分布式控制平面：并行BGP路由计算自适应负载均衡算法

江学智[1),2)] 徐明伟[1)]

[1)](清华大学计算机科学与技术系 北京 100084)

[2)](石家庄机械化步兵学院 石家庄 050083)

摘 要 下一代互联网高度可扩展支持服务动态部署.越来越多延时和抖动敏感服务(如IPTV、VoIP等)的应用对BGP路由计算的性能提出了更高的需求.路由器采用分布式控制平面和实现并行BGP路由计算克服集中控制平面的性能瓶颈是解决这个问题的有效途径.但现有并行BGP路由计算方案因负载均衡性能差影响了系统的并行性能.文中基于Hashing技术提出了并行BGP路由计算自适应负载均衡模型.通过在线统计路由更新设计了自适应负载均衡算法P-AP(Prediction-based Adaptive Partition),自适应地动态调整路由更新在处理节点间的分配.最后设计和实现了原型系统,并利用Route Views收集的BGP Update数据进行实验.实验结果表明,P-AP算法具有负载均衡性能好、负载调整频率小和路由计算加速性能好等特点,能够有效地提高并行BGP路由计算性能.

关键词 分布式控制平面;BGP;并行路由计算;负载均衡

中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2010.01591

Distributed Control Plane: Adaptive Load Balancing for Parallel BGP Route Computing

JIANG Xue-Zhi[1),2)] XU Ming-Wei[1)]

[1)](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

[2)](*Shijiazhuang Mechanized Infantry Institute, Shijiazhuang* 050083)

Abstract The high scalability of next generation Internet supports deploying Internet services dynamically. With the deployment of more and more delay- and jitter- sensitive applications such as IPTV and VoIP, more route processing power is required to improve the performance of BGP route selection. Routers adopt distributed control planes and implement parallel BGP route processing, which is a potential approach to overcome the bottleneck of centralized control planes and improve the performance of route computation. However, current BGP parallel route computing schemes cannot keep load balance well, which degrades the performance of parallel route computing. In this paper, a load balance model for parallel BGP route computing is proposed based on hashing. Through accumulating prefix updates online and reallocating them among all processing nodes adaptively, the authors propose P-AP (Prediction-based Adaptive Partition) algorithm for parallel BGP route computation, design and implement the prototype of load balancing for parallel BGP route computation. Route Views BGP Update dataset is used to verify the performance of P-AP algorithm. Experimental results show that P-AP algorithm can balance load well among all

收稿日期:2010-04-19;最终修改稿收到日期:2010-08-05.本课题得到国家"九七三"重点基础研究发展规划项目基金(2009CB320502)、国家"八六三"高技术研究发展计划项目基金(2009AA01Z251)以及国家"十一五"科技支撑计划(2008BAH37B03)资助.**江学智**,男,1975年生,博士研究生,讲师,主要研究方向为高速路由器体系结构、分布式路由协议与算法.E-mail: jxz@csnet1.cs.tsinghua.edu.cn.
徐明伟,男,1971年生,博士,教授,博士生导师,主要研究领域为计算机网络体系结构、高速路由器体系结构和互联网路由.

processing nodes, minimize load adjusting, and have the maximum speedup of route computation. It can effectively improve the performance of parallel BGP route computing.

Keywords distributed control plane; BGP; parallel route computation; load balancing

1 引 言

下一代互联网将实现电信网、广播电视网和计算机网络的融合，成为一个实时化、可扩展的网络综合业务平台. 越来越多延时敏感服务的动态部署对路由计算的实时性提出了迫切需求. 下一代互联网规模快速增长将导致 BGP(Border Gateway Protocol)[1]路由更新消息的急剧增加. Huston 等人[2]研究表明：2008 年，BGP 每天大约增加 1700000 次路由更新，900000 万次路由撤消；预计到 2010 年底，BGP 每天增加 2800000 次路由更新，1600000 次路由撤消. 下一代互联网快速发展将导致 BGP 路由计算性能与路由器路由处理能力之间的矛盾越发突出. 目前网络运营商通过部署高性能路由计算服务器[3-5]改进路由计算性能. 虽然多核技术在一定程度上可提高控制平面的路由处理能力，但是 BGP 路由更新快速增长对路由计算性能的可扩展需求降低了多核技术的效费比. 因此采用多个控制单元分布式互连的分布式控制平面[6-11]体系结构实现路由处理性能可扩展是解决这个问题的根本途径.

为了充分利用分布式控制平面的处理资源，需要在各处理节点之间均衡地分配负载以实现最佳性能. 在负载分配过程中，采用动态调整方式自适应地保持负载均衡具有更高的性能. 而现有的并行 BGP 路由计算方案[6-10]都采用静态的负载分配方法. 虽然它们在一定程度上改进了 BGP 路由计算性能，但是由于 BGP 路由更新具有突发性[12]，采用静态负载分配极易引起处理节点间的负载不均衡. 因此研究并行 BGP 路由计算自适应负载均衡算法对提高网络性能具有重要意义.

本文在介绍负载均衡相关工作的基础上，提出了并行 BGP 路由计算自适应负载均衡模型，设计了在线预测的自适应负载均衡(Prediction-Based Adaptive Partition，P-AP) 算法. 通过动态调整路由更新在处理节点间的分配，实现节点间负载动态均衡. 最后设计和实现了并行 BGP 路由计算负载均衡原型系统，并利用 Route Views[13]收集的 BGP Updates数据验证了 P-AP 算法的性能.

本文第 2 节介绍并行 BGP 路由计算和负载均衡的相关研究工作；第 3 节建立并行 BGP 路由计算负载均衡模型和评价指标；第 4 节描述负载均衡策略；第 5 节设计自适应负载均衡算法；第 6 节设计和实现原型系统并验证 P-AP 算法的性能；第 7 节分析路由计算乱序问题并提出解决方案；最后总结全文.

2 现有技术分析

2.1 并行 BGP 路由计算

2.1.1 按邻居会话划分的计算模型

Xu[6]等人基于分布式控制平面提出了按邻居会话划分的迭代树并行 BGP 路由计算模型. 在该模型中，集群路由器内部的所有处理节点组成一棵树. 每个处理节点维护与之相连的 BGP 邻居会话并交换路由信息. 当接收到路由更新消息，每个节点先计算出本地最优路由，然后将本地最优路由发送给自己的父亲节点. 沿着叶子向根的方向不断迭代，在根节点计算出每条前缀的全局最优路由. Zhang[7]等人提出集群路由器的每个处理节点计算出本地最优路由后以广播的方式同步路由信息. 与之类似，Nguyen[8-9]等人基于高性能路由器的线卡具有路由计算能力，提出每块线卡建立和维护与它相连的 BGP 邻居会话，各自存储和计算它们的本地最优路由，然后将本地最优路由发送给主控制单元，主控制单元从所有线卡发送的局部最优路由中选择全局最优路由.

