



业务优先级感知的网络流量调度机制

吕华辉, 林志达, 黄宏聪

(中国南方电网有限责任公司, 广东 广州 510623)

摘 要: 针对网络中业务数据流过大、分布不均匀所造成的网络拥塞, 提出一种优先级感知的动态网络流量调度机制。利用令牌桶算法, 根据业务优先级的不同为不同业务分配不同速率的令牌, 以实现业务优先级的划分; 综合考虑业务的优先级及用户节点剩余缓存空间, 对不同的业务采取不同的处理方式; 同时, 以流量到达因素、服务因素及节点缓存为指标定义了一种网络流量调度机制性能指标——分组丢失率。数值结果表明, 所提机制可以对网络中业务优先级进行合理的划分, 从而有效利用网络资源, 预防网络拥塞, 提升网络性能, 为用户提供更加稳定可靠的网络服务。

关键词: 网络拥塞; 流量调度; 令牌桶; 节点缓存; 分组丢失率

中图分类号: TN915

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2017224

Service priority-aware network traffic scheduling mechanism

LV Huahui, LIN Zhida, HUANG Hongcong

China Southern Power Grid Co., Ltd., Guangzhou 510623, China

Abstract: In view of the problem that network congestion which caused by the uneven distribution in large traffic data, a priority-aware dynamic network traffic scheduling mechanism was proposed. By using the token bucket algorithm, different buckets of different rates were assigned to different services according to the service priorities. The priorities of the service and the remaining buffer space of the user nodes was taken into account to deal with different services. At the same time, the traffic arrival factor, service factor and node cache were regarded as target to define a network traffic scheduling mechanism performance indicator: packet loss rate. The numerical results show that the proposed mechanism can rationally divide the service priorities in the network, effectively utilize the network resources, prevent the network congestion, enhance the network performance, and provide the users with more stable and reliable network service.

Key words: network congestion, traffic scheduling, token bucket, node cache, packet loss rate

收稿日期: 2017-05-31; 修回日期: 2017-07-09

基金项目: 国家高技术研究发展计划 (“863” 计划) 基金资助项目 (No.2012AA050801); 国家电网公司科技基金资助项目 (No.5455H7150035)

Foundation Items: The National High Technology Research and Development Program (863 Program) (No.2012AA050801), State Grid Corporation of Science and Technology Project (No.5455H7150035)



1 引言

随着互联网规模的不断扩大、通信新技术的飞速发展以及多媒体应用需求的日益增多,用户对网络服务质量(quality of service, QoS)的要求更加严格^[1,2]。然而,网络中带宽资源极其有限,日益增长的用户流量需求与紧张的网络宽带资源之间的矛盾日益凸显。网络中流量的不均匀分布,极易导致网络拥塞、增加数据传输时延、引发流量抖动甚至分组丢失现象,在对网络性能与服务质量产生负面影响的同时,严重威胁着网络安全^[3,4]。因此,如何实现网络资源的高效利用、缓和用户需求与网络资源之间的矛盾、保证核心网络业务数据稳定传输、为用户提供更好的网络服务,成为发展现代化互联网的重中之重^[5]。

网络流量调度机制^[6],通过实时分析网络中流量特征,合理地将多个用户的业务请求分配至不同的目的服务器进行访问,使不同链路上的带宽资源都能得到有效的利用,实现网络的负载均衡,提升网络吞吐量,保证网络实时高效运行,有效预防网络拥塞,降低数据传输时延,减少时延抖动^[7]。因此,在网络中设计合理、稳定、高效的流量调度机制,对业务流量路由进行调整,使得业务流量在网络中尽可能地分布均匀,提高网络资源利用率,提升网络性能及稳定性,为用户提供更好的网络服务刻不容缓^[8]。

近年来,网络中业务流量调度机制得到了人们的广泛关注。参考文献[9]提出了一种轮转调度(round robin scheduling, RRS)机制,采用轮转循环的方法,将不同用户的业务请求分配到不同的目的服务器,该机制配置简单、易于实现,且无需实时记录每台服务器的负载状态信息,但由于不同服务器的处理能力不同,该机制容易造成服务器间负载不均衡,应用范围有限;为此,参考文献[10]对上述机制进行改进,提出一种加权轮转调度(weighted round robin scheduling, WRRS)

