计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20201087
58(6):1307-1317,2021

基于预测的数据中心间混合流量调度算法

王 然[1,2] 张宇超[1] 王文东[1,2] 徐 恪[3] 崔来中[4]
[1](北京邮电大学计算机学院(国家示范性软件学院) 北京 100876)
[2](网络与交换技术国家重点实验室(北京邮电大学) 北京 100876)
[3](清华大学计算机科学与技术系 北京 100084)
[4](深圳大学计算机与软件学院 广东深圳 518060)
(wangranse@bupt.edu.cn)

Algorithm of Mixed Traffic Scheduling Among Data Centers Based on Prediction

Wang Ran[1,2], Zhang Yuchao[1], Wang Wendong[1,2], Xu Ke[3], and Cui Laizhong[4]
[1](*School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing* 100876)
[2](*State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), Beijing* 100876)
[3](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)
[4](*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong* 518060)

Abstract To handle the problem of low link utilization resulting from mixing online and offline traffic in one data center transmission network and separating them with a fix way in the same link, we propose a solution of offline traffic scheduling based on online traffic prediction. It firstly predicts online traffic needed to be guaranteed preferentially in link using an algorithm calling Sliding-k that combines EWMA and Bayesian changepoint detection algorithm. This customized algorithm can make prediction sensitive to a sudden change of network environment and reduce unnecessary re-adjustments when network environment is steady at the same time. Therefore, it can exactly meet the prediction demand under different network environments. After computing the remaining space for offline traffic according to online traffic prediction result and implementing dynamic bandwidth allocation, it uses an algorithm called SEDF that can consider both traffic deadline and size to schedule offline traffic. Experimental results reflect that Sliding-k can meet the prediction needs both when network mutation occurs and when network has no change and can simultaneously improve the accuracy of traditional EWMA algorithm. The combination of Sliding-k and SEDF can improve the utilization of data center links, so as to make full use of link resources.

Key words data center; traffic engineering (TE); prediction algorithm; scheduling; exponentially weighted moving average (EWMA)

收稿日期:2020-12-26;**修回日期**:2021-03-22

基金项目:国家重点研发计划项目(2019YFB1802603);国家自然科学基金青年科学基金项目(61802024);中央高校基本科研业务费专项资金(2020RC36);国家自然科学基金项目(62072047);国家杰出青年科学基金项目(61825204);北京高校卓越青年科学家计划项目(BJJWZYJH01201910003011)

This work was supported by the National Key Research and Development Program of China (2019YFB1802603), the National Natural Science Foundation of China for Young Scientists (61802024), the Fundamental Research Funds for the Central Universities (2020RC36), the National Natural Science Foundation of China (62072047), the National Natural Science Foundation of China for Distinguished Young Scholars (61825204), and the Beijing Outstanding Young Scientist Program (BJJWZYJH01201910003011).

通信作者:张宇超(yczhang@bupt.edu.cn)

摘 要 为解决在线流量和离线流量共用一个数据中心传输网络,且2种类型的流量在链路中的分配模式固定不变而导致的链路利用率低的问题,提出了一种基于在线流量预测的离线流量调度方式.首先使用结合了EWMA方法和贝叶斯拐点检测算法的Sliding-k算法对链路中需要优先保障的在线流量进行预测,使预测既能在网络环境突然变化时灵敏响应,又能在网络平稳时减少不必要的重调整.根据预测结果计算出离线流量的可用剩余空间,实现动态的带宽分配之后,使用能够同时考虑流量截止时间和流量大小2个维度的SEDF算法对离线流量进行调度.实验结果表明:Sliding-k能够同时满足网络突变和网络无变化情况下的预测需求,并且能够提高传统EWMA方法的准确率,它和SEDF的结合能够提高数据中心链路的利用率.

关键词 数据中心;流量工程;预测算法;调度;指数加权移动平均

中图法分类号 TP393

如今,越来越多的大型企业在世界各地构建起了自己的数据中心(data center, DC)以及跨数据中心域的数据传输平台.但是,连接每个数据中心对的长途链路十分昂贵,因此提高数据中心间链路的利用率毫无疑问能够为企业带来巨大的效益,尤其是随着5G的到来,传输数据量会急剧膨胀,数据中心间链路利用率的提升将更为紧迫,其带来的效益也将更为显著.

国内的数据中心网络一般将在线流量和离线流量混合部署,其中在线流量是为用户提供在线服务所产生的延迟敏感型的流量,而离线流量是由数据中心间数据复制等原因产生的较为不紧急的流量.文献[1]也指出,数据中心间广域网(wide area network, WAN)的流量大致可以描述为2种类型流量的混合:1)高优流量,由用户所面临的需求即时到达的流量组成.到达的需求数量是无法预测的,虽然比总容量小得多但需要被充分满足.2)数据中心之间的大规模传输的流量,是周期性时间需求的流量.这2类组成了广域网上的大部分流量.由于2种流量共享带宽,因此需要考虑带宽的分离模式,也就是决定给每种流量分配多少带宽,采用动态的还是固定的分配比例.

目前有许多关于数据中心间网络性能优化的研究[2-7],但这些传统的数据中心间的数据传输方法具有2方面的局限:

1) 固定的带宽分离模式.在这种模式中,链路总带宽被以固定的比例划分给在线流量和离线流量,这会导致在线流量很少的时候,预分配给它的那部分空闲带宽无法被数量众多的离线流量利用.现有的固定带宽分离模式是造成目前链路利用率低的最主要原因.理论上,如果我们能够实时地充分利用在线流量所剩的可用带宽,则链路利用率会大大提高.

2) 单一条件的最早截止时间(earliest deadline first, EDF)算法.传统的EDF调度方式会根据流量的截止时间对所有流量进行调度,截止时间越早的越被优先调度.在需要最大程度增加可完成的流量数量时,仅考虑流量截止时间的做法并不能达到期望的调度效果,因为在链路容量有限的情况下,当流量截止时间相差不大但流量大小相差很大时,优先调度较小但截止时间稍晚的流量可以在提高链路利用率的同时最大化完成流量的数量.

根据分析,本文提出了一种基于带宽使用情况预测的实时调度算法,能够解决传统数据复制传输方式存在的以上2方面不足.本系统采用集中式的同时考虑流量截止时间和流量大小的调度方式,通过控制器来维护链路的全局信息,同时,预测模块通过各链路的历史观测值预测出当前调度周期的在线流量(实时到达的流量)使用情况,从而将本周期内各链路的剩余带宽充分分配给离线流量(在指定时间段内传输完即可的流量),即调度器基于这些全局信息来产生全局最优的调度决策.

