

基于多种性能约束的光突发交换网络动态组装算法

牛大伟 李 岐 于卫波 米志超

(解放军理工大学通信工程学院 南京 210007)

摘要 在综合分析光突发交换网络中组装算法对控制平面处理时延、数据平面资源利用效率和数据平面突发丢失率等多种性能指标的影响的基础上,提出了一种能够同时满足多种网络性能指标约束的边缘节点动态组装算法。该算法根据全网目标性能指标的约束,动态调整组装输出的突发长度和突发时间门限,从而实现同时满足全网链路资源利用率和控制平面突发丢失概率的双重性能约束。

关键词 光突发交换网络,组装算法,资源利用率

中图分类号 TN915.26 **文献标识码** A

Assembling Algorithm of Optical Burst Switching Network under Multiple Performances Restriction

NIU Da-wei LI Qi YU Wei-bo MI Zhi-chao

(Communications Engineering Institute of PLA University of Science and Technology, Nanjing 210007, China)

Abstract The paper proposed an assembling algorithm for edge node of the optical burst switching networks. The algorithm can satisfy multiple network performances at the same time. The foundation of the algorithm is based on the analysis of the impact of assembling algorithm on control plane's end to end delay, resource utilization of data plane and the drop ratio of the burst. The algorithm can adjust the time threshold and the queue length threshold according to the network performances' restriction, and satisfies the desired performance of resource utilization and burst drop ratio at the same time.

Keywords Optical burst switching networks, Assembling algorithm, Utilization of resource

光突发交换(Optical Burst Switching, OBS)是一种在光层直接进行报文级交换的技术,具备密集波分复用系统中波长资源的统计复用能力。在向全光交换演进的过程中,OBS网络被认为是一种较为中肯的、可实现性较强的方案^[1]。

OBS网络边缘节点的组装算法是影响网络性能的一个重要因素。目前提出的组装算法主要分为两类,即固定门限组装算法^[2,3]和动态门限组装算法^[4,5]。固定门限算法不适合在未来光交换网络中应用。动态门限算法可以依据接入网输入速率或者核心网络负载状态的变化动态调整组装触发门限值,以达到自适应网络状态的变化的目的。文献[5]提出的动态组装算法以边缘节点统计的接入网络速率为输入参数,当接入流量较大时使用较大的门限以平滑输出突发流量(以Bursts/s衡量),反之则使用较小的门限以降低组装时延代价。这一类算法仅考虑组装算法对接入流量的整形作用而未考虑网络背景流量和核心节点处理能力对组装算法的约束。文献[4]提出的动态算法以核心节点控制平面所能承受流量上限为约束,根据核心节点网络流量的状态动态地调整边缘节点的组装门限以适应核心节点的承受能力。上述动态算

法仅从接入流量整形和限制突发输出速率以适应核心节点控制平面处理能力的角度对组装门限进行优化。事实上,组装算法的实际性能对网络节点结构和网络部署较为敏感,并且其参数会受到网络多种性能指标的约束。例如文献[6]对组装算法门限值与网络链路资源利用率和突发丢弃率的量化分析显示:基于无波长转换器核心节点的光突发交换网络几乎不存在有效的组装门限。在核心节点采用主流光交换矩阵且期望突碰撞概率在 10^{-4} 以下时,核心节点至少需要配置30个以上的波长转换器,才能实现期望性能指标。

本文以文献[6]的组装门限参数理论为基础,开发出一种基于多目标约束的动态组装算法MORAA(Multiple Object Restricted Assembling Algorithm)。MORAA算法能够以核心节点背景流量、核心节点波长转换器配置数量为输入条件并以目标突发丢失率、目标链路资源利用效率、目标控制平面丢失率等性能指标为约束条件,动态确定组装算法的有效门限区域,并在边缘节点队列状态进入有效组装区域时,形成突发发送,从而实现预期的网络性能指标。

本文第1节简要介绍和回顾文献[6]中提出的组装算法

到稿日期:2013-08-01 返修日期:2014-02-21 本文受国家自然科学基金重点项目:基于光学原理的射频任意波产生器及其在光纤传输系统中的应用(61032005),国家973计划项目:无线网络多域认知理论体系研究(2009CB320402),解放军理工大学优秀本科生创新基金项目资助。

牛大伟(1979—),男,博士,讲师,主要研究方向为光网络、网络优化等,E-mail:flyndw@sina.com;李岐(1994—),男,主要研究方向为光通信、光网络;于卫波(1976—),男,博士,副教授,主要研究方向为网络优化、宽带无线通信;米志超(1975—),男,副教授,主要研究方向为网络优化、宽带无线通信。

