ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.21, No.3, March 2010, pp.528−538
doi: 10.3724/SP.J.1001.2010.03489

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 一种保证时延的关键流路由调整算法*

裴育杰+, 王洪波, 程时端

(北京邮电大学 网络与交换技术国家重点实验室,北京 100876)

## Delay-Guaranteed Key Flow Routing Adjustment Algorithm

PEI Yu-Jie+, WANG Hong-Bo, CHENG Shi-Duan

(State Key Laboratory of Networking and Switching, Beijing University of Posts & Telecommunications, Beijing 100876, China)

+ Corresponding author: E-mail: yjpei51@gmail.com, http://www.bupt.edu.cn

**Pei YJ, Wang HB, Cheng SD. Delay-Guaranteed key flow routing adjustment algorithm. *Journal of Software*, 2010,21(3):528−538.** http://www.jos.org.cn/1000-9825/3489.htm

**Abstract**: The unevenly distributed traffic of Internet may lead to congestion and underutilization of network resources. Current methods of routing adjustment for load balancing often get a new path with excessively long length that can exacerbate the quality of service. This paper presents a new flow routing adjustment algorithm named LCBA (length-constrained most balanced algorithm), which can decrease the maximal bandwidth utilization rate of network with delay guaranteed. The experiment based on Abilene2 network topology and traffic data shows that LCBA can mitigate the congestion of backbone network effectively and the maximal bandwidth utilization rate can drease nearly 50% at most. This paper also evaluates the algorithm with an emulation method. Compared with current methods, the new algorithm can satisfy the demand of not only the length of key flow's path but also the maximal bandwidth utilization rate. Moreover, the theoretical analysis proves that the computational complexity of LCBA is $O(N^2\log N)$, which is better than that of a majority of methods nowaday.

**Key words**: network management; load balancing; quality of service; routing adjustment; max-min

**摘 要**: 互联网中流量分布的不均衡会导致网络拥塞、网络资源得不到有效利用.而为了负载均衡,现有算法调整路由又会造成新路径过长,服务质量降低.提出了一种路由调整算法 LCBA(length-constrained most balanced algorithm),在保证时延的基础上降低网络最大带宽利用率.基于 Abilene2 网络拓扑和真实流量的实验结果表明,LCBA 算法能够有效缓解骨干网拥塞,最多可以降低最大带宽利用率近 50%.仿真实验结果显示:与现有算法相比,该算法能够同时满足关键流路径长度和最大带宽利用率两方面的要求.此外,算法复杂度为 $O(N^2\log N)$,好于大部分路由调整算法.

**关键词**: 网络管理;负载均衡;服务质量;路由调整;最大最小化

**中图法分类号**: TP393 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.90604019, 60502037 (国家自然科学基金); the Specialized Research Fund for the Doctoral Program of Higher Education No.200800131019(高等学校博士学科点专项科研基金); the New Century Excellent Talents in University No.NECT-07-0109 (新世纪优秀人才支持计划);the National High-Tech Research and Development Plan of China under Grant Nos.2006AA01Z235, 2007AA01Z206 (国家高技术研究发展计划(863))

Received 2008-05-16; Accepted 2008-10-08; Published online 2009-04-10

近年来,以流媒体和 P2P 文件下载为代表的新业务的出现带动了互联网的快速发展.在为人们带来方便的同时,它们也对底层网络提出了更高的要求.视频业务需要网络提供实时性和可靠性的保证,网络拥塞对其影响往往是致命的;P2P 下载的自身特点决定了其一般流量较大,并且伴随着用户终端数量的增加,网络内部的流量呈指数增长.面对这样的压力,网络管理者单纯增加带宽是不够的,如何使网络中流量分布得更加合理才最为重要.因此,如何在满足用户服务质量要求的同时使网络流量分布更加合理,避免拥塞的发生,是网络管理者亟待解决的一个重要问题,也是近年来网络研究的热点.

我们知道,目前自治域(autonomous system)内部路由主要依赖 OSPF(open shortest path first)[1]等以最短路径算法为基础的链路状态协议.这类协议实现简单,不需要在中间节点维持连接所需的状态信息,因此具有很好的可扩展性,并且能够保证在自治域边缘节点之间建立最短的转发路径.但其缺点也十分明显:这类协议根据最短路径算法选择路径,得到的路径只与拓扑相关而与网络中的流量无关,因此很容易造成某些链路拥塞[2].另外,OSPF 协议只能从通往同一目的地的不同路径中选择优先级最高的那一条转发,除非存在多条优先级相同的路径,否则不能实现负载均衡.

目前,运营商在网络中普遍采用过量提供带宽的方法减少拥塞的发生[3].这种方法虽然能够在一定程度上缓解拥塞,但却不能从根本上解决问题.互联网中的流量变化十分复杂,经常会在一个大的时间尺度内波动[4].因此,过量提供带宽方法造成的结果就是网络中的一部分链路将要或者已经发生拥塞,而另外一些链路却处于空闲而得不到充分利用,这无疑加大了网络建设的成本.同时,由于互联网业务的迅猛发展导致骨干网中的流量增长过快,网络建设的速度很难跟上流量增长的速度,拥塞仍然不能得到有效避免.造成上面现象的根本原因是现有的路由协议没有把用户服务质量、网络拓扑和流量分布三者有机结合起来,导致大量流量集中到少数关键路径上,而剩余链路却很空闲.