1 相关工作

为了提高数据中心间WAN的链路利用率,Google先后提出了B4[8]以及它的改良版本B4 and After[9],此外还有支持其中TE组件的BwE组件[10].

B4整体采用软件定义网络(software-defined networking, SDN)的结构来实现,主要通过一个集中式的流量工程(traffic engineering, TE)算法来为应用分配带宽,实现最大最小公平(max-min fairness, MMF),这一方式能够使链路利用率达到接近100%.

Google通过集中式的TE来实现全局式的调度.在B4中,SDN网关将来自多个站点的拓扑整合

到 TE 控制器,带宽执行器(BwE)则收集来自不同应用的需求然后提交给控制器,然后集中式的控制器利用这些全局的信息,使用 max-min fairness 的方式对带宽进行分配,这一分配过程是站在全局的角度上进行的.

Google B4 WAN 的流量需求在 5 年内增长了 100 倍,并且从一个要求 99%可用性的批量传输、内容复制网络,发展成为一个可用性要求为 99.99%的支持交互的网络.针对带宽需求和可靠性要求的增长,Google 随后提出了 B4 and After,它改进了 B4 原有扩展方案,将每一个站点设计为 2 层拓扑抽象,同时引入了边链(sidelinks)和超节点(supernode)级别的 TE 解决容量不对称问题.

根据微软的 SWAN[2] 所描述,高容量 DC 间链路是一种昂贵的资源,它可以长距离提供 100 Gbps 到 Tbps 的容量,其每年的租赁成本高达 100 万美元.但是 DC 间 WAN 的使用效率极差,即使是繁忙链路的平均利用率也仅为 40%~60%,显然供应商目前并未充分利用这些昂贵的资源.因此微软提出了软件驱动的广域网(software-driven wide area network, SWAN),它通过协调不同服务的发送速率及集中配置网络数据平面来提供高效、灵活的数据中心互联网络.为了保持较高的使用效率,需要频繁更新网络状态,为此在链路上预留了少量带宽,并且在交换机上预留了少量内存.通过这种方式可快速地实现网络更新并且不会打断现有网络的运行.

Facebook 在 Semi-Oblivious[11] 中指出,需要在 TE 中在性能和鲁棒性之间达到平衡.大多数系统都是针对其中一种或另一种进行优化而设计的,但是很少有系统能够同时实现 2 种优化.由于操作限制而进一步加剧了这一挑战,例如路径的数量、重配置产生的开销、硬件施加的量化分割比等.Facebook 因此设计了 Smore,一种通过将仔细路径选择与动态权重适应相结合的方法.Smore 在拥塞和负载均衡指标方面实现了近乎最佳的性能,在延迟方面与最短路径的方法也堪具可比拟性.

然而这些研究的侧重点无法完全与在线流量和离线流量混合部署且要求高链路利用率的场景吻合.以 Google 为例的集中式调度对于提升链路利用率的效果立竿见影,但 Google 的场景跟国内大多数企业的场景不同,它具有 2 套独立的 WAN,除了上述连接众多数据中心的 B4 之外还有一套是链接数据中心和用户的 B2,不存在在线数据和离线数据的带宽分离问题.SWAN 在设计时主要解决如何快速实现数据平面的更新问题,而 Smore 则致力于同时兼顾性能和可靠性.尽管这些 TE 系统很好地解决了各自面临的不同需求,但它们均未深入研究链路中的 2 种流量混合部署的问题.

然而,在线流量和离线流量共用数据中心链路而引发的混合流量调度问题普遍存在并具有研究价值,理想的混合流量调度方案应该在优先调度在线流量的情况下将剩余的链路资源充分分配给离线流量需求,这样一来链路利用率就能大大提升,宝贵的数据中心链路资源就能得到高效地利用,从而为企业带来可观的经济效益.

以国内数据中心网络为例,包括百度、华为在内的数据中心网络均将在线流量和离线流量混合部署,需要进一步考虑带宽的分离模式.

百度在面临以上问题时,先后提出了 PieBridge[12] 和 BDS[13].PieBridge 通过在残存网络上进行传输,最大化通信链路的带宽使用,同时通过使用调度器选择数据传输源,显著减少了数据同步的完成时间.在 PieBridge 的理论基础上,BDS 具体通过充分利用 DC 间的覆盖路径来减少数据同步的完成时间,同时给组播流量(离线流量)和延迟敏感型流量(在线流量)固定地分配带宽占比来实现链路带宽的共享.

百度同样指出,依赖于各个服务器的本地调度决策由于缺乏全局的信息,通常都是次优的.所以百度提出了 BDS,这是一个高度集中的结构,它通过中央控制器实时地维护 agent 服务器数据传输状态,以近似实时的方式更新最新的全局信息,以便响应动态的性能变化、请求的变化、和路由决策的更新.这样一个全局式的调度系统通常比较复杂,但是,BDS 将控制算法解耦为调度和路由 2 个部分,从而减少集中控制的计算开销.系统在调度步骤中会选出 duplication data 的子集,后续路由步骤仅仅考虑这些被选中的子集,搜索空间因此大大减少,全局式的调度也因此变得更加高效.

然而,固定的带宽分离模式仍不能充分地在在线流量和离线流量之间共享带宽.在线流量的峰值和谷值相差较大,为了保障延迟敏感型的在线流量能够正常地进行传输,采用固定带宽分离模式的系统往往为在线流量分配多于其一般情况需求的带宽.这就会造成大多数情况下,分配给在线流量的带宽未能得到有效的利用而遭到浪费,此外,在线流量的突发时有发生,一旦发生在线流量的意外激增,固定分配给在线流量的带宽就会不足,因此,固定带宽

分离的模式不仅无法充分利用链路资源,也无法保障在线流量突增时的传输.而离线流量作为一种非延迟敏感型且规模庞大的流量,它可以根据链路中的在线流量占用情况来灵活调整自身的传输.

2 基于预测的调度系统概况

在传统的固定带宽分离的模式中,由于分配给延迟敏感的在线流量以及对延迟不那么敏感的离线流量的带宽是固定的,所以即使在线流量很少时,离线流量也不能利用分配给在线流量但目前空闲的带宽,这将导致很低的链路利用率.所以本系统采用了动态带宽分离模式,根据不同的网络情况来自动调整调度结果:当在线流量突发时,本系统将会缩减即将分配给离线流量的带宽以保障在线流量的性能同时避免拥堵;当在线流量缩减时,本系统允许离线流量使用更多的带宽以充分使用剩余带宽.