机制,根据每台服务器的处理能力,为每台服务器设置不同的权值,按照权值,采用RRS机制,WRRS机制一定程度上解决了RRS机制服务器间负载不均衡的问题,但在业务请求时间变化较大时,该机制表现不佳;参考文献[11]针对业务请求时间变化较大的情况,提出了一种最小连接度算法(least connectivity algorithm, LCA),以当前服务器上用户的连接数量为指标衡量当前时刻服务器上用户负载情况,LCA机制属于动态调度机制,虽然一定程度上解决了WRRS机制的缺点,但没有考虑服务器处理性能的不同,会使处理能力较高的服务器过早结束对业务请求的处理,变为空闲状态,在这种情况下,网络资源的利用率不高;除此之外,还有研究者提出随机调度机制^[12],即不考虑任何因素,将网络中用户业务请求随机分配给网络中的服务器,该方法简单、易实现,但随机性太强,可能实现负载均衡的效果也可能出现负载不均衡的情况。

针对上述问题,本文提出一种优先级感知的动态网络流量调度(priority aware traffic scheduling, PTS)机制。借助令牌桶算法,依据业务流处理优先级的不同以及其突发特性,为不同的业务数据流分配不同速率的令牌,综合考虑业务优先级及用户节点的剩余缓存容量,对不同的业务采取不同的处理方式,最后,定义了一个业务流量调度机制性能指标——分组丢失率,并通过仿真验证了所提机制的合理性及在分组丢失率性能上体现的优越性。

2 基于令牌桶算法的业务分级策略

令牌桶算法^[13]是一种典型的速率限制和流量整形算法,可用于控制网络中数据的速率以及节点缓存中数据的数量。不同于漏桶算法,令牌桶算法允许一定突发特性数据的发送,容量固定的令牌桶以恒定速率产生令牌,并配备至到达令牌桶的数据。令牌的大小通常是固定的,因此不同

大小的数据通常需消耗不同数量的令牌。此外,令牌桶中可保存的令牌数量应小于令牌桶的最大令牌限额,否则会导致令牌溢出。如果令牌桶中存在令牌,则允许数据发送;反之,数据的发送被暂时搁置。因此,当网络数据在某时刻产生突发行为时,若突发限制在一定额度,且此时令牌桶中有足够令牌,则允许数据以峰值速率发送。

网络中存在多种业务流,按照其特性和用途可划分为视频流、音频流以及普通数据流等。这些业务流都具有明显的突发特性,并且不同的业务流其突发程度以及突发频率存在明显差异。在网络中,业务流通常不是单独传输和处理,而是以混合流的方式到达某一节点。为了保障服务质量,需要对到达某一节点的多种业务流进行识别,并根据业务流自身特性以及网络需求状况对相应业务流进行优先级划分,按其优先级降序可将业务流分别表示为 $F_1, F_2, \dots, F_n$ 。不同业务流间优先级的不等式如下所示,其中 $O(F_i)$ 表示业务流 F_i 的优先级:

$$O(F_1) > O(F_2) > \dots > O(F_i) > \dots > O(F_{n-1}) > O(F_n) \quad (1)$$

为了对不同优先级业务流进行差异化处理,节点 θ 将其缓存空间划分为 N 个队列,分别存储到达节点 θ 的不同业务流。节点 θ 对 N 个队列标注不同的优先级,按优先级降序可将其表示为 $Q_1, Q_2, \dots, Q_n$, 即 Q_1 队列存储 F_1 业务流, Q_2 队列存储 F_2 业务流,以此类推。

传统的令牌桶算法以恒定的速率源源不断地产生相同的令牌。然而,节点缓存队列具有不同的优先级,因此,本文所需的令牌应具备异构性,即不同优先级的队列具有不同类型的令牌,在单位时间内,高优先级的队列应配备更多的令牌。本文将令牌划分为 N 类, TO_i 表示注入 Q_i 队列的令牌的类型,不同类型的令牌的生成速率不同。此外,不同业务流的突发特性不同,为了保障高突发性业务流的端口速率,需要对其注入更多令