针对以上问题,本文提出一种路由调整算法,在拥塞将要或已经发生时,调整关键流的路由,使其绕开拥塞链路,从而达到缓解拥塞的目的.同时,为了保证关键流的服务质量,我们还对关键流被调整后的路径长度做出一定限制.与传统方法相比,本文中的方法更加灵活,综合考虑了服务质量、网络拓扑和流量分布之间的关系,在保证了关键流服务质量的同时,最大程度地均衡了网络流量.同时,该算法计算复杂度不高,可以用于网络的实时管理.

本文第 1 节介绍相关工作.第 2 节描述问题并对问题建立模型.第 3 节提出关键流调整算法 LCBA(length-constrained most balanced algorithm).第 4 节进行实验和仿真的验证,并且与其他算法进行比较.最后一节总结全文.

## 1 相关工作

骨干网中大量分组的丢失会严重降低网络服务质量,目前的大部分解决方法都是要为分组重新选择路由.导致大量分组丢失的异常事件主要包括两类:网络故障和网络拥塞.文献[5]为自治域内的链路设定几组不同的权值,根据不同的权值分别计算出多套备用的路由方案,每个节点或链路故障都能够找到一组方案使其孤立.换句话说,系统为每个故障点提供备用路由方案,当故障发生时,网络中的节点以某种方式获取故障位置信息,统一执行使故障点孤立的那套路由方案.这种方法能够有效地解决故障造成的路由不可达,但是要求统一管理全网节点,并且可能会因为信息传达的不一致性造成路由抖动.文献[6]中提出的方法是当网络发生故障时,由分组路径上最靠近故障位置的节点把故障位置信息记录到分组头部并且重新计算下一跳接口,下游节点根据分组头部的故障位置信息重新选路.这种方法需要修改分组头,并且在发生故障时不查询路由表,而是对每个分组重新计算路由,因此复杂度高,不易实现.Zhong 等人[7]设计了一种路由方案,使得路由器在选择下一跳时,不但要考虑目的地址,还要考虑分组是从哪个接口进入,根据流入接口的信息推测网络中故障发生的位置,然后根据故障的位置信息重新计算路由.

上面的这些研究工作解决的问题都是节点或链路发生故障时网络的快速恢复,也就是说,在网络出现故障时快速调整路由,保证分组不被丢弃.这些方法只强调路由的可达性,不考虑网络流量的均衡,如果将其用于处

理拥塞事件,很容易造成新的拥塞.因此,研究人员又提出了一些专门用于处理拥塞事件的方法.

文献[8]针对每个目的地址,对每个节点按照距离递减的原则配置多个下一跳接口.用户感觉到服务质量下降时,产生一个随机数作为标记写入分组头部,该标记决定了分组所走的路由.按距离递减的原则配置下一跳接口的缺点是:它不能保证网络连通时一定可以找到一条可用路径,同时,由用户决定变更路由的方法也很难用于实际网络.文献[9]中的 KSP 算法为每条流选择前 $K$ 条最短路径,当发生拥塞时,按顺序检查这 $K$ 条路径,直到选出一条不经过拥塞点的路径为止.但是这种方法计算量较大,并且不能够保证一定可以选出符合要求的路径.文献[10,11]都是为每一条流选择两条不相交的路径,一条是主路径,另一条是备用路径,并且对路径长度作了一定限制.这种方法得到的两条路径都很固定,不能结合网络当前的流量分布选择路由.文献[12]中提出一种算法,计算分组路径上应该经过的一些中间节点,而这些中间节点之间仍然通过最短路径优先算法计算路由.该文分析指出,当中间节点的数量为 1 时,算法的效果最佳.然而,由于该文中提出的 $N$-hub 问题属于 NP 完全问题,只能求得近似解.Kar 等人[13]提出了一种最小干扰路由算法,为每个流选路时都选择对未来可能的流影响最小的路径.该算法要为每条流选路,并且要计算每对边缘节点的关键链路,因此复杂度较高.

拥塞导致网络传输质量严重降低,因此拥塞发生时需要网络及时做出调整.一方面,要使流量在网络中的分布更加均匀,缓解当前的拥塞并降低未来再次发生拥塞的概率;另一方面,服务质量主要由时延决定,时延越大则服务质量越差.端到端时延包括 3 部分:排队时延、发送时延和传输时延,骨干网中路由器接口转发速率极快,因此发送时延可以忽略.而当拥塞很少发生时,端到端时延的绝大部分由传输时延组成,而传输时延与路径长度成正比.因此,在调整过程中需要将路径长度控制在合理的范围内;另外,为了保证管理的实时性,算法复杂度不能太高.可以看到,现有方法都不能同时满足以上 3 个方面的要求.结合以上要求以及现有文献的不足之处,本文设计了一种实时路由调整算法,在拥塞将要或已经发生时为关键流调整路由,在满足关键流路径长度要求的条件下最大限度地均衡网络流量分布,缓解拥塞,使网络能够提供更好的服务.

## 2 问题建模

### 2.1 网络模型

用有向图 $G(V,E)$表示自治域内部拓扑,其中 $V$ 代表节点集合,$E$ 代表链路集合,节点和链路的数量分别用 $M=|V|$和 $N=|E|$表示.一条$(s\text{-}t)$路径 $P$ 由一组不重复的链路$(l_1,\ldots,l_n)$组成,$s$ 和 $t$ 分别是路径 $P$ 的起点和终点.