门限临界值的理论;第2节提出并详细描述了多目标约束下的组装算法 MORAA;第3节通过仿真分析了算法的整体性能;最后是结束语。

1 组装算法门限值的选取依据

1.1 组装算法的时间门限取值

OBS网络边缘节点的组装时间门限直接决定了边缘节点输出突发的速率并引入了组装时延。假设边缘节点组装算法的平均组装时间为 T_h ,则该节点输出的突发速率为 $\lambda_B = 1/T_h$ 。假设网络中流经每个核心节点的平均数据流数为 K (以数据的入口边缘节点和出口边缘节点地址对标识流),则通过核心节点的平均背景流量为 $\lambda_c = \lambda_{B1} + \dots + \lambda_{BK}$,其单位为突发/秒(bursts/s)。由于OBS网络中控制平面对资源的正确预留是成功转发数据突发的前提,因此必须保证输入核心节点的总背景流量 λ_c 低于核心节点控制平面的处理速率上限。假设核心节点处理1个突发控制报文的时间为 $1/\mu$ s,核心节点的 K 个数据流具有相互独立同分布的输入流量且要求核心节点输入的背景流量归一化负载上限为 κ ($0 < \kappa < 1$),则每个边缘节点的突发输出速率上限应为 $\lambda_B = 1/T_h \leq \kappa \cdot \mu/K$,每个边缘节点的组装算法时间门限下限如式(1)所示。

$$T_h \geq \frac{K}{\kappa \cdot \mu} \quad (1)$$

式(1)表明,组装算法时间门限下限的取值受核心节点控制平面处理能力约束。当组装算法的时间门限值低于式(1)中的 T_h 时,输入核心节点控制平面的总背景流量将超过其归一化负载上限 κ ,从而导致突发控制报文的溢出和丢弃。

另外,由于核心节点控制平面对突发的处理采用存储转发模式,因此突发控制报文在核心节点控制平面的逗留时间可利用M/G/1或M/D/1模型计算得出^[7]。假设突发逗留平均时间为 $W(\lambda_c)$ (其中 λ_c 为核心节点控制平面的背景流量),则边缘节点组装算法输出突发的平均间隔应该大于 $W(\lambda_c)$,即 $T_h \geq W(\lambda_c)$,否则核心节点的队列长度将趋于无穷大。综合式(1)可得出最小组装时间门限的取值标准,如式(2)所示。式(2)的工程含义为:边缘节点组装算法的时间门限的下限受核心节点控制平面处理能力及其背景流量约束,其中的背景流量可以采用OBS网络测量的方法实时获取^[8]。

$$T_{\min} = \text{MAX}\left(\frac{K}{\kappa \cdot \mu}, W(\lambda_c)\right) \quad (2)$$

组装算法时间门限的上限 T_{\max} 主要受端到端可接受时延的限制,例如对于一般的IP业务而言,干线网络的可接受时延约为10ms左右。因此,一般 T_{\max} 选择固定值 $T_{\max} = T_E$,其中 T_E 为数据业务端到端可接受时延的上限。

1.2 组装算法的队列门限取值

OBS网络边缘节点的队列门限决定了边缘节点输出数据突发的长度。假设边缘节点组装算法的队列门限为 L_h 且传输链路速率为Cbit/s,则边缘节点输出的数据突发的持续时间为 $T_B = L_h/C$ 。在OBS网络中,控制平面成功预留波长资源后将在数据突发时刻到达前完成光交换矩阵(也可称为光交叉连接器OXC)的配置,使输入链路的波长信道与输出链路的波长信道实现联通。OXC断开原有连接并建立新的连接需要一定的时间,在这段时间内链路无法传输数据,这段时间称为OXC配置时间 T_{oc} 。目前的OXC主要有基于

MEMS器件和基于声光器件两类^[9,10],其配置时间从10 μ s到10ms不等。由于存在OXC配置时间,因此输出链路信道的相邻占用状态之间的空闲状态的持续时间至少应大于OXC配置时间,否则信道的占用状态为无效占用,即无法完成控制平面预期的数据转发功能。假设网络中目标信道负载率下限为 η ($0 < \eta < 1$),则 $\frac{T_B}{T_B + T_{oc}} \geq \eta$,假设信道速率为 C ,则可得到边缘节点组装算法队列门限的取值下限 L_{\min} ,如式(3)所示。

$$L_{\min} = \frac{\eta T_{oc} C}{(1-\eta)} \quad (3)$$

式(3)的工程含义为:边缘节点组装算法的队列门限的下限受核心节点光交换矩阵配置器件的约束。在一定的期望波长资源利用率约束条件下,OXC设备配置时间越长,组装算法的最小队列门限值越大,反之则最小队列门限值越小。

