

**This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.**

Author(s): Nguyen, Hung Khanh; Zhang, Yanru; Chang, Zheng; Han, Zhu

Title: Parallel and Distributed Resource Allocation With Minimum Traffic Disruption for Network Virtualization

Year: 2017

Version:

Please cite the original version:

Nguyen, H. K., Zhang, Y., Chang, Z., & Han, Z. (2017). Parallel and Distributed Resource Allocation With Minimum Traffic Disruption for Network Virtualization. *IEEE Transactions on Communications*, 65(3), 1162-1175.
<https://doi.org/10.1109/TCOMM.2017.2650994>

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Parallel and Distributed Resource Allocation With Minimum Traffic Disruption for Network Virtualization

Hung Khanh Nguyen, *Student Member, IEEE*, Yanru Zhang, *Member, IEEE*,
Zheng Chang, *Member, IEEE*, and Zhu Han, *Fellow, IEEE*

Abstract—Wireless network virtualization has been advocated as one of the most promising technologies to provide multifarious services and applications for the future Internet by enabling multiple isolated virtual wireless networks to coexist and share the same physical wireless resources. Based on the multiple concurrent virtual wireless networks running on the shared physical substrate, service providers can independently manage and deploy different end-users services. This paper proposes a new formulation for bandwidth allocation and routing problem for multiple virtual wireless networks that operate on top of a single substrate network to minimize the operation cost of the substrate network. We also propose a preventive traffic disruption model for virtual wireless networks to minimize the amount of traffic that service providers have to reduce when substrate links fail by incorporating ℓ_1 -norm into the objective function. Due to the large number of constraints in both normal state and link failure states, the formulated problem becomes a large-scale optimization problem and is very challenging to solve using the centralized computational method. Therefore, we propose the decomposition algorithms using the alternating direction method of multipliers that can be implemented in a parallel and distributed fashion. The simulation results demonstrate the computational efficiency of our proposed algorithms as well as the advantage of the formulated model in ensuring the minimal amount of traffic disruption when substrate links fail.

Index Terms—Wireless network virtualization, resource allocation, routing, ADMM, distributed algorithm, preventive traffic disruption.

I. INTRODUCTION

THE rapid growth of traffic demand and application proliferation creates irresistible challenges for traditional wireless networks to ensure the quality of service (QoS) and quality of experience of subscribers [1], [2]. However, due to the inefficient resource utilization and the tightly coupling between hardware and wireless protocols caused by

the inherent design, the current wireless networks and Internet can hardly meet such great expectations without fundamentally changing network architectures [3], [4]. Recently, wireless network virtualization has been proposed as one of the key enablers to overcome the ossification of the current Internet by allowing diverse services and applications coexist on the same infrastructure [5]–[8]. In wireless network virtualization, the traditional Internet service providers are decoupled into infrastructure providers (InPs) who own and manage only infrastructure resources, and service providers (SPs) who lease resources from InPs and concentrate on providing services to subscribers [9]. The physical resources that belong to different InPs are virtualized into a single physical substrate network. Consequently, multiple virtual wireless networks are deployed and operated on top of the single substrate network [10]. As a result, multiple experiments can be performed and tested simultaneously on isolated virtual networks without affecting the operation of the others. Therefore, wireless network virtualization offers great opportunities to shorten the process of evaluating and deploying innovative technologies. Moreover, by sharing the same infrastructure resources, expenses of wireless network expansion and operation can be significantly reduced [11].

Despite wireless network virtualization has been advocated as a viable technology to enhance resource utilization [12], several research challenges remain to be studied [13]. One of the important challenges lies in how to efficiently allocate physical resources to multiple virtual wireless networks and find the optimal routing solution in each virtual network operated by SPs [14]. While the problem of resource allocation for wired network virtualization has been extensively studied in the literature (see [15] and references therein), recent research has tried to extend to wireless scenario. The work in [16] models the resource allocation problem for wireless network virtualization using a stochastic game, in which SPs bid for the resources to satisfy their service objectives. In [17], a wireless resource allocation problem in terms of transmit power and wireless spectrum is studied using game theory to maximize the aggregate spectrum efficiency. A resource allocation in virtualized small cell networks with full duplex self-backhauls is formulated in [18] as a mixed combinatorial optimization problem to maximize the total utility of all mobile virtual network operators. Zhu and Hossain [19] design a hierarchical combinatorial auction mechanism for the resource allocation problem in wireless virtualization. A resource allocation

Manuscript received January 1, 2016; revised July 2, 2016, October 7, 2016, and December 20, 2016; accepted January 2, 2017. This work was supported by the U.S. National Science Foundation under Grant CNS-1646607, Grant ECCS-1547201, Grant CCF-1456921, Grant CNS-1443917, Grant ECCS-1405121, and Grant NSFC61428101. The associate editor coordinating the review of this paper and approving it for publication was B. Liang.

H. K. Nguyen, Y. Zhang, and Z. Han are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA (e-mail: hknguyen9@uh.edu; yzhang82@uh.edu; zhan2@uh.edu).

Z. Chang is with the Department of Mathematical Information Technology, University of Jyväskylä, FIN-40014 Jyväskylä, Finland (e-mail: zheng.chang@jyu.fi).

problem for virtualized full-duplex relaying networks is formulated in [20] to maximize the total utility. In [21], the design and implementation of a network virtualization substrate in cellular networks is described.

The problem of deriving the optimal resource allocation and routing for wireless networks has been widely addressed in the literature [22], [23]. The work in [24] studies the simultaneous routing and resource allocation problem in wireless data networks and derives the optimal solution using dual decomposition framework [25]. By using the similar methodology, [26] proposes a fully distributed algorithm for the optimal routing, relay selection, and power allocation in multihop wireless networks. Recently, the alternating direction method of multipliers (ADMM) has attracted much of interest in the field of distributed and large-scale optimization due to its advantageous convergence property [27]. For instance, Tabrizi *et al.* [28] propose a semi-distributed algorithm for optimizing resource utilization in dense wireless areas using ADMM by simplifying the original non-convex problem into a convex form. A problem of joint resource allocation and routing optimization in wireless sensor networks is formulated and a distributed algorithm based on ADMM decomposition is proposed in [29]. The authors also demonstrate the faster convergence speed of the ADMM against the dual decomposition method through extensive simulation results.

Unpredictable wireless network events such as link failures may happen frequently. Any substrate link failure will affect the services of SPs, who have virtual wireless networks that operate on top of that substrate link. In order to guarantee for nonstop services even after substrate link failures, resource allocation and routing problems for virtual wireless networks need to take into account the requirement for maintaining QoS of SPs. This motivates us to consider a resource allocation and routing problem for multiple virtual wireless networks to achieve efficient resource utilization for InPs as well as ensure QoS for SPs. Even though incorporating QoS requirements for SPs against network failures can be applied for both wired and wireless networks, we primarily focus on resource allocation with QoS constraints for wireless network virtualization in this paper due to the distributed topology and independent operation nature of SPs, which make the preventive traffic disruption model is more suitable for wireless applications.

While resource allocation and routing problem may involve the optimization process from different layers such as power control, channel assignment, etc., to cope with the rapid variation of wireless resources, we instead focus on the preventive traffic disruption perspective from the unique characteristic of the resource allocation problem in wireless network virtualization since any physical network failure will affect multiple virtual networks simultaneously. Particularly, the objective is to minimize the operation cost of the substrate network. Moreover, we incorporate the preventive traffic disruption model into the resource allocation and routing problem to ensure the minimal amount of traffic reduction of SPs when substrate links fail. Due to the large number of constraints incorporated into the model, the formulated problem becomes a large-scale optimization problem, and can be intractably solved by the centralized computational framework.

Therefore, we apply the ADMM-based decomposition technique to efficiently solve the problem. By jointly tackling the above discussed challenges, the proposed model in this paper lays a fundamental theoretical framework for ensuring the QoS in resource allocation for multiple virtual wireless networks operated on top a single substrate network. In addition, the proposed distributed implementation facilitates for the practical deployment of large-scale virtual wireless networks. Specifically, in realistic applications, different SPs provide different types of services and target different end-users, and therefore will have different requirements for QoS. By applying our proposed model, the predetermined QoS levels of different virtual networks can be satisfied while still guarantee the isolating operation between different entities in wireless virtualization, i.e., each SP dictates the optimization criterion independently when participating in the resource allocation problem with InPs. The main technical contributions of this paper can be summarized as follows:

- *Preventive Traffic Disruption Modeling:* We propose a preventive traffic disruption model for virtual networks when a substrate link failure event happens. We also incorporate ℓ_1 -norm into the objective function to ensure the minimal amount of traffic reduction of SPs.
- *Parallel and Distributed Implementation:* We propose two algorithms based on the ADMM decomposition technique. The first algorithm provides a parallel computational framework that can be solved concurrently at different computing nodes, and the second algorithm allows SPs and substrate links distributively solve local problems to achieve the global optimal solution.
- *Performance Evaluation:* We evaluate the performance of our proposed algorithms using various system parameters. We also demonstrate the efficacy of our preventive model in reducing the amount of traffic reduction.

The remainder of this paper is organized as follows. We explain the network model and assumptions in Section II. The resource allocation and routing problem for virtual wireless networks is formulated in Section III. Section IV describes the preventive traffic disruption model with link failures for virtual networks. We propose two decomposition algorithms using ADMM in Section V. Simulation results are presented in Section VI. We provide some discussions about handling network failure mechanisms between wired and wireless networks in Section VII, and Section VIII concludes the paper. The key notations of our paper are summarized in Table I.