每条链路 $l\in E$ 对应一个长度参数,用 $d_l$ 表示,而路径 $P$ 的长度 $D_P$ 由组成路径的所有链路长度相加获得,即 $D_P=\sum_{l\in P}d_l$ .要调整的关键流 $F$ 的起点是 $s$,终点是 $t$.由于 $F$ 的流量一般较大,对所流经链路的带宽利用率的影响不能忽略.分别用 $u_l^F$ 和 $u_l^0$ 表示流 $F$ 经过和不经过链路 $l$ 时,$l$ 的链路带宽利用率.

### 2.2 问题描述

网络中的流量分布是不均衡的,在某些链路发生拥塞时,另外一些链路可能正处于空闲状态.因此,在网络将要发生拥塞时,可以采用对某些流量变更路由的方法.这样做,一方面改善了拥塞链路上流的服务质量,另一方面能够提高网络的利用率和吞吐量.互联网中的流大小服从重尾分布[14−16],少数字节数较大的流占据了大部分流量,同时,大部分情况下,骨干网中多处链路同时发生拥塞的可能性并不大[2].因此,本文的思路是:在拥塞将要或已经发生时,选择一条通过拥塞点的流量较大的汇聚流,重新调整其路由.我们称这条被重新调整路由的流为关键流.由于现有研究工作[17−19]已经能够识别骨干网中的大流,因此本文假定关键流已知,只考虑如何为关键流调整路由.

为关键流 $F$ 选择新路由的过程需要考虑两个原则问题:一要使网络中流量分布更加合理,二要考虑用户感受到的服务质量.一方面,为了减小拥塞发生的概率,提高网络利用率,网络流量的分布应该做到越均衡越好.我们用流量工程研究中常用的最小化最大带宽利用率(max-min)[12]原则衡量网络流量分布的均匀程度.所谓的最小化最大带宽利用率原则是指,当存在多种路由方案的时候,选择使网络中最大的带宽利用率最小的那种方案.另一方面,用户感受到的服务质量参数包括时延、时延抖动和丢包率.假设第一个问题能够被很好地解决,拥塞

很少发生,那么上面 3 个参数中时延抖动和丢包的影响可以忽略.时延主要由排队时延和传输时延组成,拥塞的减少意味着传输时延起主要作用,而传输时延和路径长度成正比,因此,本文考虑关键流被调整后的路径长度.并且我们认为,路径长度只需要满足一定条件就可以,不一定选择最短路径.这是因为在很多情况下,只要路径长度在用户能接受的范围内,差别并不明显.

我们把关键流路由调整问题抽象为下面的 LCBP(length-constrained most balanced path)问题.

LCBP 问题:在一个有向图 $G$ 中,为一条被调整的关键流 $F$ 指定一个长度上限 $T_L$,并且找到一条($s$-$t$)路径 $P$,使其满足下面条件:

1) $D_P \le T_L$;

2) $\forall P', D_{P'} \le T_L$,有 $\max\{X(i,P,F)\} \le \max\{X(i,P',F)\}, 1 \le i \le N$,其中

$$X(i,P,F)=\begin{cases} u_{l_i}^F, & l_i \in P \\ u_{l_i}^0, & l_i \notin P \end{cases}.$$

上面的限制条件 1)保证了每个关键流的路径长度不超出上限;而条件 2)确保路径 $P$ 是所有满足条件 1)的路径中,使最大的链路带宽利用率最小的那条.

## 3 算 法

我们提出一种时间复杂度为 $O(N^2 \log N)$的算法,用于求解 LCBP 问题.算法的核心思想是将 LCBP 问题转化成最短路径问题.

### 3.1 定 理

**定理 1**. 如果集合 $W=\{P_j | 1 \le j \le |W|\}$是满足 LCBP 问题条件的路径集合,$W \ne \varnothing$,令 $T_U=\max\{X(i,P_1,F)\}=\ldots=\max\{X(i,P_{|W|},F)\}$,则图 $G'=G \setminus \{L_{T_U}^+\}$ 中($s$,$t$)之间的最短路径 $P' \in W$,其中 $L_{T_U}^+ = \{l \mid u_l^F > T_U\}$.

证明:

一方面,$T_U=\max\{X(i,P_j,F)\}$,因此,对于$\forall l \in P_j$ 有 $l \notin L_{T_U}^+$,所以 $l \in G', P_j$ 也是图 $G'$中的($s$,$t$)路径.又因为 $P'$是 $G'$中最短($s$,$t$)路径,所以 $D_{P'} \le D_{P_j} \le T_L$.

另一方面,$G' \subset G$,因此 $P' \subset G$.又因为 $L_{T_U}^+ = G - G'$,所以,对于$\forall l' \in P'$有 $l' \notin L_{T_U}^+$,并且 $u_{l'}^F \le T_U$.再考虑到 $\max\{u_i^0\} \le T_U$,$\max\{i,P',F\} \le T_U = \max\{i,P_j,F\}, 1 \le i \le |E|, 1 \le j \le |W|$.

综合以上分析,$P'$也满足 LCBP 问题的条件,也是 LCBP 问题的解,因此 $P' \in W$. □

从上面的分析可以看到,只要 $T_U$ 已知,我们就可以确定图 $G'$并且计算出 $G'$内的($s$,$t$)最短路径 $P'$,$P'$同时也是 LCBP 问题的解.为了提高算法的效率,用二分法判断 $T_U$ 的准确值,而下面的定理 2 用于二分法条件的判断.

**定理 2**. 若路径集合 $W=\{P_j | 1 \le j \le |W|\}$是满足 LCBP 问题的解,$W \ne \varnothing$,$T_U=\max\{X(i,P_1,F)\}=\ldots=\max\{X(i,P_{|W|},F)\}$,并且假设对于$\forall P_j \in W, \exists l \in P_j$,使得 $u_l^F = T_U$,那么

1) 图 $G'=G \setminus \{L_T^+\}$ 中($s$,$t$)最短路径 $P'$的时延 $D_{P'} \le T_L$,其中 $T \ge T_U$.