牌。本文依据业务流处理优先级以及其突发特性,对不同业务流配备不同速率的令牌。假设某一业务流 F_i 的 Hurst 参数值为 H_i , 由于业务流的长相关特性, H_i 的值通常在 0.5~1 内。 H_i 值越大, F_i 的长相关性越强,其突发特性越明显。根据上述分析,可得业务流 F_i 在单位时间内获得的 TO_i 令牌的数量。这里只存在对数形式, $\log$ 底数不为常数:

$$H_i = \frac{\log(\frac{R_{n_i}}{S_{n_i}})}{\log n_i} - \frac{C_i}{\log n_i} \quad (2)$$

$$N(F_i) = \rho_i e^{\frac{\rho_2 H_i}{i}} \quad (3)$$

其中, ρ 为常值因子, i 为业务流 F_i 的优先级顺序。

3 动态流量调度机制及可行性分析

3.1 动态流量调度机制

业务流 F_i 到达节点 θ , 如果节点 θ 正在对其他业务流进行服务,且当前队列 Q_i 有足够空间容纳刚到达的业务流 F_i 时, F_i 将进入队列 Q_i 等待服务。当某一时刻优先级高于 Q_i 的队列中没有足够的令牌,且 Q_i 队列拥有 $N_y(F_i)$ 令牌时,位于队列 Q_i 队头的前 $N_y(F_i)$ 个 F_i 数据接受服务,进而离开节点 θ , 此时 Q_i 队列如果还有停留的 F_i 数据时,需继续等待服务,因为当前时刻缺少 TO_i 信令。如果在上述过程的持续时间内有更高优先级的业务流停留在高优先级队列内等待转发,且在这段时间已收集部分对应令牌,则在上述过程结束后,转向处理高优先级的业务流,否则,对优先级低于 Q_i 的队列进行类似的处理。

当某一时刻业务流 F_i 由于业务突发导致在节点 θ 入口处的数据到达率非常高时,需考虑相应队列是否有足够缓存空间,以存放到达节点 θ 的业务流。如果空间足够,则业务流直接进入相应队列 Q_i , 否则,需考虑优先级低于 Q_i 的其他队列剩余缓存情况。如果这些队列的剩余缓存空间足



够存放当前时间到达 F_i 的业务流,即满足式(4),则将其部分剩余空间“补偿”给 Q_i ,以存放当前时间到达的 F_i 业务流,其中 $Re(Q_x)$ 表示 Q_x 的剩余缓存容量:

$$(Re(Q_{i+1}) + \dots + Re(Q_N)) \geq F_i \quad (4)$$

当这些队列的剩余缓存空间不足以存放当前时间到达 F_i 的业务流时,需向优先级高于 Q_i 的队列“询问”其是否具有足够剩余缓存空间,若满足式(5),则向高优先级队列“索取”剩余缓存空间,而此时需分析这些高优先级业务流在接下来一段时间内到达节点 θ 的情况,可根据流量预测模型进行预测,预测式如式(6)、式(7)所示,其中 π 为系数向量,可由式(8)迭代得出, $\pi_0 = -1$ 。如果预测到下一时刻这些高优先级业务流到达率非常高,则拒绝向 Q_i “补偿”剩余缓存空间,反之,补偿部分剩余缓存空间给 Q_i 队列:

$$(Re(Q_1) + \dots + Re(Q_{i-1})) \geq F_i \quad (5)$$

$$\hat{X}(h) = \sum_{j=1}^{\infty} \pi_j^{(h)} \hat{X}_{t+h-j} \quad (6)$$

$$\pi_j^{(h)} = \pi_{j+h-1} - \sum_{i=1}^{h-1} \pi_i \pi_j^{h-1}, \pi_j^{(1)} = \pi_j \quad (7)$$

$$\pi_j = \theta_1 \pi_{j-1} + \theta_2 \pi_{j-2} + \dots + \theta_q \pi_{j-q} + \varphi_j \quad (8)$$

由于高突发性导致当前时刻节点 θ 的整个剩余缓存空间都不足以存储到达 F_i 的业务流时,此时需丢弃部分低优先级业务流中的数据,以保障高优先级业务流的服务质量。优先级感知的令牌桶算法以及缓存动态调度机制的伪代码如下所示,其中 $Re(Q_x)$ 表示 Q_x 的剩余缓存容量, Δt 表示节点 θ 服务 F_i 的 $N_\gamma(F_i)$ 个数据所持续的时间。