图 1 为本系统动态带宽分离的逻辑图.系统每 5 s 制定一次流量的调度方案,每个周期内系统各模块之间的调用关系为:

1) 网络监视器读取由代理观察到的历史流量信息(historical traffic),然后将这部分信息提供给预测模块,即由指数加权移动平均(exponentially weighted moving average, EWMA)[14]和拐点检测算法[15]组成的 Sliding-k 模块.

2) 拐点检测(change point detection)子模块依据历史在线流量信息判断当前周期是否有突变出现,并将拐点检测结果通知到 EWMA 模块.

3) EWMA 模块负责计算当前周期的在线流量预测值,它根据拐点检测的结果来调整自身的参数,并结合历史在线流量数据,计算出当前周期在线流量的预测值.

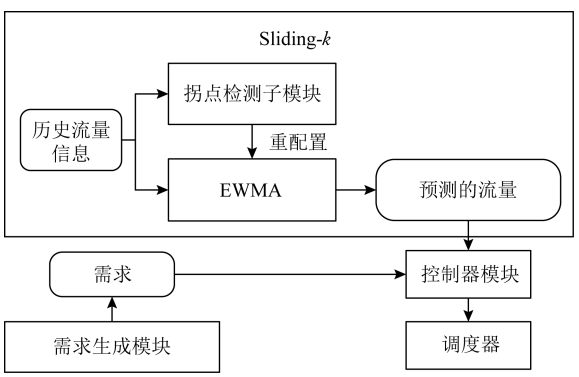

Fig. 1 Logical diagram of dynamic bandwidth separation

图 1 动态带宽分离的逻辑图

4) 控制器模块(Controller)负责收集系统的拓扑信息(包括各链路的初始容量、当前剩余容量等信息)以及收集到的离线流量需求(demand)(包括需流量需求的到达时间、截止时间、流量大小、源 DC、目的 DC),并在系统做出调度决策之后及时地更新这些信息.

5) 调度器(Scheduler)从控制器获取当前周期的拓扑信息,包括各链路内的剩余带宽大小,以及本周期内可调度的离线流量需求,通过调度算法确定离线流量的调度顺序,并使用可绕行的选路决策为每一个离线流量确定目标路径,从而完成调度.

3 基于预测的调度算法

3.1 在线流量预测机制

目前存在一些基本方法可以检测在线流量变化并动态调整调度的配置,如 EWMA,k-Sigma 等[14,16].但这些方法有时候会连续地重配置,甚至在网络很稳定的时候也会如此,这是没有必要的.因此,在预测可用带宽的时候面临着一个权衡:当我们在预测时更偏向于参考最近的数据,预测值将会出现明显的震荡,这将引起不必要的连续重调度;而当我们在预测时更偏向于参考历史数据,预测值受近期检测到的拐点的影响就较小,这使得系统对于网络变化不那么敏感.

为解决上述问题,本系统将 EWMA[14]和拐点检测算法[15]结合,并且设计了本系统定制的 Sliding-k 算法,如算法 1 伪代码所示.具体来说,我们为 EWMA 方法设置了一个边界 $K\in[0,1]$,K 为使用原有 EWMA 方法时就需要确定的经验值,一般根据数据特征选择,在本文实验部分展示了在兼具平稳和抖动数据时为了达到较好的效果,这一经验值可以设置为 0.5.若当前没有拐点,k 将被设置为 K,而当一个拐点被检测到时,k 将被设为 1,然后逐渐地降至 K.

在一个调度周期内,网络变化监视器不断地获取服务器吞吐量的一系列代理观测值,这些值在下一个调度周期内被用来预测可用带宽.

算法 1. Sliding-k 算法.

输入:拐点阈值 P、周期 t 之前的流量时间序列 T、EWMA 的最佳实践值 K;

输出:周期 t 的流量值.

① $k\leftarrow K$;

② $cp\leftarrow$上一个拐点;

③ if *ChangePointDetect*(T,P) then
　　/* 判断当前周期是否为拐点 */
④　$k=1$;
⑤　$cp=t$;
⑥　T←仅包含周期 t 的时间序列;
⑦ else
⑧　if T 包含周期 cp then
⑨　　$k=k-1$;
⑩　else
⑪　　$k=K$;
⑫　end if
⑬　T←追加周期 t;
⑭ end if
⑮ return *EWMA*(k,T). /* 使用 EWMA 方法计算当前周期的流量 */

3.1.1　EWMA

EWMA 即指数加权移动平均法,这一方法可以根据历史观测值来估计当前的值,预测时给观测值的权值随时间呈指数递减,离当前时间越近的数据由于跟当前时间的相关性越高而权重越大.

EWMA 的表达式为

$$Z_0=\mu_0, \tag{1}$$

$$Z_i=(1-k)Z_{i-1}+k\overline{X}_i, i>0, \tag{2}$$

其中,$\overline{X}_i$ 为时刻 i 的实际值,系数 k 表示加权下降的速度,其值越大表示下降得越快,也就是给最近观测值的权重更大,Z_i 为时刻 i 的 EWMA 预测值.

将 EWMA 的表达式归纳后可写成:

$$Z_i=(1-k)^i\mu_0+k(1-k)^{i-1}\overline{X}_1+k(1-k)^{i-2}\overline{X}_2+\cdots+k(1-k)\overline{X}_{i-1}+k\overline{X}_i, i\geqslant 0. \tag{3}$$

从式(3)可以看出,观测值的权值随着时间呈指数式下降.给近期观测值较大的权重是因为离当前时间点越近的观测值往往对预测值有较大的影响,更能反映近期变化的趋势.

3.1.2　贝叶斯在线拐点检测算法

为了同时保证稳定性以及对网络变化的敏感程度,预测模块引入贝叶斯拐点检测算法.

该算法使用消息传递算法计算当前时间序列的长度或自上一个拐点以来的时间概率分布.

贝叶斯拐点检测算法在单变量时间序列上以在线方式执行贝叶斯拐点检测.核心思想是在每个新数据点到达时递归计算时间序列长度的后验概率.运行长度定义为自上次拐点发生以来的时间.

