

基于约束路由的绿色虚拟拓扑设计算法

伍元胜¹, 郭兵¹, 沈艳², 王继禾¹, 刘啸滨¹

(1. 四川大学 计算机学院, 四川 成都 610065; 2. 成都信息工程学院 控制工程学院, 四川 成都 610225)

摘 要: 针对 Internet 核心网日益严重的能耗问题, 建立了绿色虚拟拓扑设计(GVTD)问题的形式化模型, 通过业务汇聚、按需配置网络资源、动态虚拟拓扑设计和多粒度睡眠机制降低网络能耗。提出了一种基于约束路由的启发式算法——CBR-GVTD 算法, 利用单跳路由和多跳路由相结合的方法构建网络虚拟拓扑, 并通过基于约束的路由实现网络功耗与路由性能的折衷。模拟结果表明, CBR-GVTD 算法可在接口平均利用率为 80%~90%和最大路由跳数不超过 5 的条件下, 最多可降低 62%~90%的网络功耗。

关键词: 绿色网络; 基于约束的路由; 虚拟拓扑设计; 多粒度睡眠; 核心网能耗

中图分类号: TN915.02

文献标识码: A

文章编号: 1000-436X(2014)04-0112-12

Green virtual topology design algorithm based on constraint-based routing

WU Yuan-sheng¹, GUO Bing¹, SHEN Yan², WANG Ji-he¹, LIU Xiao-bin¹

(1. College of Computer Science, Sichuan University, Chengdu 610065, China;

2. School of Control Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: To reduce the energy consumption of Internet core network, a formulation model for green virtual topology design (GVTD) problem was developed, which employs the following measures: traffic aggregation, network resources allocation on demand, virtual topology design dynamically and multi-granularity sleeping. A heuristic algorithm base on constraint-based routing, namely CBR-GVTD, was proposed, which combines the single-hop and multi-hop routing to establish virtual topology, and makes trade-offs between the network power consumption and routing performance via the constraint-based routing mechanism. The simulation results indicate that CBR-GVTD can achieve at most 62%~90% power saving with very high resource utilization (80%~90% for interface utilization) and excellent routing performance (not more than 5 for maximum routing hops).

Key words: green networking; constraint-based routing; virtual topology design; multi-granularity sleeping; energy consumption of core network

1 引言

近年来, Internet 的流量逐年指数增长。从 2007~2011 年, Internet 的流量和带宽的年平均增长率分别达到了 56%和 58%^[1]。流量和带宽的增长导致了 Internet 的能耗上升。据估计, 2007 年 Internet

的电力消耗达到了宽带接入国家(其平均接入带宽为 30 Mbit/s)总电量的 1%, 当平均接入带宽达到 300 Mbit/s 时, 这个比例将超过 4%^[2]。按照目前的增长速度, 到 2050 年网络领域的耗电量将达到 2006 年的 13 倍^[3]。Internet 能耗的快速增长不仅导致电力成本的持续上升, 同时也造成温室气体的加速排

收稿日期: 2012-12-20; 修回日期: 2013-10-08

基金项目: 国家自然科学基金资助项目(61332001, 61272104, 61073045); 四川省杰出青年基金资助项目(2010JQ0011); 中国科学院计算技术研究所计算机体系结构国家重点实验室开放课题基金资助项目(ICT-ARCH201003)

Foundation Items: The National Natural Science Foundation of China (61332001, 61272104, 61073045); Sichuan Science Fund for Distinguished Young Scholars(2010JQ0011); The Fund From State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences(ICT-ARCH201003)

放。因此，提高能量效率和降低能耗已成为 Internet 面临的重大研究课题。目前，随着网络的扁平化发展，Internet 的层次变得更加简单，逐渐演变成由接入网和核心网两部分组成。由于接入网的“光进铜退”以及核心网对接入网流量的汇聚，核心网正经历比接入网更快的能耗增长。研究表明到 2017 年核心网能耗将超过接入网^[4]。因此，在整个 Internet 中，核心网的节能研究正变得日益重要。

网络按照峰值业务需求超额供给^[5]网络资源，并通过冗余设计来提高网络的可靠性，这导致了网络资源的平均利用率低下，而当前利用率对网络资源的功耗影响却较小^[6-8]，因此，提高网络资源利用率并将空闲的网络资源关闭（或转入低功耗睡眠状态）是目前降低核心网能耗的一个重要途径。目前，主要有如下文献研究此问题。Chabarek 等人^[9]利用混合整数线性规划(MILP, mixed integer linear programming)技术建模功率感知的网络设计和路由问题，通过为网络节点选择合适数量和类型的线卡最小化 IP 网络的功耗，但却没有设计有效的启发式算法来求解 NP 难的 MILP 模型。文献^[10]将 IP 网络的绿色流量工程形式化为一个 MILP 模型，并提出了一种启发式算法，通过预先为每对节点计算 k 条候选最短路径来降低 MILP 问题的解空间和求解时间，但是，解空间依然随节点对的数量指数增长。文献^[11]提出了一种基于拉格朗日松弛的启发式算法，将建立的形式化模型分解为容易求解的子问题，为链路设定合适的权值进行动态路由，关闭尽可能多的链路以降低 IP 网络的能耗。与文献^[9-11]改变已有的网络拓扑不同，文献^[12,13]考虑核心网的层次结构，通过改变 IP 层的虚拟拓扑降低网络能耗。其中，文献^[12]考虑光收发器和电交换的功耗，将功率感知的虚拟拓扑设计形式化为 MILP 问题，并提出一个简单的贪婪算法和遗传算法求解。该遗传算法并没有相应的机制保证得到的后代个体满足 MILP 模型的约束条件，而是简单的舍弃不满足约束的个体，这会导致算法有时找不到可行解。文献^[13]只考虑线卡的功耗，将最小化能耗的虚拟拓扑设计问题形式化为一个简单的整数线性规划(ILP, integer linear programming)模型，并提出了一个两阶段的启发式算法。由于只考虑了线卡的功耗，建立的 ILP 模型过于简单，网络倾向于建立一个星型虚拟拓扑，最小化链路的数量。

笔者认为，现有工作存在以下不足。首先，这

些文献都只考虑网络设备的部分组件的功耗，如链路、收发器或线卡，忽略了其他部分的功耗，这样使得最终得到的解只能使整个网络中的这些组件的功耗最低，而不是整个网络的功耗最低。其次，核心网的网络设备通常采用模块化设计，现有的工作没有考虑网络设备的模块化结构。第三，由于问题的复杂性(NP 问题)，现有工作要么建立的形式化模型过于复杂，以至于不能有效求解，如文献^[9,10,12]，要么建立的模型过于简化而忽略重要特性，如文献^[11,13]只考虑链路或线卡功耗，忽略了 IP 路由器处理流量的功耗。本文研究的内容与文献^[12,13]相似，即以最小化网络功耗为目标的绿色虚拟拓扑设计(GVTD, green virtual topology design)问题。本文首先对 GVTD 进行形式化建模，该模型(即 GVTD 模型)通过业务路由实现对业务流的汇聚和疏导以提高网络资源利用率，根据网络资源的实际利用按需地配置网络资源以实现网络设备的多粒度睡眠，根据网络资源配置动态地建立虚拟拓扑，从而达到降低网络能耗的目的。GVTD 模型考虑整个网络设备的功耗，目的是降低整个网络的功耗。其次，GVTD 模型考虑了网络设备的模块化结构，利用多粒度睡眠机制配置网络资源。第三，GVTD 模型根据功耗与业务负载的依赖关系，同时考虑了不依赖业务负载的静态功耗和依赖业务负载的动态功耗。最后，GVTD 模型是一个 NP 难的 ILP 问题，只能在问题规模很小时精确求解，本文提出了一个基于约束路由的启发式算法，即 CBR-GVTD 算法。本文的主要贡献如下。