算法1 优先级感知的动态流量调度机制

Begin

While (F_i 到达 θ) {

If (θ 正在进行服务并且 Q_i 有足够的剩余空间) {

$F_i \rightarrow Q_i$; }

Else If (θ 正在进行服务并且 Q_i 没有足够的剩余空间) {

If ($(Re(Q_{i+1}) + \dots + Re(Q_N)) \geq F_i$) {
 $Q_{i+1} \dots Q_N$ 补偿剩余空间给 Q_i ; }

Else {

If ($(Re(Q_1) + \dots + Re(Q_{i-1})) \geq F_i$) {

Q_i 从 $Q_1 \dots Q_{i-1}$ 请求剩余空间;

预测下一个时间周期到达 $Q_1 \dots Q_{i-1}$ 的 $F_1 \dots F_{i-1}$ 的到达率;

If ($F_1 \dots F_{i-1}$ 在下一时间周期有较高到达率) {

$Q_1 \dots Q_{i-1}$ 拒绝补偿; }

Else {

$Q_1 \dots Q_{i-1}$ 补偿剩余空间给 Q_i ; }

End If

Else {

丢弃部分较低优先级的 F_x ; }

End If

Else {

F_i 的前 $N_\gamma(F_i)$ 个数据接受服务;

}}

End If

End While

While ($Q_1 \dots Q_{i-1}$ 没有足够的令牌并且 Q_i 有 $N_\gamma(F_i)$ 个令牌) {

F_i 的前 $N_\gamma(F_i)$ 个数据接受服务;

F_i 的前 $N_\gamma(F_i)$ 个数据离开 θ ;

}

If ($Q_1 \dots Q_{i-1}$ 有足够的令牌且在 Δt 时间内有部分 $F_1 \dots F_{i-1}$ 到达) {

处理 $F_1 \dots F_{i-1}$; }

Else {

处理 $F_{i+1} \dots F_N$; }

End If

End While

End

3.2 可行性分析

(1) 技术可行性

在用 UDP 传输数据时,由于其不可靠交付的特点,部分数据可能会被丢弃,而本文的机制可保证尽量减少丢弃发生,并优先丢弃低优先级数据,因此可将使用 UDP 传输数据所造成的损失降至最低,本文方案的技术难点主要集中在令牌桶算法以及业务流突发性的预测上,而令牌桶算法是经典算法,已有很多具体应用,本文针对令牌桶算法添加了优先级机制,因此本文基于令牌桶算法的业务分级策略的技术实现难度不大。此外,已有很多关于业务流突发性预测的机制,并已在具体应用中实现,因此业务流突发性预测的技术实现难度也不大。

(2) 仿真可行性

本文所提机制通过 MATLAB 进行仿真试验,而 MATLAB 为经典仿真平台,因此本文的方案是可通过仿真进行验证的。数值结果表明,所提方案可以对网络中业务优先级进行合理地划分,从而有效利用网络资源,预防网络拥塞,提升网络性能,为用户提供更加稳定可靠的网络服务。

(3) 实现复杂度

本文方案的算法复杂度主要集中在缓存分配以及管理过程中,而此过程的执行次数与问题规模呈线性关系,因此本文方案的实现复杂度较低。

(4) 应用可行性

本文方案可运用于很多具体应用场景中,并发挥其作用。例如:在多媒体实时应用中,视频、音频流由于其对时延的敏感性等特点,其优先级较高,因此运用本文机制能够保证在用 UDP 传输数据时,视频、音频流不会有较多分组丢失发生,以此保证其播放质量,进而提升用户体验。

4 仿真结果与分析

4.1 性能评价指标

流量数据的分组丢失率主要由流量到达因

素、服务因素以及节点缓存决定,不同业务流的优先级也会直接影响分组丢失率的大小。流量到达因素是指流量平均到达率以及其突发特性,服务因素是指令牌产生速率。对于高优先级业务流,其对应的令牌产生速率较高,即具有较高的服务速率。此外,高优先级业务流在队列剩余空间“补偿”方面有更高的主动权,因为其对优先级低于自身的队列剩余空间可以直接使用而不需要征求这些队列的“同意”。如果业务流的平均到达率低于其对应的令牌产生速率,即满足式(9),那么此业务流对应缓存队列的负载压力不会太大;反之,对应缓存队列由于较高的负载,可能最终会导致部分流量数据的丢弃。此外,流量的突发特性也会直接影响其分组丢失率的大小,较高的突发性会导致某一时间段内业务流大量到达某一节点,此时缓存空间会以较大概率在短时间内达到饱和。根据上述分析,可获得业务流 F_i 的分组丢失率量化式,如式(10)所示:

