

一种自负载降速率包列可用带宽测量算法*

张大陆¹⁺, 胡治国^{1,2}, 朱安奇¹, 张俊生¹

¹(同济大学 计算机科学与技术系, 上海 201804)

²(中国人民解放军 65583 部队, 辽宁 辽阳 111000)

Self-Loading Decreasing Rate Packet Train Method for Available Bandwidth Estimation

ZHANG Da-Lu¹⁺, HU Zhi-Guo^{1,2}, ZHU An-Qi¹, ZHANG Jun-Sheng¹

¹(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

²(65583 Troops, The People's Liberation Army, Liaoyang 111000, China)

+ Corresponding author: E-mail: daluz@acm.org

Zhang DL, Hu ZG, Zhu AQ, Zhang JS. Self-Loading decreasing rate packet train method for available bandwidth estimation. *Journal of Software*, 2012, 23(2): 335-351. <http://www.jos.org.cn/1000-9825/3965.htm>

Abstract: Based on the idea of SLoPS, a probing technique, self-loading decreasing rate train (SLDRT), is proposed. SLDRT measures the available bandwidth by using a single decreasing rate packet train. The special train can efficiently avoid bias caused by multiple sampling and makes the tool converge fast. Analysis and simulations verify the effectiveness of the method under both the single-hop and multi-hop path with the bursty cross traffic environment, which also show that SLDRT estimates available bandwidth more, accurately and with less measurement overhead and intrusiveness than other existing techniques such as Pathload, pathChirp.

Key words: network measurement; available bandwidth; multi-hop path; bursty cross traffic; SLDRT (self-loading decreasing rate train)

摘要: 基于自负载周期流技术,提出一种采用降速率包列的可用带宽测量方法 SLDRT(self-loading decreasing rate train),并全面分析了该算法在多跳网络、突发性背景流的环境下的性能.SLDRT 采用单条包列即可实现对可用带宽的高速测量,具备单次采样、准确测量的特性,可通过调整递减因子等参数,提高测量精度,降低测量负载.理论分析和不同背景流场景下的实验结果表明:在多跳、突发性背景流下,SLDRT 具有较强的健壮性;与 pathChirp,Pathload 算法相比,不仅测量精度优良,而且大量缩短了测量时间,减轻了因测量而引入的额外负载.

关键词: 网络测量;可用带宽;多跳路径;突发背景流;SLDRT(self-loading decreasing rate train)

中图法分类号: TP393 文献标识码: A

端到端路径可用带宽是动态描述网络路径传输能力的重要参数,在多媒体传输、协议设计、端到端接入控制、网络故障检测中均有广泛应用.根据是否发送探测包,网络测量技术可分为主动测量和被动测量技术.被动测量不向网络发送探测包,因此不会增加网络负荷;但被动测量只能借助网元设备去采集所需信息,主要用于单点监测,难以进行网络端到端性能分析.而主动测量只需在网络边缘引入探测分组即可获取路径中间信息,灵活

* 基金项目: 国家自然科学基金(61073154)

收稿时间: 2010-07-02; 修改时间: 2010-08-27; 定稿时间: 2010-11-17

性较大.因此,可用带宽测量通常采取主动测量方法.

目前,绝大多数可用带宽主动测量方法可分为探测速率模型(probe rate model,简称 PRM)和探测间隔模型(probe gap model,简称 PGM)^[1-5].PRM 和 PGM 都假设:(1) 先进先出(FIFO)的路由器排队;(2) 网络背景流为流体模型(设背景流发送速率为 x ,在任意时间间隔 t 内,到达某一链路背景流量均为 $x \times t$)^[6];(3) 背景流在单个测量周期中保持恒定^[4,5,7].但是,文献[8]指出:网络中的流量具有自相似性,在各时间尺度下可用带宽均有相当程度的上下浮动,流体模型作为一种简化模型,不能反映实际网络中流量的变化情况.所以,基于流体模型假设的测量方法在突发背景流量下健壮性有限.

