP message and hold the datagram until it gets an ARP reply with the necessary Ethernet address. When the ARP is implemented in the slow path, datagrams for which the destination link layer address is unknown are passed to the CPU, which does the ARP and, once it gets the ARP reply, forwards the datagram and incorporates the link-layer address into future forwarding tables in the network interfaces.

There are other functions which router designers may argue to be not critical and are more appropriate to be implemented in the slow path. IP packet fragmentation and reassembly, source routing option, route recording option, timestamp option, and ICMP message generation are examples of such functions. It can be argued that packets requiring these

functions are rare and can be handled in the slow path: a practical product does not need to be able to perform “wire-speed” routing when infrequently used options are present in the packet. Since such packets having such options comprise a small fraction of the total traffic, they can be handled as exception conditions. As a result, such packets can be handled by the CPU, i.e., the slow path. For IP packet headers with error, generally, the CPU can instruct the inbound network interface to discard the errored datagram. In some cases, the CPU will generate an ICMP message [34]. Alternatively, in the cache-based scheme [31], templates of some common ICMP messages such as the TimeExceeded message are kept in the forwarding engine and these can be combined with the IP header to generate a valid ICMP message.

An IP packet can be fragmented by a router, that is, a single packet can arrive, thereby resulting in multiple, smaller packets being transmitted onto the output ports. This capability allows a router to forward packets between ports where the output is incapable of carrying a packet of the desired length; that is, the MTU of the output port is less than that of the input port. Fragmentation is good in the sense that it allows communication between end systems connected through links with dissimilar MTUs. It is bad in that it imposes a significant processing burden on the router, which must perform more work to generate the resulting multiple output datagrams from the single input IP datagram. It is also bad from a data throughput point of view because, when one fragment is lost, the entire IP datagram must be retransmitted. The main arguments for implementing fragmentation in the slow path is that IP packet fragmentation can be considered an “exception condition”, outside of the fast path. Now that IP MTU discovery [11] is prevalent, fragmentation should be rare.

Reassembly of fragments may be necessary for packets destined for entities within the router itself. These fragments may have been generated either by other routers in the path between the sender and the router in question or by the original sending end system itself. Although fragment reassembly can be a resource-intensive process (both in CPU cycles and memory), the number of packets sent to the router is normally quite low relative to the number of packets being routed through. The number of fragmented packets destined

for the router is a small percentage of the total router traffic. Thus, the performance of the router for packet reassembly is not critical and can be implemented in the slow path.

Figure 8 further categorizes the slow path router functions into two: those performed on a packet-by-packet basis (that is, optional or exception conditions) and those performed as background tasks.

Typical Router Slow Path Functions	
Packet-by-Packet Operations	**Background Tasks**
- Fragmentation and reassembly - Source routing option - Route recording option - Timestamp option - ICMP message generation	- Routing protocols (RIP, OSPF, BGP, etc.) - Network management (SNMP) - Router configuration (BOOTP, DHCP, etc.)

Figure 8. IP router slow-path functions.

3.5.4 Protocol Entities and IP Processing in the Distributed Router Architecture

The IP protocol is the most extensive entity in the packet processing path of an IP router and, thus, IP processing typically determines the achievable performance of a router. Therefore, a decomposition of IP that enables efficient multiprocessing is needed in the distributed router architecture. An example of a typical functional partitioning in the distributed router architecture is shown in Figure 9.

This distributed multiprocessing architecture, means that the various processing elements can work in parallel on their own tasks with little dependence on the other processors in the system. This architecture decouples the tasks associated with determining routes through the network from the time-critical tasks associated with IP processing. The results of this is an architecture with high levels of aggregate system performance and the ability to scale to increasingly higher performance levels.

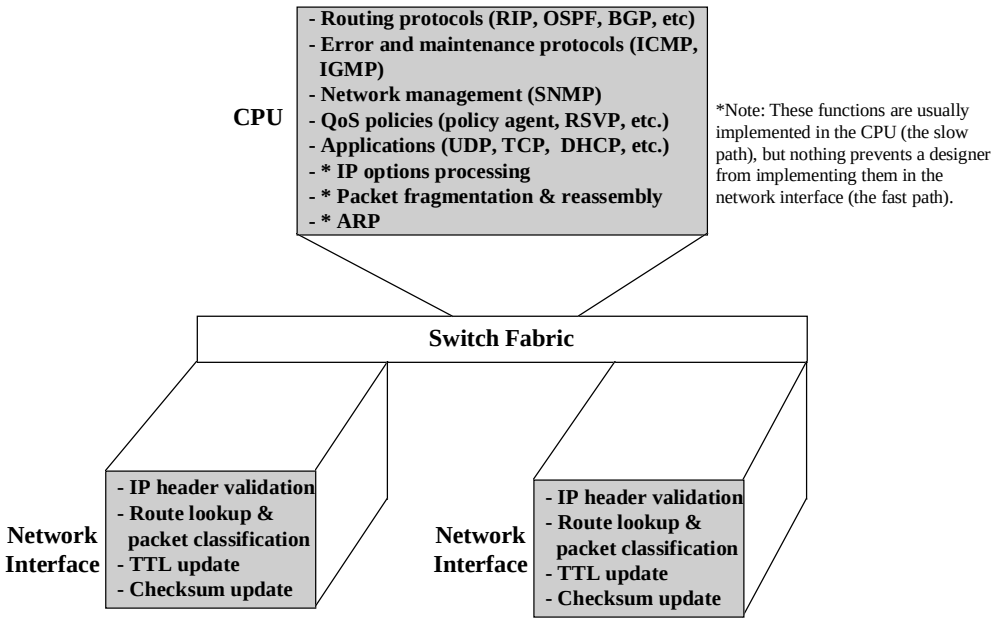


Figure 9. An example functional partitioning in the distributed router architecture.