1) 对绿色虚拟拓扑设计问题进行形式化建模，即 GVTD 模型，该模型通过业务汇聚提高网络资源利用率，按需配置网络资源和动态建立虚拟拓扑，从而达到降低网络功耗的目的。

2) 提出了一种基于约束路由的启发式算法，即 CBR-GVTD 算法，该算法使用基于约束的路由算法保证链路容量约束和最大路由跳数约束，在提高网络资源利用率的同时实现了虚拟拓扑对业务路由的动态适应。

3) 通过模拟实验从网络功耗、资源利用率和路由性能 3 个方面对 CBR-GVTD 算法的性能进行了评估，结果表明 CBR-GVTD 算法在保证路由性能的前提下获得很高的资源利用率，最多可降低 62%~90%的网络功耗。

4) 探讨了网络最大路由跳数对网络功耗的影

响,发现可在网络功耗无明显增加的条件下大大降低网络的最大路由跳数。

2 GVTD 问题的形式化描述

本节对 GVTD 问题进行形式化描述,建立 GVTD 模型。在此之前,有必要对核心网的虚拟拓扑设计、网络设备的模块化结构和多粒度睡眠机制进行介绍。在核心网中,IP 网络层通常构建于高速的 TDM(time division multiplexing)网络层或者 WDM(wavelength division multiplexing)网络层之上^[14],形成所谓的 IP over SONET/SDH/OTN 网络或 IP over WDM 网络。下层网络(即 TDM 层或 WDM 层)向 IP 层提供通道传输服务(即 TDM 电路服务和光路服务,电路和光路统称为传输通道),IP 层通过使用下层提供的通道传输服务向其上层提供分组传输服务。IP 层的链路由 1 条或多条具有相同源节点和目的节点的传输通道组成,这种链路不同于传统的物理链路,因此被称为逻辑链路,由逻辑链路构成的网络拓扑被称为虚拟拓扑(或逻辑拓扑)。根据给定的业务需求建立 IP 层逻辑链路的过程即为 IP 层的虚拟拓扑设计(VTD, virtual topology design)过程。对于 IP 层,下层提供的传输通道组成了其逻辑链路;对于下层网络,IP 层请求建立的传输通道形成了其业务需求。

根据核心网的网络设备的模块化结构把网络资源^{注1}分成接口、线卡和机框 3 类:接口由一对发送/接收端口组成,每条传输通道起始于源节点的 1 个发送端口并终止于目的节点的 1 个接收端口;线卡由一组接口所共享,能够与这组接口同时处于空闲或活跃状态;除接口和线卡以外的部分全都归入机框的范畴。基于这种分类,提出一种多粒度睡眠机制来配置网络资源:当接口空闲时让接口睡眠,当线卡的所有接口都空闲时则线卡睡眠,当所有线卡都空闲时则机框睡眠。

GVTD 模型假定网络资源按照峰值需求供给,即每个网络节点具有充足的接口、线卡和机框;下层网络可以提供 IP 层所需的传输通道。GVTD 模型主要包括以下四部分。

1) 网络功耗模型,即 GVTD 模型的目标函数。网络功耗模型同时考虑网络设备的静态功耗和动

态功耗,静态功耗指网络设备独立于业务负载的那部分功耗(即空闲状态下的功耗),动态功耗指网络设备依赖业务负载的那部分功耗。根据网络设备的模块化结构,静态功耗可以进一步细分为接口功耗、线卡功耗和机框功耗。目前利用率对网络设备功耗的影响较少,文献[14]实验测得网络交换机的功耗变动范围(即动态功耗)仅为 15%。

2) 业务路由,即 GVTD 模型的路由约束,为了避免多径路由引发的时延抖动,GVTD 模型使用单径路由。

3) 资源配置,即 GVTD 模型的资源约束,该约束确定每个节点的活跃网络资源,并按照多粒度睡眠机制使空闲的网络资源睡眠。

4) 虚拟拓扑设计,对应于 GVTD 模型的链路容量约束,该约束确定网络中需要建立的传输通道。此外,笔者通过设定接口最大利用率参数实现了对网络拥塞^{注2}的控制。

GVTD 模型的输入参数定义如下。 N 表示网络节点集合, $i, j, ii, jj \in N$ 表示网络节点, C 表示接口的容量, α 表示接口(或链路)所允许的最大利用率, p^i 表示网络资源处理单位(1 Gbit/s)业务负载的动态功耗, p^i 、 p^l 和 p^c 分别表示单个接口、线卡和机框的功耗, m^i 、 m^l 分别表示每个线卡具有的接口数量和每个机框可配备的线卡数量, D 表示网络的业务需求集合, $d(ii, jj) \in D$ 表示从节点 ii 到节点 jj 的业务需求。

GVTD 问题的决策变量(即 GVTD 模型的优化参数)定义如下。

1) $\delta(ii, jj, i, j) \in \{0, 1\}$: 业务需求 $d(ii, jj)$ 的路径是否经过逻辑链路 (i, j) , 1 表示经过, 0 表示不经过。

2) $d'(i, j) \in Z^+$: 逻辑链路 (i, j) 包含的传输通道数量,其中, Z^+ 表示正整数集合,传输通道由下层网络提供,而且每条传输通道需要消耗一个发送端口和接收端口。

3) $t(i, j) \in R^+$: 逻辑链路 (i, j) 上的总业务量,其中, R^+ 表示正实数集合。

4) $n^i(i) \in Z^+$: 网络节点 i 的活跃接口数量。

5) $n^l(i) \in Z^+$: 网络节点 i 的活跃线卡数量。

6) $n^c(i) \in Z^+$: 网络节点 i 的活跃机框数量。

其中, $\delta(ii, jj, i, j)$ 和 $t(i, j)$ 表示网络的业务路由,

注1: 网络资源通常具有很宽泛的含义,本文只关心与网络功耗相关的网络资源,因此,后文的网络资源特指接口、线卡和机框。

注2: 在网络优化中,拥塞被形式化为网络接口(或链路)的最大利用率。

$d'(i, j)$ 表示虚拟拓扑, $n^i(i)$ 、 $n^l(i)$ 和 $n^c(i)$ 表示网络的资源配置。

GVTD 问题可以形式化描述为

$$\min \sum_{i \in N} \left[n^c(i)p^c + n^l(i)p^l + n^i(i)p^i \right] + \sum_{i, j \in N: i \neq j} t(i, j)p^t \quad (1)$$