本文提出了 SLDRT(self-loading decreasing rate train)可用带宽测量方法.理论分析及实验表明,SLDRT 在单、多跳路径及流体、非流体模型的网络环境下,测量精度和测量速度均获得了较为理想的结果,因测量引入的额外负载也较小.

1 基本概念与相关工作

1.1 可用带宽定义

在流体模型下,可用带宽具有一定的连续性,设 $Capacity$ 为链路容量, $CrossTraffic_{\Delta t}$ 为 Δt 时间内通过该链路的平均背景流速率,当 Δt 足够小时,可视为定值,则链路的可用带宽为

$$avbw = Capacity - CrossTraffic_{\Delta t} \quad (1)$$

在非流体模型下,由于网络流量的突发性, $CrossTraffic_{\Delta t}$ 在任何粒度下均不可视为定值,流体模型下的可用带宽公式不再适用.假设一条链路 L 在时间段 Δt 内的平均可用带宽为 $avbw_{\Delta t}$, $LC_{\Delta t} = Capacity \times \Delta t$ 定义为在时间段 Δt 内能通过链路 L 的总比特数, $CT_{\Delta t}$ 定义为在时间段 Δt 内流过链路 L 的比特数,则突发背景流下可用带宽的定义为:在时间段 Δt 内,在链路不影响背景数据 $CT_{\Delta t}$ 发送的情况下所能提供的速率,如公式(2)所示:

$$avbw_{\Delta t} = \frac{LC_{\Delta t} - CT_{\Delta t}}{\Delta t} \quad (2)$$

1.2 可用带宽测量的研究现状

PGM 是通过分析探测包对通过路径后包对之间的间隔变化推测可用带宽.PGM 通常假设瓶颈链路(容量最小链路)和紧链路(可用带宽最小链路)为同一链路,且瓶颈链路容量已知.但在实际网络中,由于交叉背景流的存在,PGM 模型的假设条件不能总是成立.文献[9]的实验也表明,PGM 模型在多瓶颈链路下的测量结果会出现较大偏差,且会低估可用带宽.其代表算法有 spruce^[5],IGI^[10].

PRM 算法思想又称为自拥塞原则,其中自加载周期流(self-loading periodic streams,简称 SLoPS)方法最具代表性.它通过调整探测流的发送速率使其不断逼近路径可用带宽,根据探测包单向时延的上升时刻对应的探测包速率来推测可用带宽.PRM 虽然会造成路径的短暂拥塞,但不需要事先获知路径的容量,不需要 PGM 算法成立的假设条件,因此应用范围更广,典型方法有 Pathload^[11],pathChirp^[12].PRM 算法在网络紧链路定位上的应用也很广泛,如基于 Pathload 算法的 DRPS^[13]、基于 pathChirp 的 STAB^[14].Pathload 使用二分搜索策略调整探测包列速率迭代计算路径可用带宽,这种策略导致 Pathload 每次测量都发送大量探测包,因此测量速度较慢,而且会占用大量的带宽资源.PathChirp 通过一种称为 Chirp Train 的升速率探测包列进行可用带宽测量,包列的速率随时间增长,只需单条包列就可以扫描较大的带宽范围,测量速度与 Pathload 相比有显著提高,测量负载也随之降低.但 PathChirp 算法的实质是根据包列中相邻包对的速率是否大于可用带宽来进行测量,与包列速率无关.在背景流变化复杂的情况下,对 Chirp 内数据包的可用带宽估计值 E_k 的计算会出现偏差,最终使 pathChirp 在多跳突发背景流下的测量精度下降.文献[4]对基于 PGM 和 PRM 的可用带宽测量方法进行了总结,见表 1.

基于模型的测量方法也是可用带宽测量领域的一个重要方向.其基本思想是将复杂的网络进行简化建模,通过发送少量的探测流收集路径信息后,结合模型估测路径可用带宽,例如文献[19].但简化建模的这种方法难以准确地刻画网络流量全部特征,因而其测量结果与实际情况之间的偏差较大.