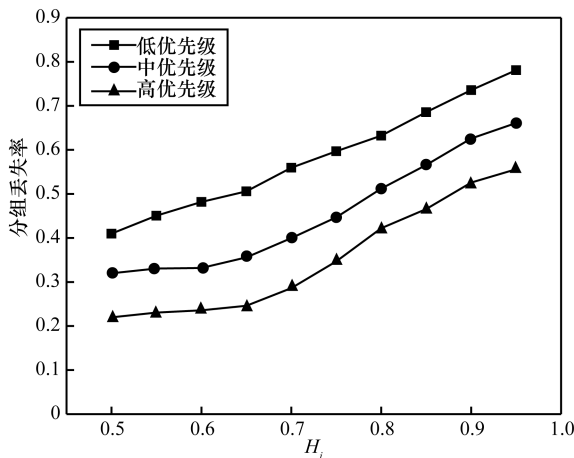
$$\frac{N(F_i)}{\lambda_i} > 1 \quad (9)$$

$$P_{\text{PLR}}(F_i) = \frac{N(F_i)}{\lambda_i} \frac{\rho_3}{\sum_{x=i+1}^N \text{Re}(Q_x) + \sum_{y=1}^{i-1} (1-\delta_y) \text{Re}(Q_y)} e^{H_i} \quad (10)$$

其中, λ_i 为业务流 F_i 的平均到达率, δ_y 为业务流 F_y 在下一时间段突发的概率, ρ_3 为常值因子, H_i 为业务流 F_i 的自相似参数值,表征其突发特性。

4.2 数值结果

本文采用 MATLAB 软件分别在业务流到达率和自相似参数变化的情况下验证本文所提算法的性能。为了验证本文所提出的业务流优先级划分的合理性,本文对 3 种不同优先级业务流(高、中、低)下的分组丢失率进行了仿真。同时,将本文所提机制 PTS 与算法 LCS 在性能指标——分组丢失率方面进行了比较。不同 H_i 下 3 种不同优先级业务流的分组丢失率变化如图 1 所示。

图1 不同 H_i 下 3 种不同优先级业务流的分组丢失率变化

从图 1 可以看出,随着 H_i 的上升,3 种不同优先级业务流的分组丢失率呈现上升的趋势。因为 H_i 增加,业务流的突发性增强,导致分组丢失率增加。高优先级业务流的分组丢失率最低,平均为 35.35%;其次是中优先级业务流,平均分组丢失率为 45.49%;最高的是低优先级业务流,平均分组丢失率为 58.39%。因为本文对不同优先级的业务流进行不同的处理,优先级高的业务流应配备更多的令牌,从而令牌产生速率较高,获得较高的服务速率,进而分组丢失率较低。而优先级低的业务流服务率较低,分组丢失率大。