2) 图 $G'=G \setminus \{L_T^+\}$ 中($s$,$t$)最短路径 $P'$的时延 $D_{P'} > T_L$,其中 $T < T_U$.

证明:

1) 因为 $T \ge T_U$,所以对于$\forall P_j \in W$, $\forall l \in P_j$,有 $l \notin L_T^+$.因为 $L_T^+ = G - G'$,所以有 $P_j \subset G', P_j$ 也是图 $G'$中的($s$,$t$)路径.又因为 $P'$是图 $G'$中的最短($s$,$t$)路径,所以 $D_{P'} \le D_{P_j} \le T_L$.

2) 这一步用反证法,假设 $D_{P'} \le T_L$.

因为 $G'=G \setminus \{L_T^+\}$,而 $T < T_U$,所以对于$\forall l' \in G'$有 $u_{l'}^F \le T$.又因为 $P' \subset G'$,所以对于$\forall l' \in P'$,有 $u_{l'}^F \le T < T_U$.因为 $G' \subset G, P'$也是图 $G$ 中的($s$,$t$)路径.又因为 $\max\{u_l^0\} \le T_U$,所以 $X(i,P',F) \le T_U$.再结合假设 $D_{P'} \le T_L, P'$也是 LCBP 问题的解,即 $P' \in W$.因此,$\exists l' \in P'$,使得 $u_{l'}^F = T_U$,又因为 $G'=G \setminus \{L_T^+\}, l' \notin G'$,这与 $P' \subset G'$矛盾.因此假设不成立,$D_{P'} > T_L$. □

### 3.2 算法描述

结合上一节中的定理,我们提出了下面的 LCBA 算法.

LCBA 算法描述:

1. 按链路带宽利用率从高到低的顺序对图 $G$ 的所有链路排序为 $l_1,\ldots,l_N$,对应的链路带宽利用率分别为 $u_{l_1}^0,\ldots,u_{l_N}^0$ .其中,前 $K$ 条链路 $l_1,\ldots,l_K$ 满足条件 $u_{l_j}^F > u_{l_1}^0$, $1 \le j \le K$
2. $Z^+=1, Z^-=K$
3. $G' = G \setminus \left\{ L_{u_{Z^-}}^+ \right\}$
4. $P'=dijkstra(G',s,t)$
5. if $D_{P'} \le T_L$
6. $P'$就是所求的路径,算法结束
7. $P'=dijkstra(G,s,t)$
8. if $D_{P'} > T_L$
9. 无解,算法结束
10. $i = \left\lceil \dfrac{Z^+ + Z^-}{2} \right\rceil$
11. $G' = G \setminus L_{u_i}^+$
12. $P'=dijkstra(G',s,t)$
13. if $D_{P'} \le T_L$
14. if $i - Z^- = 1$
15. $P'$就是所求的路径,算法结束
16. else
17. $Z^+ = i$
18. else
19. if $Z^+ - i = 1$
20. $G' = G \setminus \left\{ L_{U_{Z^+}}^+ \right\}$
21. $P'=dijkstra(G',s,t)$
22. $P'$就是所求的路径,算法结束
23. else
24. $Z^- = i$
25. goto 10

LCBA 算法通过对迪杰斯特拉算法的迭代获得 $T_U$ 的精确值,最终得到 LCBP 问题的可行解.

### 3.3 复杂度分析

整个 LCBA 算法的主要部分分为两步,下面分别计算每一步的算法复杂度.

1) 对所有链路按带宽利用率排序,本文使用目前被认为最好的排序算法快速排序,其时间复杂度为 $O(N\log N)$.
2) 对迪杰斯特拉算法的迭代.其中,最坏情况下的迭代次数为 $\log N$,而迪杰斯特拉算法的复杂度为 $O(N^2)$.因此,这一步的算法复杂度为 $O(N^2\log N)$.

综上所述,LCBA 算法总体的时间复杂度为 $O(N^2\log N)$.

文献[9]中 KSP 算法的复杂度为 $O(KN^2\log N)$,略高于 LCBA 算法.文献[10]中的方法复杂度为 $O(MN(1/\varepsilon+\log\log M))$,由于一般情况下网络中链路数大于节点数,因此该复杂度略低于 LCBA 算法的复杂度.文献[13]中的

算法需要计算每对节点的关键链路,因此需要执行 $MN(M-1)$次计算最大流的算法,而用于计算最大流的预留推进算法的复杂度为 $O(M^2\sqrt{N})$,因此,算法总体的复杂度高达 $O(M^4N^{3/2})$.可以看出,LCBA 算法的复杂度好于大部分路由调整算法.

### 3.4 实 现

本文中的算法在实际网络中有多种方式可以实现.IPv4 协议[20]中的源路由选项规定可以在分组头中写入分组路径上必须经过的节点信息,沿途每个节点都通过该信息决定分组转发的下一跳节点.但是,受 IP 分组选项长度、IP 分片以及路由器开销等条件的限制,显式路由在骨干网中并没有得到广泛支持.另一种改变路由的方法是利用多协议标签交换(MPLS)[21].分组进入 MPLS 域时在头部打标记,在网络内部沿标签交换路径转发,这种方法目前在骨干网中得到了广泛应用.因此,本文假定 LCBA 算法可以由 MPLS 协议最终实现.