给定序列的长度 r_t 可以计算出预测分布,然后对当前序列长度的后验分布进行积分,找到边际预测分布,即"$t+1$ 时刻是序列拐点"这一事件发生的概率:

$$P(x_{t+1}\mid x_{1:t})=\sum_{r_t}P(x_{t+1}\mid r_t,x_t^{(r)})P(r_t\mid x_{1:t}), \tag{4}$$

式(4)中的后验分布为

$$P(r_t\mid x_{1:t})=\frac{P(r_t,x_{1:t})}{P(x_{1:t})}, \tag{5}$$

式(5)中,序列长度和观测值之间的联合分布为

$$P(r_t,x_{1:t})=\sum_{r_{t-1}}P(r_t,r_{t-1},x_{1:t})=\sum_{r_{t-1}}P(r_t,x_t\mid r_{t-1},x_{1:t-1})P(r_{t-1},x_{1:t-1})=\sum_{r_{t-1}}P(r_t\mid r_{t-1})P(x_t\mid r_{t-1},x_t^{(r)})P(r_{t-1},x_{1:t-1}), \tag{6}$$

其中,x_t 表示时刻 t 的观测值,r_t 表示时刻 t 的时间序列的长度,$x_t^{(r)}$ 表示 r_t 所对应的观测值序列.

3.1.3　Sliding-k

在对序列进行预测时,若 EWMA 的参数 k 设置得很小,意味着给更"旧"的观测值的权重更大,预测结果会更加平稳,但对于突发的抖动不够灵敏;若 k 很大,意味着给较"新"观测值的权重越大,预测结果就会越灵敏,但是预测值会出现频繁的波动,引起连续的重配置(如果预测结果跟上一周期的预测值相差不大,则无需重新配置状态信息,减小更新压力).所以本系统采用了 Sliding-k 的方式,即当前无拐点时就给更"旧"的观测值更大的权重,以保证预测结果的平滑,而一旦检测出拐点,就给较"新"观测值更大的权重,保证预测结果的灵敏、准确.

Sliding-k 的详细处理流程如算法 1 所列,其主要步骤为:

1) 贝叶斯拐点检测算法计算当前周期为拐点的概率.首先,设定一个拐点判定的概率阈值,通过设置这一概率阈值,可以指定拐点检测模块对当前时间节点的评估概率超过多少时将该节点判定为拐点.

2) 依据贝叶斯拐点检测的原理和设置好的阈值,计算当前节点为拐点的概率.通过将上一步得出的概率值与设定好的阈值对比,判断当前周期是否为拐点.大于该阈值时,当前节点被判定为拐点;小于该阈值时,当前节点被判定为非拐点.

3) 如果当前周期为拐点,则 EWMA 的输入序列的窗口大小为 1,也就是 EWMA 输入序列仅包含

上一周期的观测值,且参数 $k=1$,即预测时仅参考上一周期的观测值,之后每经过一个周期 k 值减小0.1,直到 k 值降低到指定的稳定值 K(如0.6).

4) 如果当前周期非拐点,则 EWMA 的输入序列为上一个拐点出现时到上一周期这段时间内的一系列观测值.并且在非拐点的情况下,参数 k 有2种取值方式.①当前时刻为非拐点,且 k 值不等于指定的最佳实践值 K,说明当前正处于拐点发生之后 k 由1不断回降至 K 的过程,此时 k 应该比上一周期减少一个回降的步长 s(如0.1),例如上一周期 $k=0.8$,则本周期应为 $k=0.7$,s 的选取跟数据的特征有关,考虑到 s 过大或过小都将会使 Sliding-k 回退为接近 EWMA 的效果,根据实验经验一般将 s 设置为0.1时能够达到更好的预测效果;②当前时刻为非拐点,且 k 值等于指定的最佳实践值 K,说明当前已经从拐点中恢复过来,处于较为稳定的时期,则此时的 k 应当保持为稳定环境下的最佳实践值 K,不需要再进行更新.

5) EWMA 方法根据输入的序列值和 k 值预测当前周期的在线流量值,输出到调度模块.

上述5个步骤实现了拐点检测算法和 EWMA 方法的结合,以拐点检测的判定结果作为输入,根据结果对预测时的时间序列窗口进行调整,然后再使 EWMA 基于这一窗口内的时间序列进行后续的预测.Sliding-k 算法对简单的 EWMA 方法进行了改进,这一做法弥补了 EWMA 对突发状况不敏感的缺陷,同时使得预测使用的时间序列空间能够灵活改变,提高预测性能.

通过拐点检测算法和 EWMA 方法的有效结合,Sliding-k 根据拐点检测的结果动态地调整预测方式,使得在无拐点时能够保证预测结果的平滑,而有突发拐点时又能灵敏地响应,做出准确的预测.在时而平稳时而突变的时变网络中,简单地为 EWMA 的参数设置固定值进行预测显然无法达到根据网络环境变化而动态改变的需求.为了响应变化的预测需求,使用 Sliding-k 算法并指定稳定环境下的 K 和突变发生之后的步长 s,就能在不同的网络环境下分别达到不同的预测需求.

3.2 SEDF 调度算法

链路中同时存在着2种流量:1)实时到达的、延迟敏感的在线流量;2)要求在指定时间段内完成传输即可的离线流量,利用离线流量的这一性质,我们可以实时地将离线流量调度到在线流量使用之后仍有剩余的空闲链路上.因此在对在线流量进行了预测之后,系统需要使用合适的调度策略,将大量的离线流量分配到链路的剩余空间上.

对于具有截止时间的离线流量,传统的调度方法是常用于实时调度系统的 EDF 调度算法,即具有最早截止时间的离线流量会被优先调度,常用于各领域的实时调度系统当中[17-21].

由于调度是以流量需求为单位的,对于服务提供商来说,可完成的流量需求的个数越多意味着调度效果越佳.由于离线流量需求的大小和截止时间都不尽相同,因此,完全按照截止时间来对离线流量进行调度的方式是不合理的,无法尽可能多地完成流量需求.