OBS网络的数据平面是无缓存的全光交换,当输出链路存在争用时(资源竞争),无缓存系统将产生突发丢失现象。因此,网络规划必须设定一个最低期望突发丢失率 P_b 以满足用户业务的服务质量需求。假设输出波长资源的数目 w 为服务员的个数,输入数据突发的背景流量为 λ_c (burst/s)且满足泊松分布,由上述分析可知,数据突发的平均持续时长 $T_B = L_h/C$ 即为服务员的平均服务时间。由排队理论分析,此时的突发丢失率可由爱尔兰B公式计算。为满足期望的丢失率 P_b 的需求,组装算法的队列门限的上限 L_{\max} 如式(4)所示。其中的 $Erlb(\cdot)$ 为在一定的突发丢失率、背景流量以及波长数目的约束条件下调用爱尔兰B公式计算的最大突发持续时间。式(4)的工程意义在于,边缘节点组装算法队列门限的上限受OBS网络数据平面突发丢失率性能和背景流量状态的约束,期望丢失率越低,背景流量越大,则队列门限取值的上限约束值越小即有效取值区域越小,反之上限约束值越大即有效取值区域越大。

$$L_{\max} = CErlb(P_b, \lambda_c, w) \quad (4)$$

综上所述,OBS网络边缘节点动态组装算法的触发门限的有效取值区域受网络控制平面处理能力、数据平面的期望丢弃概率、链路资源的利用效率以及网络核心节点背景流量等多种网络性能和网络状态的约束。动态算法必须根据网络状态和约束目标的变化实时调整门限参数上限和下限,并且在有效区域内动态调整门限值。本节的组装门限区域可归纳为式(5)。

$$\left. \begin{aligned} T_{\max} &= T_E \\ T_{\min} &= \text{MAX}\left(\frac{K}{\kappa \mu}, W(\lambda_c)\right) \\ L_{\max} &= CErlb(P_b, \lambda_c, w) \\ L_{\min} &= \frac{\eta T_{oc} C}{(1-\eta)} \end{aligned} \right\} \quad (5)$$

1.3 配置波长转换器时的组装门限临界值

当核心节点配置了波长转换器时,资源预留算法需要在多个可用波长信道中选择一个最优转发信道。此时突发控制报文在控制平面的逗留时间取决于资源预留算法的时间复杂度,不同的资源预留算法,其处理时间的概率分布不同。核心节点控制平面对突发控制报文的处理时间的分布概率如式(6)所示^[11]。其前提条件是偏置时间为在区间 $[0, D]$ 内均匀分布的随机变量,其均值为 $\frac{D}{2}$ 。

$$P(t = n\tau) = \frac{(\lambda_c D/2)^n}{n!} e^{-\lambda_c D/2} \quad (6)$$

在输入背景流量为泊松分布时,可以使用 M/G/1 模型分析控制平面的排队行为。突发控制报文在核心节点的平均逗留时间如式(7)所示。其中 $N=D/\tau$ 为最大偏置时间相对于控制平面一次查表对比时间的倍数。

$$W(\lambda_c) = \tau \left(\frac{1}{2} \lambda_c N + \frac{\lambda_c N + 2}{N(2 - \lambda_c^2 N)} \right) \quad (7)$$

分析式(7)可以看到,当公式右侧第 2 项分母趋向于 0 时,平均逗留时间 $W(\lambda_c)$ 为无穷大,即控制平面所能承受的背景流量极值,如式(8)所示。

$$\lambda_p = \sqrt{\frac{2}{N\tau^2}} \quad (8)$$

可以认为当输入背景流量如式(8)所示时,控制平面的归一化负载达到 1。若核心节点允许输入的目标归一化负载为 κ ,则将式(7)和式(8)代入式(2)可得到组装算法时间门限下界,如式(9)所示。

$$T_{\min} = \text{MAX} \left(\frac{K\tau}{\kappa} \sqrt{\frac{N}{2}}, \tau \left(\frac{1}{2} \lambda_c N + \frac{\lambda_c N + 2}{N(2 - \lambda_c^2 N)} \right) \right) \quad (9)$$

假设核心节点配置了 w 个波长转换器,则存在 w 个服务员为输入链路到达的数据突发服务,此时的数据平面可采用 M/M/w/w 排队模型建模。将式(9)代入式(5)且当 OBS 网络中的核心节点配置的波长转换器数目为 w 时,其边缘节点组装算法的组装门限取值区域如式(10)所示。

$$\left. \begin{aligned} T_{\max} &= T_E \\ T_{\min} &= \text{MAX} \left(\frac{K\tau}{\kappa} \sqrt{\frac{N}{2}}, \tau \left(\frac{1}{2} \lambda_c N + \frac{\lambda_c N + 2}{N(2 - \lambda_c^2 N)} \right) \right) \\ L_{\max} &= \text{CErlb}(P_o, \lambda_c, w) \\ L_{\min} &= \frac{\eta T_{\text{acc}} C}{(1 - \eta)} \end{aligned} \right\} \quad (10)$$