A high level diagram of a distributed router architecture is shown in Figure 10. Network interface cards built with general-purpose processors and complex communication protocols tend to be more expensive than those built using ASICs and simple communication protocols. Choosing between ASICs and general-purpose processors for an interface card is not straightforward. General-purpose processors tend to be more expensive, but allow extensive port functionality. They are also available off-the-shelf, and their price/performance ratio improves yearly [47]. ASICs are not only cheaper, but can also provide operations that are specific to routing, such as traversing a Patricia tree. Moreover, the lack of flexibility with ASICs can be overcome by implementing functionality in the route processor (e.g., ARP, fragmentation and reassembly, IP options, etc.).

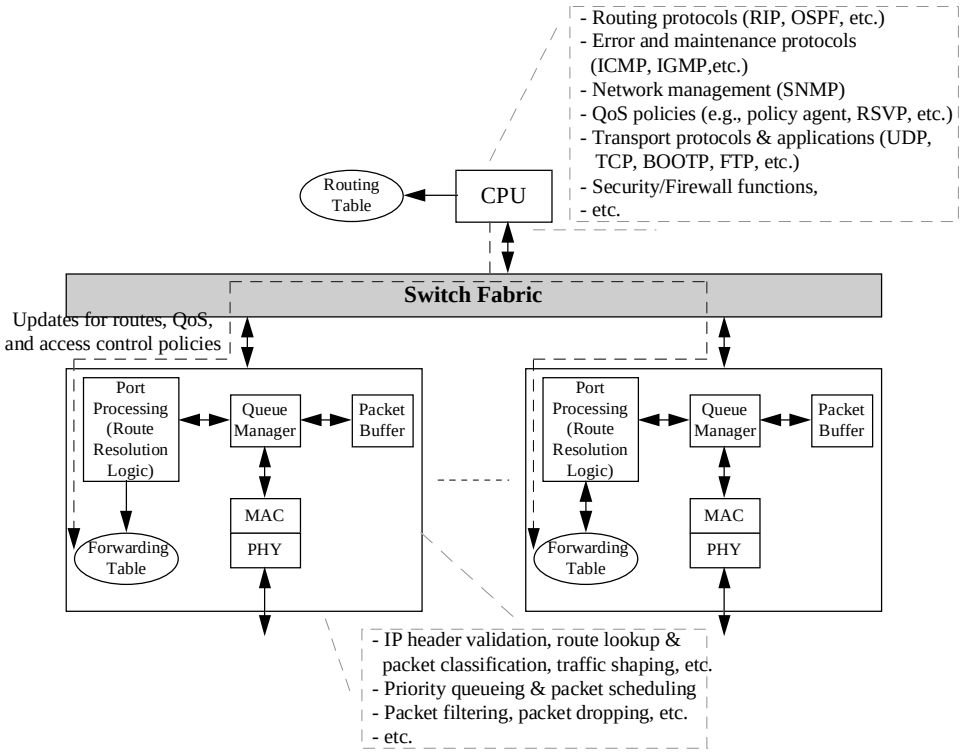


Figure 10. A high level functional diagram of a distributed router architecture.

Some router designers often observe that the IPv4 specification is very stable and say that it would be more cost effective to implement the forwarding engine in an ASIC. It is argued that ASIC can reduce the complexity on each system board by combining a number of functions into individual chips that are designed to perform at high speeds. Other designers also observe that the Internet is constantly evolving in a subtle way that require programmability and as such a fast processor is appropriate for the forwarding engine.

The forwarding database in a network interface consists of several cross-linked tables as illustrated in Figure 11. This database can include IP routes (unicast and multicast), ARP tables, and packet filtering information for QoS and security/access control.

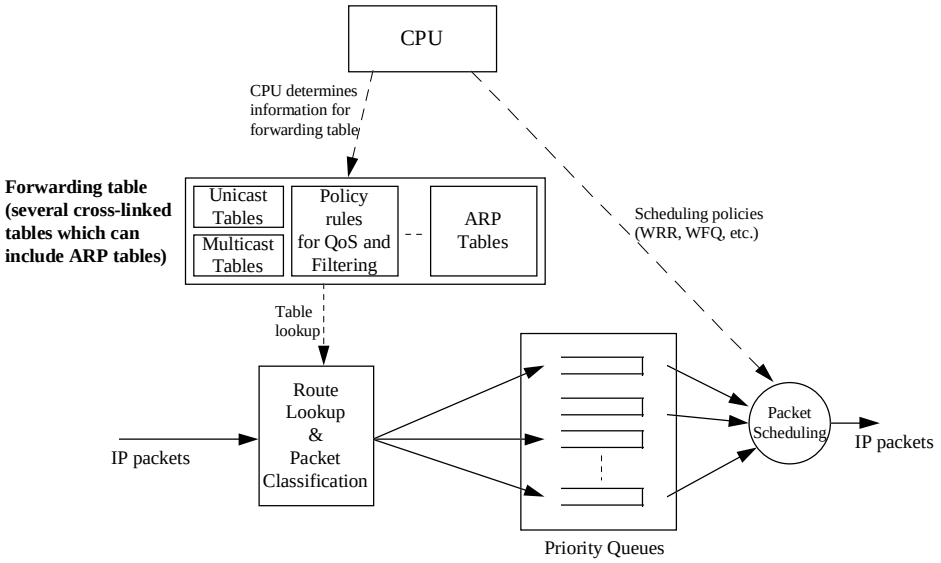


Figure 11. Forwarding database consisting of several cross-linked tables.

Now, let us take a generic shared memory router architecture and then trace the path of an IP packet as it goes through an ingress port and out of an egress port. The IP packet processing steps are shown in Figure 12.

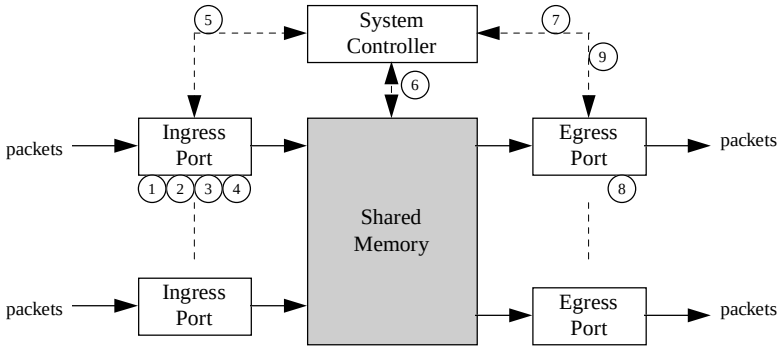


Figure 12. IP packet processing in a shared memory router architecture.