II. NETWORK MODEL AND ASSUMPTIONS

A. Wireless Network Virtualization

We consider a wireless network with a set of InPs. Each InP possesses and operates a physical network, also call substrate network. The physical network is composed of physical nodes connected by physical links that form the physical topology. Based on the virtualization frameworks, the physical networks of all InPs are virtualized into a unique physical topology, denoted by a directed graph $G_s = (\mathcal{N}_s, \mathcal{L}_s)$, where \mathcal{N}_s is the set of physical nodes and \mathcal{L}_s is the set of physical links.

TABLE I
NOTATION DEFINITIONS

Symbol	Physical Meaning
$G_s = (\mathcal{N}_s, \mathcal{L}_s)$	topology of substrate network
$G_k = (\mathcal{N}_k, \mathcal{L}_k)$	topology of virtual network k
$L_s \triangleq \mathcal{L}_s $	number of substrate links
$L_k \triangleq \mathcal{L}_k $	number of virtual links in virtual network k
$K \triangleq \mathcal{K} $	number of service providers or virtual networks
$w_{l,k}$	bandwidth that substrate link l allocates for virtual link in virtual network k
W_l^{max}	maximum bandwidth capacity of substrate link l
A_k	node-link incidence matrix of virtual network k
\mathcal{J}	set of links can possibly failed
A_k^j	node-link incidence matrix of virtual network k when link j fails
f_{k,l_k}	traffic flow on virtual link l_k in virtual network k in the normal state
f_{k,l_k}^j	traffic flow on virtual link l_k in virtual network k in failure state j
\mathbf{r}_k	traffic demand vector of virtual network k in the normal state
\mathbf{r}_k^j	traffic demand vector of virtual network k in the failure state j
$C_l(\cdot)$	operating cost function of substrate link l
Δ_k^j	amount of traffic reduction of SP k in link failure state j
Δ_k^{max}	maximum allowable traffic reduction of SP k
λ, μ	Lagrangian multipliers
ρ, γ	penalty parameters

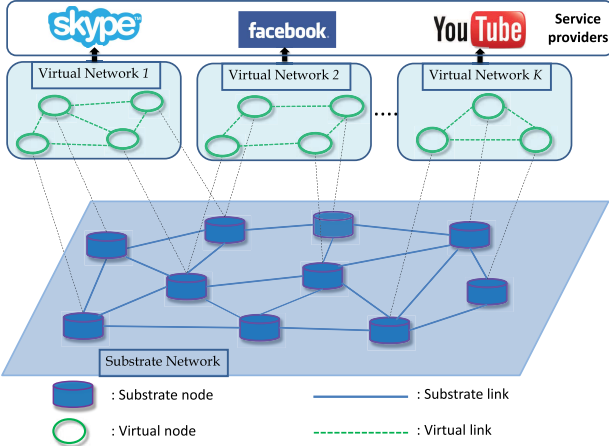


Fig. 1. The model of multiple virtual networks operate on top of a single substrate network.

Suppose that there is a set $\mathcal{K} \triangleq \{1, 2, \dots, K\}$ of SPs request K different virtual networks,¹ which is composed of a set of virtual nodes and virtual links, each established over the same physical network, denoted by a directed graph $G_k = (\mathcal{N}_k, \mathcal{L}_k)$. K virtual networks coexist and operate over the same physical network as illustrated in Fig. 1. In this paper, we assume the virtual network mapping result from each G_k to G_s is already known and focus on resource allocation for virtual networks.

Depending on the resource request from SPs, InPs will allocate bandwidth capacity of each substrate link $l \in \mathcal{L}_s$ to virtual links of SPs. For each substrate link $l \in \mathcal{L}_s$, let $w_{l,k}$ be the bandwidth that substrate link l allocates to virtual link of

¹Since each virtual network is operated by each SP, we use virtual network index and SP index interchangeably

virtual network k . Then, we have bandwidth allocation vector

$$\mathbf{w}_l \triangleq \{w_{l,1}, w_{l,2}, \dots, w_{l,K}\}. \quad (1)$$

For any virtual network k that does not have virtual link operates on top of the substrate link l , the bandwidth allocation must be equal to zero

$$w_{l,k} = 0, \quad l \notin \mathcal{L}_k. \quad (2)$$

The total bandwidth allocated to all virtual links must be less than the bandwidth capacity of the physical link, which can be expressed as the following constraint

$$\sum_{k=1}^K w_{l,k} \leq W_l^{max}, \quad (3)$$

where W_l is the maximum capacity of physical link l . Note that the capacity of each physical link will change over time, which depends on the power control and adaptive modulation scheme deployed at the physical layer. However, we consider capacity of physical links as the achievable average capacity, which is assumed to achieve through optimizing the physical layer. Therefore, the maximum capacity of each physical link changes slowly and is treated as a constant during the period of study.

B. Routing Model for Virtual Network

Each virtual network k , denoted by a directed graph $G_k = (\mathcal{N}_k, \mathcal{L}_k)$, has a collection of N_k virtual nodes that can send, receive, and relay data across virtual communication links. The network topology with respect to the interactions between virtual nodes and virtual links of virtual network k can be compactly represented by a node-link incidence matrix $A_k \in R^{N_k \times L_k}$. An entry $A_k[n_k][l_k]$ of the matrix A_k associated with node $n_k \in \mathcal{N}_k$ and link $l_k \in \mathcal{L}_k$, is given by [24]

$$A_k[n_k][l_k] = \begin{cases} 1 & \text{if node } n_k \text{ is the start node of link } l_k, \\ -1 & \text{if node } n_k \text{ is the end node of link } l_k, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

We consider a network flow model for routing data to a single destination in each virtual network. The data flows are assumed to be lossless in each virtual link and flow conservation law is assumed to be satisfied at each virtual node in each virtual network. In addition, a virtual source node may need a number of relay nodes to route the data stream to its destination node. We assume each SP uses multi-path routing protocol where the traffic from each virtual source node is split into several flows which follow different multi-hop paths to reach the desired destination.

We assume that d_k is the destination node of virtual network k . Each source node $n_k \in \mathcal{N}_k$ ($n_k \neq d_k$) generates data with an average rate of r_{n_k} to destination d_k . Then the total data rate at the destination node d_k is

$$r_{d_k} = - \sum_{n_k \neq d_k} r_{n_k}. \quad (5)$$

For each virtual network $k \in \mathcal{K}$, we also define a source-sink vector $\mathbf{r}_k \in R^{N_k}$ as

$$\mathbf{r}_k \triangleq \{r_1, r_2, \dots, r_{n_k}, \dots, r_{N_k}\}, \quad (6)$$

whose the n_k -th ($n_k \neq d_k$) entry r_{n_k} denotes the amount of data that virtual source node n_k injects into the network and destined for virtual destination nodes d_k , and the d_k -th entry is the total data rate at the destination node d_k , determined as in (5).

On each virtual link $l_k \in \mathcal{L}_k$ of virtual network k , we let $f_{k,l_k} \geq 0$ be the aggregate flow for destination node d_k . The aggregate on each link may come from different virtual source nodes under the multi-path routing model. At each virtual node $n_k \in \mathcal{N}_k$, the total flow going into a virtual node is the same as the total flow going out of that virtual node

$$\sum_{l_k \in O(n_k)} f_{k,l_k} - \sum_{l_k \in I(n_k)} f_{k,l_k} = r_{n_k}, \quad (7)$$

where $O(n_k)$ be the set of outgoing links of node n_k , and $I(n_k)$ be the set of incoming links to virtual node n_k . The compact expression for the flow conservation law across the whole virtual network k can be expressed as

$$A_k \mathbf{f}_k = \mathbf{r}_k, \quad (8)$$

where \mathbf{f}_k be the flow vector in virtual network k , which can be defined as

$$\mathbf{f}_k \triangleq \{f_{k,l_k}\}_{\forall l_k \in \mathcal{L}_k}. \quad (9)$$

The total amount of traffic on each virtual link must be no more than the bandwidth that substrate link allocates to virtual links in virtual network k

$$f_{k,l_k} \leq w_{l,k}, \quad \forall l \in \mathcal{L}_k. \quad (10)$$

III. JOINT RESOURCE AND ROUTING OPTIMIZATION FOR VIRTUAL WIRELESS NETWORKS

Based on the above definitions, a resource and routing optimization problem for multiple virtual networks operate on top of a single substrate network can be formulated. In the considered system, each InP has a cost function for operating substrate link $l \in \mathcal{L}_s$, which is assumed to be a strictly convex function on the total bandwidth, motivated by energy consumption cost [30]. Then the total cost for operating the substrate network is $\sum_{l=1}^{L_s} C_l(\mathbf{w}_l)$, where $C_l(\cdot)$ denotes the cost function for operating physical link l .

Given K virtual networks operate on top of the substrate network and fixed traffic demand from source nodes of each virtual network, the objective is to find an optimal bandwidth allocation for virtual links such that all traffic demand injected from source nodes is delivered to the desired destination in each virtual network with a minimum operation cost of the substrate network. The operation cost minimization problem can be formulated as

$$\min \sum_{l=1}^L C_l(\mathbf{w}_l) \quad (11)$$

$$\text{s.t. } A_k \mathbf{f}_k = \mathbf{r}_k, \quad \forall k \in \mathcal{K}, \quad (12)$$

$$f_{k,l_k} \leq w_{l,k}, \quad \forall l_k \in \mathcal{L}_k, \quad \forall k \in \mathcal{K}, \quad (13)$$

$$\sum_k w_{l,k} \leq W_l^{\max}, \quad \forall l \in \mathcal{L}_s, \quad (14)$$

variables: $\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l}$.

The constraints in (12) represent the flow conservation law for each virtual network. The constraints in (13) ensure the amount of traffic on each virtual link to be less than the bandwidth that substrate link allocates for that virtual link, and the constraints in (14) represent the maximum amount of bandwidth of each substrate link.