Subject to:

$$\sum_{j \in N: j \neq i} \delta(ii, jj, i, j) - \sum_{j \in N: j \neq i} \delta(ii, jj, j, i) = 0, \quad \forall ii, jj, i \in N: d(ii, jj) > 0 \wedge i \neq ii \wedge i \neq jj \quad (2a)$$

$$\sum_{j \in N: j \neq ii} \delta(ii, jj, ii, j) - \sum_{j \in N: j \neq ii} \delta(ii, jj, j, ii) = 1, \quad \forall ii, jj \in N: d(ii, jj) > 0 \quad (2b)$$

$$\sum_{i \in N: i \neq jj} \delta(ii, jj, i, jj) - \sum_{i \in N: i \neq jj} \delta(ii, jj, jj, i) = 1, \quad \forall ii, jj \in N: d(ii, jj) > 0 \quad (2c)$$

$$\sum_{j \in N: j \neq i} \delta(ii, jj, i, j) \leq 1, \quad \forall ii, jj, i \in N: d(ii, jj) > 0 \quad (2d)$$

$$\sum_{j \in N: j \neq i} \delta(ii, jj, j, i) \leq 1, \quad \forall ii, jj, i \in N: d(ii, jj) > 0 \quad (2e)$$

$$t(i, j) = \sum_{ii, jj \in N: d(ii, jj) > 0} \delta(ii, jj, i, j) d(ii, jj), \quad \forall i, j \in N: i \neq j \quad (3)$$

$$t(i, j) \leq d'(i, j) \alpha C, \quad \forall i, j \in N: i \neq j \quad (4)$$

$$n^i(i) \geq \sum_{j \in N: j \neq i} d'(i, j), \quad \forall i \in N \quad (5a)$$

$$n^i(i) \geq \sum_{j \in N: j \neq i} d'(j, i), \quad \forall i \in N \quad (5b)$$

$$n^l(i)m^l \geq n^i(i), \quad \forall i \in N \quad (6)$$

$$n^c(i)m^c \geq n^i(i), \quad \forall i \in N \quad (7)$$

目标函数(式(1))最小化整个网络的功耗。网络功耗由静态功耗和动态功耗两部分组成, 静态功耗可进一步分为机框功耗 $n^c(i)p^c$ 、线卡功耗 $n^l(i)p^l$ 和接口功耗 $n^i(i)p^i$; 笔者使用线性函数^[15]近似动态功耗与业务负载的关系, 即 $t(i, j)p^t$ 。由于网络资源的睡眠功耗远远小于活跃功耗, 本文忽略网络资源的睡眠功耗。

路由约束(式(2a)~式(2e))为经典的多品种流守恒^[16]约束, 业务需求 $d(ii, jj)$ 经过单条路径从源节点 ii 到达目的节点 jj 。式(3)计算经过逻辑链路 (i, j) 的业务负载(即链路流量), 链路的容量约束(式(4))确保链路的流量不超过链路的容量。IP 层

的链路 (i, j) 为逻辑链路, 由下层网络提供的 $d'(i, j)$ 条传输通道组成, 每条传输通道的容量为 C , 并引入最大利用率 α 以应对 IP 业务的动态特性^{注3}。IP 层虚拟拓扑的确立过程即为决策变量 $d'(i, j)$ 的值的确立过程, 因此, 式(4)在实现链路容量约束的同时还实现了 IP 层的虚拟拓扑设计, 实现了 IP 层的逻辑链路与下层网络提供的传输通道的映射(即 IP 层与下层网络的层间约束)。式(5)~式(7)为资源约束。其中, 式(5a)和式(5b)可确保节点提供足够的活跃接口, 即活跃接口不应该少于传输通道所需的发送端口(式(5a))和接收端口(式(5b))。式(6)和式(7)确保节点能提供足够的活跃线卡和活跃机框, 即活跃线卡和活跃机框提供的接口数不少于活跃接口数。

3 GVTD 模型的求解

GVTD 模型是一个 ILP 模型, 由于 ILP 是 NP 难问题, 在计算资源有限的条件下, 目前只可在问题规模很小时进行精确求解。为此, 本文提出一种快速有效的启发式算法, 将该算法命名为 CBR-GVTD 算法。CBR-GVTD 算法主要考虑和解决以下 2 个问题。

1) 如何构造可行解, 即构造解时应满足 GVTD 模型的所有约束条件。

2) 如何提高解的质量, 即构造的最终解尽可能地近似最优解。

对于第一个问题, 首先需要分析各个约束条件以及各个决策变量之间的依赖关系。在 GVTD 模型中, 根据约束条件间的关系可知决策变量存在以下依赖关系: $n^l(i)$ 和 $n^c(i)$ 都依赖 $n^i(i)$, $n^i(i)$ 依赖 $d'(i, j)$, $d'(i, j)$ 依赖 $t(i, j)$, $t(i, j)$ 依赖 $\delta(ii, jj, i, j)$ 。因此, 根据以上依赖关系, 本文使用单跳路由和基于约束的路由(多跳路由)确立 $\delta(ii, jj, i, j)$, $t(i, j)$ 和 $d'(i, j)$ 的值, 并同时满足式(2a)~式(2e)所示的路由约束、式(3), 和式(4)所示的链路容量约束。然后使用以式(8)~式(10)确立 $n^i(i)$, $n^l(i)$ 和 $n^c(i)$ 的值。

$$n^i(i) = \max \left(\sum_{j \in N: j \neq i} d'(i, j), \sum_{j \in N: j \neq i} d'(j, i) \right), \quad \forall i \in N \quad (8)$$

$$n^l(i) = \lceil n^i(i) / m^l \rceil, \quad \forall i \in N \quad (9)$$

$$n^c(i) = \lceil n^i(i) / (m^l \cdot m^c) \rceil, \quad \forall i \in N \quad (10)$$

注3: 业务需求的动态特性可能导致网络拥塞, 将链路的最大利用率限定得越低, 产生网络拥塞的概率也越低。GVTD 模型通过参数 α 调节链路的最大利用率以实现网络拥塞与功耗的折衷。


```

18) }else if hops(v)=hops(u)+1 then{
19) if min(pb(u),b(u,v))<pb(v) then{
20) Q←Q∪{v}
21) predecessor(v)←u
22) pb(v)←min(pb(u),b(u,v)) }end if}end if}end
if }end for
23) if Q=∅ then break
24) u←first element of Q
25) Q←Q−{u} }end while
26) if u=i and hops(j)≤H then{
27) return predecessor
28) }else return NIL

```

此算法的输入包括业务需求 $d(i, j)$ 、网络节点集合 N 、网络链路集合 E 、网络链路的可用带宽集合 B 和网络的最大路由跳数 H 。算法输出为 $d(i, j)$ 的路径。在算法中, $predecessor(v)$ 记录节点 v 的前驱节点, $hops(v)$ 记录从源节点 i 到节点 v 的最小跳数, $visited(v)$ 标识节点 v 是否已被访问, $pb(v)$ 表示从节点 i 到节点 v 的路径的可用带宽, $b(u, v) \in B$ 表示链路 (u, v) 的可用带宽, Q 为候选节点列表。算法的 1)~7) 行首先进行一些初始化工作, 然后从源节点 i 开始(第 8)行), 计算源节点 i 的最短路径生成树, 直到计算出到目标节点 j 的路径时停止(第 9)~25)行)。算法只考虑可用带宽大于业务需求的链路(第 12)行), 当跳数相同时, 优先选择可用带宽最小的路径(第 18)~22)行)。算法最后判定计算的最短路径是否小于网络的最大跳数 H (第 26)行), 如果路由成功, 则返回前驱节点表示的最短路径(第 27)行), 否则返回空值 NIL(第 28)行)。