## 4 实验仿真

我们使用两种方法验证 LCBA 算法的效果.第 4.1 节使用 LCBA 算法对 Abilene2 网络(如图 1 所示)中流量矩阵数据[22]的路由作调整.第 4.2 节利用文献[13]中的拓扑 MIRANet(如图 2 所示)作仿真,并且选取几种具有代表性的路由调整算法与 LCBA 算法作比较.

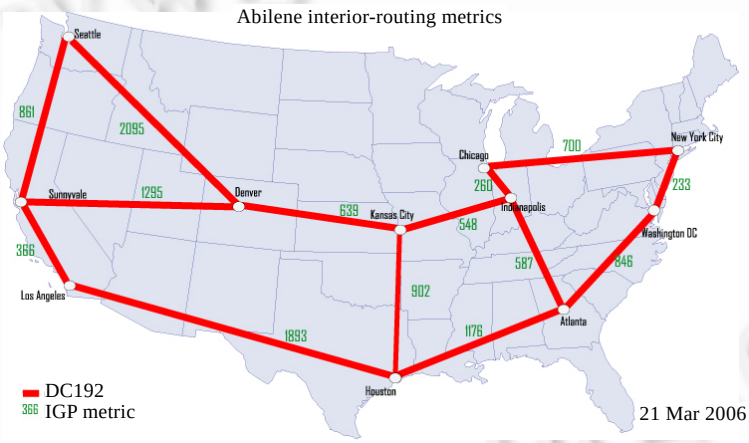



Fig.1 Topology of Abilene2

图 1 Abilene2 拓扑

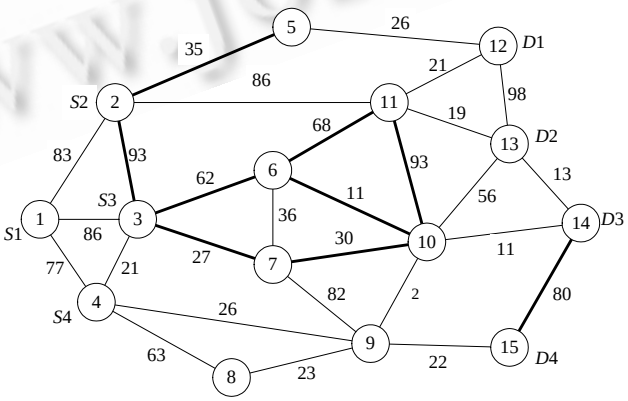



Fig.2 Topology of MIRANet

图 2 MIRANet 拓扑

### 4.1 Abilene2数据仿真

受条件所限,我们无法在真实网络中验证 LCBA 算法的有效性,但是却可以通过对真实网络数据的模拟获得近似的效果.本节的实验选取美国 Abilene2 网络真实流量矩阵数据和网络拓扑.

图 1 是 Abilene2 网络拓扑,图中共有 12 个节点,分别位于美国的 11 个城市,其中 Atlanta 市含有两个节点.除了 Atlanta 与 Indianapolis 之间的双向链路带宽为 2.5G 外,图中剩余链路带宽均为 10G.Texas 大学计算机系的 Zhang 等人[22]在做流量矩阵的研究工作中,通过采集 netflow 数据并加以分析的方法获得了 Abilene2 网络 6 个月的流量矩阵数据.

我们利用上面的拓扑和数据做实验,验证 LCBA 算法的有效性.受篇幅所限,本文不能列出全部实验结果.为了最大程度地减小实验数据的偶然性带来的影响,本节等间隔地选取其中 5 周的实验结果呈现出来,所选取数据的详细描述见表 1.

**Table 1** Data description of Abilene2

**表 1** Abilene2 数据描述

| Dataset | Time | Number of nodes | Number of links | Interval (m) |
|---|---|---|---|---|
| Abilene2-I | Apr. 9th, 2004~Apr. 21st, 2004 | 12 | 30 | 5 |
| Abilene2-II | May 22nd, 2004~May 28th, 2004 | 12 | 30 | 5 |
| Abilene2-III | June 26th, 2004~July 2nd, 2004 | 12 | 30 | 5 |
| Abilene2-IV | July 31st, 2004~Aug. 6th, 2004 | 12 | 30 | 5 |
| Abilene2-V | Sept. 4th, 2004~Sept. 10th, 2004 | 12 | 30 | 5 |

实验过程如下:首先获取每对边缘节点之间的流量,然后通过路由信息计算每条链路的带宽利用率,最后通过对网络中负载最大的链路上的流量作路由调整来降低网络中的最大链路带宽利用率.由于文献[22]中的流量矩阵数据的统计周期为 5 分钟,因此,我们也以 5 分钟作为路由调整的基本时间单位.具体来说,从每 5 分钟的数据中找出平均带宽利用率最高的链路,选择通过该链路的所有边缘节点之间汇聚流中流量最大的那条,将其一半流量按照 LCBA 算法调整到其他路径上.

目前已有大量关于大流检测的算法[17−19],因此这一过程是容易实现的.

实验完成后,计算 LCBA 算法对网络最大带宽利用率减小的比例.

图 3~图 7 为实验结果,横坐标是统计时间点,单位是 5 分钟,纵坐标是采用 LCBA 算法后网络最大带宽利用率减小的部分与未采用 LCBA 算法时网络最大带宽利用率的比值.

在网络状况良好的情况下,由于对新路径长度进行了限制,因此 LCBA 算法只能将最大带宽利用率降低 10%甚至更少.但是,当网络拥塞比较严重时(例如,图 3 中横坐标为 500~750 的一段时间),LCBA 算法对拥塞状况的改善效果很明显,最大带宽利用率最多可以降低近 50%.