图 2 反映了在不同到达率的情况下,3 种不同优先级业务流分组丢失率的变化趋势。

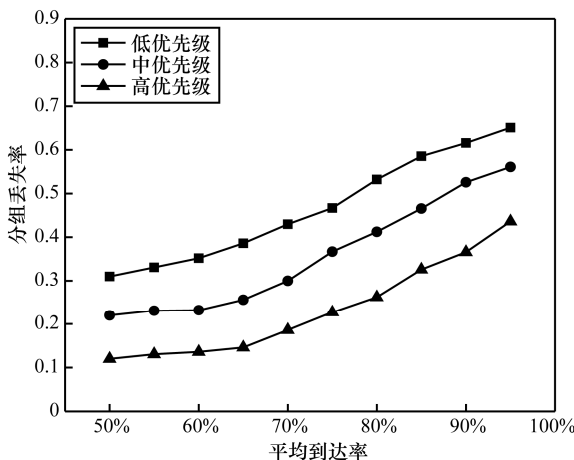
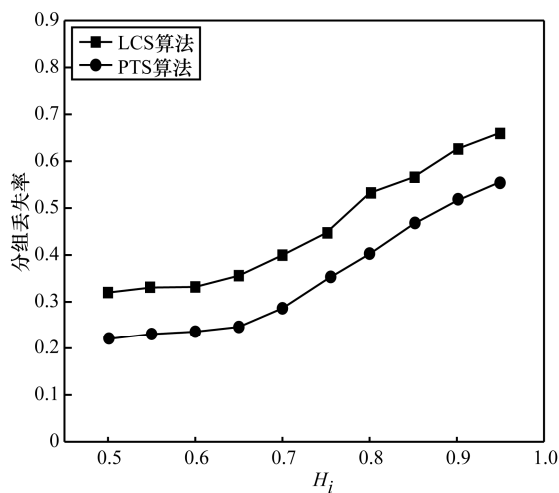


图2 不同到达率下分组丢失率变化

由于业务流的平均到达率增加会导致高于其对应的令牌产生速率,业务流对应缓存队列的负载压力增加,进而导致流量数据丢弃。因此,3 种不同业务流的分组丢失率随着业务流的平均到达率的增加呈现上升的趋势。然而,从图 2 中可以看出,平均到达率为 70%时,高优先级业务流的分组丢失率为 18.61%,中优先级业务流为 29.96%,低优先级业务流为 42.96%。因为不同优先级的业务流所对应的令牌产生速率存在差异,高优先级业务流对应的令牌产生速率高,中优先级业务流令牌产生速率次之,低优先级业务流令牌产生速率最低,从而优先级较高的业务流分组丢失率较低。

图 3 描述两种算法的分组丢失率在不同 H_i 下的变化情况。

图3 不同 H_i 下两种算法的分组丢失率变化

可以看出,节点的分组丢失率会随着 H_i 的增加而不断上升。这是因为业务流自相似参数 H_i 描述业务流的长相关特性, H_i 值越大,业务流的长相关性越强,突发特性就越明显,使得节点的分组丢失率上升。然而,相比于算法 LCS,本文所提的机制 PTS 性能优于 LCS,其分组丢失率平均低于 24.76%。

图 4 展示了在业务流平均到达率变化的情况下,两种算法的分组丢失率变化趋势。

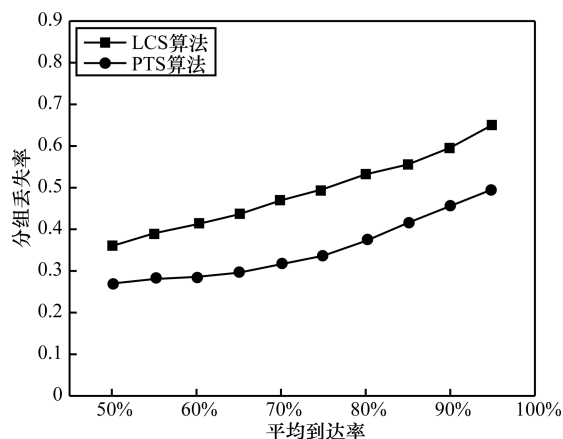


图4 不同到达率下两种算法的分组丢失率变化

随着业务流平均到达率的增加,两种算法的分组丢失率均呈现上升的趋势。之所以分组丢失率上升,原因与之前描述的一样,平均到达率增加,导致比令牌产生速率高,使得业务流队列负载压力增加,以至于会丢弃一些部分流量数据。此外,本文所提机制的性能优于算法 LCS,其分组丢失率平均低于算法 LCS 的分组丢失率 13.74%。原因在于相比于算法 LCS,本文所提机制对节点的缓存进行了动态调度。通过对优先级较高的业务流进行动态的缓存补偿,保证高优先级业务流的服务能力,减少流量数据的丢失,保证了流量数据的完整性。对于优先级较低的业务流,通过向高优先级业务流的队列进行缓存索取,尽可能降低优先级较低的业务流的分组丢失率,实现对整个业务流的质量保障。