The IP packet processing steps are as follows:

1. *IP Header Validation*: As a packet enters an ingress port, the forwarding logic verifies all Layer 3 information (header length, packet length, protocol version, checksum, etc.).

2. *Route Lookup and Header Processing*: The router then performs an IP address lookup using the packet's destination address to determine the egress (or outbound) port, and performs all IP forwarding operations (TTL decrement, header checksum, etc.).

3. *Packet Classification*: In addition to examining the Layer 3 information, the forwarding engine examines Layer 4 and higher layer packet attributes relative to any QoS and access control policies.

4. With the Layer 3 and higher layer attributes in hand, the forwarding engine performs one or more parallel functions:

    - associates the packet with the appropriate priority and the appropriate egress port(s) (an internal routing tag provides the switch fabric with the appropriate egress port information, the QoS priority queue the packet is to be stored in, and the drop priority for congestion control),
    - redirects the packet to a different (overridden) destination (ICMP redirect),
    - drops the packet according to a congestion control policy (e.g., RED, WRED, etc.), or a security policy, and
    - performs the appropriate accounting functions (statistics collection, etc.).

5. The forwarding engine notifies the system controller that a packet has arrived.

6. The system controller reserves a memory location for the arriving packet.

7. Once the packet has been passed to the shared memory, the system controller signals the appropriate egress port(s). For multicast traffic, multiple egress ports are signalled.

8. The egress port(s) extracts the packet from the known shared memory location using any of a number of algorithms: Weighted Fair Queueing (WFQ), Weighted Round-Robin (WRR), Strict Priority (SP), etc.

9. When the destination egress port(s) has retrieved the packet, it notifies the system controller, and the memory location is made available for new traffic.

4. Typical Switch Fabrics of Routers

Switch fabric design is a very well studied area, especially in the context of ATM switches [48][49] so in this section, we examine briefly the most common fabrics used in router design. The switch fabric in a router is responsible for transferring packets between the other functional blocks. In particular, it routes user packets from the input modules to the appropriate output modules. The design of the switch fabric is complicated by other requirements such as multicasting, fault tolerance, and loss and delay priorities. When these requirements are considered, it becomes apparent that the switch fabric should have additional functions, e.g., concentration, packet duplication for multicasting if required, packet scheduling, packet discarding, and congestion monitoring and control.

Virtually all IP router designs are based on variations or combinations of the following basic approaches: shared memory; shared medium; distributed output buffered; space division (e.g., crossbar). Some important considerations for the switch fabric design are: throughput, packet loss, packet delays, amount of buffering, and complexity of implementation. For given input traffic, the switch fabric designs aim to maximize throughput and minimize packet delays and losses. In addition, the total amount of buffering should be minimal (to sustain the desired throughput without incurring excessive delays) and implementation should be simple.

4.1 Shared Medium Switch Fabric

In a router, packets may be routed by means of a shared medium e.g., bus, ring, or dual bus. The simplest switch fabric is the bus. Bus-based routers implement a monolithic backplane comprising a single medium over which all inter-module traffic must flow. Data is transmitted across the bus using Time Division Multiplexing (TDM), in which each module is allocated a time slot in a continuously repeating transmission. However, a bus is limited in capacity and by the arbitration overhead for sharing this critical resource. In a typical shared memory bus architecture, all ports access a central memory pool via a shared bus. An arbitration mechanism is used to control port access to the shared memory. The challenge is that it is almost impossible to build a bus arbitration scheme fast enough to provide nonblocking performance at multigigabit speeds.

Another example of a fabric using a time-division multiplexed (TDM) bus is shown in Figure 13. Incoming packets are sequentially broadcast on the bus (in a round-robin fashion). At each output, address filters examine the internal routing tag on each packet to determine if the packet is destined for that output. The address filters passes the appropriate packets through to the output buffers.

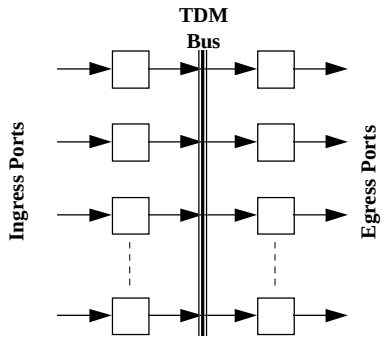


Figure 13. Shared medium bus: a TDM bus.

It is apparent that the bus must be capable of handling the total throughput. For discussion, we assume a router with *N* input ports and *N* output ports, with all port speeds equal to *S* (fixed size) packets per second. In this case, a packet time is defined as the time required to receive or transmit an entire packet at the port speed, i.e., 1/*S* sec. If the bus operates at a sufficiently high speed, at least *NS* packets/sec, then there are no conflicts for bandwidth and all queueing occurs at the outputs. Naturally, if the bus speed is less than *NS* packets/sec, some input queueing will probably be necessary.

The outputs are modular from each other, which has advantages in implementation and reliability. The address filters and output buffers are straightforward to implement. Also, the broadcast-and-select nature of this approach makes multicasting and broadcasting natural. For these reasons, the bus type switch fabric has found a lot of implementation in routers. However, the address filters and output buffers must operate at the speed of the shared medium, which could be up to *N* times faster than the port speed. There is a physical limit to the speed of the bus, address filters, and output buffers; these limit the scalability of this approach to large sizes and high speeds. Either the size *N* or speed S can be large, but there is a physical limitation on the product *NS*. As with the shared memory

approach (to be discussed next), this approach involves output queueing, which is capable of the optimal throughput (compared to simple FIFO input queueing). However, the output buffers are not shared, and hence this approach requires more total amount of buffers than the shared memory fabric for the same packet loss rate.