采用邻居会话划分方案，由于每个处理节点维护的邻居数量不同而导致接收的路由更新不同，容易造成处理节点间负载不均衡，降低了系统的并行性能.

2.1.2 按前缀的前 8 位划分的计算模型

与邻居会话划分算法不同，Hidell[10]等人根据前缀前 8 位对应的十进制整数(范围是 0～255)划分路由计算任务. 例如：前缀 212.96.38.18/24 和前缀 213.88.52.3/16 前 8 位对应的十进制整数分别为 212 和 213，所有 212.x.x.x 的前缀分配给 1 号控制单元，所有 213.x.x.x 的前缀分配给 2 号控制单

元，以此类推. 由于在实际网络中，一个 AS 故障通常影响一段连续范围的前缀[14]，这些前缀中的大部分的前 8 位相同，因此按前缀的前 8 位划分负载容易导致某个控制单元的负载突发，造成节点间负载不均衡.

2.2 负载均衡

负载均衡主要讨论任务在各处理节点之间的分配和调度. 现有的路由器负载均衡方案主要利用并行处理缓解数据平面的高速报文流与转发能力之间的矛盾.

Lukas Kencl[15]等人提出了路由器内部各转发引擎间的自适应负载均衡模型. 通过反馈控制，利用扩展的 HRW(Highest Random Weight)算法[16]动态修改转发引擎的权重向量分配报文，实现了转发引擎间的负载均衡. Shi[17]等人通过 Hashing 技术对流量进行划分，每次动态调整时将流量最大的流分配给当前负载最轻的处理节点实现负载均衡. Shaikh Anees[18]等人采用动态路由转发长期的流，采用静态路由转发短期的流. 通过动态调整对负载敏感的长流减少链路状态信息的频繁更新，在链路之间平衡流量. Dittmann Gero[19]等人采用 Hashing 技术将流量映射到相应的转发单元，记录具有相同索引的流最近到达报文的时间戳，通过将时间戳最小的流分配给当前负载最轻的转发单元实现负载均衡.

2.3 负载均衡与并行 BGP 路由计算

基于以上分析，实现负载均衡是提高 BGP 并行路由计算性能的有效途径. 从处理粒度看，并行 BGP 路由计算可分为报文级和流级两种. BGP 的每个路由更新消息是一个报文，每个路由更新消息中通告和撤消的每条前缀可以看作一个流. 与之相对应，按照邻居会话划分负载是一种报文级的调度，它以路由更新消息为粒度，将每个邻居通告的路由更新消息分配给对应的处理节点. 由于多个邻居通告的同一条前缀的候选路由分布在不同的控制单元，因此每个控制单元只能计算局部最优路由. 按照前缀的前 8 位划分负载是一种基于流的调度. 它以前缀为粒度，将同一条前缀的所有路由更新采用 Hashing 技术映射到同一个节点处理，每个节点能够独立计算相应前缀的最优路由. 由于每条前缀的路由更新动态变化，这种粗粒度分配导致节点间负载不均衡.

从调度方式看，负载均衡主要分为静态调度和自适应调度. 静态调度没有考虑网络流量的突发性，采用固定的负载分配方法，可能导致负载分配不均衡. 现有的并行 BGP 路由算法都采用静态负载分配，但 BGP 路由更新具有突发性，静态划分 BGP 负载容易导致负载不均衡. 自适应调度根据处理节点的负载大小自适应地分配负载，有更好的性能. 因此，研究并行 BGP 路由计算自适应负载分配对提高网络性能具有重要的意义.

3 并行 BGP 路由计算负载均衡模型和评价指标

在分布式控制平面内部，为了便于区分每个处理节点的功能，我们按功能将它们分为主控制单元和从控制单元. 主控制单元负责建立与维护 BGP 邻居会话，当接收到路由更新消息时，将负载分配给相应的从控制单元. 如果从控制单元之间的负载超过阈值 T，启动负载调整，重新调整任务分配，构建新的任务分配方案(Task Table，TT)，并根据新的任务分配方案分配负载. 为了便于描述，除特别说明，控制单元都指从控制单元.

3.1 负载均衡模型

BGP 路由计算是对每一条路由发生更新的前缀，从所有 BGP 邻居通告的候选路由中选择一条最优的路由，每一条前缀的路由计算相互独立. 如果根据前缀划分负载，由于前缀的数量远大于控制单元的数量，因此每个控制单元需要处理一部分前缀. 为了区分每个控制单元处理的前缀集合，需要建立前缀与控制单元间的映射关系.

定义 1. 前缀 ID. 将一条前缀 S 利用函数 H 映射成的整数 $H(\cdot):S\rightarrow\{0,1,2,3,\cdots,N\}$，定义为这条前缀的 ID. ID 相同的前缀属于同一个前缀组. N 表示前缀组的数量.

例如，将每条前缀的第 9、13、17 和 18 位组合的实数映射成前缀的 ID，则每条前缀所属前缀组 ID 的长度 $B=4$，可将前缀分成($2^B=16$)个前缀组. 则前缀〈112.147.48.0/24〉的前缀组 ID 是 8(1000)；〈148.24.86.0/22〉的前缀组 ID 是 5(0101). 当第 i 条前缀 $Prefix_{[i]}$ 的前缀 ID 等于前缀组 j 的 ID，则前缀 $Prefix_{[i]}$ 属于前缀组 j，表示为 $Prefix_{[i]}\in j$. 通过前缀组 ID 可区分每个控制单元处理的前缀.

定义 2. 存储冗余率 $B_{[i]}$ 定义为某一条前缀(或某一个前缀组)在系统中存储的份数.

为了便于本文描述，下面给出相关的变量定义，如表 1 所示.

表 1 相关变量及定义

M	控制单元数量；
T	两个控制单元路由更新次数相差阈值；
B	前缀 ID 的长度；
S	前缀组的集合；$S=\{1,2,\cdots,2^B\}$；
P_j	分配在编号为 j 的控制单元上的前缀组集合；$\bigcup P_j=S(j=1,2,3,\cdots,M)$；
$P\_id_{[i]}$	ID 为 i 的前缀组的路由更新次数；
$B_{[i]}$	ID 为 i 的前缀组的存储冗余率；
$L_{[i]}$	ID 为 i 的前缀组分配到一个控制单元时，该控制单元增加的负载；$L_{[i]}:=P\_id_{[i]}$；
$BOOL(i,j)$	前缀组 i 属于控制单元 j 上的前缀集合 P_j；
$R_{[k]}$	控制单元 k 上所有前缀的路由更新次数；$R_{[k]}:=\sum_{i\in P_k}L_{[i]}(k=1,2,\cdots,M)$.