Table 1 Comparison among different types of available bandwidth estimation methods^[4]

表 1 可用带宽测量方法比较^[4]

Method	Single-Bottleneck only	Fluid cross traffic only
PGM ^[5,10,15,16]	Yes	Yes
PRM ^[11,12,17,18]	No	Yes

近年来,国内在可用带宽测量方面也有一些研究,比如,文献[1]提出了 LinkPPQ 测量方法,可对每条链路的可用带宽进行测量;文献[2]用随机发送单个小探测包的方法获得链路空闲率,计算路径可用带宽;文献[3]用包距离对称度来衡量所探测网络路径的负载程度,依据负载程度的变化针对性地进行可用带宽测量。

目前,可用带宽测量方法大多基于流体模型假设,在复杂网络环境中的应用范围受限,因此,有必要研究一种适用于多跳、突发背景流环境下的可用带宽测量方法,并具备高速、准确、低测量负载的特点。

2 可用带宽测量算法 SLDRT

2.1 SLDRT算法设计

本文提出一种可用带宽测量算法:自负载降速率探测包列 SLDRT.该算法首先背靠背地发送 d 个负载包 (loading packet),随后以指数形式逐步降低包列的输入速率,当包列的输入速率等于输出速率时测量结束,以此时的输入速率作为测量结果.SLDRT 的包列构造如图 1 所示。

背对背的负载包发送速率大于可用带宽时,在测量过程中,由于排队的原因,报文的相对单向延迟值会逐渐增加.随着包列输入速率的降低,链路拥塞随之不断减轻直至消除,报文的相对单向延迟值逐渐变小直至呈现一种平稳的趋势.在 NS-2 仿真环境下记录了 SLDRT 包列的单向时延变化情况,如图 2 所示。

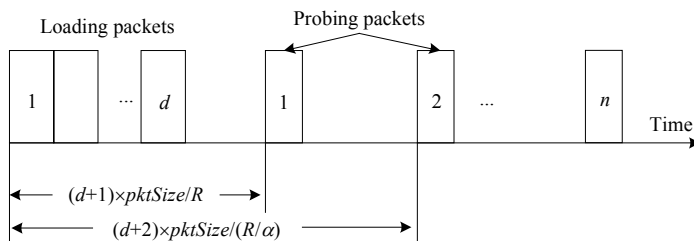


Fig.1 Structure of the probe consisting of SLDRT

图 1 SLDRT 包列构造

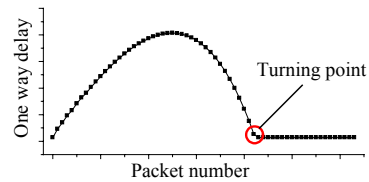


Fig.2 One-Way delay variation of packet train

图 2 包列单向时延变化

在算法具体实现中,有两种方式可作为 SLDRT 算法收敛的判断条件:(1) SLDRT 探测包传输时延 OWD 是否已等于最小传输时延 $MinOWD$ (即在排队时延为 0 的情况下,探测报文传输需要的时间^[2]).在测量时,可先发送离散的数个探测包,假定在所测得的传输时延中,最小值即记为 $MinOWD$;(2) 探测包的 OWD 是否已基本保持不变,以 OWD 等于 $MinOWD$ 或 OWD 基本保持不变的起始点时的 SLDRT 包列速率作为测量结果输出。

另外,SLDRT 算法是否收敛的依据是探测包间延迟抖动变化情况.即,当前探测包时延值与 $MinOWD$ 时延值相比是否相等,或者当前探测包的时延值与前一个探测包的时延值相比是否基本保持不变.以拐点出现时的探测包速率作为可用带宽值输出,可用带宽测量结果与单向时延值的精度或大小无关.因此,以上两种方法均可避免发送端和接收端时钟不同步对测量造成的影响。

SLDRT 探测包列满足以下特性:第 i 个探测包(probing packet)与第 1 个负载包(loading packet)之间的发送间隔为 $(i+d) \times pktSize / R_i$,其中, R_i 为第 i 个探测包发送时的包列平均发送速率,紧接其后的探测包 $i+1$ 发送时的包列速率为 R_i / α (α 为递减因子, $\alpha > 1$).探测包初始发送速率 R_{init} 可设为网络路径最大容量。