例如,有3个流量需求 A,B,C,它们都需要使用同一段链路进行传输.由于一个离线流量需求通常无法在一个调度周期内完成而需要在多个周期进行传输,在本例中为了方便说明,对于使用同一段链路的流量需求,使用调度周期(scheduling period,SP)来衡量流量需求的大小,且 A,B,C 三个流量需求的大小分别为 5SP,3SP,2SP.同时假设 A,B,C 均在第1个周期开始时到达,截止时间分别为第5,6,7个周期.即这3个流量需求按照截止时间从早到晚排序同时按照流量大小由大到小排序.由于链路容量大小(一个周期对应1SP)的限制,按照 EDF 的调度方式,将会依次调度 A,B,C 三个需求.第5个周期结束时,A 完成传输,B 开始传输;第6个周期结束时,B 由于达到截止时间而终止传输,C 开始传输;第7个周期结束时,C 由于达到截止时间而终止传输.截至第7个周期结束,仅 A 能够完成全部传输,而 B 和 C 均为部分完成,因此不能达到尽可能多地完成流量需求的要求.

而如果对 A,B,C 三个流量需求按照流量大小进行调度,则会优先调度 B 和 C 这2个相对较小的流量,B,C 均能全部完成,而 A 由于链路容量限制而无法全部完成,此时可完成的流量需求数量将会增加.并且随着流量需求规模的增加,增加的可完成传输的流量需求数目会更多.然而,仅考虑流量需求大小的调度顺序显然会使部分截止时间靠前的大型流量得不到调度.

因此本系统提出了这种基于 EDF 算法的 SEDF(Smart EDF)算法,在每个调度周期中,调度顺序将会综合考虑流量需求的大小和截止时间.算法首先按照流量需求截止时间将需求分为指定数量的优先级段,截止时间越早的段优先级越高,将被优先调度,而同一优先级段内的需求按需求由小到大的顺序调度,如算法2伪代码所示.

算法 2. SEDF 算法.

输入:流量需求集合 R、优先级段的数量 n.

① $R_1,R_2,\cdots,R_n \leftarrow SegmentEDF(R)$; /* 按照流量需求截止时间将需求分为指定数量的优先级段 */

② for 优先级从高到低的每一个优先级段 R_i

③　$R_i' \leftarrow SortBySize(R_i)$; /* 按需求由小到大的顺序对 R_i 段内的需求排序 */

④　for R_i 段内的每一个需求 r

⑤　　$SelectPath(r)$;
　　/* 为需求 r 选择目标路径 */

⑥　end for

⑦ end for

当离线流量的负载增加和离线流量需求的数量增加时,如果仅仅按照 EDF 的顺序进行调度意味着有些本可以完成传输的离线流量需求无法完成传输而被丢弃.而如果在离线流量需求的截止时间相差不大,且同处于一个截止时间优先级段内的情况下,传输多个粒度较小但截止时间稍晚的离线流量需求,比传输一个截止时间较早但粒度较大的离线流量需求要划算,因为此时有更多的需求能被满足.因此本文提出了 SEDF 算法,它在调度时同时考虑离线流量需求的截止时间和大小,从而尽可能地增加可全部传完的离线流量需求数目.

4　实验评估

4.1　实验配置

本文在华为数据中心场景中验证了基于预测的数据中心间混合流量调度算法的有效性.实验的数据中心网络划分为北、东、南 3 个数据中心域,每个数据中心域内包含 8～10 个 DC,且每个域中包含一个称为 DC Core 的特殊 DC,它作为域内和域外传输的唯一出入口.域间为全网状的连接形式,且域间链路的带宽均为 50 Gbps;域内仅存在由普通 DC 到 DC Core 的连接,且域内链路的带宽为数十至数百 Mbps.

实验的时间长度为某天 20:00 开始之后的 24 h,调度周期为 5 s.各周期各链路上的在线流量大小由网络监视器实时测量并提供.离线流量需求根据链路容量和使用情况模拟生成,每一个离线流量需求信息为一个六元组,包含需求的编号、到达时间、截止时间、流量大小、源 DC、目的 DC,如(1,1,79,8350112502.68,Ningbo,Fuzhou).其中到达时间和截止时间均为从实验开始时间(20:00)算起的累计分钟数,流量大小的单位为字节(B).在实际场景中,离线流量需求量远大于在线流量,且离线流量的时间跨度不尽相同,为了满足离线流量的这一特征,实验在生成模拟的离线流量的同时需要考虑链路使用情况,使离线流量需求稍大于剩余链路容量,并且到达时间、截止时间、源 DC 和目的 DC 均采用随机的方式确定.

4.2　实验结果和分析

为了验证 Sliding-k 算法的预测效果,我们随机生成了分段的带"毛刺"的序列,分段平稳的数据模拟平稳网络下的数据,分段交界处的突增或突降模拟突变网络下的数据.实验首先使用已有的 EWMA 方法对这部分模拟生成的在线流量序列进行预测,当 k 设置为 0.3,0.5,0.9 时预测效果分别如图 2～4 所示,其中黑线代表真实值,灰线代表预测值.

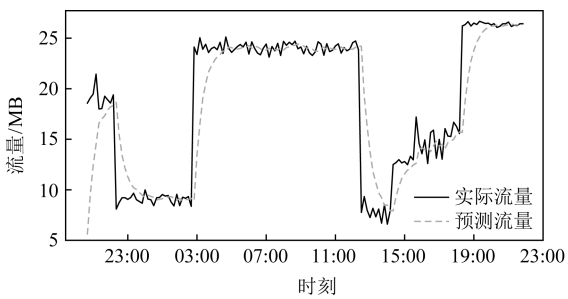

Fig. 2　Prediction effect when the EWMA parameter k is 0.3

图 2　EWMA 参数 $k=0.3$ 时的预测效果

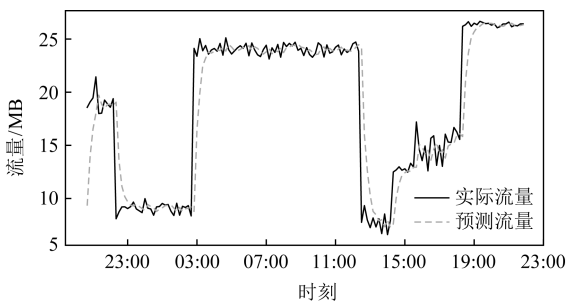

Fig. 3　Prediction effect when the EWMA parameter k is 0.5

图 3　EWMA 参数 $k=0.5$ 时的预测效果

从图 2 可以看到,EWMA 方法的参数 $k=0.3$ 时预测结果比较平滑但不够准确,特别是在拐点出现的时间点上预测效果较为不理想,如图 2 中 03:00,13:00,19:00 时刻左右,预测值偏离真实值的程度较大.而当 EWMA 方法的参数 $k=0.9$ 时,不论是在时间序列走势较为平稳的周期还是在拐点出现的周期,预测值均十分接近真实值,但这样的预测效果