Remark 1: The optimization problem in (11)-(14) involves resource allocation and routing, which may be executed at different time scales. Traditionally, the resource allocation process needs to run at a fast time scale due to the rapid time-variation of wireless channels. The routing process, by contrast, may have a slower time scale due to the low dynamic of data traffic flows. However, the underlying channel capacity in this paper is treated as the achievable average capacity, which changes slowly and is assumed to be constant during the period of study. Therefore, the joint resource allocation and routing can be performed on the same time scale as in [31], [32].

The problem in (11)-(14) is convex and can be solved using the convex optimization techniques to obtain the optimal solution. The optimal solution will fully satisfy for all traffic demands to be delivered from sources to destinations in all virtual networks. However, when there is a substrate link fails, all traffic on virtual links that operate on top of that substrate link will be disrupted. This influences multiple virtual networks that have virtual links mapped on the failure link, and leads to discontinuation on service to end-users. In the next section, we will address how SPs can avoid this service discontinuation when substrate links fail.

IV. PREVENTIVE TRAFFIC DISRUPTION WITH LINK FAILURES

The bandwidth allocation model in previous section can satisfy traffic demands for all virtual networks only when all substrate links are fully available, which we will refer as the normal state. However, unpredictable wireless network events such as link failures may occur anytime. Although when a link failure event happens, the network controller can reformulate the problem in (11)-(14) with new system parameters to reallocate bandwidth for all virtual networks, it will take a certain amount of time to wait for network re-convergence. Since different SPs target different types of services and may have stringent reliability and QoS requirements, this performance degradation and severe discontinuation will be intolerable to end-users. Therefore, in this section, we propose a preventive traffic disruption model for virtual networks to provide nonstop reliable services to end-users. Particularly, the system allows virtual networks to continue operation, possibly at an allowable reduced performance level, rather than failing completely, when some part of substrate links fail.

Let \mathcal{J} be a set of substrate links may possibly be failed. When a substrate link in the substrate network fails, all virtual links that operate on top of that substrate link are

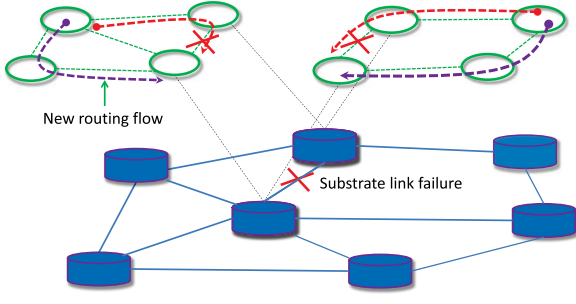


Fig. 2. The model of substrate link failure.

no longer available. Consequentially, the physical topology of each virtual network also changes. SPs have to find another routing path to carry data traffic across virtual networks as illustrated in Fig. 2. However, SPs may not be able to fully satisfy all data demand as in the normal state. Therefore, SPs may reduce a fraction amount of traffic demand.

Let A_k^j be the node-link incidence matrix of virtual network k when substrate link j fails.² In this paper, we consider a set of $J \triangleq |J|$ substrate links can be possibly failed. However, we assume that only one substrate link fails at a time. Multiple-link failure scenarios can be applied without changing the structure of this paper by constructing all corresponding constraints associated with multiple-link failure scenarios and incorporating into the original problem. The only difference is the node-link incidence matrix will be more sparse, which makes the routing solution has limited routing paths to carry data traffic from the source to the destination, and consequently may lead to the larger amount of traffic disruption. We define \mathbf{r}_k^j be the new traffic demand vector that SP k can support when substrate link j fails. Similar to the normal state, the new traffic demand must satisfy flow conservation law in all virtual networks as

$$A_k^j \mathbf{f}_k^j = \mathbf{r}_k^j, \quad \forall k \in \mathcal{X}. \quad (15)$$

Moreover, the new traffic flow across each virtual network must satisfy virtual link capacity in the normal state, i.e.,

$$f_{k,l_k}^j \leq w_{l,k}, \quad \forall l_k \in \mathcal{L}_k, \quad \forall k \in \mathcal{X}. \quad (16)$$

Note that, in constraints (15) and (16), only traffic demand and flow across virtual links change. The bandwidth that substrate links allocate to virtual links does not change since when a substrate link fails, SPs still use the existing available resource that has been already allocated to virtual networks in the normal state to continue operation in failure states. Since SPs have to decrease demand to satisfy with the current available resource, we can calculate the amount of traffic reduction at each virtual network $k \in \mathcal{X}$ when substrate link j fails as

$$\Delta_k^j = \sum_{n_k \neq d_k} (r_{n_k} - r_{n_k}^j). \quad (17)$$

²We use superscript j to denote all variables and system parameters associated with link j failure event, which we also refer as link failure state j .

We further define the traffic reduction vector of all virtual networks when substrate link j fails as

$$\Delta^j \triangleq \{\Delta_1^j, \dots, \Delta_k^j, \dots, \Delta_K^j\}. \quad (18)$$

Since SPs target different services and typically have different requirements for traffic satisfaction when a substrate link fails, each SP has a predefined maximum threshold for data reduction. Let Δ_k^{max} be the maximum allowable traffic reduction of SP k . Then we have a constraint for SP k when substrate link j fails

$$\Delta_k^j \leq \Delta_k^{max}. \quad (19)$$

We can express the above constraint for all SPs compactly in vector-form as

$$\Delta^j \leq \Delta^{max}. \quad (20)$$

The objective is to find the optimal resource allocation that can minimize the cost of operating substrate network in the normal state. Moreover, when any substrate link fails, we also want to guarantee for minimal amount of traffic disruption in virtual networks. The optimization problem for the preventive traffic disruption model can be formulated as

$$\min \sum_{l=1}^{L_s} C_l(\mathbf{w}_l) + \tau \sum_{j \in \mathcal{J}} \|\Delta^j\|_0 \quad (21)$$

$$\text{s.t. } A_k \mathbf{f}_k = \mathbf{r}_k, \quad \forall k, \quad (22)$$

$$f_{k,l_k} \leq w_{l,k}, \quad \forall l_k \in \mathcal{L}_k, \quad \forall k \in \mathcal{X}, \quad (23)$$

$$\sum_k w_{l,k} \leq W_l^{max}, \quad \forall l \in \mathcal{L}_s, \quad (24)$$

$$A_k^j \mathbf{f}_k^j = \mathbf{r}_k^j, \quad \forall k \in \mathcal{X}, \quad \forall j \in \mathcal{J}, \quad (25)$$

$$f_{k,l_k}^j \leq w_{l,k}, \quad \forall l_k \in \mathcal{L}_k, \quad \forall k \in \mathcal{X}, \quad \forall j \in \mathcal{J}, \quad (26)$$

$$\Delta_k^j = \sum_{n_k \neq d_k} (r_{n_k} - r_{n_k}^j), \quad \forall k \in \mathcal{X}, \quad \forall j \in \mathcal{J}, \quad (27)$$

$$\Delta_k^j \leq \Delta_k^{max}, \quad \forall k \in \mathcal{X}, \quad \forall j \in \mathcal{J}, \quad (28)$$

where τ is a positive parameter to capture the trade-off between cost minimization and minimal traffic disruption. The second term in (21), $\|\Delta^j\|_0$, is ℓ_0 -norm of vector Δ^j , which determines the number of nonzero entries in Δ^j

$$\|\Delta^j\|_0 \stackrel{\text{def}}{=} \#\{k : \Delta_k^j \neq 0\}. \quad (29)$$

Incorporating ℓ_0 -norm $\|\Delta^j\|_0$ into the objective function allows us to obtain the optimal solution that leads to a minimal number of SPs have to reduce traffic in link failure states. Constraints (22), (23), and (24) are for the normal state, while constraints (25), (26), (27), and (28) are for all link failure states.

Due to the non-convexity of $\|\Delta^j\|_0$, exactly solving the problem in (21) is computationally difficult. To avoid computational burdens, we use ℓ_1 -norm, which is the convex approximation to ℓ_0 -norm

$$\|\Delta^j\|_1 \stackrel{\text{def}}{=} \sum_k |\Delta_k^j|. \quad (30)$$

The problem in (21)-(27) can be reformulated as

$$\begin{aligned} \min \quad & \sum_{l=1}^{L_s} C_l(\mathbf{w}_l) + \tau \sum_{j \in \mathcal{J}} \|\Delta^j\|_1 \\ \text{s.t.} \quad & (22), (23), (24), (25), (26), (27), (28). \end{aligned} \quad (31)$$

The problem in (31) is convex and can be solved using several convex optimization techniques. However, directly solving (31) can be intractable due to the large-scale nature of the original problem.

Remark 2: By incorporating the preventive traffic disruption constraint into the resource allocation and routing problem in (31), the optimal solution provides a proactive mechanism to cope with physical network failures, i.e., reserves the backup resource before any failure happens. Specifically, InPs will allocate a certain amount of redundant bandwidth for virtual links so that when physical links fail, SPs still can use the redundant bandwidth from the other virtual links to flow the data traffic.

V. DECOMPOSITION ALGORITHMS USING ADMM

In this section, we propose two algorithms to solve the centralized problem in (31) using the ADMM-based decomposition technique. The first algorithm provides a parallel computational framework, and the second one can be implemented in a distributed fashion at each SP and each substrate link.

A. Parallel Algorithm Using ADMM

In this subsection, we use the ADMM decomposition method to propose a parallel algorithm. This algorithm decomposes the original problem in (31) into a master problem corresponding to the normal state, and J problems corresponding to J link failure states, which can be solved concurrently at different computing facilities.