在复杂多跳的网络环境中,路径最大容量与可用带宽值会有较大差距.在这种情况下,对网络路径进行持续

测量时,可利用前次可用带宽测量值($avbw_{last}$)结果设置 SLDRT 探测包的初始速率 R_{init} .为此,本文引入了初始速率递增因子 $\gamma(\gamma>1)$.即,连续测量时,SLDRT 探测包初始速率设为上一次可用带宽测量值的 γ 倍.考虑到 γ 取值过大则会失去存在的意义,过小则可能导致探测包初始速率小于可用带宽从而使算法不能收敛,所以,本文采用循环增大的方式设置 SLDRT 探测包的初始速率,具体实现如图 3 所示.

```

1   $R_{init} = avbw_{last} \times \gamma;$ 
2  if ( $OWD_{init} > MinOWD$ ) /*  $OWD_{init}$  is the one way delay of initial probing packet */
3     $flag = PROBE;$  /* PROBE denotes estimation is under way */
4  else
5    return  $R_{init} = R_{init} \times \gamma;$ 

```

Fig.3 Initial rate setting algorithm of probing packet

图 3 探测包的初始速率设置算法

γ 的引入,使 SLDRT 算法在可用带宽与路径容量偏离较大的情况下,避免了测量包的初始速率总是从链路最大容量下降的缺点,加快了测量收敛时间,减少了因测量而引入的额外负载.在可用带宽与路径容量相差不大的情况下,则无需引入初始速率递增因子 γ .

SLDRT 采用双终端方式实现,其程序分为两部分: $sldrt_snd$ 和 $sldrt_rcv$.其中, $sldrt_snd$ 安装在发送端, $sldrt_rcv$ 安装在接收端.发送端负责发送探测包列,接收端则负责判断算法收敛情况.如果已满足收敛条件,则接收端通知发送端结束此次测量,将此时探测包列平均速率作为测量结果输出.发送端主要算法如图 4 所示.

```

/* SLDRT algorithm */
1 send()
2 send d loading packets;
3 if ( $flag == RESTART$ ) /* RESTART:开始新一次测量 */
4    $R_{init} = avbw_{last} \times \gamma;$  /*  $R_{init}$ :探测包初始速率;  $avbw_{last}$ :上一次可用带宽测量结果 */
5    $flag = PROBE;$  /* PROBE:测量开始标志位 */
6   restart();
7 end
8 else if ( $flag == PROBE$ ) /* 调整探测包初始速率 */
9    $R_i = R_{init};$  /* 设置探测包初始速率 */
10  while ( $OWD > MINOWN$ ) /* 在接收端进行判断 */
11     $T_i = (pktsize \times (i + d)) / R_i;$ 
12     $R_{i+1} = R_i / \alpha$ 
13     $T_{i+1} = (pktsize \times (i + 1 + d)) / R_{i+1};$ 
14    sendpkt( $T_{i+1} - T_i$ ); /* 过  $T_{i+1} - T_i$ 时间发送下一个探测包 */
15     $R_i = R_{i+1};$ 
16     $i ++;$ 
17  end
18 end
19 end

```

Fig.4 SLDRT algorithm

图 4 SLDRT 算法

2.2 SLDRT算法性能分析

2.2.1 前提假设

文献[1,4,5,7]总结了已有 PGM 和 PRM 模型成立的前提假设:(1) 先进先出(FIFO)的路由器排队;(2) 网络背景流为流体模型;(3) 背景流在单个测量周期中基本保持恒定.本文提出的算法无须假设网络背景流为流体模型,但假设条件(1)和条件(3)仍是本文分析在流体和非流体模型下算法性能的前提假设.

2.2.2 基于流体模型分析

1. 单跳网络路径分析

根据流体背景流模型,网络流量被建模为无限可分的比特流.在网络链路上注入输入速率为 R_{in} 的探针包列,记输出速率为 R_{out} ,有:

引理 1.

(1) $R_{in}=R_{out} \Leftrightarrow R_{in} \leq avbw$;

(2) $R_{in} > R_{out} \Leftrightarrow R_{in} > avbw$.