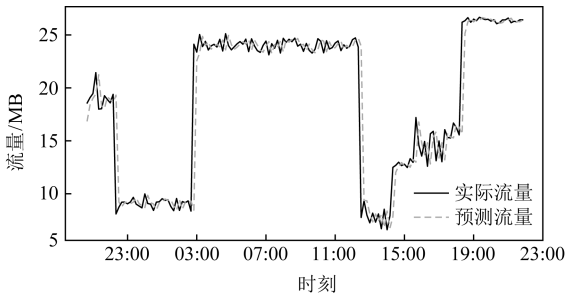

Fig. 4 Prediction effect when the EWMA parameter k is 0.9

图 4 EWMA 参数 $k=0.9$ 时的预测效果

却会带来另一个问题,即预测结果在时间序列走势较为平稳的一段时间内会出现频繁的抖动现象,由于重配置会给系统带来更新的开销,为了提高处理效率,仅在当前周期在线流量的大小跟上一周期的在线流量大小的变化量超过某一固定的阈值时,才会触发带宽分配情况的重配置,一旦出现了图 4 中 03:00～12:00 期间的抖动较大的预测,将会引起频繁的重配置,增加更新压力,即使预测结果十分准确,也不是本文期望达到的最佳效果.而当 EWMA 方法的参数 $k=0.5$ 时,虽然拐点处的预测值不十分贴近真实值,但是比 $k=0.3$ 时预测得更准确,同时,虽然在 03:00～12:00 期间仍然存在着多次抖动,但相较于 $k=0.9$ 的情况有所减少,能够在一定程度上兼顾预测平滑和预测灵敏这 2 点需求.

由此得出了针对这一特定在线流量时间序列的相对最佳 EWMA 参数 0.5,能够使得 EWMA 方法在整段序列上都能得到相对理想的预测效果.但是,为所有的网络情况采用统一固定的参数来进行预测无疑是一个不够灵活的做法.最理想的模式是使 EWMA 方法能够针对不同的网络环境动态地调整其参数 k.例如,对于图 2～4 模拟生成的在线流量序列而言,有 23:00,03:00,12:00,14:00,18:00 这 5 个数据拐点,拐点情况下 EWMA 的参数设置得越大,预测的效果越准确,在前面讨论的 k 分别设置为 0.3,0.5,0.9 这 3 种选择中,显然 $k=0.9$ 是拐点发生时最适合 EWMA 方法采用的参数.而对于 23:00～03:00,03:00～12:00 这 2 段波动较小的数据段,为了达到尽量减少重配置开销的目的,在前面所讨论的 3 个 k 值中,显然应该选择 $k=0.3$.

由于想要实现预测值足够平滑,且在拐点处的预测足够灵敏准确,因此要使用本文提出的 Sliding-k 算法.当设置 EWMA 的参数值 $k=0.3$ 时,使用 Sliding-k 算法得到的预测效果如图 5 所示,对比图 2 所示的 k 同样为 0.3 但仅使用 EWMA 方法的预测效果可以发现,Sliding-k 能够弥补 EWMA 的不足,使拐点出现时(如 03:00 时刻)的预测效果更加准确而贴近真实值.当 $k=0.5$ 时,Sliding-k 的整体预测效果更好,在无拐点的时段(如 03:00～12:00),Sliding-k 能够达到平滑的预测效果,鲜有预测值出现较大落差的情况发生,而在拐点出现的时刻(如 03:00 和 12:00),Sliding-k 能够作出敏感且准确的预测,因而在整个时段内的不同场景下均达到了预期的预测效果,如图 6 所示:

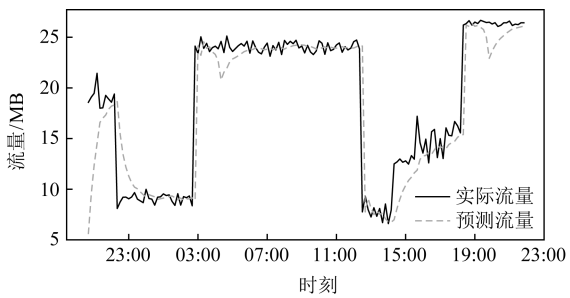

Fig. 5 Prediction effect with Sliding-k when k is 0.3

图 5 Sliding-k 算法在 $k=0.3$ 时的预测效果

由此可以总结出 Sliding-k 的预测准确率和单独使用 EWMA 方法相比有所提升,尤其是在拐点出现时,准确率提升较为明显.这不仅能在图 3 和图 6 的对比中直观地展现,而且能通过准确率的统计来进行验证.为了进一步评估 Sliding-k 算法相较 EWMA 方法的预测准确率提升,在 2 种算法的 k 均设置为 0.5 且均选择了有拐点出现的 01:30～04:30 这一时间段时,分别统计使用 EWMA 和 Sliding-k 算法时的准确率 CDF 图.这一区间的数据同时包含平稳的数据段以及突变的数据拐点,具有普遍性和代表性.单独使用 EWMA 方法时这一拐点区间的准确率 CDF 如图 7 所示,而使用 Sliding-k 算法的准确率 CDF 如图 8 所示.使用 Sliding-k 算法在拐点处进行

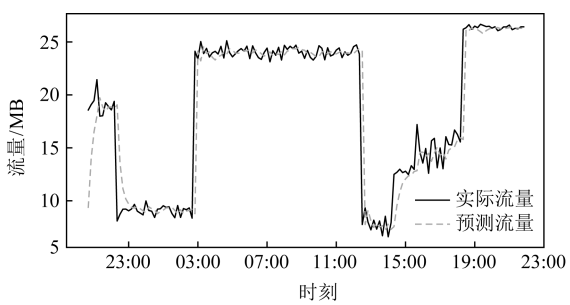

Fig. 6 Prediction effect with Sliding-k when k is 0.5

图 6 Sliding-k 算法在 $k=0.5$ 时的预测效果

预测的准确率明显提高,这一拐点区间内95%的预测的准确率大于90%,而EWMA方法在这一拐点区间内仅有83%左右的预测准确率大于90%.