The problem in (31) contains a large number of constraints. However, we realize that almost all the constraints are separable into the normal state and different link failure states. Particularly, constraints (22), (23), and (24) are associated with the normal state only, while constraints (25), (26), (27), and (28) can be separated into each link failure state, except for variables $\{w_{l,k}\}_{\forall l, \forall k}$ belong to the normal state. In order to make constraints (26) to be separable from the normal state, we define auxiliary variables

$$\mathbf{w}_l = \mathbf{w}_l^j, \quad \forall l \in \mathcal{L}_s, \quad \forall j \in \mathcal{J}, \quad (32)$$

where each auxiliary variable \mathbf{w}_l^j can be interpreted as the local copies of \mathbf{w}_l in the normal state at each link j failure state. We now can rewrite constraint (26) with respect to only local variables at each link failure state as

$$f_{k,l}^j \leq w_{l,k}^j, \quad \forall l_k \in \mathcal{L}_k, \quad \forall k. \quad (33)$$

The link failure state constraints (25), (27), (28), and (33) are now decoupled from the normal state. To facilitate for presentation, we further define the feasible set for the normal state, \mathcal{F}^0 , and each link failure state, \mathcal{F}^j as

$$\mathcal{F}^0 = \{(\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l}) | (22), (23), (24)\},$$

$$\mathcal{F}^j = \{(\{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall k}, \{\mathbf{w}_l^j\}_{\forall l}) | (25), (27), (28), (33)\}, \quad \forall j \in \mathcal{J}.$$

Then the problem in (31) can be rewritten as

$$\begin{aligned} \min \quad & \sum_{l=1}^{L_s} C_l(\mathbf{w}_l) + \tau \sum_{j \in \mathcal{J}} \|\Delta^j\|_1 \\ \text{s.t.} \quad & (\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l}) \in \mathcal{F}^0, \\ & (\{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall k}, \{\mathbf{w}_l^j\}_{\forall l}) \in \mathcal{F}^j, \quad \forall j \in \mathcal{J}, \\ & \mathbf{w}_l = \mathbf{w}_l^j, \quad \forall l \in \mathcal{L}_s, \quad \forall j \in \mathcal{J}. \end{aligned} \quad (34)$$

The set of equality constraints in (35) represents the consensus constraints, i.e., it enforces the local copies of bandwidth allocation variables to be agreement with the corresponding variables in the normal state. The augmented Lagrangian function of problem in (34) with respect to consensus constraints in (35) is given by

$$\begin{aligned} \mathcal{L}_1 &= \sum_{l=1}^{L_s} C_l(\mathbf{w}_l) + \tau \sum_{j=1}^J \sum_{k=1}^K \Delta_k^j \\ &+ \sum_{j=1}^J \sum_{l=1}^{L_s} (\lambda_l^j)^T (\mathbf{w}_l^j - \mathbf{w}_l) + \frac{\rho}{2} \sum_{j=1}^J \sum_{l=1}^{L_s} \|\mathbf{w}_l^j - \mathbf{w}_l\|^2 \\ &= \sum_{l=1}^{L_s} C_l(\mathbf{w}_l) - \sum_{j=1}^J \sum_{l=1}^{L_s} (\lambda_l^j)^T \mathbf{w}_l \\ &+ \sum_{j=1}^J \left[\tau \sum_{k=1}^K \Delta_k^j + \sum_{l=1}^{L_s} (\lambda_l^j)^T \mathbf{w}_l^j + \frac{\rho}{2} \sum_{l=1}^{L_s} \|\mathbf{w}_l^j - \mathbf{w}_l\|^2 \right], \end{aligned} \quad (36)$$

where $\{\lambda_l^j\}_{\forall l, \forall j}$ is the Lagrangian multiplier, and $\rho > 0$ is a penalty parameter.

Defining the primal variables $\mathbf{x} = (\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l})$, which is the decision variable vector in the normal state, and $\mathbf{z}^j = (\{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall k}, \{\mathbf{w}_l^j\}_{\forall l})$ is the decision variable vector in link j failure state. Then the ADMM decomposition technique can be used to solve the problem in (34) in an iterative procedure. Specifically, at the t -th iteration, the primal variables and dual variables are updated sequentially as

$$\mathbf{x}[t+1] = \arg \min \mathcal{L}_1(\mathbf{x}, \mathbf{z}[t], \boldsymbol{\lambda}[t]), \quad (37)$$

$$\mathbf{z}^j[t+1] = \arg \min \mathcal{L}_1(\mathbf{x}[t+1], \mathbf{z}^j, \boldsymbol{\lambda}^j[t]), \quad \forall j, \quad (38)$$

$$\lambda_l^j[t+1] = \lambda_l^j[t] + \rho (\mathbf{w}_l^j[t+1] - \mathbf{w}_l[t+1]), \quad \forall l, \quad \forall j. \quad (39)$$

Based on the Lagrangian function (36), we decompose the problem in (34) into the following $J+1$ optimization problems. The first sub-problem is associated with variables in the normal state only and corresponding to the primal variables update in (37)

$$\begin{aligned} \min \quad & \sum_{l=1}^{L_s} C_l(\mathbf{w}_l) - \sum_{j=1}^J \sum_{l=1}^{L_s} (\lambda_l^j)^T \mathbf{w}_l + \frac{\rho}{2} \sum_{j=1}^J \sum_{l=1}^{L_s} \|\mathbf{w}_l^j - \mathbf{w}_l\|^2 \\ \text{s.t.} \quad & (\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l}) \in \mathcal{F}^0. \end{aligned} \quad (40)$$

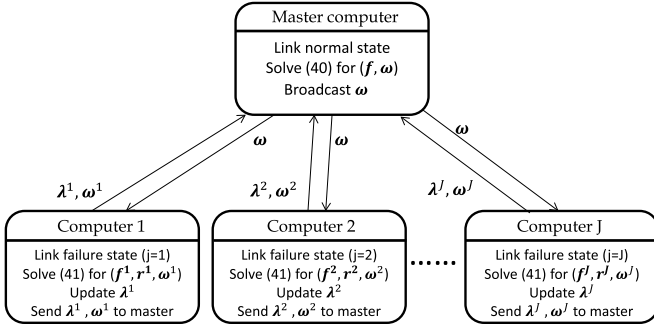


Fig. 3. The illustration of information exchange between the normal state sub-problem and link failure state sub-problems.

By fixing the values of $\{\mathbf{w}_l^j, \boldsymbol{\lambda}_l^j\}$, and then solving the problem in (40), we obtain the optimal solution for $(\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l})$.

The remaining J sub-problems are associated with variables in link failure states and corresponding to primal variables update in (38). For each link $j \in \mathcal{J}$ failure state, we decompose into the following problem

$$\begin{aligned} \min \quad & \tau \sum_{k=1}^K \Delta_k^j + \sum_{l=1}^{L_s} (\boldsymbol{\lambda}_l^j)^T \mathbf{w}_l^j + \frac{\rho}{2} \sum_{l=1}^{L_s} \|\mathbf{w}_l^j - \mathbf{w}_l\|^2 \\ \text{s.t.} \quad & (\{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall k}, \{\mathbf{w}_l^j\}_{\forall l}) \in \mathcal{F}^j. \end{aligned} \quad (41)$$

By fixing $(\mathbf{w}_l, \boldsymbol{\lambda}_l^j)_{\forall l}$ and then solving the problem in (41), we obtain the optimal solution for $(\{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall k}, \{\mathbf{w}_l^j\}_{\forall l})$.

Algorithm Implementation: The whole procedure for parallel algorithm using ADMM is described in Algorithm 1. First, a master computing node solves the optimization problem in (40) to obtain the optimal solution $(\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l})$. Then it will broadcast the bandwidth allocation solution of the whole network $\{\mathbf{w}_l\}_{\forall l}$ to all J distributed computing nodes. Each distributed computing node solves the optimization problem in (41) to obtain the optimal solution $(\{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall k}, \{\mathbf{w}_l^j\}_{\forall l})$. Finally, based on the local value of $\{\mathbf{w}_l^j\}_{\forall l}$ and $\{\mathbf{w}_l\}_{\forall l}$ received from the master node, the dual variables can be updated as in line 16 in Algorithm 1. Note that, J optimization problems associated with link failure states are decoupled and can be solved in a parallel fashion at different computing nodes without affecting the others. This parallel implementation reduces the computation time for the proposed Algorithm 1.

The amount of information exchange between the master node and distributed computing nodes is depicted in Fig. 3. The master node broadcasts the bandwidth allocation solution in the normal state, which is same to all distributed nodes. Each distributed node needs to send the local information including dual variable $\{\boldsymbol{\lambda}_l^j\}_{\forall l}$ and $\{\mathbf{w}_l^j\}_{\forall l}$ to the master node. This can be done by using Message Passing Interface (MPI), which is widely used for high-performance computing paradigm [33]. Note that in Algorithm 1, all SPs and InPs need to exchange information with the master computing node to perform computation in each sub-problem. This can be done via certain entities that are proposed to operate the wireless network virtualization such as mobile virtual network operators (MVNOs), who lease the network resources from

Algorithm 1 Parallel Algorithm Based on ADMM Decomposition

- 1: Initialize: $t = 1, \{\boldsymbol{\lambda}_l^j\}_{\forall l, \forall j} = 0$
 - 2: **repeat**
 - 3: **At master computing node:**
 - 4: **repeat**
 - 5: wait
 - 6: **until** receive updates $\mathbf{w}_l^j, \boldsymbol{\lambda}_l^j$ from all J distributed computing nodes
 - 7: step 1: solve (40) for optimal solution $(\{\mathbf{f}_k\}_{\forall k}, \{\mathbf{w}_l\}_{\forall l})$
 - 8: step 2: broadcast $\{\mathbf{w}_l\}_{\forall l}$ to all J distributed computing nodes
 - 9: step 3: $t \leftarrow t + 1$
 - 10:
 - 11: **At each distributed computing node:**
 - 12: **repeat**
 - 13: wait
 - 14: **until** receive the update $\{\mathbf{w}_l\}_{\forall l}$ from the master
 - 15: step 1: solve (41) for optimal solution $(\{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall k}, \{\mathbf{w}_l^j\}_{\forall l})$
 - 16: step 2: update dual variables:

$$\boldsymbol{\lambda}_l^j[t+1] = \boldsymbol{\lambda}_l^j[t] + \rho (\mathbf{w}_l^j[t+1] - \mathbf{w}_l[t+1]), \forall l$$
 - 17: step 3: send $(\boldsymbol{\lambda}_l^j, \mathbf{w}_l^j)_{\forall l}$ to the master
 - 18: **until** a stopping criterion is met
-

InPs and create virtual resources based on the requests from SPs [5].