因为每条前缀的 BGP 路由计算相互独立，要提高并行 BGP 路由计算性能，需要合理划分前缀，实现控制单元间负载均衡. 前缀在控制单元间分配需要根据控制单元的数量和每条前缀的路由更新次数，将这些前缀分配到相应的控制单元，并通过冗余存储实现控制单元彼此分担负载，使负载最大的控制单元与负载最小的控制单元之间的负载差值最小. 因此，前缀在控制单元间的分配可用表达式(1)表示：

$$Min(Max(R_{[k]})-Min(R[l])),\ k,l=1,\cdots,M \quad (1)$$

为实现表达式(1)的优化目标，前缀在控制单元间的分配和冗余存储必须满足式(2)～(7)的约束条件：

$$\bigcup_j P_j=S,\ j=1,2,\cdots,M \quad (2)$$

$$R_{[i]}-R_{[j]}\le T,\ 0<i\le M,\ 0<j\le M \quad (3)$$

$$BOOL(i,j):=\begin{cases}1, & i\in P_j\\ 0, & i\notin P_j\end{cases} \quad (4)$$

$$B_{[i]}:=\sum_{j=1}^{k}BOOL(i,j),\ i\in S \quad (5)$$

$$B_{[i]}=M,B_{[i]}\in Z^+,\ j\in S \quad (6)$$

$$R_{[k]}:=\sum_{i\in P_k}L_{[i]},\ k=1,2,\cdots,M \quad (7)$$

式(2)～(7)表明：每个控制单元存储的前缀组是所有前缀集合的子集，所有控制单元存储的前缀的并集包含所有的前缀. 任意两个控制单元路由更新次数的差值不超过阈值 T. 当前缀组 i 存储在控制单元 j 上，则 $BOOL(i,j)=1$；前缀组 i 在控制单元上存储的份数表示它的存储冗余率 $B_{[i]}$. 为实现自适应负载均衡，所有前缀组的冗余存储率均为 M. 前缀组 i 分配到控制单元 K 上，控制单元 K 增加的负载为 $L_{[i]}$. 控制单元 K 的负载 $R_{[K]}$ 等于它需要处理的前缀组的负载总和 $\sum_{i\in P_K}L_{[i]}$.

在 M 个从控制单元组成的具有负载均衡能力的分布式并行 BGP 处理系统 $\{P_1,P_2,P_3,\cdots,P_M\}$ 中，当 $P_i(1\le i\le M)$ 空闲时，分配的负载可以立即处理，否则报文缓冲在它的输入缓冲区 $Q_i(i=1,2,\cdots,M)$ 中. 并行 BGP 路由计算负载均衡模型如图 1 所示.

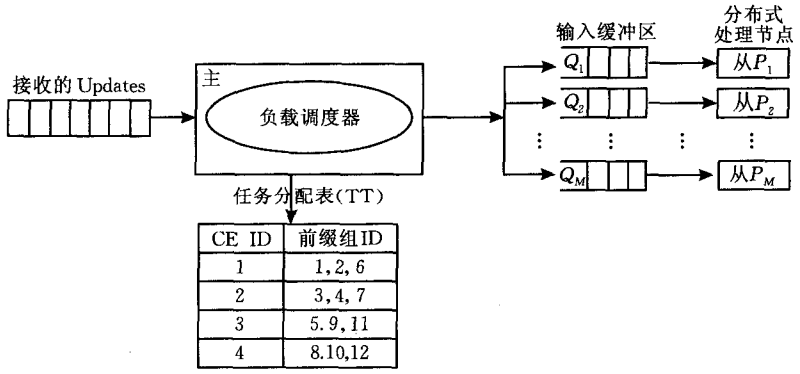


图 1 并行 BGP 路由计算负载均衡模型

3.2 并行 BGP 路由计算负载均衡评价指标

由于 BGP 路由计算的对象是每条前缀，而 Shi[17] 等人和 Chen Yi-Jiao[20] 等人提出的性能评价指标不适合评价并行 BGP 路由计算的性能. 我们对文献[17,20]中的负载均衡评价指标进行了修改和扩充，它主要由前缀负载均衡度、路由更新消息负载均衡度、负载调整频率和路由计算加速比等构成.

定义 3. 负载均衡度(Load Balancing Degree，LBD). 设 $p_i(t)$ 表示时间 t 内，节点 $P_i(i=1,2,\cdots,M)$ 处理的前缀更新所属的路由更新消息，$|p_i(t)|$ 表示时间 t 内，处理节点 $P_i(i=1,2,\cdots,M)$ 处理的前缀更新所属的路由更新消息数；$|Pre_i(t)|$ 表示时间 t 内，处理节点 $P_i(i=1,2,\cdots,M)$ 处理的前缀更新的总数，且 $p(t)=\bigcup_{i=1}^{M}p_i(t)$，$Pre(t)=\bigcup_{i=1}^{M}Pre_i(t)$，则有 $|Pre_i(t)|$ 的自协方差：

$$Cov[Pre_i(t)]=\frac{E(|Pre_i(t)|^2)-(E(|Pre_i(t)|))^2}{(E(|Pre_i(t)|))^2} \tag{8}$$

将式(8)归一化到[0,1]区间，得到前缀负载均衡度(Prefix Load Balancing Degree) $PLBD(t)$：

$$PLBD(t)=\frac{Cov[|Pre_i(t)|]}{Cov[|Pre_i(t)|]+1}=1-\frac{(E(|Pre_i(t)|))^2}{E(|Pre_i(t)|^2)} \tag{9}$$

由于$|Pre_i(t)|$和$|Pre_i(t)|^2$的数学期望可分别用它们$(t-\Delta t,t)$时间的数量表示：

$$E(|Pre_i(t)|)=\sum_{i=1}^{m}|Pre_i(t)|,$$

$$E(|Pre_i(t)|^2)=\frac{\sum_{i=1}^{m}|Pre_i(t)|^2}{M},$$

因此，式(9)可转化为

$$PLBD(t)=1-\frac{\left(\sum_{i=1}^{m}|Pre_i(t)|\right)^2}{M\sum_{i=1}^{m}|Pre_i(t)|^2} \tag{10}$$

$PLBD(t)$以前缀为单位统计了负载分配的均衡性，负载均衡时 $PLBD(t)\to 0$. 对于式(10)，用$|p_i(t)|$代替$|Pre_i(t)|$作为负载均衡的衡量标准可得到路由更新消息负载均衡度 $ULBD(t)$：

$$ULBD(t)=1-\frac{\left(\sum_{i=1}^{m}|p_i(t)|\right)^2}{M\sum_{i=1}^{m}|p_i(t)|^2} \tag{11}$$

路由更新消息负载均衡度以路由更新消息为粒度衡量分配的宏观负载均衡度. 在以路由更新消息为粒度的分配方案中，好的路由更新消息负载均衡度可以获得好的处理性能.