2 MORAA 算法

第 1 节中给出了在多种网络性能指标约束条件下,组装算法时间和队列门限的临界值表达式,即以组装时间门限值为横轴、以组装队列门限值为纵轴时,存在一个有效区域 $[T_{\min}, T_{\max}, L_{\min}, L_{\max}]$ 。当边缘节点形成突发发送时,其队列状态必须落入上述边界值所形成的象限内,否则无法满足上述多种网络性能约束条件。

本节根据上述组装参数的分析,构造了一种基于多目标约束的动态组装算法(MORAA-Multiple Object Restricted Assembling Algorithm)。MORAA 算法能够以核心节点背景流量、核心节点波长转换器配置数量为输入条件并以目标突发丢失率、目标链路资源利用效率、目标控制平面丢失率等性能指标为约束条件,动态确定组装算法有效区域 $[T_{\min}, L_{\min}, T_{\max}, L_{\max}]$ (见图 1)。

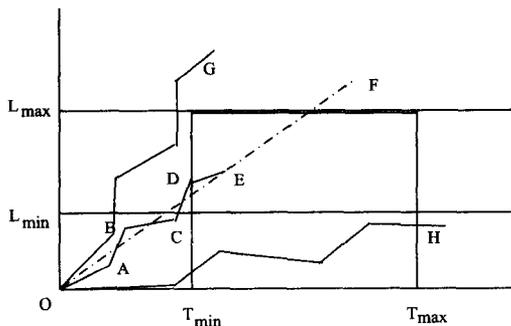


图 1 MORAA 算法示意图

如图 1 所示,边缘节点较为适中的接入流量的轨迹如曲线 A-B-C-D-E(实际上,这个轨迹是由排队时间和队列长度等队列状态体现出来的)。由于接入流量瞬时状态的突发特性,其轨迹曲线并不是一条直线,而是具有一定的波动性。图中所示的 E 点即为队列状态首次进入有效组装区域,从组装时延代价最小的角度衡量,E 点应为其最佳突发组装点。E 点的横坐标分量为该突发的组装时延而纵坐标分量为其突发长度。MORAA 算法即是基于这种在满足网络多目标约束条件前提下获取最小组装时延原则的动态组装算法。该算法的基本思想如下:

当边缘节点队列状态首次进入有效组装区域内时(见图 1 中的 E 点),组装形成突发发送;

当边缘节点接入流量过大(图 1 中左侧的曲线)而导致其队列状态轨迹无法进入有效区域时,节点将在其队列组装时延首次达到最小组装门限 T_{\min} 时(图中 G 点)形成突发发送。由图中可以看出,此时的突发长度将超过最大长度门限 L_{\max} ,所以边缘节点将利用随机丢弃策略,丢弃部分报文,从而使突发长度降至 L_{\max} ;

当边缘节点流量过小(图 1 中下部的曲线)而导致其队列无法进入有效组装区域时,节点将在其队列组装时延首次达到最大组装门限 T_{\max} 时(图中 H 点)形成突发发送。由图中可以看到,此时的突发长度尚未达到多目标约束条件下的最小突发长度门限 L_{\min} ,所以边缘节点将利用填充报文对其进行填充,从而使突发长度达到 L_{\min} 。

图 2 与图 3 为 MORAA 算法的伪代码描述,与第 2 节和第 3 节介绍的算法不同,MORAA 算法包括 2 个进程。Process1(见图 2)为有效组装区间计算进程而 Process2(图 3)为动态组装进程。

Process1 分为 2 步执行。步骤 1 为初始化阶段,其中的 get() 指代边缘节点由网络管理系统或者网络规划过程获取干线网端到端最大时延目标 T_E 、相对于核心节点资源预留算法的 1 次查表对比时间的归一化最大偏置时间 N 、核心节点控制平面的目标归一化负载 κ 、目标链路资源利用效率 η 、目标突发冲突概率 P_o 、核心节点配置的波长转换器数目 w 以及单信道容量 $C(\text{bits/s})$ 。如前所述,上述参数在网络运行周期中基本为预设量和常量,所以其可以通过网络规划过程获取。另外,步骤 1 中的 test() 表示获取算法需要的一些稳定参数,这些参数值在网络运行期间一般是相对稳定的,但是对边缘节点而言无法在网络运行前先验获取,而只能在网络运行初期的测试阶段获取,这些参数包括:核心节点资源预留算法的 1 次查表对比时间 τ 、与本队列(队列一般由入口出口边缘节点对唯一标识)共享瓶颈核心节点的数据流数目 K 、核心节点光交换矩阵配置时间 T_{acc} 。MORAA 算法经过一段时间的步骤 1 执行并获取所需的参数后将循环执行步骤 2。