B. Distributed Algorithm Using ADMM

Even Algorithm 1 decomposes the original problem in (31) into $J + 1$ sub-problems and provides a parallel computational framework, it requires to have a central controller to collect all information of the whole network to solve each sub-problem. However, in practice, it is hard to be fulfilled due to enormous amount of signaling. Therefore, in this subsection, we propose a fully decentralized algorithm for solving the problem in (31), in which each SP and each substrate link independently solve their own problems to obtain the optimal solution, while requires a limited amount of information exchange between SPs and substrate links.

The objective function in (31) involves operation cost of substrate links and amount of traffic reduction of SPs. However, we can separate into the individual cost for each substrate link as well as the amount of traffic reduction of each SP. Besides, constraints (22), (23), (25), (26), (27), and (28) are separable into each SP. The only constraints in (24) are coupled between substrate links and SPs. In order to make the constraints in (24) are decoupled between individual substrate link and SPs, we consider $w_{l,k}$ as the local variables at SP k and define auxiliary variables as

$$\mathbf{w}_l = \mathbf{w}_l^s, \quad \forall l \in \mathcal{L}_s. \quad (42)$$

Each $w_{l,k}$ has a local copy $w_{l,k}^s$ at the substrate link, as illustrated in Fig. 4. We can interpret $w_{l,k}$ as the bandwidth

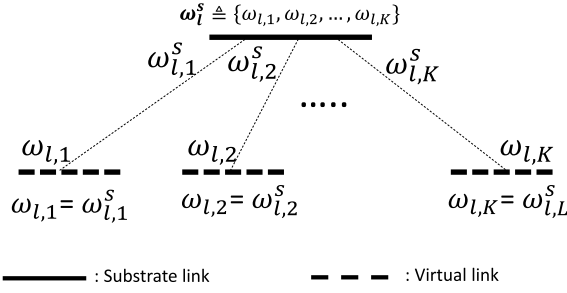


Fig. 4. The structure of auxiliary variables define for each substrate link and corresponding virtual links operate on top of the substrate link.

that virtual link requests to satisfy its service, while $w_{l,k}^s$ as the true bandwidth that the substrate link can allocate to virtual link. Then, by adding consensus constraints, the requested bandwidth and allocated bandwidth reach an agreement.

The problem in (31) can be rewritten as

$$\min \sum_{l=1}^{L_s} C_l(\mathbf{w}_l^s) + \tau \sum_{k \in \mathcal{X}} \sum_{j \in \mathcal{J}} \Delta_k^j \quad (43)$$

$$[-1pt] \text{ s.t. } A_k \mathbf{f}_k = \mathbf{r}_k, \quad \forall k, \quad (44)$$

$$f_{k,l} \leq w_{l,k}, \quad \forall l \in \mathcal{L}_k, \quad \forall k, \quad (45)$$

$$A_k^j \mathbf{f}_k^j = \mathbf{r}_k^j, \quad \forall j, \quad \forall k \quad (46)$$

$$f_{k,l}^j \leq w_{l,k}, \quad \forall l \in \mathcal{L}_k, \quad \forall j, \quad \forall k, \quad (47)$$

$$\Delta_k^j = \sum_{n_k \neq d_k} (r_{n_k} - r_{n_k}^j), \quad \forall j, \quad \forall k, \quad (48)$$

$$\Delta_k^j \leq \Delta_k^{\max}, \quad \forall k, \quad \forall j, \quad (49)$$

$$\mathbf{w}_l^s = \mathbf{w}_l, \quad \forall l \in \mathcal{L}_s, \quad (50)$$

$$\sum_{k=1}^K w_{l,k}^s \leq W_l^{\max}, \quad \forall l \in \mathcal{L}_s. \quad (51)$$

The cost function in (43) is substituted by local variables of each substrate link. The equality constraints in (50) represent the consensus constraints. Constraints (51) is maximum bandwidth capacity constraints with local variables at each substrate link. We further define feasible set for each SP as

$$\mathcal{F}_k = \{(\mathbf{f}_k, \{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall j}, \{w_{l,k}\}_{\forall l}) | (44) - (49)\}, \quad \forall k \in \mathcal{X}.$$

Then, the problem in (43)-(51) can be compactly expressed as

$$\begin{aligned} \min \quad & \sum_{l=1}^{L_s} C_l(\mathbf{w}_l^s) + \tau \sum_{k \in \mathcal{X}} \sum_{j \in \mathcal{J}} \Delta_k^j \\ \text{s.t.} \quad & (\mathbf{f}_k, \{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall j}, \{w_{l,k}\}_{\forall l}) \in \mathcal{F}_k, \quad \forall k, \\ & \mathbf{w}_l^s = \mathbf{w}_l, \quad \forall l \in \mathcal{L}_s, \\ & \sum_{k=1}^K w_{l,k}^s \leq W_l^{\max}, \quad \forall l \in \mathcal{L}_s. \end{aligned} \quad (52)$$

The Lagrangian function of the problem in (52) with respect to consensus constraints is given by

$$\begin{aligned} \mathcal{L}_2 &= \sum_{l=1}^{L_s} C_l(\mathbf{w}_l^s) + \tau \sum_{k=1}^K \sum_{j=1}^J \Delta_k^j \\ &+ \sum_{l=1}^{L_s} (\boldsymbol{\mu}_l)^T (\mathbf{w}_l^s - \mathbf{w}_l) + \frac{\gamma}{2} \sum_{l=1}^{L_s} \|\mathbf{w}_l^s - \mathbf{w}_l\|^2 \\ &= \sum_{l=1}^{L_s} \left\{ C_l(\mathbf{w}_l^s) + (\boldsymbol{\mu}_l)^T \mathbf{w}_l^s \right\} \\ &+ \tau \sum_{k=1}^K \sum_{j=1}^J \Delta_k^j - \sum_{l=1}^{L_s} (\boldsymbol{\mu}_l)^T \mathbf{w}_l + \frac{\gamma}{2} \sum_{l=1}^{L_s} \|\mathbf{w}_l^s - \mathbf{w}_l\|^2, \end{aligned} \quad (53)$$

where $\{\boldsymbol{\mu}_l\}_{\forall l}$ is the Lagrangian multiplier, and $\gamma > 0$ is a penalty parameter.

Defining primal variable $\mathbf{u} = (\mathbf{f}_k, \{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall j}, \{w_{l,k}\}_{\forall l})_{\forall k}$, which is the decision variable vector for SPs, and $\mathbf{v} = \{\mathbf{w}_l\}_{\forall l}$, which is the decision variable vector for substrate links. Then the ADMM decomposition technique can be applied to solve the problem in (52) in an iterative procedure. Specifically, at the t -th iteration, the primal variables and dual variables are updated sequentially as

$$\mathbf{u}[t+1] = \arg \min \mathcal{L}_2(\mathbf{u}, \mathbf{v}[t], \boldsymbol{\mu}[t]), \quad (54)$$

$$\mathbf{v}[t+1] = \arg \min \mathcal{L}_2(\mathbf{u}[t+1], \mathbf{v}, \boldsymbol{\mu}[t]), \quad (55)$$

$$\boldsymbol{\mu}_l[t+1] = \boldsymbol{\mu}_l[t] + \gamma (\mathbf{w}_l^s[t+1] - \mathbf{w}_l[t+1]), \quad \forall l. \quad (56)$$

From the Lagrangian function in (53), we decompose the problem in (52) into the service provider-level problem and substrate link-level problem. The service provider-level problem is associated with primal variable update in (54) and can be expressed as

$$\begin{aligned} \min \quad & \tau \sum_{k=1}^K \sum_{j=1}^J \Delta_k^j - \sum_{l=1}^{L_s} (\boldsymbol{\mu}_l)^T \mathbf{w}_l + \frac{\gamma}{2} \sum_{l=1}^{L_s} \|\mathbf{w}_l^s - \mathbf{w}_l\|^2 \\ \text{s.t.} \quad & (\mathbf{f}_k, \{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall j}, \{w_{l,k}\}_{\forall l}) \in \mathcal{F}_k, \quad \forall k. \end{aligned} \quad (57)$$

The substrate link-level problem is associated with primal variable update in (55) and can be expressed as

$$\begin{aligned} \min \quad & \sum_{l=1}^{L_s} \left\{ C_l(\mathbf{w}_l^s) + (\boldsymbol{\mu}_l)^T \mathbf{w}_l^s + \frac{\gamma}{2} \|\mathbf{w}_l^s - \mathbf{w}_l\|^2 \right\} \\ \text{s.t.} \quad & \sum_{k=1}^K w_{l,k}^s \leq W_l^{\max}, \quad \forall l. \end{aligned} \quad (58)$$