定义 4. 负载调整频率(Load Adjusting Frequency，LAF). 设 $N(t)$表示时间 t 内处理的前缀更新数量，$|L(t)|$表示在时间 t 内的负载调整次数. 负载调整频率表示为

$$LAF=\frac{|L(t)|}{N(t)} \tag{12}$$

定义 5. 路由计算加速比(Route Computing Speedup，RCS). 设时间 t 内，路由器接收到 $pre(t)$ 条路由更新，用 T_S 表示单节点串行处理 $pre(t)$ 条路由更新的时间，T_P 表示 M 个处理节点 $P_i(i=1,2,\cdots,M)$ 并行处理 $pre(t)$ 条路由更新的时间. 路由计算加速比表示为

$$RCS=\frac{T_S}{T_P} \tag{13}$$

4　负载均衡策略

由于 Update 消息中通告的前缀数量动态变化，因此，按 Update 消息划分负载无法有效地实现负载均衡. 虽然文献[21]中综述了许多自适应负载均衡策略，但这些自适应负载均衡策略由于复杂度高和开销大难以应用于实际. Guo[22]等人根据扩展的 HRW 自适应负载均衡策略提出多媒体集群服务器的负载均衡机制经过扩展可应用于并行 BGP 路由计算.

根据前缀 ID，每一条前缀的路由更新可映射到相应的控制单元 j 处理. 系统根据每个控制单元的负载大小动态调整前缀与控制单元的映射关系，它可以表示为

$$x_j\eta(s,j)=\min_{k\in\{1\cdots M\}}x_p\eta(s,p) \tag{14}$$

其中 s 是前缀组 ID，j 表示这个前缀组映射的控制单元 ID. $(x_1,x_2,\cdots,x_M)$表示各控制单元负载大小的权重系数. 主控制单元根据所有从控制单元的负载信息判断是否超过阈值 T，当超过阈值 T，启动负载调整，调整每个从控制单元的权重系数，重新分配负载.

4.1　最小负载优先轮询分配(Least-Load First Round-Robin，LLF-RR)

采用轮询分配机制，主控制单元根据从控制单元数量将前缀组轮询分配给所有的从控制单元. 为避免反馈带来的内部开销，主控制单元在本地为每个从控制单元维护一个分配队列指示它们各自的负载大小. 每当负载超过阈值 T，主控制单元根据本地分配队列长度调整各从控制单元负载权重，按照最小负载优先的原则从小到大轮询本地分配队列，将前缀组轮询分配给所有的从控制单元，重建任务分配表.

4.2　离线预测最小负载优先分配(Offline-Prediction Least-Load First，OP-LF)

虽然采用 Hashing 技术[23]等映射机制分配路由计算任务具有开销低、效率高等优点，但因路由更新动态变化，采用轮询分配无法较好地保持负载均衡.

利用 Hashing 技术将每条更新前缀映射为前缀 ID，根据一段历史时间的 BGP Updates 数据可离线地统计每条前缀的路由更新次数，从而预测它们

未来的路由更新变化.根据预测结果将它们均匀地分配到 M 个控制单元,构建任务分配表,实现控制单元间的负载均衡.主控制单元周期性收集每个从控制单元的负载反馈信息.当负载超过阈值 T,主控制单元对从控制单元的负载进行排序,重新调整每个处理节点的权重系数,按照负载从小到大的方式选择从控制单元,重建任务分配表.

4.3 离线预测加权优先分配(Offline-Prediction Weight First,OP-WF)

与 OP-LF 不同的是当负载超过阈值 T,主控制单元只对当前负载最大的控制单元上的负载进行调整.首先,主控制单元对从控制单元的负载进行排序,按负载大小反向调整每个处理节点的权重系数,将负载最大的控制单元的权重系数设为最小值.然后将负载最大的从控制单元的需要处理的前缀组 ID 按照调整后的权重系数分配给所有的从控制单元,重建任务分配表.

4.4 在线预测自适应负载分配

由于 BGP 更新具有突发性,并且不稳定前缀在不断变化.虽然离线统计能够对未来路由更新变化做出预测,但是离线预测和轮询负载分配算法会引起处理节点间的负载不均衡,降低系统的并行性能.如果能够在线统计得到每个前缀组的路由更新次数,可对未来路由更新变化做出更准确的预测,实现负载均衡.

5 P-AP 自适应负载分配算法

在并行 BGP 路由计算负载均衡系统中,我们提出在线预测自适应负载均衡算法 P-AP 实现各控制单元间负载均衡.如图 3 所示,算法由负载自适应、负载选择、前缀分类、在线统计和任务分配表 5 个模块组成.负载自适应模块调整负载分配和建立任务分配表.负载选择模块根据任务分配表将负载加入相应处理节点的输入队列.前缀分类模块区分每条路由更新的前缀 ID.在线统计模块统计每条前缀的路由更新次数.任务分配表模块记录每个控制单元需要处理的前缀组 ID.

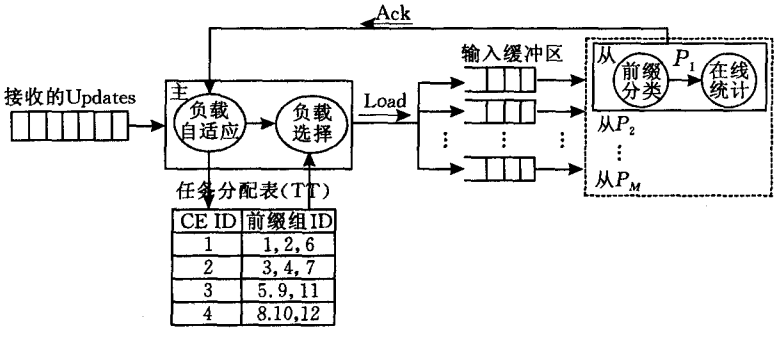


图 2 并行 BGP 路由计算负载均衡调度框图

主控制单元每接收一个 Update 消息将其封装成 Load 消息发送给所有从控制单元,Load 消息格式如图 3 所示.每个从控制单元根据 Load 消息中各自需要处理的前缀计算这些前缀的路由,并统计各自处理的前缀组的路由更新次数.主控制单元定期收集每个处理节点的负载反馈信息.