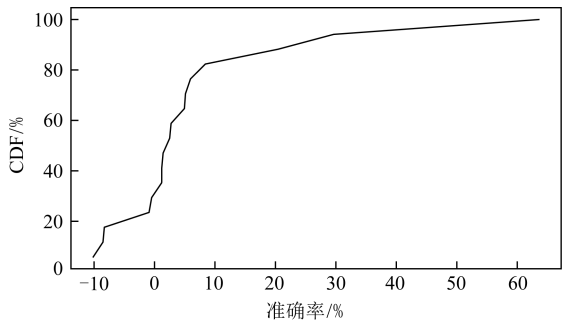

Fig. 7　Accuracy with EWMA when parameter k is 0.5

图7　使用EWMA方法预测且$k=0.5$的准确率

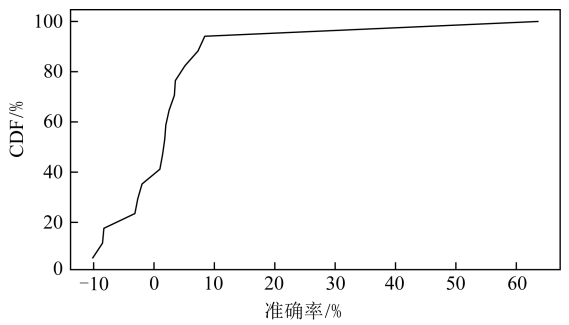

Fig. 8　Accuracy with Sliding-k when parameter k is 0.5

图8　使用Sliding-k算法预测且$k=0.5$的准确率

为了验证改良版的EDF算法(SEDF)的调度效果,在本系统的调度模块中分别应用简单的EDF算法和SEDF进行对比.部分模拟生成的离线流量需求如表1所示.表1中3个离线流量需求均要求从长春数据中心传输到武汉数据中心,到达时间和截止时间的大小是以20:00为原点的累计分钟数,如221即为从某日20:00开始算起的第221分钟.

Table 1　Part of Offline Traffic Demand

表1　部分离线流量需求

No.	到达时间	截止时间	流量大小/B	源DC	目的DC
222	2	221	77 703 485	长春	武汉
301	2	223	36 208 364	长春	武汉
302	2	224	32 986 515	长春	武汉

如果使用简单EDF算法对离线流量进行调度,则编号为222的需求将被优先调度,而表1(表项含义见4.1节有关需求信息的介绍)的其余2个需求由于链路容量有限而无法完成传输,此时3个需求中仅有一个需求能被满足.而如果使用本系统定制的SEDF算法,编号为302和301的需求被先后调度,编号为222的需求则由于链路容量有限而无法完成,但此时3个需求中有2个需求被满足.

从对比实验中观察到,与传统的EDF算法相比,采用可以同时感知流量需求截止时间和流量大小的SEDF算法对离线流量进行调度能够增加可完成传输的离线流量需求的数目.

同时,为了验证Sliding-k预测模块和SEDF调度模块共同协作的效果,实验对比了BDS采用的固定带宽分离方案与本系统的动态带宽分离方案下的24 h内的链路使用情况.实验主要是在系统中为在线流量和离线流量在链路中的比值设定一个固定值来模拟已有的固定带宽分离模式的效果.而本系统的动态带宽分离方案则通过动态地预测当前周期的在线流量需求大小,然后将剩下的空闲带宽都分配给离线流量.

如图9所示,实验结果中,"在线流量占比"代表了在线流量占总带宽的百分比,"固定模式"代表固定带宽分离方案下总的带宽利用率(在线流量和离线流量占总带宽的百分比),"动态模式"代表本系统方案下的总的带宽利用率.从图9可以看到,为在线流量与离线流量设置固定的比值2:3时,离线流量最多只能使用完分配给它的60%的总带宽,尽管在线流量远未使用完分配给它的总带宽的40%,离线流量也不能去借用这部分空闲流量,这就造成带宽的浪费.其中,图9中"在线流量占比"曲线与"固定模式"曲线形状相似是因为离线流量在每周期内都使用了60%的带宽(其上限),也就是说"固定模式"曲线是"在线流量占比"曲线往上平移了60%,这与前面的设想是一致的.

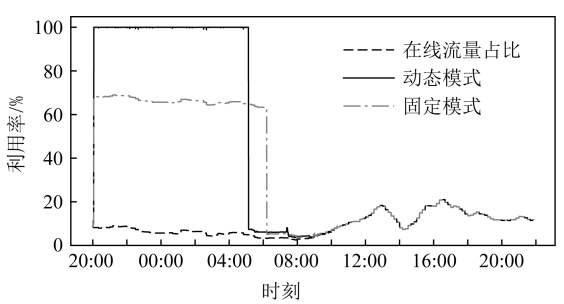

Fig. 9　Bandwidth usage on Nanjing-Wuhan link

图9　南京-武汉链路上的带宽使用情况

此外,我们还通过对链路容量和在线流量进行扩增以模拟业务量不断扩增的场景.如4.1节所述,实验的数据中心网络共分3个区域,为了验证业务量扩增时本方案的效果,在对链路容量进行扩增时,

先找出域内链路带宽总量居中的一个域,然后将其扩增至域间链路容量大小(50 Gbps),记录需要扩增的倍数,然后各个域中各 IDC 的带宽按照这一倍数进行扩增.同时,实验根据在线流量的历史观测值将每条域内链路的在线流量按倍数扩增,使扩增后的在线流量峰值至少达到该链路容量的 60%.从实验结果中发现,当在线流量的需求量不断扩增时,基于预测的调度方案仍然能够优先保障在线流量不受影响的情况下,充分利用链路带宽.

5 结束语

传统固定带宽分离的方式会造成这样一种现象,当在线流量处于谷值的时候,那些预留给它的带宽由于已经被固定地分配给了在线流量而不能被大量离线流量充分利用,从而造成数据中心间链路带宽的浪费.

而本文使用了动态带宽分离的方式,使用将 EWMA 和拐点检测算法结合的 Sliding-k 算法预测在线流量的带宽占用情况,在保证延迟敏感型的在线流量不受影响的条件下将剩余可用的链路带宽容量分配给离线流量.Sliding-k 算法在预测时先判定当前是否出现拐点,然后再决定预测时"回看"多远,这样既能够保证预测的灵敏,又能够在网络变化不大的时候避免频繁的重配置.这种基于预测的流量调度方式可以使链路利用率接近 100%,充分地利用了链路带宽.而离线流量的调度则使用了基于 EDF 的定制算法,从离线流量需求的截止时间和流量需求大小 2 个维度考虑,使成功完成的离线流量需求数尽可能增多.