After solving (57) and (58), the dual variable is update as in (56). Moreover, we realize that the problem in (57) is completely separable into each SP and can be solved by each individual SP. Each SP $k \in \mathcal{X}$ solves it own problem as follow

$$\begin{aligned} \min \quad & \tau \sum_{j=1}^J \Delta_k^j - \sum_{l=1}^{L_s} \mu_{l,k} w_{l,k} + \frac{\gamma}{2} \sum_{l=1}^{L_s} (w_{l,k}^s - w_{l,k})^2 \\ \text{s.t.} \quad & (\mathbf{f}_k, \{\mathbf{f}_k^j, \mathbf{r}_k^j\}_{\forall j}, \{w_{l,k}\}_{\forall l}) \in \mathcal{F}_k. \end{aligned} \quad (59)$$

Algorithm 2 Distributed Algorithm Based on ADMM Decomposition

```

1: initialize:  $t = 1, \boldsymbol{\mu}_l = 0, \forall l$ 
2: repeat
3:   At each virtual network:
4:   repeat
5:     wait
6:   until receive updates  $w_{l,k}^s, \mu_{l,k}$  from all substrate links
7:     1) solve (59)
8:     2) broadcast  $w_{l,k}$  to substrate link
9:   

---


10:  At each substrate link:
11:  repeat
12:    wait
13:  until receive the update  $w_{l,k}$  from all SPs
14:    1) solve (60) for optimal solution  $\boldsymbol{w}_l^s$ 
15:    2) update dual variables:
16:      
$$\boldsymbol{\mu}_l[t + 1] = \boldsymbol{\mu}_l[t] + \gamma (\boldsymbol{w}_l^s[t + 1] - \boldsymbol{w}_l[t + 1])$$

17:    

---


18:     $t \leftarrow t + 1$ 
19:  until a stopping criterion is met
  
```

Similarly, the problem in (58) can be decomposed to solve at each substrate link as

$$\begin{aligned}
 & \min C_l(\boldsymbol{w}_l^s) + (\boldsymbol{\mu}_l)^T \boldsymbol{w}_l^s + \frac{\gamma}{2} \|\boldsymbol{w}_l^s - \boldsymbol{w}_l\|^2 \\
 & \text{s.t. } \sum_{k=1}^K w_{l,k}^s \leq W_l^{\max}. \quad (60)
 \end{aligned}$$

Algorithm Implementation: We propose a distributed algorithm to solve the problem in (31) as in Algorithm 2. The decomposition structure of the solution process is clearly visible: SPs and substrate links perform optimization independently. Particularly, SPs solve optimization problem in (59) simultaneously and send the updated value of $w_{l,k}$ to the corresponding substrate link. Each substrate link will solve its local optimization problem in (60) after receiving all information from SPs, and then updates dual variable $\boldsymbol{\mu}_l$. The new values of $\{\boldsymbol{w}_l^s, \boldsymbol{\mu}_l\}$ will be sent to SPs. This process is repeated until convergence. Note that, each substrate link sends exchange information to only SPs who have virtual links operate on top of it. The distributed nature of Algorithm 2 in which substrate links and SPs are completely decoupled facilitates for the capability of adapting the dynamic behavior of wireless links. When physical link capacity varies, each substrate link updates the constraint in sub-problem (60) and performs the calculation independently. SPs continue executing the computation in (59) without any modification.

VI. SIMULATION RESULTS

In this section, we use the computational experiment to evaluate the performance of our proposed algorithms. We generate a random substrate network topology comprising 20 physical nodes and 70 physical links. The bandwidth

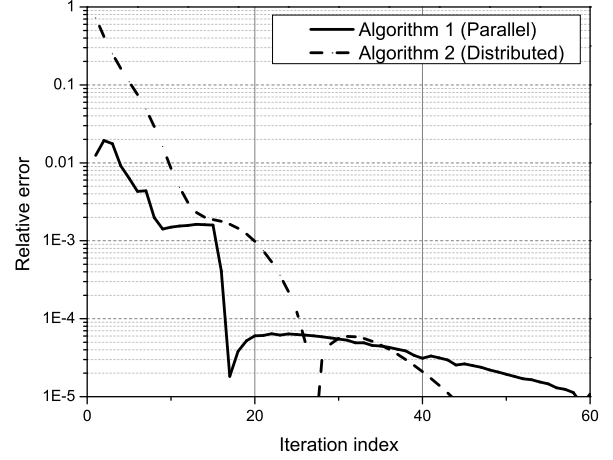


Fig. 5. The convergence performance of the proposed algorithms.

capacity of substrate links are generated randomly with a uniform distribution from [50, 100] Mb/s. We also deploy 10 virtual networks on top of the substrate network, each virtual network has the random number of nodes from [5, 10], and the random number of links from [10, 15]. We select one node as a destination for each virtual network, and source nodes inject data with an average rate randomly generated from [10, 15] Mb/s. We assume the convex operating cost function of each substrate link is $C_l(\boldsymbol{w}_l) = a_l (\sum_k w_{l,k})^2$, where a_l is generated randomly for each substrate link from a uniform distribution $a_l \in [0.002, 0.004]$ \$/Mb². The link failure set \mathcal{J} is selected randomly from 10% of the total substrate links and $\tau = 0.3$ in all simulations, unless otherwise stated. Virtual networks are allowed to reduce half of the traffic data when substrate links fail. All tests are conducted on a Windows 7 64-bit personal computer with Intel i7-4770 3.4 GHz CPU and 16GB of RAM. Each sub-problem in our proposed algorithms is solved using CVX [34].

A. Convergence and Computational Performance

We show the convergence behavior of our proposed algorithms in Fig. 5. Since both algorithms are implemented and performed on a single computer, the delay time due to local information exchange can be ignored. We plot the relative error of the objective function versus the number of iterations without applying any specific termination condition for the proposed algorithms. For parallel Algorithm 1, the relative error approaches to 10^{-3} in about 15 iterations, while distributed Algorithm 2 needs about 20 iterations to yield the same relative error. The faster convergence behavior of Algorithm 1 is due to the smaller number of sub-problems compared to those of distributed Algorithm 2, and consequently reach to an agreement at a faster speed. Specifically, in this simulation, Algorithm 1 produces a master problem and 7 parallel problems corresponding to 7 link failure states, while for Algorithm 2, the number of decomposed problems is 80 (10 problems for SPs and 70 problems for substrate links).

A larger set of link failure \mathcal{J} leads to a greater number of constraints in the optimization problem, and generally increases the computation time of two algorithms. Therefore, we investigate the computation time of the proposed

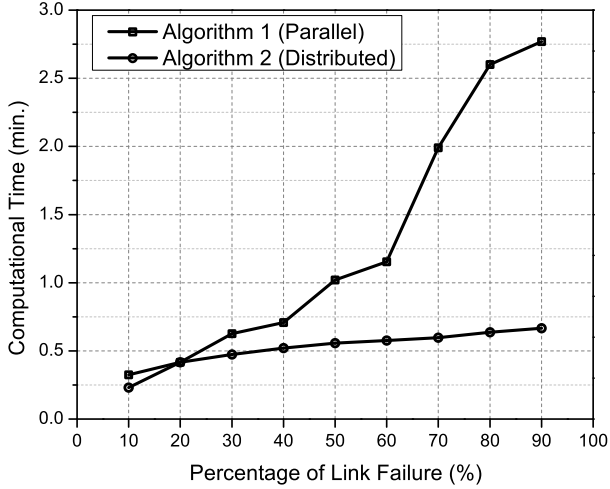


Fig. 6. Computation time of two proposed algorithms versus percentage of link failure.

algorithms by varying the percentage of link failure. For each percentage of link failure, we select the set of substrate links can possibly failed randomly and run 100 realizations to average out the result. In Fig. 6, we plot the computation time of two algorithms to reach the relative error $< 10^{-2}$. Since Algorithm 1 is proposed to be implemented in a parallel fashion, the total computation time of each iteration is determined as the total computation time in the normal state sub-problem and the maximum amount of computation time among J different link failure sub-problems. For Algorithm 2, the computation time of each iteration is the maximum amount of computational time among SPs and the maximum computation time among substrate links. We ignore the delay time due to information exchange. From the result in Fig. 6, a higher percentage of link failure increases the computational time of both algorithms, especially in Algorithm 1. This due to the fact that a larger percentage of link failure does not increase the number of consensus constraints in Algorithm 2; while in Algorithm 1, the number of consensus variables increases significantly. Therefore, much more iterations are required to reach the global solution.

We further plot the computation time for different layers of decomposition structure in two algorithms versus the percentage of link failure. In Fig. 7, we show the average computation time of the master node, who solves the normal state problem, and average computation time of distributed computing nodes, who solve the link failure state problems in parallel fashion. As it can be seen in Fig. 7, the computation time of the master node increases as the percentage of link failure increases, while the computation time at each distributed computing node is not affected. This can be explained by the decomposition structure in Algorithm 1, when the number of link failure states increase, the number of consensus constraints in (35) increase accordingly. This leads to a higher complexity of the normal state problem in (40), and a larger number of sub-problems in (41). However, J sub-problems in (41) are performed in parallel fashion by distributed computing nodes, the complexity does not grow.