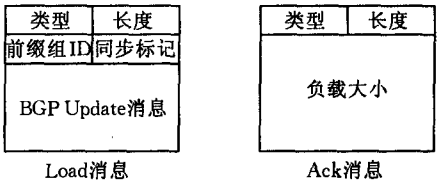


图 3 主、从控制单元通信消息格式

当两个控制单元接收的路由更新次数超过阈值 T,触发负载调整.主控制单元对从控制单元的负载进行排序,重新调整每个处理节点的权重系数.负载自适应模块根据在线统计的每个前缀组的路由更新次数,将所有前缀组重新分配给从控制单元,并新建任务分配表.P-AP 算法的伪代码如下.

P-AP 算法.

Input:

1. the number of CEs: M; //从控制单元数量
2. the total updates of every prefix group(ID): $P\_id_{[j]}(j=1,2,\cdots,2^B)$; //前缀组路由更新
3. $R_{[k]}\leftarrow\varnothing, P_j\leftarrow\varnothing$; //将每个控制单元的前缀组集全置为空,负载清零

begin:

while (控制单元间的路由更新>T) do

{

1. Sort $P\_id_{[i]}(i=1,2,3,\cdots,2^B)$ in decreasing order, and record the result as $\{S_{[1]}, S_{[2]}, \cdots, S_{[2^B]}\}$;

2. for i from 1 to 2^B do
3. for j from 1 to M do
4. Sort $R_{[j]}$ ($j=1,2,3,\cdots,M$) in increasing order, and record as $\{Sd_{[1]},Sd_{[2]},\cdots,Sd_{[M]}\}$
//将控制单元的负载按升序排列，更新每个控制单元的权重系数($x_1,x_2,\cdots,x_M$)
5. endfor
6. select minimum $R_{[j]}$;
//选择权重系数最小的控制单元
7. $R_{[j]}=Sd_{[j]}+S_{[i]}$; //将前缀组分给负载最小控制单元 j，增加其负载值
8. $P_j=P_{[j]}\cup P_{[i]}$; //将分配的前缀组加入控制单元 j 上的前缀组集合
9. endfor
}
end while
end
Output: $\{P_i,\ i=1,2,3\cdots M\}$; //输出每个控制单元需要处理的前缀组 ID

6　性能评价

为了真实地评价 P-AP 自适应负载均衡算法的性能，首先，我们选取一台 1000Mbps 交换机和 5 台具有 1000Mbps 网卡的 PC 机，基于 linux 内核，设计和实现了并行 BGP 路由计算自适应负载均衡原型系统；其次，通过重放 Route Views 真实的 BGP Updates 数据验证 LLF-RR、OP-LF、OP-WF 和 P-AP 4 种负载均衡算法的性能；最后，在大量的数据中随机选择了 10 天的数据进行实验.

6.1　原型系统设计

并行 BGP 路由计算自适应负载均衡原型系统由 5 个物理节点组成，如图 4 所示.

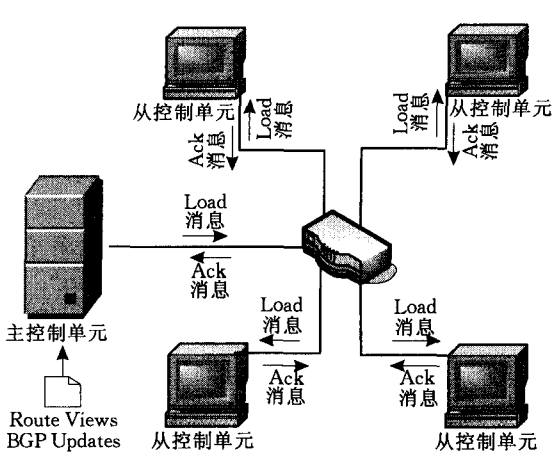


图 4　并行 BGP 路由计算负载均衡原型系统

一个节点充当主控制单元，其它 4 个节点充当从控制单元. 它们通过交换机相连. 主控制单元：(1) 还原 Route Views BGP Update 消息，模拟接收路由更新；(2) 确定每个 Update 消息中路由计算负载的分配，将每个 Update 消息封装为内部通信的 Load 消息，发送给每个从控制单元；(3) 接受从控制单元的负载反馈信息，自适应调整负载分配，重建任务分配表. 从控制单元：(1) 接收主控制单元发送的每个 Load 消息；(2) 根据每个 Load 消息头部的前缀组 ID 计算对应前缀的路由；(3) 在线统计每条前缀的路由更新次数；(4) 向主控制单元发送 Ack 消息反馈自身的负载信息和每条前缀的路由更新次数. 原型系统配置及实验参数如表 2 所示.

表 2　原型系统硬件性能及实验参数

分类		配置参数	
	名称	CPU	内存
硬件	主控制单元	Intel(R)Core(TM) 2.66GHz	8GB
	从控制单元	Intel(R)Dual Core 2.50GHz	3GB
	接口速率	1000Mbps	
实验参数设置	负载调整阈值 T	500	
	时间间隔 t	5min	
	控制单元数量	主:1(个)，从:4(个)	
	权重系数	0.4,0.3,0.2,0.1	

在表 2 中，负载调整阈值、时间间隔和权重系数可根据实际需要进行设置. 负载调整阈值用于触发负载调整. 时间间隔用于量化一段时间内的 Update 数量和路由更新次数. 权重系数主要用于区分每个控制单元的负载大小，每次负载调整时，我们按照 4 个从控制单元负载的大小，分别设置其权重系数为 0.4,0.3,0.2,0.1. 最后的实验结果是每天的路由更新数据经过 288(24h×60/5min=288)次实验的平均值.

6.2　内部通信机制

为实现自适应负载均衡，主、从控制单元之间采用内部通信协议进行通信.

6.2.1　主控制单元

每当接收到 Update 消息，主控制单元不修改 Update 消息，只根据任务分配表，在接收的 Update 消息头部添加相应从控制单元需要处理的前缀 ID，封装成 Load 消息，然后发送给从控制单元，并接收控制单元发送的 Ack 消息. 主控制单元根据反馈信息判断是否调整负载. 每次负载调整后，重建任务分配表，并且在向所有从控制单元发送的第 1 个 Load 消息中将“同步标记”字段设为“同步”，指示每个从控制单元重新开始统计每个前缀组的路由更新次

数. 否则"同步标记"字段设为"异步".