在 CBR-Dijkstra 算法伪代码中, 第一个 for 循环的执行次数等于网络的节点数 $|N|$, while 循环中的 for 循环最多执行次数等于网络有向边数 $|E|$, 对于连通图有 $|N| \leq |E|$, 因此, CBR-Dijkstra 算法的时间复杂度仅为 $O(|E|)$ 。

3.2 基于约束路由的绿色虚拟拓扑设计算法——CBR-GVTD 算法

基于约束路由的绿色虚拟拓扑设计(CBR-GVTD, constraint-based routing green virtual topology design) 算法的伪代码如下所示。

Algorithm CBR-GVTD($N, D, \alpha, C, p^t, p^i, p^l, p^c, m^i, m^l, H$)

```

1) Initialize  $n^i(i)$  for each node  $i \in N$  with the
lower bound  $n^i(i)_{lb}$  according to formula (11)

```

```

2) do{
3) VTDCChange←FALSE
4) for each demand  $d(i, j) \in D$  in descending
order do{
5) Compute the number of transport channels  $k$ 
should be established for  $d(i, j)$ , that is  $k \leftarrow \lceil d(i, j) / (\alpha C) \rceil$ 
6) if there are enough unused interfaces at node  $i$ 
and  $j$  then{
7) Establish  $k$  transport channels between  $i$  and  $j$ 
and route  $d(i, j)$  via single hop routing
8)  $k$  sending ports and receiving ports are used at
node  $i$  and  $j$  respectively }end if }end for
9) for each demand  $d(i, j) \in D$  in descending or-
der which have not been routed yet do{
10) Route  $d(i, j)$  over the virtual topology using
CBR-Dijkstra algorithm with the maximum hops con-
straint  $H$ 
11) if failed to route  $d(i, j)$  then{
12) Establish  $k \leftarrow \lceil d(i, j) / (\alpha C) \rceil$  transport chan-
nels between  $i$  and  $j$  and route  $d(i, j)$  via single hop
routing
13) Increase  $n^i(i)$  and  $n^i(j)$  to make sure node  $i$  and
 $j$  can provide  $k$  sending and receiving interfaces
14) VTDCChange←TRUE
15)  $k$  sending ports and receiving ports are used at
node  $i$  and  $j$  respectively }end if }end for
16) }while VTDCChange
17) Initialize  $E'$  with current logic link set  $E$ , that
is  $E' \leftarrow E$ 
18) for each logic link  $(i, j) \in E'$  satisfying  $b(i,
j) = \max(u, v) \in E' b(u, v)$  do{
19)  $E' \leftarrow E' - \{(i, j)\}$ 
20) Remove one channel of logic link  $(i, j)$ , that is
 $d^r(i, j) \leftarrow d^r(i, j) - 1$ 
21) if the associated traffic demands of logic link
 $(i, j)$  can be rerouted over the remaining virtual topol-
ogy using CBR-Dijkstra algorithm with the maximum
hops constraint  $H$  then{
22) Release one sending and receiving port at
node  $i$  and  $j$  respectively
23) }else recovery the removed transport channel
of logic link  $(i, j)$ , that is  $d^r(i, j) \leftarrow d^r(i, j) + 1$  }end for
24) Remove the unused interfaces at each node  $i$ 

```

$\in N$, and get the final $n^i(i)$

25) Compute $t(i,j)$ for each logic link according to formula (3)

26) Compute $n^l(i)$ and $n^c(i)$ for each node according to formula (9) and (10)

27) Compute the total power consumption of network according to formula (1)

此算法的输入包括 N 、 D 、 α 、 C 、 p^l 、 p^i 、 p^l 、 p^c 、 m^i 、 m^l 和 H ，其中， H 为网络的最大路由跳数，其他符号已在第 3 节中定义。算法的输出为网络功耗和第 3 节定义的决策变量。CBR-GVTD 算法分 2 个阶段，即：1) 从目标函数下界值构造可行解（第 1)~16)行）；2) 通过移除利用率低的传输通道对可行解进行优化（第 17)~27)行）。

在第 1 阶段，算法首先根据式(11)用下界值 $n^i(i)_{lb}$ 初始化每个网络节点的接口数 $n^i(i)$ (第 1)行)，然后分两步建立网络的虚拟拓扑。第 1 步按照从大到小的顺序为每个业务需求 $d(i,j)$ 建立从 i 到 j 的传输通道，并进行单跳路由（第 4)、5)、7)行）。每个传输通道消耗源节点 i 的一个发送端口（sending port）和目的节点 j 的一个接收端口（receiving port）（第 8)行），该过程将持续到接口耗尽、没有传输通道可以建立为止（第 6)行）。第 2 步在第 1 步建立的虚拟拓扑的基础上，利用单跳路由的剩余带宽，对剩余的业务需求（第 1 步中未能完成单跳路由的业务需求）按照从大到小的顺序进行 CBR-Dijkstra 路由（第 9)、10)行）。该路由必然是多跳路由，为了防止路由跳数过大，本文限制网络的最大跳数为 H 。如果业务需求 $d(i,j)$ 的路由失败（第 11)行），则增加节点 i 和 j 的接口（因为在第 1 步中已经耗尽）（第 13)行），为 i 到 j 建立传输通道并消耗相应的接口数（第 12)、15)行），这也意味着网络的虚拟拓扑发生了变化（第 14)行）。通过第 2 步，所有的业务需求都成功完成路由，但是网络的虚拟拓扑可能发生了变化，通过反复迭代使这种虚拟拓扑对业务路由的适应性变化收敛（第 2)、3)、14)、16)行）。

在第 2 阶段，算法初始化链路集合 E' 为当前的逻辑链路集合 E (第 17)行)，然后每次从 E' 中取出剩余带宽最大的逻辑链路 (i,j) (第 18)行)并将 (i,j) 从 E' 中去除（第 19)行)，尝试移除从 i 到 j 的一条传输通道（第 20)行)，并对经过逻辑链路 (i,j) 的所

有业务需求进行 CBR-Dijkstra 重路由，如果重路由成功，则释放该传输通道消耗的接口（第 21)、22)行)，否则恢复移除的传输通道（第 23)行)。其中，链路的剩余带宽 $b(i,j)$ 在每次路由后进行更新，最大的 $b(i,j)$ 表明链路 (i,j) 的某条传输通道的利用率最低。至此，虚拟拓扑设计和业务路由已经完成，即 $\delta(ii,jj,i,j)$ 和 $d'(i,j)$ 的值已经确定。算法移除每个节点剩余（即没用使用）的接口，确立 $n^i(i)$ 的最终值（第 24)行)，通过式(3)、式(9)和式(10)分别确定 $t(i,j)$ 、 $n^l(i)$ 和 $n^c(i)$ （第 25)和 26)行)，最后通过式(1)计算整个网络的功耗（第 27)行)。

CBR-GVTD 算法通过单跳路由建立虚拟拓扑，对剩余的业务需求使用 CBR-Dijkstra 算法进行多跳路由。由于单跳路由（算法的第 5)、7)和 8)行)的时间复杂度为常数级，低于多跳路由（即 CBR-Dijkstra 算法)的时间复杂度，因此，以 CBR-Dijkstra 算法为基本操作分析 CBR-GVTD 算法的时间复杂度。假设第 2)~16)行的 do-while 循环了 k 次，则第 10)行的 CBR-Dijkstra 算法最多执行 $k \cdot |D|$ 次（ $|D|$ 为业务需求数）。假设网络的平均路由由路数为 h ，所有业务需求共 $h \cdot |D|$ 跳，因此，第 21)行共执行 $h \cdot |D|$ 次，因此 CBR-GVTD 算法的时间复杂度为 $O((k+h) \cdot |D|)$ 。 k 和 h 通常都很小，在第 4 节的实验中， k 不超过 4， h 不超过 3。