MORAA 算法 Process1 的步骤 2 的执行如下:

调用文献[7]提出的核心节点背景流量估计算法(图中的 estiamte())实时获取瓶颈核心节点的背景流量 λ_c ;

根据波长转换器的配置情况(w),计算动态组装算法的有效组装区域 $[T_{\min}, L_{\min}, T_{\max}, L_{\max}]$;

返回步骤 2 的起始位置循环执行。

Process1(effective zone);

step1(init);

```

get( $T_E, N, \kappa, \eta, P_b, C, w$ );
test( $\tau, K, T_{occ}$ );
step2(calculation);
 $\lambda_c = estimate()$ ;
If( $w == 0$ )
{
 $T_{max} = T_E$ 
 $T_{min} = MAX(\frac{K}{\kappa \times \mu}, \frac{\lambda - 2 \times \mu}{2 \times \mu (\lambda - \mu)})$ 
 $L_{max} = C \times Erlb(P_b, \lambda_c, 1)$ 
 $L_{min} = \frac{\eta \times T_{occ} \times C}{(1 - \eta)}$ 
}
else
{
 $T_{max} = T_E$ 
 $T_{min} = MAX(\frac{K \times \tau}{\kappa} \times \sqrt{\frac{N}{2}}, \tau \times (\frac{1}{2} \times \lambda_c \times N + \frac{\lambda_c \times N + 2}{N \times (2 - \lambda_c^2 \times N)})$ 
 $L_{max} = C \times Erlb(P_b, \lambda_c, 1)$ 
 $L_{min} = \frac{\eta \times T_{occ} \times C}{(1 - \eta)}$ 
}
end if
goto step2
end process(effective zone)

```

图2 MORAA 算法伪代码描述

Process1 在计算出有效组装区域后将 $[T_{min}, L_{min}, T_{max}, L_{max}]$ 实时与 Process2 共享, Process2 在收到接入网络的报文后将依据 Process1 共享的有效组装区域动态决定突发组装的时机, 从而满足网络性能的多目标需求。

```

Process2(dynamical assembling);
step1(init);
queue_len=0, queue_time=0;
step2(assembling);
wait until a packet arriving
queue_len += pkt_len;
if(queue_time == 0)
queue_time=current_time;
end if
get( $[T_{min}, L_{min}, T_{max}, L_{max}]$ );
if((queue_time, queue_len)  $\subset$   $[T_{min}, L_{min}, T_{max}, L_{max}]$ )
queue_len=0;
queue_time=0;
assemble burst;
else
if ((queue_time, queue_len)  $\subset$   $[T_{min}, L_{max}, \infty, \infty]$ )
discard pkts until( queue_len ==  $L_{max}$ )
queue_len=0;
queue_time=0;
assemble burst;
else
if((queue_time, queue_len)  $\subset$   $[T_{max}, 0, \infty, L_{min}]$ )
padding until( queue_len ==  $L_{min}$ )
queue_len=0;
queue_time=0;
assemble burst;
else
goto step2
end process(dynamical assembling)

```

图3 MORAA 算法伪代码描述

MORAA 算法的 Process2 中步骤 1 初始化的变量 queue_len 为实时队列长度指示变量, queue_time 为队列存在的时间。Process2 中的步骤 2 是算法的主要执行体, 当边缘节点接收到接入报文后将执行如下操作:

更新队列长度, 如果该报文为队列中的首个报文, 则将当前时间(current_time)赋值给全局变量 queue_time;

获取 Process1 共享的有效组装区域 $[T_{min}, L_{min}, T_{max}, L_{max}]$;

若当前队列状态(queue_time, queue_len)处于有效组装区域内, 则形成突发发送;

若接入流量过大而导致当前队列状态(queue_time, queue_len)已经超过有效组装区域, 则对队列进行随机丢弃, 直到其队列长度达到 L_{max} , 并形成突发发送;

若接入流量过小而导致当前队列状态(queue_time, queue_len)在 T_{max} 到达之前无法进入有效组装区域, 则对队列进行填充, 直到其队列长度达到 L_{min} , 并形成突发发送;

否则, 返回步骤 2, 等待下一次接入报文的到达。

综上所述, MORAA 算法以边缘节点组装队列首次进入有效组装区域为触发突发发送的条件(其中的有效组装区域随核心节点负载等网络状态和配置参数的变化而动态变化), 当队列状态无法进入有效区域时, 采用随机丢弃(接入负载过大)或者填充(接入负载过小)的方式使其输出的突发长度和组装时间满足目标丢弃率、信道利用率以及端到端时延等多目标的性能需求。