当输入速率小于可用带宽时,所有探测包都将不受拥塞地传输出去,因而输出速率等于输入速率.假设输入速率大于可用带宽,则探测包列将由于排队延迟而被延迟.SLDRT 包列首先发送 d 个背靠背的负载包,然后以逐渐增长的间隔发送测量包,整条包列的速率从一个相对较高的值开始逐渐递减.当 SLDRT 包列到达接收端时,接收端计算出包列的输出速率,并与输入速率进行比较,直到两者相等.此时, $R_{in}=R_{out} \Leftrightarrow R_{in} \leq avbw$.而上一个包发送时由于 $R'_m > R'_{out} \Leftrightarrow R'_m > avbw$,且上一个包发送时包列的整体速率为 $R'_m = R_{in} \times \alpha$,因此将当前包列的速率 R_{in} 作为可用带宽的测量值返回.此时,可用带宽的真实值落在 $[R_{in}, R_{in} + R_{in} \times (\alpha - 1)]$ 内,则测量的绝对误差范围为 $[0, R_{in} \times (\alpha - 1)]$,相对误差小于 $\alpha - 1$.

2. 多跳网络路径分析

将测量扩展到多跳场景.设 L 为路径 P 的紧链路,显然, $L \in P$ 且 $avbw_p = avbw_L, \forall L' \neq L$ 且 $L' \in P, avbw_{L'} \geq avbw_L$.这条路径可以建模为 3 个部分:紧链路前的子路径、紧链路后的子路径与紧链路本身.如图 5 所示.

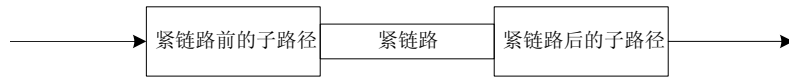


Fig.5 Three-Hop network model

图 5 三段网络模型

显然可以得到:

$$avbw_{pre} = \min\{avbw_l | \forall l \in P_{pre}\} \geq avbw_L.$$

$avbw_{pre}$ 是紧链路前的子路径可用带宽,而 P_{pre} 表示该子路径所包含的所有链路集合.类似地,可以得到:

$$avbw_{post} = \min\{avbw_l | \forall l \in P_{post}\} \geq avbw_L.$$

当注入的探针包速率为 R_{in} 时:

- (1) 当 $R_{in} > avbw_{pre}$ 时,探测包列将在紧链路前的子路径被延迟;
- (2) 当 $avbw_{pre} > R_{in} > avbw_L$ 时,包列将在紧链路被延迟;
- (3) 当 $R_{in} \leq avbw_L$ 时,包列将不会被延迟.

引理 1 将被扩展到多跳的场景:

引理 2.

(1) $R_{in}=R_{out} \Leftrightarrow R_{in} \leq avbw_L$;

(2) $R_{in} > R_{out} \Leftrightarrow R_{in} > avbw_L$.

引理 1 和引理 2 保证了 SLDRT 在基于流体背景流的单跳和多跳网络路径上的测量准确性.

2.2.3 基于非流体背景流分析

1. 单跳网络路径分析

文献[8]指出:网络中的流量具有突发性,流体模型作为一种简化模型,不能反映实际网络中可用带宽的变化情况.突发背景流下的可用带宽测量同样首先假设测量路径为单跳路径,则可用带宽的测量被简化为链路可用带宽的测量.路径中的节点均做 FIFO(先进先出)假设. $CT_{\Delta t}$ 表示 $(t, t + \Delta t)$ 时间段内通过链路的背景流比特数, $LC_{\Delta t}$ 为 Δt 时间段内能通过的链路的最大比特数.

在突发流量模型下,链路在 $(t, t + \Delta t)$ 时间段内的平均可用带宽满足以下公式:

$$avbw_{\Delta t} = \frac{LC_{\Delta t} - CT_{\Delta t}}{\Delta t} \quad (3)$$

由此推出引理 3.