未来可以考虑使用 EWMA 之外更好的时间序列预测算法结合 Sliding-k 算法提高预测效果,使链路利用率更高而且更加稳定.

参 考 文 献

[1] Kandula S, Menache I, Schwartz R, et al. Calendaring for wide area networks [C] //Proc of the 2014 ACM Conf on SIGCOMM. New York: ACM, 2014: 515-526

[2] Hong Chiyao, Kandula S, Mahajan R, et al. Achieving high utilization with software-driven WAN [C] //Proc of the 2013 ACM Conf on SIGCOMM. New York: ACM, 2013: 15-26

[3] Zhang Yuchao, Nie Xiaohui, Jiang Junchen, et al. BDS+: An inter-datacenter data replication system with dynamic bandwidth separation [J]. IEEE/ACM Transactions on Networking, 2021, PP(99):1-17

[4] Lin Xiao, Ji Shuo, Yue Shengnan, et al. Node-constraint store-and-forward scheduling method for inter-datacenter networks [J]. Journal of Computer Research and Development, 2021, 58(2): 319-337 (in Chinese)
(林霄, 姬硕, 岳胜男, 等. 面向跨数据中心网络的节点约束存储转发调度方法[J]. 计算机研究与发展, 2021, 58(2): 319-337)

[5] Zhang Yuchao, Tian Ye, Wang Wendong, et al. Federated routing scheme for large-scale cross domain network [C] // Proc of IEEE INFOCOM 2020—IEEE Conf on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway, NJ: IEEE, 2020: 1358-1359

[6] Zhang Yuchao, Xu Ke, Wang Haiyang, et al. Going fast and fair: Latency optimization for cloud-based service chains [J]. IEEE Network, 2017, 32(2): 138-143

[7] Jiang Jingjie, Ma Shiyao, Li Bo, et al. Adia: Achieving high link utilization with coflow-aware scheduling in data center networks [J]. IEEE Transactions on Cloud Computing, 2016, 7(2): 431-441

[8] Jain S, Kumar A, Mandal S, et al. B4: Experience with a globally-deployed software defined WAN [C] //Proc of the 2013 ACM Conf on SIGCOMM. New York: ACM, 2013: 3-14

[9] Hong Chiyao, Mandal S, Al-Fares M, et al. B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in Google's software-defined WAN [C] //Proc of the 2018 Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2018: 74-87

[10] Kumar A, Jain S, Naik U, et al. BwE: Flexible, hierarchical bandwidth allocation for WAN distributed computing [J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 1-14

[11] Kumar P, Yuan Yang, Yu C, et al. Semi-oblivious traffic engineering: The road not taken [C] //Proc of the 15th USENIX Conf on Networked Systems Design and Implementation. Berkeley, CA: USENIX Association, 2018: 157-170

[12] Zhang Yuchao, Xu Ke, Yao Guang, et al. PieBridge: A cross-DR scale large data transmission scheduling system [C] //Proc of the 2016 ACM Conf on SIGCOMM. New York: ACM, 2016: 553-554

[13] Zhang Yuchao, Jiang Junchen, Xu Ke, et al. BDS: A centralized near-optimal overlay network for inter-datacenter data replication [C] //Proc of the 13th EuroSys Conf. New York: ACM, 2018: 1-14

[14] Lucas J M, Saccucci M S. Exponentially weighted moving average control schemes: Properties and enhancements [J]. Technometrics, 1990, 32(1): 1-12

[15] Adams R P, MacKay D J C. Bayesian online changepoint detection [J]. arXiv preprint arXiv:0710.3742, 2007

[16] Roberts S W. Control chart tests based on geometric moving averages[J]. Technometrics, 1959, 1(3): 239-250

[17] Jiang Xu, Sun Jinghao, Tang Yue, et al. Utilization-tensity bound for real-time DAG tasks under global EDF scheduling [J]. IEEE Transactions on Computers, 2019, 69(1): 39-50

[18] Casini D, Biondi A, Buttazzo G. Handling transients of dynamic real-time workload under EDF scheduling[J]. IEEE Transactions on Computers, 2018, 68(6): 820-835

[19] Peng Yuhan, Varman P. Fair-EDF: A latency fairness framework for shared storage systems[C] //Proc of the 11th USENIX Conf on Hot Topics in Storage and File Systems. Berkeley, CA: USENIX Association, 2019: 6-6

[20] Muwumba A M, Justo G N, Massawe L V, et al. Priority EDF scheduling scheme for MANETs[C] //Proc of Int Conf on Communications and Networking in China. Berlin: Springer, 2019: 66-76

[21] Yang Kecheng, Guo Zhishan. EDF-based mixed-criticality scheduling with graceful degradation by bounded lateness[C] //Proc of 2019 IEEE 25th Int Conf on Embedded and Real-Time Computing Systems and Applications (RTCSA). Piscataway, NJ: IEEE, 2019: 1-6

Wang Ran, born in 1996. Master candidate. Her main research interests include traffic engineering (TE) and datacenter resources management.

王　然,1996 年生.硕士研究生.主要研究方向为流量工程和数据中心资源管理.

Zhang Yuchao, born in 1989. PhD. Associate professor. Member of CCF, IEEE and ACM. Her main research interests include large scale datacenter networks, content delivery networks, data-driven networks and edge computing.

张宇超,1989 年生.博士,副教授.CCF,IEEE 和 ACM 会员.主要研究方向为大型数据中心网络、内容交付网络、数据驱动网络和边缘计算.

Wang Wendong, born in 1963. Master. Full Professor. Member of IEEE. His main research interests include the next generation network architecture, network resources management and QoS, and mobile Internet.

王文东,1963 年生.硕士,教授.IEEE 会员.主要研究方向为下一代网络体系结构、网络资源管理和 QoS 以及移动网络.

Xu Ke, born in 1974. PhD. Full professor. Fellow of the Chinese Institute of Electronics. His main research interests include computer network architecture, network security and blockchain systems.

徐　恪,1974 年生.博士,教授.中国电子学会会士.主要研究方向为计算机网络体系结构、网络安全和区块链系统.

Cui Laizhong, born in 1984. PhD, professor. Senior member of IEEE and CCF. His main research interests include future Internet architecture and protocols, edge computing, multimedia systems and applications, blockchain, Internet of things.

崔来中,1984 年生.博士,教授.IEEE 和 CCF 高级会员.主要研究方向为未来互联网体系结构和协议、边缘计算、多媒体系统和应用程序、区块链、物联网.