3 算法仿真与性能分析

在第 2 节, 我们开发了一种能够满足多种目标性能约束的动态 MORAA 算法。本节利用 OPNET 仿真平台实现了该动态组装算法并与现有的动态组装算法进行比较, 从而评估 MORAA 算法的性能。仿真所基于的拓扑为图 4 所示的 NSFNET。

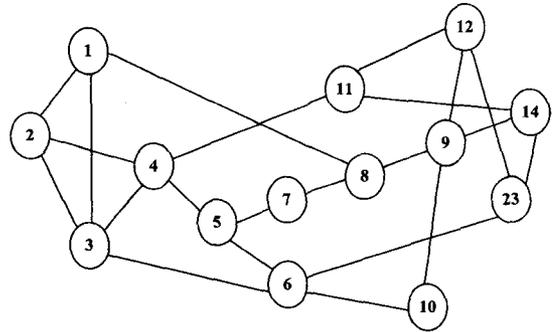


图4 NSFNET 拓扑

网络中的参数配置如下:

- 网络中所使用的光纤链路的单信道速率为 $C = 10\text{Gb/s}$;
- 核心节点资源预留算法采用 LAUC-VF 算法;
- 资源预留算法的一次查表比对时间为 $\tau = 10^{-5}\text{s}$;
- 核心节点控制平面的目标最大归一化负载上限 $\kappa = 0.8$;
- 最大偏执时间 $D = 100 \cdot \tau$, 即 $K = 100$;
- 核心节点光交换矩阵配置时间 $T_{acc} = 10^{-3}\text{s}$;
- 目标突发冲突概率上限为 $P_b = 10^{-4}$;

- 边缘节点最大组装时延上限 $T_E = 0.01s$;
- 目标信道资源利用率下限为 $\eta = 90\%$ 。

为比较 MORAA 算法与其他动态算法的性能,本节也在 OPNET 上实现了另外一种 DTP(Data-length Time-lag Product)算法^[12],该算法以边缘节点队列的长度和排队时延的乘积为判断突发组装的门限。假设在第 i 个接入报文到达时,边缘节点队列长度为 L_i 比特,此时排队时间为 T_i 秒,则此时边缘节点计算其队列长度与时延乘积 $B_i = T_i \times L_i$,当 B_i 大于乘积门限 B_n 时,边缘节点形成突发发送。假设接入边缘节点的 IP 报文的平均速率为 ρ_a ,则边缘节点的队列长度为 $L_i = \rho_a \times T_i$,边缘节点的排队时延乘积 $B_i = \rho_a \times T_i^2$ 。当 $B_i = \rho_a \times T_i^2 = B_n$ 时,形成突发发送,则可得到边缘节点突发组装时的排队时延为 $T = \sqrt{\frac{B_n}{\rho_a}}$ 。DTP 算法的核心思想如式(11)所示,可见这种算法可以使边缘节点的突发输出速率与接入边缘节点的 IP 报文的速率变化相适应,从而实现边缘节点输出流量的平滑。算法的优点在于其可以自动跟踪接入流量的变化,而无需专门的估计和跟踪过程。其缺点在于以下两点:

DTP 算法没有考虑核心节点背景流量的影响,无法根据网络中瓶颈核心节点的处理能力动态调整组装算法的输出速率;

DTP 算法门限值 B_n 的选取标准较难界定,不同的网络配置和不同的节点处理能力需要设置不同的 B_n ,但是文献^[12]没有给出具体的参数设置准则,离未来 OBS 网络实用化距离较远。

$$\frac{d(T)}{dt} = \sqrt{\frac{B_n}{\rho_a^3}} \times \frac{d(\rho_a)}{dt} \quad (11)$$

3.1 输出链路资源利用率

图 5 为核心节点配置 30 个波长转换器时的输出链路平均资源利用率曲线。由图可见:MORAA 能够获取目标链路资源利用率性能($\eta = 90\%$)。DTP 算法的性能取决于门限值 B_n 的设定,当门限值设定较高时,可以接近 MORAA 算法性能,当门限较低时,所获取的链路资源利用率性能远低于目标约束性能。然而 DTP 算法的门限值 B_n 的设定与具体的网络配置有关,不同的网络参数配置(如输出链路速率、波长转换器数量以及控制平面处理速度等)的最佳 B_n 值有所不同,并且很难找到有效的计算方式,只能依据经验测试的方法获取。MORAA 算法除必要的参数输入外无需额外的计算和处理。另外,MORAA 算法考虑了核心节点的背景流量和控制平面处理能力的影响,而 DTP 算法无法实现核心节点的适应性。

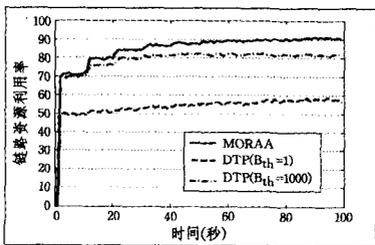


图 5 输出链路信道资源利用率($w=30$)