5 结束语

本文针对网络中由于网络业务流量过大、分布不均匀的问题,提出了一种优先级感知的网络流量调度机制,可以更加高效地利用有限的网络资源,有效地预防网络拥塞,提升系统性能,为用户提供更高质量、更稳定的网络服务。本文所提网络流量调度机制需要流量识别技术作为支撑,网络流量识别是流量分配与管理、为用户提供精准服务的关键技术,因此未来工作将会集中于网络流量识别领域。

参考文献:

- [1] WU D P, ZHANG H P, WANG H G, et al. Quality of protection(QoP)-driven data forwarding for intermittently connected wireless networks[J]. IEEE Wireless Communication, 2015, 22(4): 66-73.
- [2] 魏峰, 施唯佳, 祝谷乔. 互联网视频中多屏互动技术的应用[J]. 电信科学, 2014, 30(5): 27-32, 39.
WEI Z, SHI W J, ZHU G Q. Multi-screen interaction technologies on internet streaming video[J]. Telecommunications Science, 2014, 30(5): 27-32, 39.
- [3] WU D P, ZHANG P N, WANG H G, et al. Node service ability aware packet forwarding mechanism in intermittently connected wireless networks[J]. IEEE Transaction on Wireless Communications, 2016, 15(12): 8169-8181.
- [4] 王崇霞, 高美真, 刘倩, 等. 混合云联合身份认证与密钥协商协议设计[J]. 电信科学, 2014, 30(4): 95-99, 108.
WANG C X, GAO M Z, LIU Q, et al. Design of united identity authentication and key agreement protocol for hybrid cloud[J]. Telecommunications Science, 2014, 30(4): 95-99, 108.
- [5] WANG H, XIE H, QIU L, et al. COPE: traffic engineering in dynamic networks[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(4): 99-110.
- [6] XION T E, LEANG H P. Traffic congestion controller in multihop communication network[C]//International Conference on Information Technology: New Generations, April 10-12, 2006, Las Vegas, NV, USA. New Jersey: IEEE Press, 2006: 730-735.
- [7] 张平, 李建武, 冯志勇, 等. 认知无线网络基础理论与关键技术研究[J]. 电信科学, 2014, 30(2): 1-13.
ZHANG P, LI J W, FENG Z Y, et al. Research on basic theory and key technology of cognitive radio network[J]. Telecommunications Science, 2014, 30(2): 1-13.
- [8] XU K, ZHANG Z L, BHATTACHARYA S. Profiling internet backbone traffic: behavior models and applications[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(4): 169-180.
- [9] GRUDENIC I, BOGUNOVIC N. Computer cluster scheduling algorithm based on time bounded dynamic programming[C]//Mipro, International Convention, May 23-27, 2011, Opatija, Croatia. New Jersey: IEEE Press, 2011: 722-726.
- [10] 向建军, 白欣, 左继章. 一种用于实时集群的多任务负载均衡算法[J]. 计算机工程, 2003, 29(12): 36-38.
XIANG J J, BAI X, ZUO J Z. A multipletask load balancing algorithm used in real-time cluster system[J]. Computer Engineering, 2003, 29(12): 36-38.
- [11] BRYHNI H, KLOVNING E, KURE O. A comparison of load balancing techniques for scalable Web servers[J]. IEEE Network, 2000, 14(4): 58-64.
- [12] LEI Y C, GONG Y, ZHANG S, et al. Research on scheduling algorithms in Web cluster servers[J]. Journal of Computer Science & Technology, 2003, 18(6): 703-716.



- [13] 张晓彤, 李培娅, 王沁, 等. 接入网 MAC 层 QoS 系统的多业务令牌桶流量整形算法[J]. 计算机科学, 2009, 36(1): 68-70.

ZHANG X T, LI P Y, WANG Q, et al. Multi-service token bucket algorithm of MAC QoS system in access network[J]. Computer Engineering, 2009, 36(1): 68-70.

[作者简介]



吕华辉 (1968-), 男, 中国南方电网有限责任公司高级工程师, 主要研究方向为网络安全。



林志达 (1983-), 男, 中国南方电网有限责任公司高级工程师, 主要研究方向为信息系统运行和安全管理。



黄宏聪 (1987-), 男, 中国南方电网有限责任公司高级工程师, 主要研究方向为信息网络设计、建设与运维管理。