4 性能评估

通过一系列模拟实验从网络功耗、网络资源（即接口、线卡和机框）利用率和路由性能（即平均跳数和最大跳数）3 个方面对 CBR-GVTD 算法进行性能评估，并将 CBR-GVTD 算法和其他算法的实验结果进行比较，包括：GVTD 模型的精确解、网络功耗的下界和上界以及基于拉格朗日松弛的启发式算法（即 LR 算法^[12]）。其中，笔者使用数学工具 GAMS/CPLEX^[17]求解 GVTD 模型的精确解，CPLEX 是一种高性能的 ILP 求解器，采用分支定界算法求解 ILP 问题，由于 ILP 问题是 NP 难的，只能对规模较小的网络精确求解。网络功耗下界 LB 由式(11)和式(12)求得，网络的功耗上界 UB 由式(13)和式(14)求得。在式(13)中，每个业务需求 $d(i,j)$ 通过 $\lceil d(i,j)/(\alpha C) \rceil$ 条传输通道进行单跳路由，因此得到节点的接口上界 $n^i(i)_{ub}$ 。与 LB 的计算类似，将 $n^i(i)_{ub}$ 代入式(9)和式(10)可求得 $n^l(i)_{ub}$ 和 $n^c(i)_{ub}$ ，最后将这些值代入目标函数即得到上界值 UB （即式

(14)。LR 算法使用拉格朗日松弛技术将 ILP 模型分解成容易求解的子问题，然后对每个子问题求解并使用次梯次算法求解拉格朗日对偶问题，最后对得到的解进行可行化处理。

$$n^i(i)_{ub} = \max \left(\sum_{j \in N: j \neq i} \lceil d(i,j)/(\alpha C) \rceil, \sum_{j \in N: j \neq i} \lceil d(j,i)/(\alpha C) \rceil \right), \forall i \in N \quad (13)$$

$$UB = \sum_{i \in N} \left\{ \lceil n^i(i)_{ub} / (m^i m^i) \rceil \cdot p^c + \lceil n^i(i)_{ub} / m^i \rceil \cdot p^l + n^i(i)_{ub} p^j \right\} + \sum_{i,j \in N: i \neq j} d(i,j) p^j \quad (14)$$

为了全面评估 CBR-GVTD 算法的性能，实验充分考虑了多种不同大小的网络、多种不同大小的业务需求和多种不同大小的网络最大跳数，实验参数设定如下所示。

1) 网络规模。GVTD 问题不涉及网络的物理拓扑，因此，只需节点数量即可表示各种大小的网络。实验网络的节点数量分别为 10、20、30、40 和 50。

2) 业务需求。Internet 的流量具有重尾分布特性，本文使用重力模型^[17]生成重尾分布的业务需求，网络节点间的平均业务需求分别为 1 Gbit/s、5 Gbit/s、10 Gbit/s、15 Gbit/s、20 Gbit/s、25 Gbit/s、30 Gbit/s、35 Gbit/s 和 40 Gbit/s。

3) 网络设备的相关参数以 Cisco CRS3-8/S 路由器的规格说明为参考，具体设定如表 1 所示。

表 1 网络设备相关的参数

参数	取值
p^c	1183.2 watt
p^l	379.1 watt
p^j	42.5 watt
p^i	0.825 watt/Gbit/s
m^l	8
m^i	3
C	40 Gbit/s
α	0.95

4) 网络最大路由跳数 H 的取值分别为 1、2、3、4、5 和 ∞ ，其中， $H = \infty$ 表示不限制网络的最大路由跳数。

本文对网络规模、业务需求和网络最大路由跳数的多种取值的各种组合（共 $5 \times 9 \times 6 = 270$ 种）分别进行实验。为了减小重力模型生成业务需求过程中随机因素可能对实验结果的影响，以上每种组合的实验重复

进行 10 次（共 2700 次），每次实验的业务需求使用不同的随机数种子生成，结果取平均值。为了公平地比较 CBR-GVTD 与 GAMS/CPLEX、LU、UB 和 LR 的实验结果，这些算法都使用完全相同的实验参数（包括网络规模、业务需求和网络设备相关参数）。网络最大跳数 H 为 CBR-GVTD 算法所独有，在 5.1 节和 5.2 节的实验中，网络最大跳数都不受限，即 $H = \infty$ ，在 5.3 节中将展示 H 的其他取值下的实验结果，并探讨如何设定 H 的值。

4.1 网络功耗

GVTD 的目标是最小化网络的功耗，考虑到网络业务需求的变化，在理想的情况下，网络功耗应该与网络业务需求成正比，即网络应该具有能量匀增特性^[9]。为了评估 CBR-VTD 算法求得的网络功耗与 GVTD 形式化模型最优值的差距，首先对规模较小（即 10 个节点）的网络进行模拟实验，使用 GAMS/CPLEX 计算 GVTD 模型的最优解（相对误差限为 2%），并将求得的网络功耗与 CBR-VTD 算法、LR 算法以及网络功耗的下界(LB)和上界(UB)进行比较，结果如图 1 所示。在图 1 中，网络功耗的上界 (UB) 和下界 LB，CBR-GVTD、LR 算法和 GAMS/CPLEX 求得的网络功耗与业务需求呈现几乎线性增长的函数关系，即表现出一定的能量匀增特性。其中，网络功耗下界(LB)与最优解 CPLEX 比较接近，这表明 LB 作为网络功耗下界是十分紧密的，因此，在网络规模更大时（此时，无法用 GAMS/CPLEX 获得最优解），LB 可用于评估 CBR-GVTD 算法的性能。在图 1 中，CBR-GVTD 介于 LR 与 CPLEX 之间，这表明 CBR-GVTD 算法求得的网络功耗比 LR 算法的更低且更接近于 GAMS/CPLEX 求得的最优值。