4.2 Shared Memory Switch Fabric

A typical architecture of a shared memory fabric is shown in Figure 14. Incoming packets are typically converted from a serial to parallel form which are then written sequentially into a (dual port) random access memory. Their packet headers with internal routing tags are typically delivered to a memory controller, which decides the order in which packets are read out of the memory. The outgoing packets are demultiplexed to the outputs, where they are converted from parallel to serial form. Functionally, this is an output queueing approach, where the output buffers all physically belong to a common buffer pool. The output buffered approach is attractive because it can achieve a normalized throughput of one under a full load [50]. Sharing a common buffer pool has the advantage of minimizing the amounts of buffers required to achieve a specified packet loss rate. The main idea is that a central buffer is most capable of taking advantage of statistical sharing. If the rate of traffic to one output port is high, it can draw upon more buffer space until the common buffer pool is (partially or) completely filled.

Because the buffer space can be shared, this approach requires the minimum possible amount of buffering and has the most flexibility to accommodate traffic dynamics, in the sense that the shared memory can absorb large bursts directed to any output. For these reasons it is a popular approach for router design (e.g., [37][40][42][43][45]).

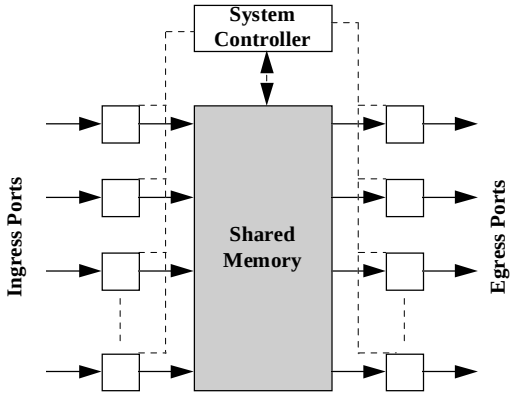


Figure 14. A shared memory switch fabric.

Unfortunately, the approach has its disadvantages. As the packets must be written into and read out from the memory one at a time, the shared memory must operate at the total throughput rate. It must be capable of reading and writing a packet (assuming fixed size packets) in every 1/*NS* sec, that is, *N* times faster than the port speed. As the access time of random access memories is physically limited, this speed-up factor *N* limits the ability of this approach to scale up to large sizes and fast speeds. Either the size *N* or speed *S* can be large, but the memory access time imposes a limit on the product *NS*, which is the total throughput. Moreover, the (centralized) memory controller must process (the routing tags of) packets at the same rate as the memory. This might be difficult if, for instance, the controller must handle multiple priority classes and complicated packet scheduling. Multicasting and broadcasting in this approach will also increase the complexity of the controller.

In shared memory switches, a single point of failure is invariably introduced in the design because adding a redundant switch fabric to this design is so complex and expensive. As a result, shared memory switch fabrics are best suited for small capacity systems.

4.3 Distributed Output Buffered Switch Fabric

The distributed output buffered approach is shown in Figure 15. Independent paths exist between all N^2 possible pairs of inputs and outputs. In this design, arriving packets are broadcast on separate buses to all outputs. Address filters at each output determine if the

packets are destined for that output. Appropriate packets are passed through the address filters to the output queues.

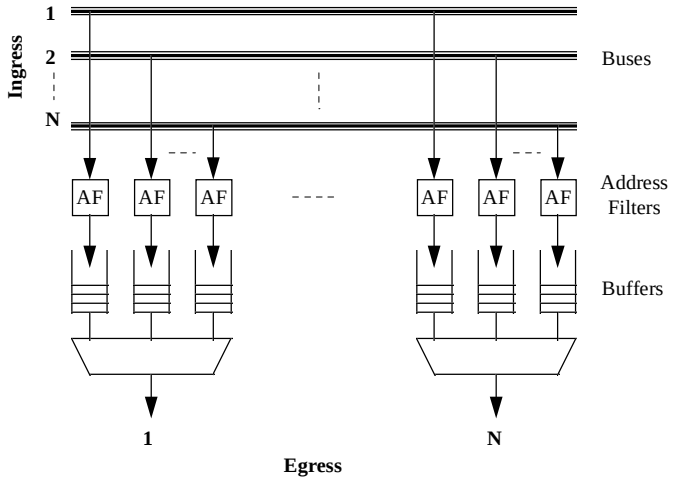


Figure 15. A distributed output buffered switch fabric.

This approach offers many attractive features. Naturally there is no conflict among the N^2 independent paths between inputs and outputs, and hence all queueing occurs at the outputs. As stated earlier, output queueing achieves the optimal normalized throughput compared to simple FIFO input queueing [50]. Like the shared medium approach, it is also broadcast-and-select in nature and, therefore, multicasting is natural. The address filters and output buffers are simple to implement. Unlike the shared medium approach, the address filters and buffers need to operate only at the port speed. All of the hardware can operate at the same speed. There is no speed-up factor to limit scalability in this approach. For these reasons, this approach has been taken in some commercial router designs (e.g., [41]).

Unfortunately, the quadratic N^2 growth of buffers means that the size N must be limited for practical reasons. However, in principle, there is no severe limitation on S. The port speed S can be increased to the physical limits of the address filters and output buffers. Hence, this approach might realize a high total throughput NS packets per second by scaling up the port speed S. The Knockout switch was an early prototype that suggested a trade-off to reduce the amount of buffers at the cost of higher packet loss [51]. Instead of

N buffers at each output, it was proposed to use only a fixed number L buffers at each output (for a total of NL buffers which is linear in N), based on the observation that the simultaneous arrival of more than L packets (cells) to any output was improbable. It was argued that $L = 8$ is sufficient under uniform random traffic conditions to achieve a cell loss rate of 10^{-6} for large N.