A similar behavior can be observed for Algorithm 2 in Fig. 8, we can see that the computation time at each substrate

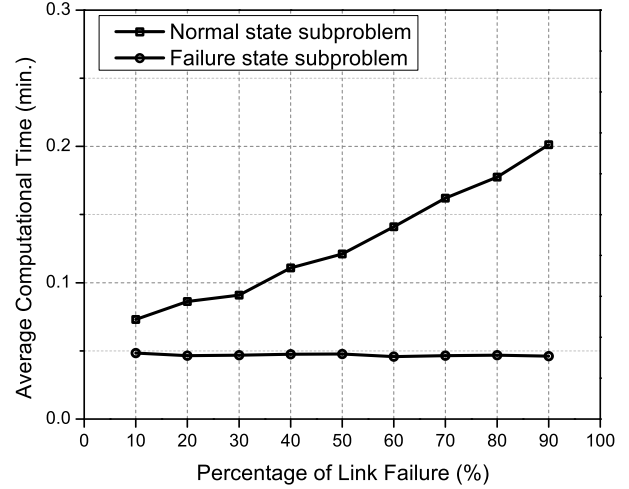


Fig. 7. Average computation time of the normal state sub-problem and link failure state sub-problem in Algorithm 1.

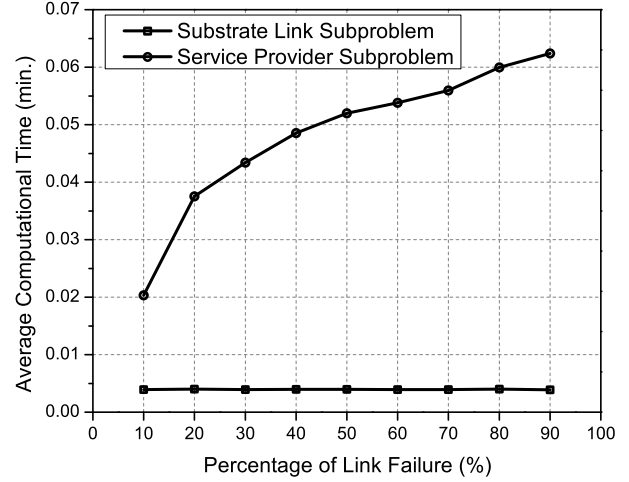


Fig. 8. Average computation time of each substrate link and each service provider in Algorithm 2.

link is not affected by the percentage of link failure, while the computation time for each SP increases accordingly. This can be explained by the independence of the substrate link-level problem in (60) to link failure states. Particularly, the decomposition structure in Algorithm 2 produces the optimization problem for each substrate link in (60) whose complexity does not depend on the percentage of link failure. In contrast, the optimization problem that each SP performs in (59) grows the complexity as the percentage of link failure increases. Note that, this property can facilitate for the practical implementation of Algorithm 2 since each substrate link normally has less computation capability, while SPs with better computation capability can perform higher complexity computing tasks.

B. System Performance

We evaluate the efficacy of our model in guaranteeing the minimal amount of traffic reduction when substrate links fail by plotting the percentage of traffic reduction of individual SP in Fig. 9. We compare the result with and without incorporating preventive traffic disruption model into the resource

TABLE II
OPERATION COST AND BANDWIDTH UTILIZATION COMPARISON

	Problem in (11)	Problem in (31)
Bandwidth (Mb/s)	1504	1513
Cost (\$/s)	223.8	230.9

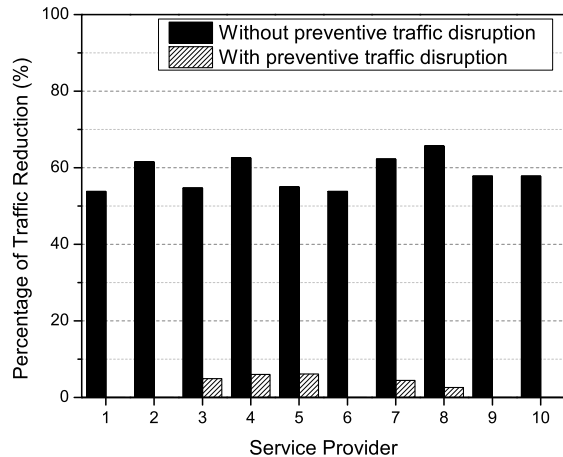


Fig. 9. The comparison of percentage of traffic reduction at individual service provider when substrate links fail with and without incorporating preventive traffic disruption model.

allocation problem. We realize that incorporating preventive traffic disruption model can significantly reduce the amount of traffic reduction. Moreover, the result also indicates that ℓ_1 -norm in the objective function ensures a sparse number of SPs have to reduce traffic in link failure states. Particularly, only SPs 3, 4, 5, 7, and 8 have to reduce traffic demand when substrate links fail.

Furthermore, we examine the effect of preventive traffic disruption model to the operation cost of the substrate network. Note that the problem in (11)-(14) and the problem in (31) can both fully satisfy for traffic demand of all virtual networks in the normal state, i.e., all substrate links are fully operational. However, the problem in (31) takes into account the constraints that ensure the minimal amount of traffic disruption of SPs when substrate links fail. This enforces the substrate network to allocate a certain amount of surplus bandwidth to some virtual links in each virtual network so that when substrate links fail, SPs can find another routing path to deliver data from source to destination. Table II lists the operation cost of the substrate network, the total bandwidth capacity allocated to virtual networks for the optimization problem in (11)-(14) and the problem in (31). It can be easily inferred from Table II that the optimal solution of the problem in (11)-(14) utilizes less bandwidth capacity, and consequently the lower operation cost of the substrate network is achieved.

We next evaluate the effect of parameter τ on the operation cost of the substrate network as well as the amount of traffic disruption of SPs. In general, higher values of τ increase the weight for ℓ_1 -norm term in the objective function, and the resultant optimal solution will ensure smaller amount of traffic disruption of SPs when substrate links fail as illustrating in Fig. 10. However, it also leads to higher operation costs for the substrate network. Based on the result from the figure,

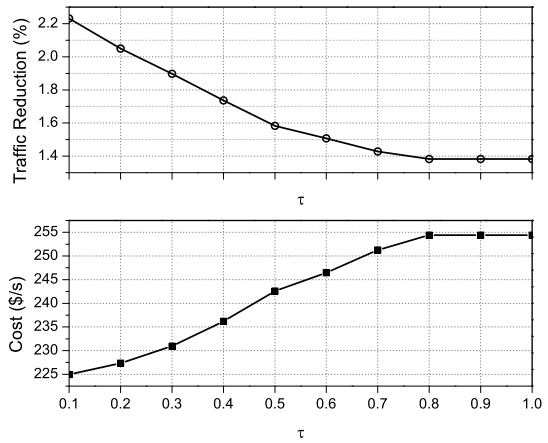


Fig. 10. The effect of τ on the operation cost of the substrate network and the percentage of traffic disruption of SPs.

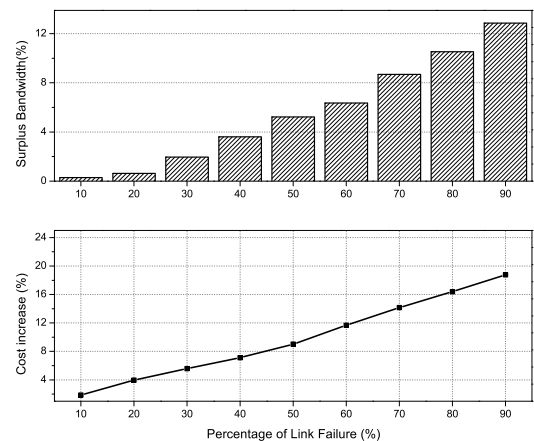


Fig. 11. The effect of percentage of link failure incorporated into the model on the total bandwidth allocated to virtual networks and the operation cost of the substrate network.

the network operator can select appropriate values of τ to satisfy the design criteria for the whole system. The selection will obtain the trade-off between the operation cost for InPs and quality of service for SPs.

Finally, we study the effect of percentage of link failure on the operation cost of the substrate network. Although incorporating a higher percentage of link failure will provide a better preventive level for traffic reduction, it may incur in general a higher operation cost in the normal state. In Fig. 11, we show the amount of surplus bandwidth that the substrate network allocates to virtual networks when we increase the percentage of link failure. We can see from Fig. 11 that the larger amount of surplus bandwidth the substrate network has to allocate to virtual networks to ensure the minimal amount of traffic disruption when we incorporate higher percentage of link failure. However, the operation cost of the substrate network increases accordingly.

VII. DISCUSSIONS

Link failure survivability and survivable routing for network virtualization (both wired and wireless networks) has been an important problem to ensure the quality of service for end-users. Due to the coexistence of multiple virtual networks on the same substrate network, even a single failure in

the substrate network can affect a large number of virtual networks simultaneously. Therefore, resource allocation and routing problems for virtual networks need to take into account substrate link failure constraints to guarantee performance of service providers.

Due to the distributed and independent operation nature of SPs in wireless networks, the post-link failure protection mechanism, i.e., handling protection after network failures happen, may not be practical to guarantee QoS for critical applications. Wireless networks are more vulnerable to network failures, and physical link outage may happen more frequently. Moreover, the dynamic and distributed topology of wireless networks make the recovery mechanism to run longer time to reconfigure the system to the new optimal operation point, and consequently lead to intolerable period of service discontinuation for multiple SPs. Therefore, the proposed preventive approach in this paper is more realistic for wireless network virtualization since it can provide better protection scheme by pre-allocating redundant resources for each virtual networks before any failure happens. Even though this method incurs a higher operation cost for InPs during the normal operation state, it provides higher performance satisfaction for SPs without stopping operation of each virtual network immediately after failures occurs.

However, in wired network virtualization, the primary approach for handling network failures is reactive mechanism, which is performed after the failure occurs by a centralized entity [35]–[38]. After any physical link is broken, failures in virtual networks can be repaired by migrating virtual links operated on top of the original broken links to other available substrate links dedicated for backup purposes. The backup resources can be shared by other backup purpose or will be used when there are new virtual network requests [39]. Since the restoration is carried out after the link failures happen, this method is more cost-effective and achieves lower operation cost during the normal operation state. However, it may lead to a high possibility of huge amount of data loss in virtual networks during the failure.