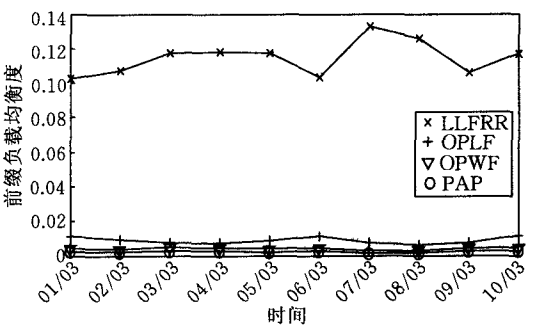
6.2.2 从控制单元

因为每个从控制单元只有存储了对应前缀的所有候选路由才能独立地计算出全局最优路由. 为了实现自适应负载均衡,从控制单元存储主控制单元发送的所有 Load 消息中的路由信息. 收到 Load 消息后,先检查"同步标记"字段,如果为"同步",将每条前缀的路由更新计数清零,重新统计每个前缀组的路由更新次数. 如果为"异步",计算每个 Load 消息中属于自己的前缀的路由,并累计这些前缀的路由更新次数. 忽略不属于自己处理的路由更新. 每接收到一个 Load 消息,从控制单元向主控制单元发送 Ack 消息反馈当前的负载大小.

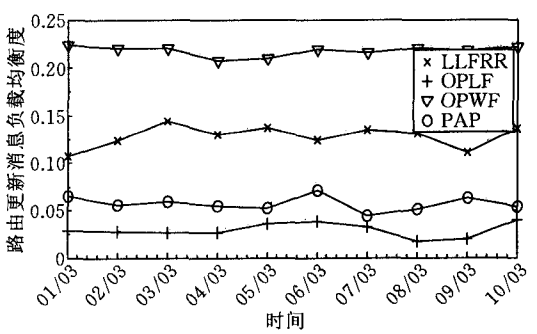
6.3 实验结果

6.3.1 负载均衡度

负载均衡度体现了一段时间内各个从控制单元的负载均衡情况,我们通过实验分别模拟了系统的前缀负载均衡度和路由更新消息负载均衡度. 实验结果如图 5 所示.



(a) 前缀负载均衡度比较



(b) 路由更新消息负载均衡度比较

图 5

图 5(a)的实验结果表明,从前缀数量比较,PAP 算法的前缀负载均衡度优于其它 3 种算法,前缀在从控制单元间的分配比较均衡. 而 LLFRR 算法采用轮询的方式调整负载,从控制单元间的前缀负载均衡度差. OPWF 算法只对负载最大的从控制单元的负载进行调整,其前缀均衡性能优于 LLFRR 和 OPLF 算法.

从路由更新消息数量比较,OPLF 算法能够很好地实现 Update 消息在从控制单元间的均匀分配,而 PAP 算法也能获得较好的路由更新消息均衡度. 由于每个 Update 消息中通告的前缀数量不断变化,OPWF 算法和 LLFRR 算法的路由更新消息均衡度较差.

6.3.2 负载调整频率

负载调整频率越高,表明系统的负载调整次数越多,负载最大与负载最小控制单元间的负载差异超过阈值 T 的次数越多. 负载调整频率越低,表明负载最大与负载最小控制单元间的负载越均衡,系统的并行性能越好. 为了评价系统的负载均衡性能,我们测量了它们各自的负载调整频率. 实验结果如图 6 所示.

实验结果表明 PAP 算法的负载调整频率小,系统具有最好的负载均衡性能. OPLF 算法也较好地实现了从控制单元之间的负载均衡,但其负载调整频率在一定范围内变化,与 PAP 算法相比,影响了并行性能. 由于 OPWF 算法每次只将负载最大的从控制单元上的负载调整到其它的从控制单元上,这种局部负载调整策略加剧了负载最大与负载最小节点间的负载不均衡,负载调整次数多. LLFRR 算法的负载调整也比较频繁,负载均衡性能差.

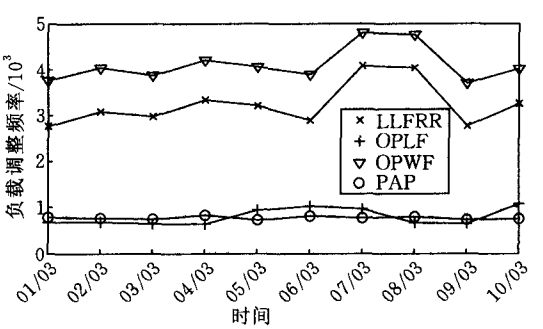


图 6 负载调整频率比较

6.3.3 路由计算加速比

路由计算加速比反映了系统的路由计算加速性能. 路由计算加速比越大,系统各处理节点间的负载越均衡,系统的并行性能越好. 路由加速比越小,各处理节点间的负载不均衡程度越高,系统的并行性能越差. 路由计算加速比的实验结果如图 7 所示.

实验结果表明,PAP 算法的路由计算加速比最大,系统的并行性能最好. 这表明 PAP 算法能够很好地保持从控制单元间负载均衡,提高系统的并行路由计算性能. 虽然 OPWF 算法的负载调整频率

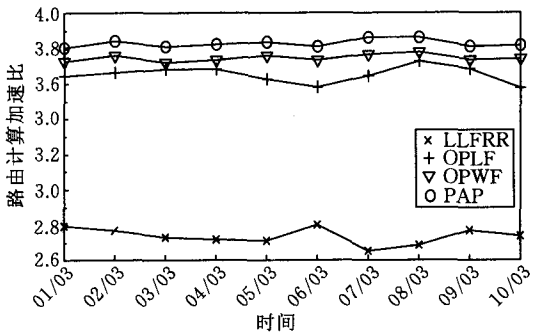


图 7　路由计算加速比

较高，但 OPWF 算法也具有较好的并行性能. 我们分析这种原因是由于负载最大的从控制单元与负载最小的从控制单元之间的局部负载差异造成的，而系统中其余节点间的负载比较均衡，这种分析我们可以从图 5(a)中 OPWF 算法的前缀均衡度得到验证. 因此 OPWF 算法的路由计算加速性能比较好. 但是与 PAP 算法相比，因负载最大的从控制单元与负载最小的从控制单元间负载不均衡，降低了系统的并行性能. 而 LLFRR 算法的并行性能相对较差.

6.3.4　内部通信开销和传输延时

分布式控制平面处理节点间的通信流量反映了系统从控制单元间的负载变化. 我们分别测量了分布式控制平面采用 LLFRR、OPLF、OPWF 和 PAP 算法每秒钟从控制单元间的通信流量. 在我们选取的 BGP Update 数据中，这个数据收集点维护了 36 个BGP 邻居. 从控制单元间每秒钟的通信流量如图 8 所示.