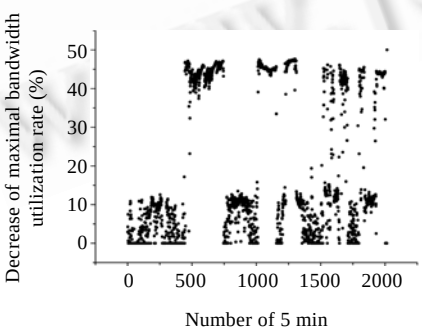



Fig.3 Experimental result of Abilene2 I

图 3 Abilene2 实验结果-I

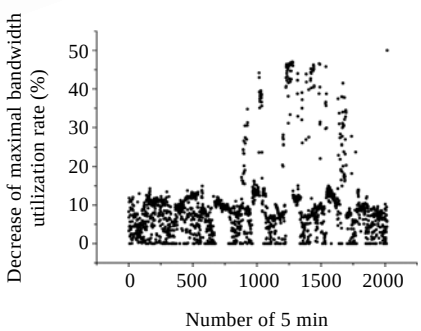



Fig.4 Experimental result of Abilene2 II

图 4 Abilene2 实验结果-II

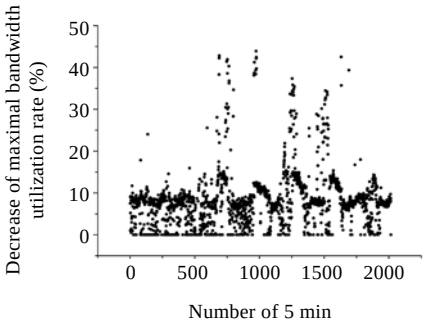



Fig.5 Experimental result of Abilene2 III

图 5 Abilene2 实验结果-III

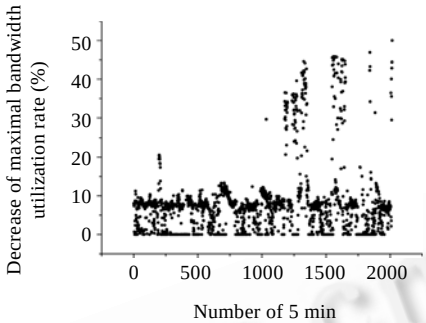



Fig.6 Experimental result of Abilene2 IV

图 6 Abilene2 实验结果-IV

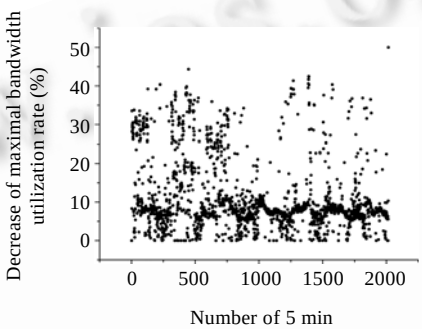



Fig.7 Experimental result of Abilene2 V

图 7 Abilene2 实验结果-V

### 4.2 MIRANet拓扑仿真

本节的仿真使用文献[13]中的网络拓扑(如图 2 所示).图中共有 15 个节点,每条链路都是双向链路,相当于容量相等的两条单向链路,其中,粗线代表的链路容量为 4 800,细线代表的链路容量为 1 200.文献[13]并没有对图 2 的拓扑定义链路长度,本文假定所有链路等长,每条链路的长度为 1.因为仿真实验需要验证 LCBA 算法在网络拥塞或者负载较高情况下的性能,因此采用与文献[13]类似的实验方法:向网络中不断加入静态流,直到网络彻底拥塞.所谓静态流,是指一旦加入就不再离开并持续到实验结束的流.流的源点从节点 1~节点 4 中随机选择,终点从节点 12~节点 15 中随机选择,流的大小服从(1,3)之间的均匀分布.一旦网络中某处链路的负载达到饱和,实验便宣告结束.所有仿真实验程序代码由 C 语言编写.

作为对比,我们选取了其他几种有代表性的路由调整算法,分别是:最短路径优先算法(简称 SP 算法)、N-hub 最短路径算法(简称 N-hub 算法)[12]、最小干扰路由算法(简称 MIRA 算法)[13].其中,最短路径优先算法使用迪杰斯特拉算法[23]求解,而 N-hub 算法选取文献[12]中 3 种算法里实验效果最好的算法 1.LCBA 算法在每条新流到达时执行,并且选取网络中所有带宽利用率最高的链路上流量最大的汇聚流作为关键流.仿真实验中,分别选择路径长度上限(下文用符号 LU 表示)为最短路径长度的 1.2 倍、1.5 倍和 2 倍.

我们对每种算法分别做 20 次实验,并且通过下面几个参数比较上面算法的差异.

1) 网络所能承载的流个数,也就是实验终止时网络中所加入的流的个数.

2) 最大流长.由于 MIRANet 网络中并没有定义每条链路的长度,因此,用路径包含的链路数表示路径的长度.

3) 加权平均流长,定义是: $L=\sum_{i=1}^{n}s_i l_i \Big/ \sum_{i=1}^{n}s_i$ .其中,$s_i$ 是流 $f_i$ 的大小,$l_i$ 是流 $f_i$ 的路径长度,$n$ 是网络所能承载

的流个数.