4.4 Space Division Switch Fabric: The Crossbar Switch

Optimal throughput and delay performance is obtained using output buffered switches. As long as input port and output port is under-subscribed, 100% throughput is achieved. Moreover, since upon arrival, the packets are immediately placed in the output buffers, it is possible to better control the latency of the packet. This helps in providing QoS guarantees. While this architecture appears to be especially convenient for providing QoS guarantees, it has serious limitations: the output buffered switch memory speed must be equal to at least the aggregate input speed across the switch. To achieve this, the switch fabric must operate at a rate at least equal to the aggregate of all the input links connected to the switch. However, increasing line rate (S) and increasing switch size (N) make it extremely difficult to significantly speedup the switch fabric, and also build memories with a bandwidth of order $O(NS)$.

At multigigabit and terabit speeds it becomes difficult to build output buffered switches. As a result some high-speed implementations are based on the input buffered switch architecture. One of the most popular interconnection networks used for building input buffered switches is the crossbar because of its (i) low cost, (ii) good scalability and (iii) non-blocking properties. Crossbar switches have an architecture that, depending on the implementation, can scale to very high bandwidths. Considerations of cost and complexity are the primary constraints on the capacity of switches of this type. The crossbar switch (see Figure 16) is a simple example of a space division fabric which can physically connect any of the N inputs to any of the N outputs. An input buffered crossbar switch has the crossbar fabric running at the link rate. In this architecture buffering occurs at the inputs, and the speed of the memory does not need to exceed the speed of a single port. Given the current state of technology, this architecture is widely considered to be substantially more

scalable than output buffered or shared memory switches. This has renewed interest in switches with lower complexity (and cost) such as input buffered switches despite their deficiencies. However, the crossbar architecture presents many technical challenges that need to be overcome in order to provide bandwidth and delay guarantees. Examples of commercial routers that use crossbar switch fabrics are [38][39][45].

We start with the issue of providing bandwidth guarantees in the crossbar architecture. For the case when there is a single FIFO queue at each input, it has long been known that a serious problem referred to as head-of-line (HOL) blocking [50] can substantially reduce achievable throughput. In particular, the well-known results of [50] is that for uniform random distribution of input traffic, the achievable throughput is only 58.6%. Moreover, Li [52] has shown that the maximum throughput of the switch decreases monotonically with increasing burst size. Considerable amount of work has been done in recent years to build input buffered switches that match the performance of an output buffered switch. One way of reducing the effect of HOL blocking is to increase the speed of the input/output channel (i.e., the speedup of the switch fabric). Speedup is defined as the ratio of the switch fabric bandwidth and the bandwidth of the input links. There have been a number of studies such as [53][54] which showed that an input buffered crossbar switch with a single FIFO at the input can achieve about 99% throughput under certain assumptions on the input traffic statistics for speedup in the range of 4 - 5. A more recent simulation study [55] suggested that speedup as low as 2 may be sufficient to obtain performance comparable to that of output buffered switches.

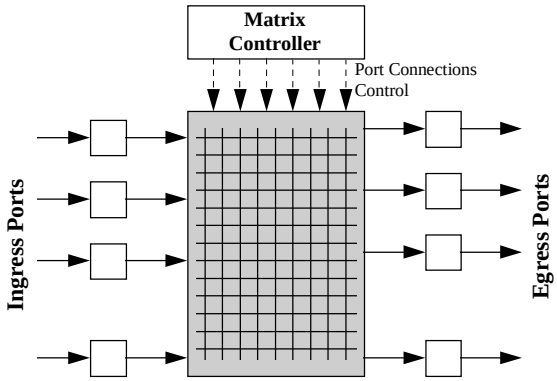


Figure 16. A crossbar switch.

Another way of eliminating the HOL blocking is by changing the queueing structure at the input. Instead of maintaining a single FIFO at the input, a separate queue per each output can be maintained at each input. To eliminate HOL blocking, virtual output queues (VOQs) were proposed at the inputs. However, since there could be contention at the inputs and outputs, there is a necessity for an arbitration algorithm to schedule packets between various inputs and outputs (equivalent to the matching problem for bipartite graphs). It has been shown that an input buffered switch with VOQs can provide asymptotic 100% throughput using a maximum matching algorithm [56]. However, the complexity of the best known maximum match algorithm is too high for high speed implementation. Moreover, under certain traffic conditions, maximum matching can lead to starvation. Over the years, a number of maximal matching algorithms have been proposed [57][58][59][60][61].

As stated above, increasing the speedup of the switch fabric can improve the performance of an input buffered switch. However, when the switch fabric has a higher bandwidth than the links, buffering is required at the outputs too. Thus a combination of input buffered and output buffered switch is required, i.e., CIOB (Combined Input and Output Buffered). The goal of most designs then is to find the minimum speedup required to match the performance of an output buffered switch using a CIOB and VOQs. McKeown *et al*. [62] shown that a CIOB switch with VOQs is always work conserving if speedup is greater $N/2$. In a recent work, Prabhakar *et al*. [63] showed that a speed of 4 is sufficient to

emulate an output buffered switch (with an output FIFO) using a CIOB switch with VOQs.

4.5 Other Issues in Router Switch Fabric Design

We have described above four typical design approaches for router switch fabrics. Needless to say, endless variations of these designs can be imagined but the above are the most common fabrics found in routers. There are other issues applicable to understanding the trade-offs involved in any new design. We discuss some of these issues next.

4.5.1 Construction of Large Router Switch Fabrics

With regards to the construction of large switch fabrics, most of the four basic switch fabric design approaches are capable of realizing routers of limited throughput. The shared memory and shared medium approaches can achieve a throughput limited by memory access time. The space division approach has no special constraints on throughput or size, only physical factors do limit the maximum size in practice. There are physical limits to the circuit density and number of input/output (I/O) pins. Interconnection complexity and power dissipation become more difficult issues with fabric size. In addition, reliability and repairability become difficult with size. Modifications to maximize the throughput of space division fabrics to address HOL blocking increases the implementation complexity.

It is generally accepted that large router switch fabrics of 1 terabits per second (Tbps) throughput or more cannot be realized simply by scaling up a fabric design in size and speed. Instead, large fabrics must be constructed by interconnection of switch modules of limited throughput. The small modules may be designed following any approach, and there are various ways to interconnect them.