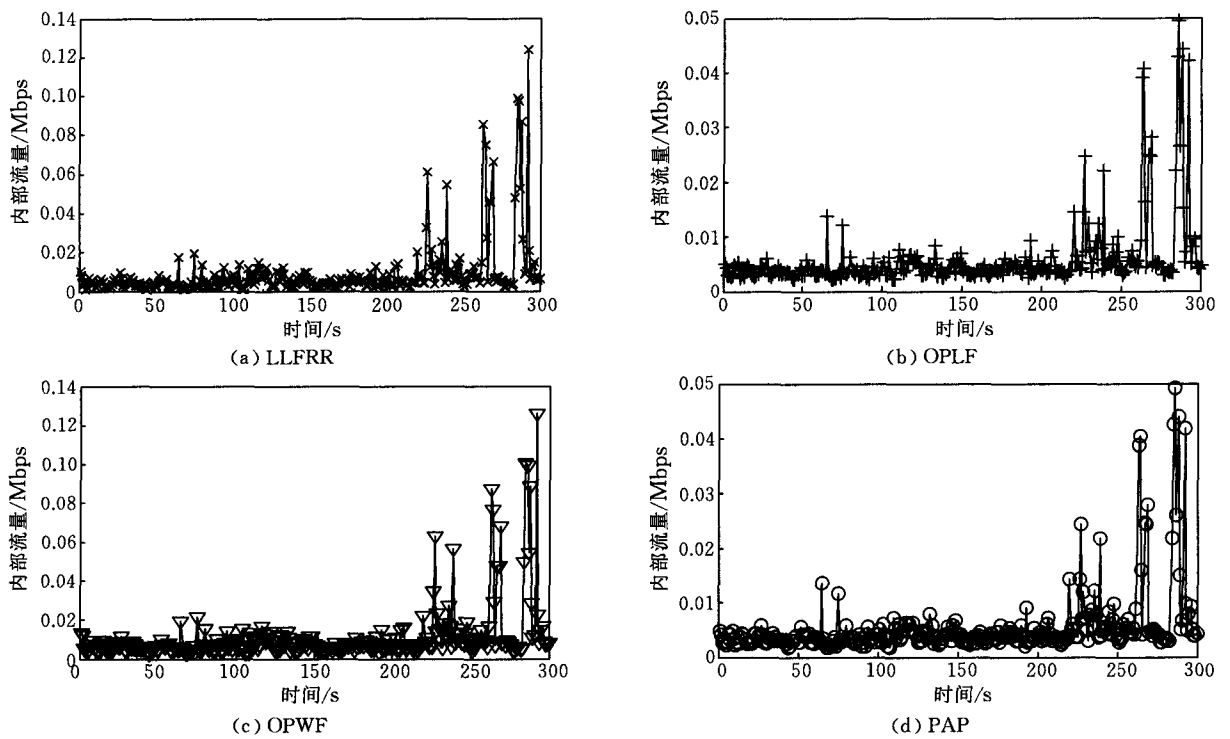


图 8　内部通信流量

实验结果表明，采用 PAP 算法处理节点间的通信流量最小，系统具有较好的并行路由计算性能. 由于 OPWF 频繁地触发负载调整实现负载均衡，其内部通信流量最大，平均流量为 0.01Mbps. 但与分布式控制平面内部 1000Mbps 交换网络相比，从控制单元间的平均通信流量只占总带宽的 0.01%. 我们分析这是因为在一段时间，路由器只能从邻居接收一定数量的路由更新. 这表明分布式控制平面内部采用 1000Mbps 高速交换网络，具有足够的带宽. 虽然 OPWF 频繁地触发负载调整，但其具有较好的并行路由计算加速比，这表明路由计算消耗了控制平面大量的处理资源，实现负载均衡可有效地提高系统的并行路由计算性能.

设每条路由更新的平均大小为 30 字节，则每条路由更新在分布式控制平面内部的传输延时为 0.24μs. Kevin Fall[24] 等人的研究表明，一台 BGP 路由器有 100 个邻居，0.5 万条前缀，每条路由更新在单核 3GHz 处理器上的路由计算时间为 33μs. 因为每条路由更新的内部同步传输时间远小于路由计算时间，所以可以忽略每条路由更新的内部传输延时. 这进一步表明 BGP 路由计算消耗了控制平面大量的处理资源. 因此分布式控制平面内部节点间互

连网络的传输速率越高，系统的并行路由计算性能主要取决于处理节点间的负载均衡性能.

7 路由计算乱序

按照 FIFO 的顺序对邻居通告的路由更新进行处理的结果能及时反映网络链路的状态变化，符合实际网络的路由改变. 自适应负载均衡方案在自适应调整负载时可能将同一条前缀的路由更新分布到两个处理节点并行处理. 由于每个处理节点的负载大小不同，可能导致同一条前缀最后到达的路由更新先于最早到达的路由更新完成计算，我们称同一条前缀的路由更新不按接收的时间顺序进行的路由计算为路由计算乱序. 如果一个处理节点根据一条前缀最新通告的路由更新计算出到这条前缀的路由并更新了转发表(FIB)，而另一个处理节点根据同一条前缀以前通告的路由更新计算出到这条前缀的路由，并更新 FIB，这将造成用无效/次优路由替换 FIB 中正确的路由，造成路由不可达或非最优路由.

为了有效地解决路由计算乱序，在收到一个路由更新消息时，标记这个更新消息接收的时间. 在 FIB 增加一个记录接收时间的域. 每个处理节点并行地计算各自的路由，并根据计算结果独立地更新 FIB 中的路由. 在更新 FIB 时，对路由的时间戳进行比较. 当 FIB 中到同一条目的前缀的路由的时间早于计算好的路由的时间，则对 FIB 中的这条路由进行更新. 如果计算好的路由的接收时间早于 FIB 中这条路由的接收时间，则丢弃当前的计算结果.

8 结束语

基于现有的并行 BGP 路由算法的负载均衡性能差，无法充分利用分布式控制平面的处理资源，本文采用前缀划分和动态调整相结合的自适应调度思想，通过在线统计每条前缀的路由更新次数，并根据每个处理节点的负载信息动态调整前缀在处理节点间的分配，实现系统负载均衡，有效地提高了并行 BGP 路由计算性能. 但自适应负载均衡可能导致路由计算乱序，因此在具体实现时需要记录每条路由更新的到达时间. 通过设计和实现原型系统，并用真实网络数据验证了 P-AP 算法的负载均衡性能好，负载调整频率小和路由计算加速比大，能够满足下一代互联网快速发展对 BGP 路由计算性能可扩展的需求.