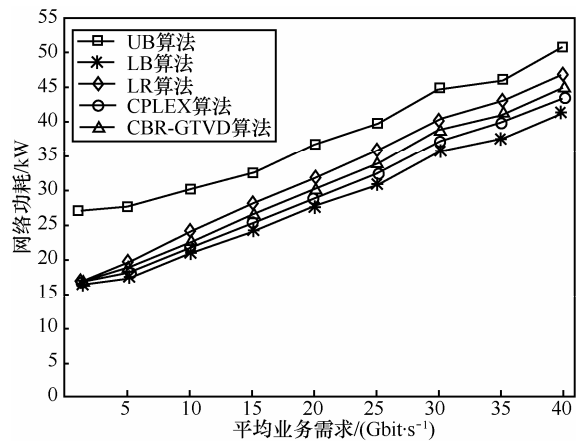


图 1 10 个节点的网络功耗：CBR-GVTD 算法与 CPLEX、LR、LB 和 UB 算法的比较

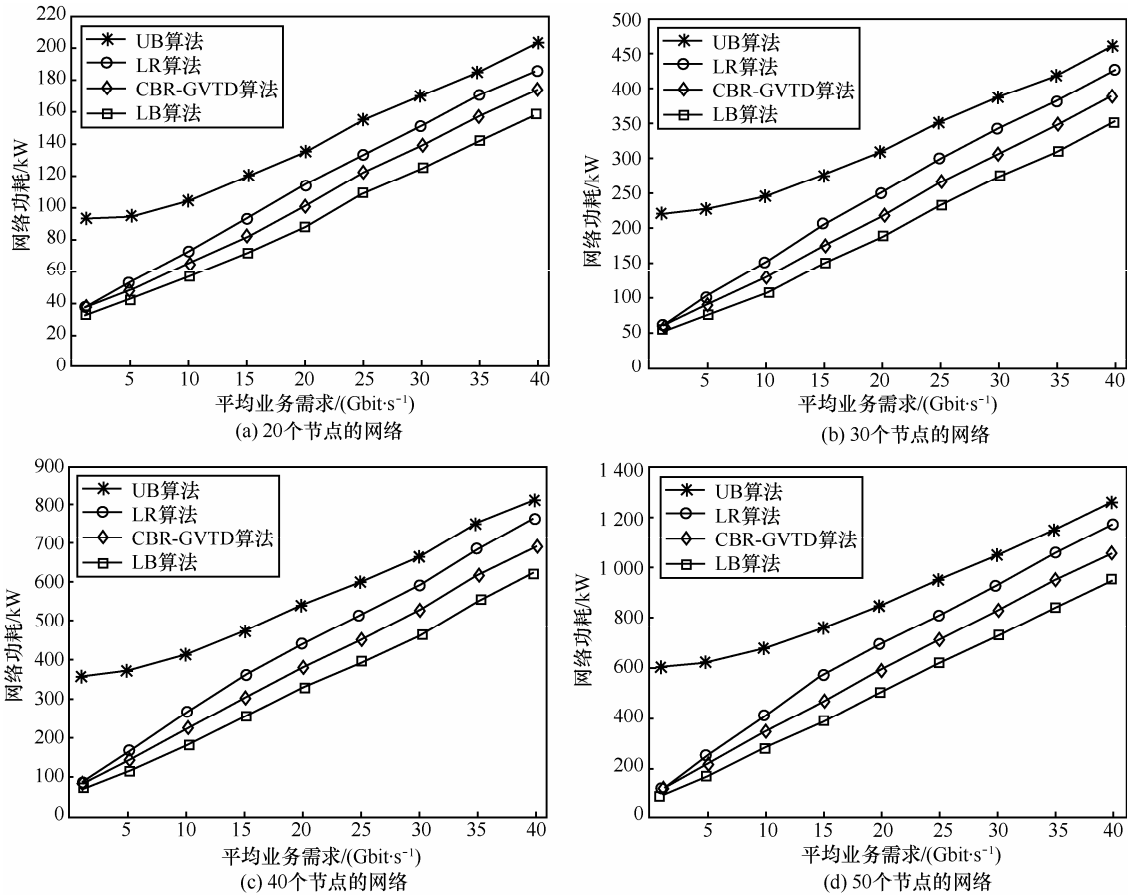


图 2 不同规模的网络功耗: CBR-GVTD 算法与 LR、LB 和 UB 算法的比较

为了进一步评估在其他规模的网络上, CBR-GVTD 算法求得的网络功耗的性能, 本文分别在节点数为 20、30、40 和 50 的网络上进行模拟实验, 实验结果如图 2 所示。由图 2 可知, 无论网络节点数为多少, CBR-GVTD 都位于 LR 与 LB 之间, 这说明 CBR-GVTD 算法求得的网络功耗始终比 LR 算法的更低, 更接近于网络功耗下界(LB)。总之, 在网络功耗方面, 图 1 和图 2 表明 CBR-GVTD 算法具有比 LR 算法更优的性能, 能够获得接近最优值的网络功耗。Internet 按照峰值需求供应网络资源(即超额供给)导致网络资源的平均利用率较低, 因此, 在大多数时间里业务需求都处于非高峰期, 图 3 为各种规模的网络在业务非高峰期时 CBR-GVTD 算法节省的最大功耗, 即 CBR-GVTD 算法在业务低峰期(1 Gbit/s)时与业务高峰期(40 Gbit/s)时相比节省的网络功耗。由图 3 可知, 在业务非高峰期时, CBR-GVTD 算法最多可节省 62%~90% 的网络功耗, 且该比例随着网络规模的增大不断上升。

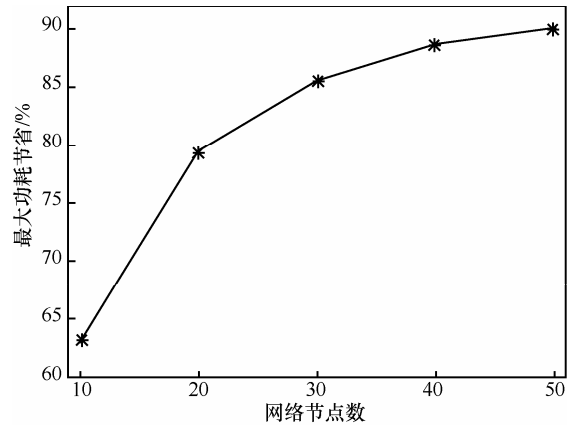


图 3 不同规模网络在业务非高峰期时节省的最大功耗

4.2 网络资源利用率

GVTD 降低网络功耗的一个主要原因在于提高网络资源的利用率, 下面从资源利用率方面评估 CBR-GVTD 算法的性能。图 4 为 CBR-GVTD 算法和 LR 算法在 30 个节点的网络上求得的网络资源平均利用率, 由于节点数为 10、20、40 和 50 的网络的资源利用率与图 4 类似, 本文仅以 30 个节点的网络为例说

明。在图 4 中，CBR-GVTD 算法和 LR 算法求得的网络资源的平均利用率都随着网络业务需求的增加而增大，而且，接口和线卡的平均利用率都保持在较高的水平。对于机框和线卡的平均利用率，CBR-GVTD 算法和 LR 算法差别不大，但是，CBR-GVTD 算法对应的接口平均利用率比 LR 算法更高，基本维持在 80%~90% 的范围内，而 LR 算法对应的接口平均利用率基本在 70%~80% 左右，这解释了为什么 CBR-GVTD 对应的网络功耗小于 LR 算法的网络功耗。总之，图 4 表明在资源利用率方面，CBR-GVTD 算法的性能也优于 LR 算法，CBR-GVTD 算法可以获得很高的资源利用率。