4.5.2 Fault Tolerance and Reliability

With the rapid growth of the Internet and the emergence of growing competition between Internet Service Providers (ISPs), reliability has become an important issue for IP routers. In addition, multigigabit routers will be deployed in the core of enterprise networks and the Internet. Traffic from thousands of individual flows pass through the switch fabric at

any given time [32]. Thus, the robustness and overall availability of the switch fabric becomes a critically important design parameter. As in any communication system, fault tolerance is achieved by adding redundancy to the crucial components. In a router, one of the most crucial components is the packet routing and buffering fabric. In addition to redundancy, other considerations include detection of faults and isolation and recovery.

4.5.3 Multicasting

New applications or services are emerging that utilize multicast transport. These applications include distribution of news, financial data, software, video, audio and multi-person conferencing. These services or applications will require a router to multicast an incoming packet to a number of selected outputs or broadcast it to all outputs. Multicasting is inherently natural to the shared medium and distributed output-buffered approaches. Both approaches consist of broadcasting incoming packets and selecting the appropriate packets with address filters at the output buffers. For multicasting, an address filter can recognize a set of multicast addresses as well as output port addresses. As a result, multicasting is natural in these two broadcast-and-select approaches.

Multicasting is not natural to the shared memory approach but can be implemented with additional control circuitry. A multicast packet may be duplicated before the memory or read multiple times from the memory. The first approach obviously requires more memory because multiple copies of the same packet are maintained in the memory. In the second approach, a packet is read multiple times from the same memory location. The control circuitry must keep the packet in memory until it has been read to all the output ports in the multicast group.

Multicast in the space division fabrics is simple to implement but has some consequences. For example, a crossbar switch (with input buffering) is naturally capable of broadcasting one incoming packet to multiple outputs. However, this would aggravate the HOL blocking at the input buffers. Approaches to alleviate the HOL blocking effect would increase the complexity of buffer control. Other inefficient approaches in crossbar switches require an input port to write out multiple copies of the packet to different output ports

one at a time. This does not support the one-to-many transfers required for multicasting as in the shared bus architecture and the fully distributed output buffered architectures. The usual concern about making multiple copies is that it reduces effective switch throughput. Several approaches for handling multicasting in crossbar switches have been proposed [64]. Generally, multicasting increases the complexity of space division fabrics.

4.5.4 Buffer Management and Quality of Service (QoS)

The prioritization of mission critical applications and the support of IP telephony and video conferencing create the requirement for supporting QoS enforcement with the switch fabric. These applications are sensitive to both absolute latency and latency variations.

Beyond best-effort service, routers are beginning to offer a number of QoS or priority classes. Priorities are used to indicate the preferential treatment of one traffic class over another. The switch fabrics must handle these classes of traffic differently according to their QoS requirements. In the output buffered switch fabric, for example, typically the fabric will have multiple buffers at each output port and one buffer for each QoS traffic class. The buffers may be physically separate or a physical buffer may be divided logically into separate buffers.

Buffer management here refers to the discarding policy for the input of packets into the buffers (e.g., Drop Tail, Drop-From-Front, Random Early Detection (RED), etc.), and the scheduling policy for the output of packets from the buffers (e.g., strict priority, weighted round-robin (WRR), weighted fair queueing (WFQ), etc.). Buffer management in the IP router involves both dimensions of time (packet scheduling) and buffer space (packet discarding). The IP traffic classes are distinguished in the time and space dimensions by their packet delay and packet loss priorities. We therefore see that buffer management and QoS support is an integral part of the switch fabric design.

5. Conclusions and Open Problems

IP provides an amazing degree of flexibility in building large and arbitrary complex networks. Interworking routers capable of forwarding aggregate data rates in the

multigigabit and terabit per second range are required in emerging high performance networking environments. This paper has presented an evaluation of typical approaches proposed for designing high speed routers. We have focused primarily on the architectural overview and the design of the components that have the highest effect on performance.

First, we have observed that high-speed routers need to have enough internal bandwidth to move packets between its interfaces at multigigabit and terabit rates. The router design should use a switched backplane. Until very recently, the standard router used a shared bus rather than a switched backplane. While bus-based routers may have satisfied the early needs of IP networks, emerging demands for high bandwidth, QoS delivery, multicast, and high availability place the bus architecture at a significant disadvantage. For high speeds, one really needs the parallelism of a switch with superior QoS, multicast, scalability, and robustness properties. Second, routers need enough packet processing power to forward several million packets per second (Mpps). Routing table lookups and data movements are the major consumers of processing cycles. The processing time of these tasks does not decrease linearly if faster processors are used. This is because of the sometimes dominating effect of memory access rate.

Experience has shown that while an IP router must, in general, perform a myriad of functions, in practice the vast majority of packets need only a few operations performed in real-time. Thus, the performance critical functions can be implemented in hardware (the fast path) and the remaining (necessary, but less time-critical) functions in software (the slow path). IP contains many features and functions that are either rarely used or that can be performed in the background of high-speed data forwarding (for example, routing protocol operation and network management). The router architecture should be optimized for those functions that must be performed in real-time, on a packet-by-packet basis, for the majority of the packets. This creates an optimized routing solution that route packets at high speed at a reasonable cost.

It has been observed in [47] that the cost of a router port depends on, 1) the amount and kind of memory it uses, 2) its processing power, and 3) the complexity of the protocol

used for communication between the port and the route processor. This means the design of a router involve trade-offs between performance, complexity, and cost.

Router ports built with general-purpose processors and complex communication protocols tend to be more expensive than those built using ASICs and simple communication protocols. Choosing between ASICs and general-purpose processors for an interface card is not straightforward. General-purpose processors tend to be more expensive, but allow extensive port functionality. They are also available off-the-shelf, and their price/performance ratio improves yearly. ASICs are not only cheaper, but can also provide operations that are specific to routing, such as traversing a Patricia tree. Moreover, the lack of flexibility with ASICs can be overcome by implementing functionality in the route processor.