网络所能承载的流个数能够反映网络资源的利用率以及流量分布的均衡程度,流个数越多,流量分布越合理;而路径长度反映的是服务质量以及用户感受,长度越小越好.另外还有一个影响算法实现的重要因素,即算法复杂度,只有复杂度合理的算法才能够用于实时路由调整.

图 8 和图 9 分别是加权平均流长和最大流长曲线.从图中可以看到,当 $L_U$ 较小时,LCBA 算法的路径长度与最短路径算法接近,而随着 $L_U$ 的增加,LCBA 算法的路径长度也逐渐增加,最终介于 *N*-hub 算法与 MIRA 算法之间.MIRA 算法由于过分考虑流量分布的均衡,从而忽视了路径长度,最大流长和平均流长均明显大于其他算法.

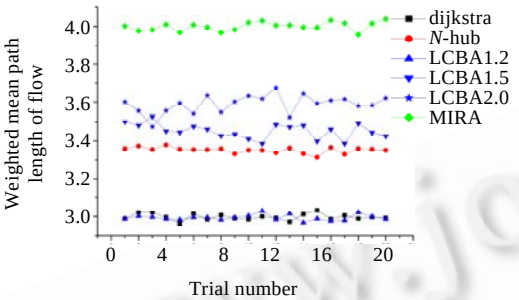



Fig.8 Weighted mean flow length

图 8 加权平均流长曲线

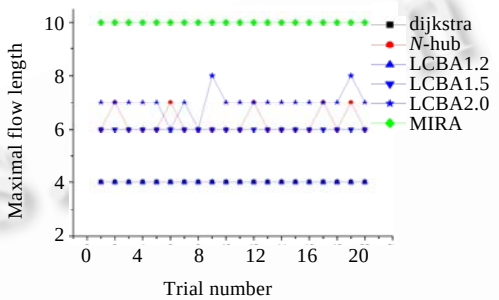



Fig.9 Maximal flow length

图 9 最大流长曲线

图 10 是流个数曲线.从图中可以看出,MIRA 算法能够加入的流最多,SP 算法最少.造成这一现象的原因是,SP 算法只考虑路径长度最短,没有让流量在网络中分布得更加均衡;而 MIRA 算法为每条流选择对以后可能加入的流影响最小的路径,因此使网络中的流量分布更加均衡.随着路径长度上限的增加,LCBA 算法中的流个数从少于 SP 算法逐渐增加到接近 MIRA 算法.LCBA 算法虽然在流数量方面不及 MIRA 算法,但是当参数设置的当时,却明显好于 *N*-hub 算法和 SP 算法.另一方面,MIRA 算法虽然能够加入更多流,但是需要为每一条流选择路由.由于 MIRA 算法中涉及算法复杂度极高的关键路径计算,因此并不适合高速网络中的实时应用.而在实际网络中,LCBA 算法只需要在网络发生拥塞时启用,而实际网络中的拥塞事件并不经常发生.并且 LCBA 复杂度较低,执行一次的开销较小,可以用于大型网络的实时应用.

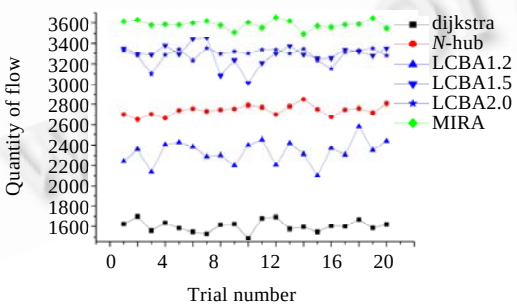



Fig.10 Quantity of flow

图 10 流个数曲线

综上所述,LCBA 算法是一种比最短路径算法、*N*-hub 算法和 MIRA 算法更加适合网络拥塞时的实时路由调整算法.并且,通过对算法设置不同的参数,算法可以变得更加灵活:对于文件下载等时延要求不高的应用,可以对 $L_U$ 设置较大的参数,尽可能地使网络中的流量分布更加均衡;而对于流媒体等对时延要求较高的应用,就应该对 $L_U$ 设置较小的参数,满足实时性的要求.

## 5 结 论

由于网络拓扑和流量分布的不均衡以及终端传输协议和路由协议自身存在的缺陷,骨干网中的一些链路会在某一个时间段内发生拥塞,同时,另外的一些链路长时间处于空闲状态.这样的结果一方面使网络状况恶化,服务质量变差,另一方面也降低了网络的资源利用率.当拥塞出现时,网络管理者必须尽快采取措施缓解拥塞,避免更大的损失.现有针对拥塞的路由调整算法都不够全面,它们或者不考虑路径长度和服务质量,或者不考虑流量分布,或者计算代价过高,因而都不能满足网络管理者的需要.

本文提出了一种针对关键流重路由的 LCBA 算法,在满足了关键流路径长度要求的前提下,使网络中最大链路带宽利用率最小.通过对所求问题的转化,我们得到了一种多项式时间复杂度的求解算法.理论证明,LCBA 算法能够达到预期目标.仿真及实验结果表明,该算法能够在保证关键流路径长度在一定范围内的基础上,降低网络中最大链路带宽利用率.另外,LCBA 算法计算复杂度低,可以用于实时处理网络中将要和已经发生的拥塞事件.

本文中的算法最适合处理网络中单个位置发生拥塞的情况,当网络中多处同时发生拥塞时,算法只能够按顺序依次处理每个拥塞点.这种方法并没有把多个拥塞点的位置及流量信息综合考虑,因此,当这种情况发生时, LCBA 算法效果并不是最好的.我们下一步的工作就是要对本文中所讨论的问题进行扩展,在网络中多个位置将要或已经发生拥塞时,通过调整多个关键流的路径缓解拥塞,并且同时满足多个关键流的路径长度要求,使网络流量分布得最为合理.