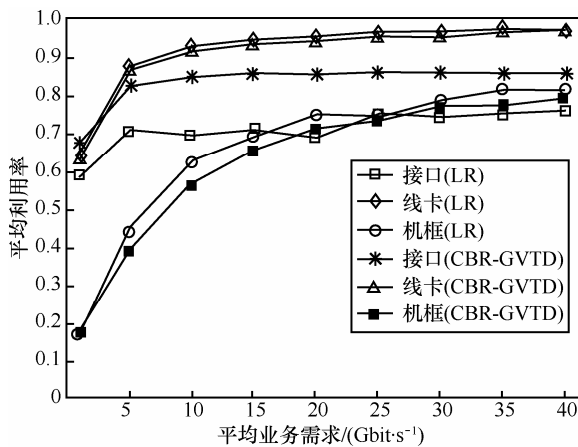


图 4 网络资源利用率：CBR-GVTD 算法与 LR 算法比较

4.3 路由性能

通常，业务的传输路径越长，端到端的网络时延也越大，因此，路径长度（即路由跳数）可用于评估 CBR-GVTD 算法的路由性能。图 5 为 30 个节点的网络使用 CBR-GVTD 算法和 LR 算法进行 10 次重复实验得到的平均跳数、最大跳数和每次实验的最大跳数的平均值（即平均最大跳数），图 5 中，分别用 avgHop、maxHop 和 avgMaxHop 标识，CBR-GVTD 算法和 LR 算法的平均跳数都很小（都小于 3），其中，CBR-GVTD 算法的平均跳数略大于 LR 算法的，且都随着网络业务需求的增加而减小，表现出较好的路由性能。由图 5 可知，总体上，最大跳数也随网络业务需求的增加而减小。由于 maxHop 标识的为 10 次重复实验中的最大跳数，可能受随机因素的干扰，因而表现出较大的起伏；而 avgMaxHop 标识的是平均最大跳数，因此更加平滑，反映出最大跳数随网络业务需求增加而变小的趋势。此外，在图 5 中，CBR-GVTD 算法的最大跳

数在 6~10 的范围内变动，明显高于 LR 算法的最大跳数（变化范围为 3~7）。图 6 为不同节点的网络在各种大小的业务需求下的平均跳数、最大跳数和平均最大跳数。在图 6 中，LR 算法的平均跳数、最大跳数和平均最大跳数随网络节点数的增加而减小，CBR-GVTD 算法的平均跳数随着网络节点数的增加几乎保持不变，但是最大跳数和平均最大跳数随着网络节点数的增加而增大。

从平均跳数来看，图 5 和图 6 都表明 CBR-GVTD 算法具有接近 LR 算法的路由性能，但是最大跳数却比 LR 算法的大，且随着网络节点数的增加而增大。图 7 进一步描述了 30 个节点的网络运行 CBR-GVTD 算法的跳数分布。在图 7 中，绝大部分路径的跳数都很小，只有极少数路径的跳数较大。随着网络业务需求的增加，跳数较小的路径所占比例逐渐增大。值得注意的是，在以上实验中，CBR-GVTD 算法的网络最大跳数都没有限制，即实验参数 $H = \infty$ ，因此，实验结果也代表了 CBR-GVTD 算法的最坏路由性能。通过参数 H ，可对 CBR-GVTD 算法的最大跳数进行控制。本文对节点数分别为 10, 20, ..., 50 的网络，使用 CBR-GVTD 算法在 $H = 1, 2, 3, 4, 5$ 时进行实验，并用 LR 算法和下界 LB 作比较。由于不同节点的网络的实验结果相似，下面仅以 30 个节点的网络为例说明，网络功耗如图 8 所示。在图 8 中，CBR-GVTD 算法在 $H = 3, 4, 5, \infty$ 时的网络功耗基本相同，且小于 LR 算法的功耗。这表明 CBR-GVTD 算法可在网络功耗基本不变的情况下减小网络的最大跳数，同时获得比 LR 算法更低的网络功耗和更好的路由性能。

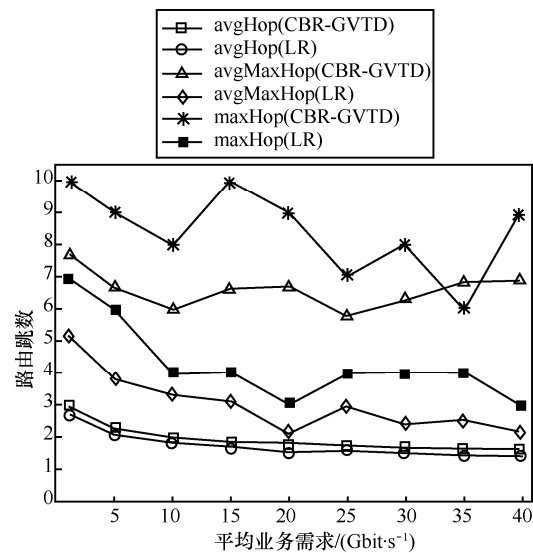


图 5 路由跳数与业务需求的关系：CBR-GVTD 算法与 LR 算法比较

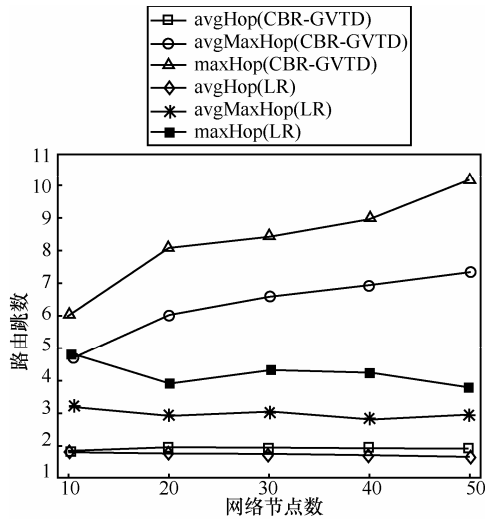


图 6 路由跳数与网络规模的关系：CBR-GVTD 算法与 LR 算法比较

当最大跳数 H 进一步减少时，通过图 8 可以知道，当 $H=2$ 时，网络节点间的平均业务需求为 1 Gbit/s 和 5 Gbit/s 时，CBR-GVTD 算法的功耗比 $H=3,4,5,\infty$ 时略高，但随着网络业务需求的增大， $H=2$ 时的网络功耗与 $H=3,4,5,\infty$ 时的功耗基本相同。由图 5 可知，当 $H=\infty$ ，网络节点间的平均业务需求为 1 Gbit/s 和 5 Gbit/s 时，CBR-GVTD 算法的平均跳数大于 2，因此，在图 8 中，最大跳数限定为 2（即 $H=2$ ）时，网络功耗会有所增加。当 $H=1$ 时，此时网络的所有业务都使用单跳路由，CBR-GVTD 算法的网络功耗即为式(14)的网络功耗上界值 UB。因此，通过调节参数 H 的取值，可以实现网络功耗与路由性能的折衷，为了保证 CBR-GVTD 算法的性能，建议参数 H 不应该小于 $H=\infty$ 时 CBR-GVTD 算法的平均跳数。