The cost of a router port is also proportional to the type and size of memory on the port. SRAMs offer faster access times, but are more expensive than DRAMs. Buffer memory is another parameter that is difficult to size. In general, the rule of thumb is that a port should have enough buffers to support at least one bandwidth-delay product worth of packets, where the delay is the mean end-to-end delay and the bandwidth is the largest bandwidth available to TCP connections traversing that router. This sizing allows TCP to increase their transmission windows without excessive losses.

The cost of a router port is also determined by the complexity of the internal connections between the control paths and the data paths in the port card. In some designs, a centralized controller sends commands to each port through the switch fabric and the port's internal buffers. Careful engineering of the control protocol is necessary to reduce the cost of the port control circuitry and also the loss of command packets which will certainly need retransmission.

Significant advances have been made in router designs to address the most demanding customer issues regarding high speed packet forwarding (e.g., route lookup algorithms, high-speed switching cores and forwarding engines), low per-port cost, flexibility and programmability, reliability, and ease of configuration. While these advances have been

made in the design of IP routers, some important open issues still remain to be resolved. These include packet classification and resource provisioning, improved price/performance router designs, "active networking" [65] and ease of configuration, reliability and fault tolerance designs, and Internet billing/pricing. Extensive work is being carried out both in the research community and industry to address these problems.

References

[1]. P. Newman, T. Lyon, and G. Minshall, "Flow Labelled IP: A Connectionless Approach to ATM," *Proc IEEE Infocom'96*, San Francisco, CA, March 1996, pp. 1251 - 1260.

[2]. Y. Katsube, K. Nagami, and H. Esaki, "Toshiba's Router Architecture Extensions for ATM: Overview," *IETF RFC 2098*, April 1997.

[3]. Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow, "Cisco Systems' Tag Switching Architecture Overview," *IETF RFC 2105*, Feb. 1997.

[4]. F. Baker, "Requirements for IP Version 4 Routers," *IETF RFC* 1812, Jun. 1995.

[5]. W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Reading, MA: Addison-Wesley, 1994.

[6]. C. Huitema, *Routing in the Internet*, Prentice Hall, 1996.

[7]. J. Moy, *OSPF: Anatomy of an Internet Routing Protocol*, 1998.

[8]. R. Braden, D. Borman, and C. Partridge, "Computing the Internet Checksum," *IETF RFC 1071*, Sept. 1988.

[9]. T. Mallory and A. Kullberg, "Incremental Updating of the Internet Checksum," *IETF RFC 1141*, Jan. 1990.

[10]. C. A. Kent and J. C. Mogul, "Fragmentation Considered Harmful," *Computer Commun. Rev.*, Vol. 17, No. 5, Aug. 1987, pp. 390 - 401.

[11]. J. Mogul and S. Deering, "Path MTU Discovery" *IETF RFC 1191*, April 1990.

[12]. V. Fuller et al. "Classless Inter-Domain Routing," *IETF RFC 1519*, Jun. 1993.

[13]. K. Sklower, "A Tree-Based Packet Routing Table for Berkeley Unix," *USENIX, Winter'91*, Dallas, TX, 1991.

[14]. W. Doeringer, G. Karjoth, and M. Nassehi, "Routing on Longest-Matching Prefixes," *IEEE/ACM Trans. on Networking*, Vol. 4, No. 1, Feb. 1996, pp. 86 - 97.

[15]. D. C. Feldmeier, "Improving Gateway Performance with a Routing-Table Cache," *Proc. IEEE Infocom'88*, New Orleans, LI, Mar. 1988.

[16]. C. Partridge, "Locality and Route Caches," *NSF Workshop on Internet Statistics Measurement and Analysis*, San Diego, CA, Feb. 1996.

[17]. D. Knuth, *The Art of Computer Programming, Vol. 3. Sorting and Searching*, Addison-Wesley, 1973.

[18]. M. Degermark, *et al*., "Small Forwarding Tables for Fast Routing Lookups," *Proc. ACM SIGCOMM'97*, Cannes, France, Sept. 1997.

[19]. H.-Y. Tzeng, "Longest Prefix Search Using Compressed Trees," *Proc. Globecom'98*, Sydney, Australia, Nov. 1998.

[20]. M. Waldvogel, G. Varghese, J. Turner, and B. Plattner, "Scalable High Speed IP Routing Lookup," *Proc. ACM SIGCOMM'97*, Cannes, France, Sept. 1997.

[21]. V. Srinivasan and G. Varghese, "Faster IP Lookups using Controlled Prefix Expansion," *Proc. ACM SIGMETRICS*, May 1998.

[22]. S. Nilsson and G. Karlsson, "Fast Address Look-Up for Internet Routers," *Proc. of IEEE Broadband Communications'98*, April 1998.

[23]. E. Filippi, V. Innocenti, and V. Vercellone, "Address Lookup Solutions for Gigabit Switch/Router," *Proc. Globecom'98*, Sydney, Australia, Nov. 1998.

[24]. A. J. McAuley and P. Francis, “Fast Routing Table Lookup using CAMs,” *Proc. IEEE Infocom’93*, San Francisco, CA, Mar. 1993, pp. 1382 - 1391.

[25]. T. B. Pei and C. Zukowski, “Putting Routing Tables in Silicon,” *IEEE Network*, Vol. 6, Jan. 1992, pp. 42 - 50.

[26]. M. Zitterbart et al., “HeaRT: High Performance Routing Table Lookup,” *4th IEEE Workshop on Architecture & Implementation of High Performance Communications Subsystems*, Thessaloniki, Greece, Jun. 1997.

[27]. P. Gupta, S. Lin, and N. McKeown, “Routing Lookups in Hardware at Memory Access Speeds,” *Proc. IEEE Infocom’98*, Mar. 1998.

[28]. S. F. Bryant and D. L. A. Brash, “The DECNIS 500/600 Multiprotocol Bridge/Router and Gateway,” *Digital Technical Journal*, Vol. 5, No. 1, 1993.

[29]. P. Marimuthu, I. Viniotis, and T. L. Sheu, “A Parallel Router Architecture for High Speed LAN Internetworking,” *17th IEEE Conf. on Local Computer Networks*, Minneapolis, Minnesota, Sept. 1992.