**References**:


[1] Moy J. OSPF Version 2. RFC 2328, 1998.

[2] Iyer S, Bhattacharyya S, Taft N, Diot C. An approach to alleviate link overload as observed on an IP backbone. In: Proc. of the 22nd Annual Joint Conf. of the IEEE Computer and Communications Societies. San Francisco: IEEE INFOCOM, 2003. 406−416. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1208692

[3] Huang Y, Guerin R. Does over-provisioning become more or less efficient as networks grow larger? In: Proc. of the 13th IEEE Int'l Conf. on Network Protocols. Washington: IEEE Computer Society, 2005. 225−235. http://portal.acm.org/citation.cfm?id=1100354

[4] Feldmann A, Greenberg A, Lund C, Reingold N, Rexford J. NetScope: Traffic engineering for IP networks. IEEE Network Magazine, 2000,14(2):11−19.

[5] Kvalbein A, Hansen AF, Cicic T, Gjessing S, Lysne O. Fast IP network recovery using multiple routing configurations. In: Proc. of the 25th IEEE Int'l Conf. on Computer Communications. Barcelona: IEEE INFOCOM, 2006. 1−11. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4146880

[6] Lakshminarayanan K, Caesar M, Rangan M, Anderson T, Shenker S, Stoica I. Achieving convergence-free routing using failure-carrying packets. In: Proc. of the 2007 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM, 2007. 241−252. http://portal.acm.org/citation.cfm?id=1282380.1282408

[7] Zhong Z, Nelakuditi S, Yu Y, Lee S, Wang J, Chuah CN. Failure inferencing based fast rerouting for handling transient link and node failures. In: Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies. Miami: IEEE INFOCOM, 2005. 2859−2863. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1498576

[8] Yang X, Wetherall D. Source selectable path diversity via routing deflections. In: Proc. of the 2006 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2006. 159−170. http://portal.acm.org/citation.cfm?id=1159933

[9] Hershberger J, Maxel M, Suri S. Finding the *k* shortest simple paths: A new algorithm and its implementation. ACM Trans. on Algorithms, 2007,3(4):45. http://portal.acm.org/citation.cfm?doid=1290672.1290682

[10] Orda A, Sprintson A. Efficient algorithms for computing disjoint QoS paths. In: Proc. of the 23rd AnnualJoint Conf. of the IEEE Computer and Communications Societies. Hong Kong: IEEE INFOCOM, 2004. 738. http://ieeexplore.ieee.org/xpl/ freeabs_all.jsp?arnumber=1354543

[11] Xu D, Chen Y, Xiong Y, Qiao C, He X. On finding disjoint paths in single and dual link cost networks. In: Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies. Hong Kong: IEEE INFOCOM, 2004. 715. http://ieeexplore.ieee.org/iel5/9369/29751/01354541.pdf?arnumber=1354541

[12] Cohen R, Nakibly G. On the computational complexity and effectiveness of *N*-hub shortest path routing. IEEE/ACM Trans. on Networking, 2008,16(3):691−704.

[13] Kar K, Kodialam M, Lakshman TV. Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications. IEEE Journal on Selected Areas in Communications, 2000,18(12):2566−2579.

[14] Fang W, Peterson L. Inter-As traffic patterns and their implications. In: Proc. of Global Telecommunications Conf. 1999. Rio de Janeiro: IEEE GLOBECOM, 1999. 1859−1868. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=832484

[15] Feldmann A, Greenberg A, Lund C, Reingold N, Rexford J, True F. Deriving traffic demands for operational IP networks: Methodology and experience. IEEE/ACM Trans. on Networking, 2001,9(3):265−280.

[16] Zhang Y, Breslau L, Paxson V, Shenker S. On the characteristics and origins of internet flow rates. In: Proc. of the 2002 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM, 2002. 309−322. http://portal.acm.org/citation.cfm?id=633025.633055

[17] Estan C, Varghese G. New directions in traffic measurement and accounting. In: Proc. of the 2002 SIGCOMM Conf. New York: ACM, 2002. 323−336. http://portal.acm.org/citation.cfm?id=505212

[18] Smitha, Kim I, Reddy AL. Identifying long-term high-bandwidth flows at a router. Lecture Notes in Computer Science, 2001, 2228(1):361−371.

[19] Mori T, Uchida M, Kawahara R, Pan J, Goto S. Identifying elephant flows through periodically sampled packets. In: Proc. of the 4th ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2004. 115−120. http://portal.acm.org/citation.cfm?id=1028788.1028803

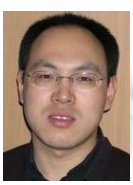
[20] Postel J. Internet protocol. RFC 791, 1981.

[21] Rosen E, Viswanathan A, Callon R. Multiprotocol label switching architecture. RFC 3031, 2001.

[22] Zhang Y. 6 months of Abilene traffic matrices. http://www.cs.utexas.edu/~yzhang/research/AbileneTM/

[23] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. 2nd ed., Cambridge: MIT Press, 2001.

**裴育杰**(1977－),男,天津人,博士生,主要研究领域为 IP 网网络测量,P2P 计算.



**王洪波**(1975－),男,博士,副教授,主要研究领域为 IP 网网络测量与网络安全,P2P 计算,下一代互联网体系结构.

**程时端**(1940－),女,教授,博士生导师,主要研究领域为互联网性能分析与服务质量控制,P2P 计算,传感器网络.