引理 3.

$$AB_{\Delta t} = avbw_{\Delta t} \times \Delta t = LC_{\Delta t} - CT_{\Delta t} \quad (4)$$

其中, $AB_{\Delta t}$ 为时间段 Δt 内, 在不影响背景流的情况下链路所能处理的总比特数, 即必须满足 Δt 内所有到达链路的 $CT_{\Delta t}$ 必须发送完毕. 如果能求出 $LC_{\Delta t}$ 和 $CT_{\Delta t}$, 则可以通过公式(3)计算可用带宽. 但是在实际测量中, 尤其是多跳网络路径中, $LC_{\Delta t}$ 不容易获得, 同样地, $CT_{\Delta t}$ 也很难观测到. 但反过来, 如果能求得 $AB_{\Delta t}$, 那么同样也可以很容易地计算出可用带宽.

如果在测量时间段 Δt 内向链路中注入 $AB_{\Delta t}$ 比特的探针包, 并同时满足以下 3 个限制条件, 则 $PB_{\Delta t}$ 是所要求的 $AB_{\Delta t}$, 并可以依此计算出时间段 Δt 内的平均可用带宽为 $avbw_{\Delta t}$:

- (1) 在 Δt 内保持链路为忙状态(不间断地发送数据);
- (2) $CT_{\Delta t}$ 要在 Δt 内全部处理完毕(背景流不受影响);
- (3) 背景流在单个测量时间段 Δt 内基本保持恒定.

为了满足条件(1), 首先往链路里注入背靠背的负载包, 由于流量的到达速率大于链路处理能力, 因此必然发生拥塞, 在拥塞期间链路必处于忙状态, 也即满足了条件(1). 同时注意到, 当链路队列被清空时, 链路已经在测量结束时处理完了所有的分组, 包括 $CT_{\Delta t}$ 和 $PB_{\Delta t}$, 这满足条件(2). 在背靠背的负载包后紧跟降速率的测量包, 当观测到单向时延为 MinOWD 的探针包时, 表明链路队列已没有排队的探测包. 以这个测量包作为包列的结尾, 包列的平均速率 $R_{in} = PB_{\Delta t} / \Delta t = AB_{\Delta t} / \Delta t$ 即为可用带宽的测量值.

2. 多跳网络路径分析

当测量的路径扩展到多跳时, 以上推论仍然满足. 假设链路 L 为路径 P 的紧链路, 则 $L \in P, avbw_p = avbw_L$. 且 $\forall L' \neq L, L' \in P$, 必有 $avbw_{L'} > avbw_L$. 当探针包列速率大于可用带宽时, 包列必然在紧链路 L 延展. 当探针包列速率小于可用带宽时, 包列不会在紧链路 L 延展; 又由 $avbw_{L'} > avbw_L$, 包列也不会再在路径中其他链路上被延展. 引理 1 仍满足. 这说明, 在非流体模型下, 无论是单跳还是多跳, 降速率探针包列均能准确地测量出路径可用带宽值.

2.3 SLDRT算法健壮性分析

2.3.1 测量过程中背景流流量大小发生变化的情况

SLDRT 算法探测包的相对单向时延值的总体变化趋势是先上升后下降至恒定, 在此过程中, 若受到突发背景流的影响, 会造成上升或下降的趋势发生变化. 若背景流流量增加, 可用带宽值就会变小, 则探测流排队加剧, 单向时延下降变缓, 那么 SLDRT 探测包单向时延开始保持恒定的包的序列号增大, 所测得的可用带宽值会相应减小; 反之, 若背景流流量减小, 可用带宽值就会增大, 则探测流排队减缓, 探测包的单向时延下降加快, 则单向时延下降至开始保持恒定的探测包的序列号就会相应减小, 测得的可用带宽值则增大. 这就保证了 SLDRT 在突发背景流下进行可用带宽测量具有较高的健壮性, 得到较高精度的测量值.

2.3.2 测量过程中背景流流量大小不变的情况

定理 1. 在测量过程中, 只要经过链路的背景流流量总量一定, 那么无论此间背景流如何变化, SLDRT 算法的测量结果都保持不变.