[30]. S. Asthana, C. Delph, H. V. Jagadish, and P. Krzyzanowski, “Towards a Gigabit IP Router,” *Journal of High Speed Networks*, Vol. 1, No. 4, 1992.

[31]. C. Partridge et al., “A 50Gb/s IP Router,” *IEEE/ACM Trans. on Networking*, Vol. 6, No. 3, Jun 1998, pp. 237 - 248.

[32]. K. Thomson, G. J. Miller, and R. Wilder, “Wide-Area Traffic Patterns and Characteristics,” *IEEE Network*, Dec. 1997.

[33]. V. P. Kumar, T. V. Lakshman, and D. Stiliadis, “Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow’s Internet,” *IEEE Commun. Mag.*, May 1998, pp. 152 - 164.

[34]. A Tantawy, O Koufopavlou, M. Zitterbart, and J. Abler, "On the Design of a Multigigabit IP Router," *Journal of High Speed Networks*, Vol. 3, 1994, pp. 209 - 232.

[35]. O Koufopavlou, A. Tantawy, and M. Zitterbart, "IP-Routing among Gigabit Networks," *Interoperability in Broadband Networks*, S. Rao (Ed.), IOS Press, 1994, pp. 282 - 289.

[36]. O Koufopavlou, A. Tantawy, and M. Zitterbart, "A Comparison of Gigabit Router Architectures," *High Performance Networking*, E. Fdida (Ed.), Elsevier Science B. V. (North-Holland), 1994.

[37]. "Implementing the Routing Switch: How to Route at Switch Speeds and Switch Costs", *White Paper*, Bay Networks, 1997.

[38]. "Cisco 12000 Gigabit Switch Router," *White Paper*, Cisco Systems, 1997.

[39]. "Performance Optimized Ethernet Switching," *Cajun White Paper #1*, Lucent Technologies.

[40]. "Internet Backbone Routers and Evolving Internet Design," *White Paper*, Juniper Networks, Sept. 1998.

[41]. "The Integrated Network Services Switch Architecture and Technology," *White Paper*, Berkeley Networks, 1997.

[42]. "Torrent IP9000 Gigabit Router," *White Paper*, Torrent Networking Technologies, 1997.

[43]. "Wire-Speed IP Routing," *White Paper*, Extreme Networks, 1997.

[44]. "PE-4884 Gigabit Routing Switch," *White Paper*, Packet Engines, 1997.

[45]. "GRF 400 White Paper: A Practical IP Switch for Next-Generation Networks," *White Paper*, Ascend Communications, 1998.

[46]. "Rule Your Networks: An Overview of StreamProcessor Applications," *White Paper*, NEO Networks, 1997.

[47]. S. Keshav and R. Sharma, "Issues and Trends in Router Design," *IEEE Commun. Mag.*, May 1998, pp. 144 - 151.

[48]. H. Ahmadi and W. Denzel, "A Survey of Modern High-Performance Switching Techniques," *IEEE J. on Selected Areas in Commun.*, Vol. 7, Sept. 1989, pp. 1091 - 1103.

[49]. F. Tobagi, "Fast Packet Switch Architectures for Broadband Integrated Services Digital Networks," *Proc. of the IEEE*, Vol. 78, Jan. 1990, pp. 133 - 178.

[50]. M. Karol, M. Hluchyj, and S. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Trans. on Commun.*, Vol. COM-35, Dec. 1987, pp. 1337 - 1356.

[51]. Y.-S. Yeh, M. Hluchyj and A. S. Acampora, "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching," *IEEE J. on Selected Areas in Commun.* Vol. SAC-5, No. 8, Oct. 1987, pp. 1274 - 1282.

[52]. S.-Q. Li, "Performance of a Non-blocking Space-division Packet Switch with Correlated Input Traffic," *Proc. IEEE Globecom'89*, 1989, pp. 1754 - 1763.

[53]. C.-Y. Chang, A. J. Paulraj, and T. Kailath, "A Broadband Packet Switch Architecture with Input and Output Queueing," *Proc. Globecom'94*, 1994.

[54]. I. Iliadis, and W. Denzel, "Performance of Packet Switches with Input and Output Queueing," *Proc. ICC'90*, 1990.

[55]. R. Guerin and K. N. Sivarajan, "Delay and Throughput Performance of Speed-Up Input-Queueing Packet Switches," *IBM Research Report RC 20892*, Jun. 1997.

[56]. N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch," *Proc. IEEE Infocom'96*, 1996, pp. 296 - 302.

[57]. T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High Speed Switch Scheduling for Local Area Networks," *ACM Trans. on Computer Systems*, Vol. 11, No. 4, Nov. 1993, pp. 319 - 352.

[58]. D. Stiliadis and A. Verma, "Providing Bandwidth Guarantees in an Input-Buffered Crossbar Switch," *Proc. IEEE Infocom'95*, 1995, pp. 960 - 968.

[59]. N. McKeown, "Scheduling Algorithms for Input-Queued Cell Switches," *Ph.D. Thesis*, UC Berkeley, May 1995.

[60]. C. Lund, S. Phillips, and N. Reingold, "Fair Prioritized Scheduling in an Input-Buffered Switch," *Proc. Broadband Communications*, 1996.

[61]. A. Mekkittikul and N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches," *Proc. IEEE Infocom'98*, Mar. 1998.

[62]. N. McKeown, B. Prabhakar, and M. Zhu, "Matching Output Queueing with Combined Input and Output Queueing," *Proc. 35th Annual Allerton Conf. on Communications, Control and Computing*, Oct. 1997.

[63]. B. Prabhakar and N. McKeown, "On the Speedup Required for Combined Input and Output Queueing Switching," Computer Systems Lab, *Technical Report CSL-TR-97-738*, Stanford University.

[64]. N. McKeown, "Fast Switched Backplane for a Gigabit Switched Router," *Technical Report*, Dept. of Elect. Eng., Stanford University.

[65]. D. Tennenhouse, et al., "A Survey of Active Network Research," *IEEE Commun. Mag.*, Jan. 1997.