参 考 文 献

[1] Rekhter Y, Li T, Hares S. A border gateway protocol 4 (BGP-4). IETF RFC 4271, 2006

[2] Huston G, Armitage G. Projecting future IPv4 router requirements from trends in dynamic BGP behavior//Proceedings of the Australian Telecommunication Networks and Application Conference 2006. Melbourne, Australia, 2006: 189-193

[3] Caesar M, Caldwell D, Feamster N, Rexford J, Shaikh A, van der Merwe J. Design and implementation of a routing control platform//Proceedings of the NSDI'05. Boston, USA, USENIX Association, 2005: 15-28

[4] Bates T, Chen E, Chandra R. BGP route reflection: An alternative to full mesh internal BGP (iBGP). IETF RFC 4456, 2006

[5] Hitesh B, Paul F, Cao T, Wang J. Making routers last longer with ViAggre//Proceedings of the NSDI'09. Boston, USA, USENIX Association, 2009: 453-466

[6] Xu Ke, He Huan. BGP parallel computing model based on the iteration tree. The Journal of China Universities of Posts and Telecommunications, 2008, 15(18): 1-8

[7] Zhang Xiao-Zhe, Zhu Pei-dong, Lu Xi-cheng. Fully-distributed and highly-parallelized implementation model of BGP4 based on clustered routers//Proceedings of the 4th International Conference on Networking. Reunion-Island, France, 2005: 433-441

[8] Nguyen K, Jaumard B, Agarwal A. A distributed and scalable routing table manager for the next generation of ip routers. IEEE Network, 2008, 22(2): 6-14

[9] Nguyen K, Mahkoum H. Jaumard B, Assi C, Lanoue M. Toward a distributed control plane architecture for next generation routers//Proceedings of the 4th European Conference on Universal Multiservice Networks. Washington, DC, 2007: 173-182

[10] Hidell Markus. Decentralized modular router architectures [Ph. D. dissertation]. KTH-Royal Institute of Technology, Sweden, 2006

[11] Yang L, Dantu R, Gopal R. Forwarding and control element separation (ForCES) framework. IETF RFC 3746, 2004

[12] Zhao X, Massey D, Lad M, Zhang L. ON/OFF model: A new tool to understand bgp update burst. USC-CSD, Technical Report 04-819, 2004

[13] Route Views project: Advanced network technology center. University of Oregon. http://www.routeviews.org/

[14] Athina Markopoulou et al. Characterization of failures in an operational IP backbone. IEEE/ACM Transactions on Networking, 2008, 16(4): 749-762

[15] Lukas Kencl, Boudec Le Jean-Yves. Adaptive load sharing for network processors. IEEE/ACM Transactions on Networking, 2008, 16(2): 293-306

[16] David G, Ravishankar C V. Using name-based mappings to increase hit rates. IEEE/ACM Transactions on Networking, 1998, 6(1): 1-14

[17] Shi W G, MacGregor M H, Gburzynski P. Load balancing for parallel forwarding. IEEE/ACM Transactions on Networking, 2005, 13(4): 790-801

[18] Shaikh Anees, Rexford Jennifer, Shin Kang G. Load-Sensitive routing of long-lived IP flows. ACM SIGCOMM Computer Communication Review, 1999, 29(4): 215-226

[19] Dittmann Gero, Herkersdorf Andreas. Network processor load balancing for high-speed links//Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'2002). San Diego, CA, 2002: 727-735

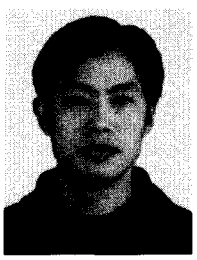
[20] Chen Yi-Jiao, Lu Xi-Cheng, Sun Zhi-Gang. MSF: A session-oriented adaptive load balancing algorithm//Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops. Dalian, China, 2007: 657-663

[21] Shirazi B A, Hurson A R, Kavi K M. Scheduling and Load Balancing in Parallel and Distributed Systems. Wiley: CS Press, 1995

[22] Guo Jia-Ni, Laxmi N Bhuyan. Load balancing in a cluster-based web server for multimedia applications. IEEE Transactions on Parallel Distributed System, 2006, 17(11): 1321-1334

[23] Cao Z, Wang Z, Zegura E. Performance of hashing-based schemes for internet load balancing//Proceedings of the 19th IEEE INFOCOM. Tel-Aviv, Israel, 2000: 332-341

[24] Kevin Fall, Brighten Godfrey P. Routing Tables: Is smaller really much better?//Proceedings of the 8th ACM Workshop on Hot Topics in Networks (HotNets'09). New York, USA, 2009



JIANG Xue-Zhi, born in 1975, Ph. D. candidate. His research interests include high-speed router architecture and parallel route computation.

XU Ming-Wei, born in 1971, Ph. D., professor, Ph. D. supervisor. His research interests focus on computer network architecture, high-speed router architecture and Internet routing.

Background

The explosive growth of the Internet and delay- and jitter- sensitive applications such as VoIP and IPTV requires real-time route calculation. However, current routers are restricted by the bottleneck of a single control element (CE) on centralized control plane. With the exponentially growth of BGP updates and hardware limitations of current router control plane, next generation routers demand exploiting distributed control plane that consist of multiple CEs to implement parallel BGP route processing. But the burst of BGP updates impedes load balance. Traditional BGP parallel route processing schemes always allocated route updates unreasonably and load unbalanced among CEs. It seriously degraded the performance of parallel route processing.

With the support of National Basic Research Program (973 Program) of China under grant (No. 2009CB320502) and the National Science & Technology Pillar Program during the Eleventh Five-Year Plan Period under grant (No. 2008BAH37B03), the authors have researched on the distributed control plane architecture and parallel route table computation.

To improve the performance of parallel BGP route computation, we propose a load balance model for BGP distributed parallel route computing based on hashing and devise an adaptive load balance scheme that dynamically divides prefixes into groups with hashing and adjusts the allocation of the prefix groups adaptively. Through accumulating prefix updates online, we propose P-AP (Prediction-Based Adaptive Partition) algorithm to allocate prefix updates among all CEs adaptively.

With the support of National High Technology Research and Development Program (863 Program) of China under grant (No. 2009AA01Z251), the authors designed and implemented the prototype of load balancing for parallel BGP route computing. Using Route Views BGP Updates dataset, we verify the performance of P-AP algorithm. The experimental results show that P-AP algorithm keeps load balanced, minimizes the frequency of load adjusting, and has the maximum speedup of route computing. It effectively improves the performance of parallel BGP route computing.