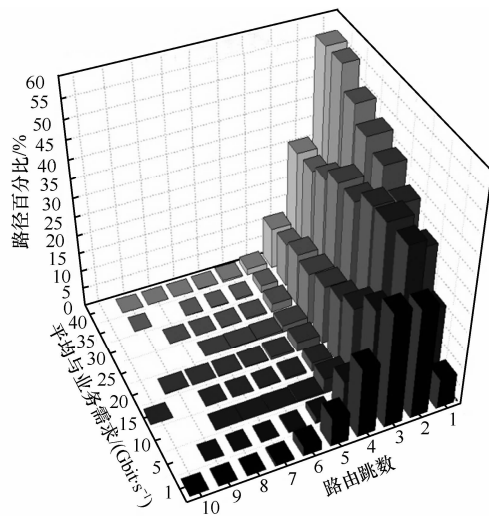


图 7 CBR-GVTD 算法在 30 个节点的网络的跳数分布

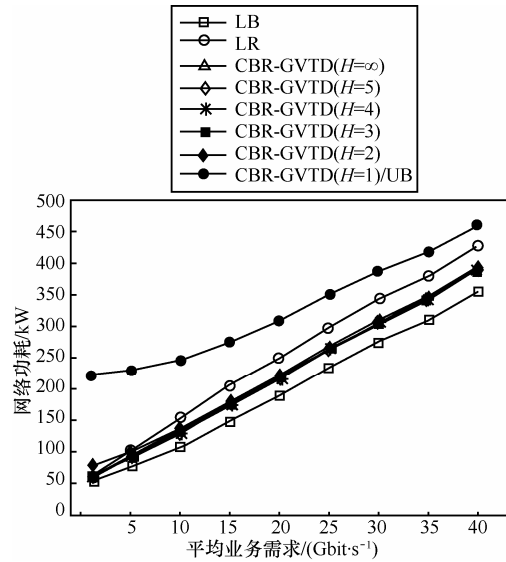


图 8 CBR-GVTD 算法在 H 不同取值下的网络功耗

5 结束语

本文研究 Internet 核心网的绿色虚拟拓扑设计 (GVTD) 问题。GVTD 通过业务汇聚提高网络资源利用率、按需配置网络资源实现多粒度睡眠和动态建立网络的虚拟拓扑，以达到降低网络能耗的目的。笔者对 GVTD 问题进行了形式化描述，建立了 GVTD 模型。由于 GVTD 问题是 NP 难的，笔者提出一个基于约束路由的启发式求解算法，即 CBR-GVTD 算法。CBR-GVTD 算法以网络功耗的下界为基础构造 GVTD 问题的解，使用 CBR-Dijkstra 算法以保证链路容量约束和网络最大路由跳数约束。CBR-GVTD 算法按照从大到小的顺序处理网络的业务需求：对大的业务需求建立点到点的传输通道并进行单跳路由，从而确定网络的虚拟拓扑；对小的业务需求使用 CBR-Dijkstra 算法在虚拟拓扑的剩余带宽上进行多跳路由，通过业务疏导^[19]提高网络资源利用率。CBR-GVTD 算法按需配置网络资源，动态地建立虚拟拓扑，实现了虚拟拓扑对网络业务需求的动态适应，并且可通过调节网络最大路由跳数来实现对路由性能的控制。在不同大小的网络、不同大小的业务需求下，从网络功耗、资源利用率和路由性能 3 个方面通过模拟实验对 CBR-GVTD 算法进行了性能评估。实验结果表明，CBR-GVTD 算法可在优异的业务路由性能和很高的资源利用率的条件下，获得接近 GVTD 模型最优值的解；在业务非高峰期时，对于 10 和 50 个节点的网络最多可分别降低 62% 和 90% 的网络功耗，该比例随着网络规模的增大而上升。

参考文献：

- [1] Telegeography[EB/OL].<http://www.telegeography.com/product-info/gig/index.php>, 2012.
- [2] BALIGA J, HINTON K, TUCKER K R S. Energy consumption of the internet[A]. Proc of COIN-ACOFIT[C]. Melbourne, AU, 2007. 1-3.
- [3] YUN D, LEE J. Research in green network for future Internet[J]. Journal of KIISE, 2010, 28(1):41-51.
- [4] LANGE C. Energy-related aspects in backbone networks[A]. Proceedings of 35th European Conference on Optical Communication (ECOC2009)[C]. Wien, AU, 2009.1-8.
- [5] LIN C, TIAN Y, YAO M. Green network and green evaluation: mechanism, modeling and evaluation[J]. Chinese Journal of Computers, 2011, 34(4): 593-612.
- [6] HLAVACS H, COSTA G D, PIERSON J. Energy consumption of residential and professional switches[A]. Int Conf on Computational Science and Engineering (CSE'09)[C]. Vancouver, Canada, 2009. 240-246.
- [7] MELLAH H, SANSO B. Review of facts, data and proposals for a greener internet[A]. Proc of Sixth International Conference on Broadband Communications, Networks, and Systems[C]. Madrid, Spain, 2009.1-5.
- [8] BARROSO L A, HOLZLE U. The case for energy-proportional computing[J]. Computer, 2007, 40(12): 33-37.
- [9] CHABAREK J, SOMMERS J, BARFORD P, *et al.* Power awareness in network design and routing[A]. Proc of the 27th IEEE Conference on Computer Communications (INFOCOM'08)[C]. Phoenix, AZ, USA, 2008.457-465.
- [10] ZHANG M, YI C, LIU B, *et al.* GreenTE: power-aware traffic engineering[A]. IEEE Int Conference on Network Protocols (ICNP)[C]. Kyoto, Japan, 2010.21-30.
- [11] LEE S S W, TSENG P K, CHEN A. Link weight assignment and loop-free routing table update for link state routing protocols in energy-aware internet[J]. Future Generation Computer Systems, 2011, 28(2): 437-439.
- [12] AHMAD A, BIANCO A, BONETTO E, *et al.* Power-aware logical topology design heuristics in wavelength-routing networks[A]. 2011 15th International Conference on Optical Network Design and Modeling (ONDM)[C]. Bologna, Italy, 2011.1-6.
- [13] COIRO A, LISTANTI M, SQUARCIA T, *et al.* Energy-minimized virtual topology design in IP over WDM backbone networks[J]. Optoelectronics, IET, 2012,6(4): 165-172.
- [14] BERTHOLD J, SALEH A A M, BLAIR L, *et al.* Optical networking: past, present, and future[J]. Lightwave Technology, 2008, 26(9): 1104-1118.
- [15] SIVARAMAN V, VISHWANATH A, ZHAO Z, *et al.* Profiling per-packet and per-byte energy consumption in the NetFPGA gigabit router[A]. IEEE INFOCOM 2011 Workshop on Green Communications and Networking[C]. Shanghai, China, 2011.331-336.
- [16] JENSEN P A, BARNES J W. Network Flow Programming[M]. Beijing: Science Press, 1988.
- [17] GAMS/CPLEX[EB/OL].<http://www.gams.com/solvers/solvers.htm#CPLEX>, 2012.
- [18] NUCCI A, SRIDHARAN A, TAFT N. The problem of synthetically generating IP traffic matrices: initial recommendations[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(2) :19-32.
- [19] YETGINER E, ROUSKAS G N. Power efficient traffic grooming in optical WDM networks[A]. Proceedings of the 52nd Annual IEEE Global Telecommunications Conference Workshop (GLOBECOM'09)[C]. Honolulu, Hawaii, USA, 2009.1-6.

作者简介：



伍元胜（1986-），男，四川广安人，四川大学博士生，主要研究方向为IP网络能耗优化、嵌入式实时系统。

郭兵（1970-），男，山东泰安人，博士，四川大学教授、博士生导师，主要研究方向为嵌入式实时系统、绿色计算和软件工程。

沈艳（1973-），女，湖南永州人，博士，成都信息工程学院教授，主要研究方向为分布式测量系统、嵌入式系统开发、无线传感器网络。

王继禾（1986-），男，陕西西安人，四川大学博士生，主要研究方向为高能效的可定制异构多核体系结构、嵌入式实时系统。

刘啸滨（1987-），男，回族，河北唐山人，四川大学博士生，主要研究方向为软件体系结构的能耗建模与优化、嵌入式实时系统。