The reactive approach for handling network failures is more suitable for wired network virtualization due to its efficient resource utilization and advantages in computation. By relocating failed links to backup resource after failures happens, this method can significantly reduce the operation cost and avoid resource wastage due to the low probability of link failures in wired networks. Moreover, by performing centralized computation to migrate and map broken links to another available links, this approach can result in an overall network optimization and faster computation time due to the fixed topology of wired networks.

Even though we primarily focus on incorporating QoS requirements for SP against network failures for wireless scenarios in this paper, the proposed formulation model can also be applied to wired network virtualization with some modifications of the problem in (21)–(28) as follow

$$\min \sum_{l=1}^{L_s} C_l(\mathbf{w}_l) + \tau \sum_{j \in \mathcal{J}} \|\Delta^j\|_1 \quad (61)$$

$$\text{s.t. } A_k \mathbf{f}_k = \mathbf{r}_k, \quad \forall k, \quad (62)$$

$$f_{k,l_k} \leq w_{l,k}, \quad \forall l_k \in \mathcal{L}_k, \quad \forall k \in \mathcal{K}, \quad (63)$$

$$\sum_k w_{l,k} \leq W_l^{\max}, \quad \forall l \in \mathcal{L}_s, \quad (64)$$

$$A_k^j \mathbf{f}_k^j = \mathbf{r}_k^j, \quad \forall k \in \mathcal{K}, \quad \forall j \in \mathcal{J}, \quad (65)$$

$$f_{k,l_k}^j \leq w_{l,k}^j, \quad \forall l_k \in \mathcal{L}_k, \quad \forall k \in \mathcal{K}, \quad \forall j \in \mathcal{J}, \quad (66)$$

$$\sum_k w_{l,k}^j \leq W_l^{\max}, \quad \forall l \in \mathcal{L}_s, \quad (67)$$

$$\Delta_k^j = \sum_{n_k \neq d_k} (r_{n_k} - r_{n_k}^j), \quad \forall k \in \mathcal{K}, \quad \forall j \in \mathcal{J}, \quad (68)$$

$$\Delta_k^j \leq \Delta_k^{\max}, \quad \forall k \in \mathcal{K}, \quad \forall j \in \mathcal{J}. \quad (69)$$

The difference between the models in (61)–(69) and in (21)–(28) is the bandwidth allocation vectors in (66) are allowed to reschedule to the new optimal operation point after link failures happen instead of using the pre-allocated bandwidth resource as in the preventive model. In this model, when any physical link is broken, the central entity will reallocate the bandwidth to virtual networks, which can achieve the global resource utilization and minimal operation cost.

VIII. CONCLUSIONS

In this paper, a resource allocation and routing for wireless network virtualization has been studied. We propose a new formulation for preventive traffic disruption for SPs in wireless network virtualization. The proposed model minimizes the operation cost of the substrate network in the normal state while still guarantees for the minimal amount of traffic reduction when substrate links fail. Due to the large-scale nature of the formulated model, directly solving the optimization problem can be intractable. We then apply the ADMM decomposition technique to propose two algorithms, namely parallel algorithm and distributed algorithm. The parallel algorithm decomposes the centralized problem into multiple sub-problems that can be solved concurrently at different computing nodes, while the distributed algorithm allows each SP and substrate link distributively solve the local problem to converge to the global optimal solution. Numerical results are conducted to demonstrate the convergence behavior as well as computational efficiency of the proposed algorithms. Moreover, the results also show that our model ensures a sparse number of SPs have to reduce traffic when substrate links fail.

REFERENCES

- [1] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Netw. Appl.*, vol. 20, no. 1, pp. 4–18, Feb. 2015.
- [2] A. Alexiou, "Wireless world 2020: Radio interface challenges and technology enablers," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 46–53, Mar. 2014.
- [3] Q. Zhou, C.-X. Wang, S. McLaughlin, and X. Zhou, "Network virtualization and resource description in software-defined wireless networks," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 110–117, Nov. 2015.
- [4] A. Tzanakaki, M. P. Anastasopoulos, G. S. Zervas, B. R. Rofoee, R. Nejabati, and D. Simeonidou, "Virtualization of heterogeneous wireless-optical network and IT infrastructures in support of cloud and mobile cloud services," *IEEE Commun. Mag.*, vol. 51, no. 8, pp. 155–161, Aug. 2013.

- [5] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.
- [6] C. Liang, F. Yu, and X. Zhang, "Information-centric network function virtualization over 5G mobile wireless networks," *IEEE Netw.*, vol. 29, no. 3, pp. 68–74, May 2015.
- [7] Z. Feng, C. Qiu, Z. Feng, Z. Wei, W. Li, and P. Zhang, "An effective approach to 5G: Wireless network virtualization," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 53–59, Dec. 2015.
- [8] C. Liang and F. R. Yu, "Wireless virtualization for next generation mobile cellular networks," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 61–69, Feb. 2015.
- [9] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1114–1128, 4th Quart., 2012.
- [10] N. Feamster, L. Gao, and J. Rexford, "How to lease the Internet in your spare time," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 61–64, Jan. 2007.
- [11] N. M. M. K. Chowdhury and R. Boutaba, "Network virtualization: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20–26, Jul. 2009.
- [12] H. Wen, P. K. Tiwary, and T. Le-Ngoc, *Wireless Virtualization* (Springer Briefs in Computer Science). New York, NY, USA: Springer-Verlag, 2013.
- [13] H. Wen, P. Tiwary, and T. Le-Ngoc, "Current trends and perspectives in wireless virtualization," in *Proc. Int. Conf. Select. Topics Mobile Wireless Netw. (MoWNeT)*, Montreal, QC, Canada, Aug. 2013, pp. 62–67.
- [14] G. Liu, F. Yu, H. Ji, and V. C. M. Leung, "Virtual resource management in green cellular networks with shared full-duplex relaying and wireless virtualization: A game-based approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7529–7542, Sep. 2016.
- [15] A. Fischer, J. F. Botero, M. Till Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.
- [16] F. Fu and U. C. Kozat, "Stochastic game for wireless network virtualization," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 84–97, Feb. 2013.
- [17] Q. Zhu and X. Zhang, "Game-theory based power and spectrum virtualization for maximizing spectrum efficiency over mobile cloud-computing wireless networks," in *Proc. 49th Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2015, pp. 1–6.
- [18] L. Chen, F. R. Yu, H. Ji, G. Liu, and V. C. M. Leung, "Distributed virtual resource allocation in small-cell networks with full-duplex self-backhauls and virtualization," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5410–5423, Jul. 2016.
- [19] K. Zhu and E. Hossain, "Virtualization of 5G cellular networks as a hierarchical combinatorial auction," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2640–2654, Oct. 2016.
- [20] G. Liu, F. R. Yu, H. Ji, and V. C. M. Leung, "Distributed resource allocation in full-duplex relaying networks with wireless virtualization," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 4959–4964.
- [21] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.
- [22] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 2185–2193, Aug. 2006.
- [23] A. El-Sherif and A. Mohamed, "Joint routing and resource allocation for delay minimization in cognitive radio based mesh networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 186–197, Jan. 2014.
- [24] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136–1144, Jul. 2004.
- [25] 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.
- [26] L. Le and E. Hossain, "Cross-layer optimization frameworks for multi-hop wireless networks using cooperative diversity," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2592–2602, Jul. 2008.
- [27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
- [28] H. Tabrizi, B. Peleato, G. Farhadi, J. Cioffi, and G. Aldabbagh, "Spatial reuse in dense wireless areas: A cross-layer optimization approach via admm," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 7083–7095, Dec. 2015.
- [29] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed joint resource and routing optimization in wireless sensor networks via alternating direction method of multipliers," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5454–5467, Nov. 2013.
- [30] Y. Chen *et al.*, "Fundamental trade-offs on green wireless networks," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 30–37, Jun. 2011.
- [31] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–13.
- [32] A. Zhou, M. Liu, Z. Li, and E. Dutkiewicz, "Joint traffic splitting, rate control, routing, and scheduling algorithm for maximizing network utility in wireless mesh networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2688–2702, Apr. 2016.
- [33] *MPI: A Message-Passing Interface Standard. Version 2.2*, MPI Forum, Sep. 2009.
- [34] M. Grant and S. Boyd. (Mar. 2014). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1*. [Online]. Available: <http://cvxr.com/cvx>
- [35] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, "Adaptive virtual network provisioning," in *Proc. 2nd ACM SIGCOMM Workshop Virtualized Infrastruct. Syst. Architect. (VISA)*, New York, NY, USA, 2010, pp. 41–48.
- [36] M. R. Rahman, I. Aib, and R. Boutaba, *Survivable Virtual Network Embedding*. Berlin, Germany: Springer, 2010, pp. 40–52.
- [37] B. Guo *et al.*, "A novel virtual node migration approach to survive a substrate link failure," in *Proc. Opt. Fiber Commun. Conf. Expo. Nat. Fiber Opt. Eng. Conf. (OFC/NFOEC)*, Mar. 2012, pp. 1–3.
- [38] H. Yu, V. Anand, C. Qiao, and H. Di, "Migration based protection for virtual infrastructure survivability for link failure," in *Proc. Opt. Fiber Commun. Conf. Expo. Nat. Fiber Opt. Eng. Conf. (OFC/NFOEC)*, Mar. 2011, pp. 1–3.
- [39] T. Guo, N. Wang, K. Moessner, and R. Tafazolli, "Shared backup network provision for virtual network embedding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.