3.2 核心节点背景流量

图 6 为仿真中统计的瓶颈核心节点背景流量的均值曲线。可以看到在门限设置较小时 DTP 算法的突发输出速率很大,在核心节点引入的背景流量(接近 10^5 pkts/s)较大。由

于核心节点资源预留算法的单个查表对比时间为 $\tau = 10^{-5}s$,而处理一个 BCP 的时延将远大于 τ ,因此当 $B_n = 1$ 时,会因控制平面队列溢出而引起大量的 BCP 丢失和 IOT 丢弃。当边缘节点使用 MORAA 算法或者 DTP 算法的 $B_n = 1000$ 时,可以看到其在核心节点控制平面引入的 BCP 速率较小,基本能够适应控制平面的性能需求,满足控制平面的目标约束。值得注意的是 MORAA 算法是在背景流量测量的基础上动态调整输出速率以适应核心节点的控制平面性能约束,而 DTP 算法并没有核心节点控制平面当前流量的知识,所以其在网络条件较为复杂的情况下,无法实现网络状态的跟踪。

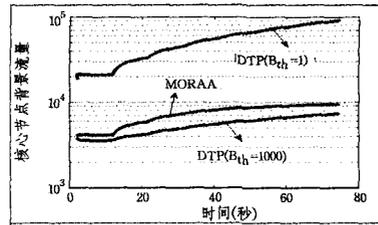


图 6 核心节点背景流量统计

3.3 突发端到端时延

图 7 为采用 MORAA 算法和 DTP 算法统计的数据报文端到端时延。仿真中计算的端到端时延主要包括边缘节点组装时延以及核心节点资源预留算法搜索资源空隙的时延(核心节点处理时延),而忽略了传输时延和传播时延。可以看到:应用 MORAA 算法时,报文的端到端时延较为稳定,其组装时延与核心节点处理时延之和低于 $0.02s$ 。DTP 算法的端到端时延明显高于 MORAA 算法,原因在于其组装时延和资源预留算法时延随流量的增加而快速增加。

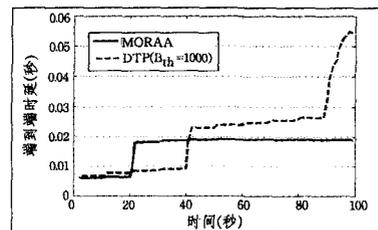


图 7 数据报文端到端时延

综上所述,仿真证明 MORAA 算法能够满足网络控制平面和数据平面的多种性能约束目标,实现依据接入网络流量和背景流量的状态实时动态调整组装输出速率和输出突发长度的功能。相对于其他动态算法,MORAA 算法具有动态地同时满足多种性能约束目标且对门限设置不敏感、有效组装区域的计算准则清晰等优点。

结束语 本文提出了一种多目标约束条件下的动态边缘节点组装算法(MORAA 算法)。该算法能够根据网络状态的变化,以控制平面和数据平面性能的联合约束为优化目标,动态调整组装算法的时间门限和队列门限,以达到适应网络性能需求的目的。仿真结果表明,MORAA 算法能够满足多种网络预期性能指标约束。下一步的工作将重点研究多瓶颈核心节点条件下的算法性能分析。

参考文献

[1] 管爱红,张波云,张元,等.光突发交换网络中基于优先级与突发包分割的光缓存方法[J].激光与光电子学进展,2011,48(6):1-6

(下转第 103 页)

离,但是在许多室内定位的实际应用中,只要定位到移动用户在哪间教室即可,因此当前的精度已经可以满足定位需求。

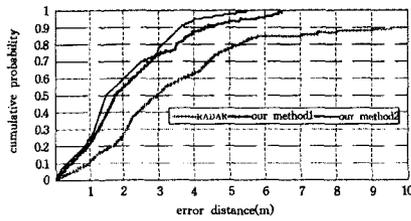


图6 定位误差比较图

表1 25%、50%和75%的测试点的误差距离

方法	25th(\leq (m))	50th(\leq (m))	75th(\leq (m))
RADAR	1.93	2.94	4.69
Our Method1	1.14	1.75	3.05
Our Method2	1.05	1.5	2.95

结束语 本文利用边界盒算法和改进的二分范围搜索算法,给出了一个基于RSSI的WLAN定位方法。所提出的方法通过线性二分范围搜索减小了查找空间,提高了实时定位的效率和精度。同时从3个方面对RADAR系统和本文设计的两个方法进行比较,即搜索空间中的样本数、定位时间以及定位误差。实验结果表明,提出的方法可以以较高的精度更好地估算移动用户的位置。

但是正如前面所讨论的,仍然存在一些需要深入研究和解决的问题,这也是我们未来的工作方向,比如接入点的安装位置与接收信号强度之间的关系(可以考虑使用数据挖掘技术进一步研究),以及如何抽象指纹数据来提高搜索效率和定位精度等等。

参考文献

[1] Enge P, Misra P. Special Issue on GPS: The Global positioning system[C]//Proc. of the IEEE, January 1999, 3-127
 [2] Wang S S, Green M, Malkawa M. E-911 location Standard and

Location Commercial Services[C]//Proceedings of IEEE Emerging Technologies Symposium on Broadband, Wireless Interact Access, April, 2000
 [3] Bahl P, Padmanabhan V. RADAR: An in-building RFbased user location and tracking system[C]//Proceedings of IEEE INFOCOM, March 2000, 2: 775-784
 [4] Simic S N, Sastry S. Distributed Location in Wireless Ad-hoc Networks[EB/OL]. www.ee.iitb.ac.in/student/~sripada/papers/simiknshastry.pdf, 2002-04-10
 [5] Cormen T H, Leiserson C E, Rivest R L. Introduction to Algorithms[M]. The MIT Press, 1990
 [6] Xiang Z, Song S, Chen J, et al. A wireless LAN-based indoor positioning technology[J]. IBM Journal of Research and Development, 2010, 48(5/6): 617-626
 [7] Roos T, Myllymaki P, Tirri H, et al. A Probabilistic Approach to WLAN User Location Estimation[J]. Int. Journal of Wireless Information Networks, 2002, 9(3): 155-164
 [8] Youssef M, Agrawala A K. Handling Samples Correlation in the Horus System[C]//IEEE Info Com, Hong Kong, March 2004
 [9] Robinson M, Psaromiligkos I. Received Signal Strength-based Location Estimation of a Wireless LAN Client[C]//Wireless Communications and Networking Conference. IEEE, March, 2005: 2350-2354
 [10] Kotanen A, Hannikainen M, Leppakoski H, et al. Positioning with IEEE 802.11b wireless lan[C]//14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications (PIMRC 2003), 2003: 2218-2222
 [11] Wang Y, Jia X, Lee H K. An Indoors Wireless Positioning System Based on Wireless Local Area Network Infrastructure[C]//The 6th International Symposium on Satellite Navigation (SatNav 2003), Melbourne, Australia, July 2003
 [12] Wang Xing-fu, Liu Zhi-qiang, et al. Improved Bounding-box Localization Algorithm in WSN[J]. Computer Engineering, 2011, 37(20): 57-59

(上接第83页)

[2] Ge A, Callegati F, Tamil L. On optical burst switching and self-similar traffic[J]. IEEE Communication Letters, 2000, 4(3): 98-100
 [3] Vokkarane V M, Haridoss K, Jue J P. Threshold-based burst assembly policies for QoS support in optical burst-switched networks[C]//SPIE Optical Communication, Boston Mass USA, 2002: 125-136
 [4] Toksoz M A, Akar N. Dynamic threshold-based assembly algorithms for optical burst switching networks subject to burst rate constraints[J]. Springer Photonic Network Communication, 2010, 20(2): 120-130
 [5] Yuan C, Zhang Z, Li Z, et al. A unified study of burst assembly in optical burst switching networks[J]. Springer Photo Netw Commun, 2011, 21(5): 228-237
 [6] 牛大伟, 于卫波, 米志超, 等. 多目标约束下的光突发交换网络组装参数分析[J]. 电子与信息学报, 2013, 35(2): 314-319

[7] 牛大伟, 王海, 于卫波, 等. 一种适用于光突发交换网络的背景流量估计模型[J]. 光学学报, 2012, 32(11): 1-5
 [8] 牛大伟, 彭来献, 于卫波, 等. 一种基于控制平面测量的光突发交换网络动态偏置时间算法[J]. 电子与信息学报, 2012, 34(4): 776-781
 [9] 乐孜纯, 陈君, 付明磊, 等. 一种新型结构光交叉连接节点及其联网性能分析[J]. 光学学报, 2011, 31(3): 1-7
 [10] 胡卫生, 孙卫强, 何浩, 等. 光交换的时间及空间结构分析[J]. 激光与光电子学进展, 2012, 49(1): 1-6
 [11] de Pedro L, Aracil J, Hernandez J A, et al. Analysis of the processing and sojourn times of burst control packets in optical burst switches[C]//Proceedings of International Conference on Optical Networks Design and Modeling, Catalonia Spain, 2008: 1-3
 [12] Yuan Chi, Zhang Zhen-rong, Li Zheng-bin, et al. A unified study of burst assembly in optical burst switching networks[J]. Photon Netw Commun, 2011, 21: 228-237