证明: 不失一般性, 设 SLDRT 在 t_0 时刻开始对某一链路进行测量, 测量结果为 $avbw$, 测量结束时间为 t_1 , 由可用带宽定义可知下式成立: $R_c + avbw = C; R_c = C - avbw = \frac{M}{t_1 - t_0}$. 其中, R_c 为背景流的平均速率, M 为在背景流的总流量大小, C 为链路容量, $avbw$ 为可用带宽测量值. 记 SLDRT 算法收敛时探测包的序号为 n .

反证, 前提条件: 测量过程中背景流流量大小 M 不变.

假设(1): 通过改变链路中背景流的波动特征, 测得可用带宽值 $avbw' > avbw$. 当 $avbw' > avbw$ 时, 由 SLDRT 算法可知, 算法收敛时的探测包的序号必小于 n , 则此时的测量时间 $t_1' - t_0 > t_1 - t_0$.

因测量时间段内背景流 $R'_c = \frac{M}{t'_1 - t_0} > R_c$, 故 $R'_c + avbw' > R_c + avbw = C$ 与 $R'_c + avbw' = C$ 矛盾, 假设(1)不成立.

假设(2): 改变链路中背景流的波动特征, 测得可用带宽值 $avbw'' < avbw$, 由测量方案可知, 此时 SLDRT 算法收敛时的探测包序号必大于 n , 即测量时间 $t''_1 - t_0 < t_1 - t_0$.

此次测量的平均背景流 $R''_c = \frac{M}{t''_1 - t_0} < R_c$, 故 $R''_c + avbw'' < R_c + avbw = C$ 与 $R''_c + avbw'' = C$ 矛盾, 假设(2)不成立.

因此, 定理得证. 即, 当测量阶段经过链路背景流流量大小不变的情况下, 背景流特性的变化不能使测量结果产生变大或变小的情况. \square

由以上证明过程容易得出以下推论:

推论 1. SLDRT 测量结果只与背景流的大小相关. 即测量过程中, 无论背景流速率是突发还是恒定, 只要总量不变, SLDRT 算法测量结果就没有变化.

3 算法参数选择

SLDRT 算法中, 测量包的组成特性由初始速率因子 γ 、递减因子 α 决定. γ 的存在只是为了利用上次可用带宽测量的结果, 减小因网络测量而引入的额外负载, 特别是在带宽利用率较低的情况下, 避免 SLDRT 算法总是以最大速率开始递减. 因此, SLDRT 算法中最主要的参数是测量包列的递减因子 α .

3.1 性能评价标准

为了评价 SLDRT 算法性能, 定义算法性能的评价指标如下:

定义 1(测量误差 ME(measurement error)). 测量得到的可用带宽为 MV(measured value)的测量值与路径实际可用带宽 RV(real value)之间的差值.

$$ME = \frac{1}{n} \sum_{i=1}^n \frac{|MV - RV|}{RV} \times 100\% \quad (5)$$

定义 2(测量时间 MT(measurement time)). 从算法发起测量到得出待测路径可用带宽值所需的时间, 单位为 s. 测量持续的时间越长, 反映可用带宽动态变化情况的能力就越差, 测量负载对网络应用造成的影响也就越大^[20].

定义 3(测量负载 MO(measurement overhead)). 完成一次测量所需发送探测包的总字节数, 单位设为 MB. 测量负载越小, 意味着测量工具对网络运行的影响越小, 该测量工具也就越适合大规模的采用^[20].

3.2 算法参数选择分析

3.2.1 递减因子与测量误差

设 R_1, R_2, \dots, R_i 为 SLDRT 算法在测量过程中发送探测包 1, 2, ..., i 时的包列速率, 出现单向时延为 MinOWD 的阈值点 R_t 应在 R_i 与 R_{i-1} 之间, 即 $R_{i-1} > R_t \geq R_i$, 取 $R_t = (R_i + R_{i-1})/2$ 作为可用带宽真实值.

定义 4. 测量相对误差率 ρ

$$\rho = \frac{R_{i-1} - R_i}{(R_i + R_{i-1})/2} = 2 \left(\frac{\alpha - 1}{\alpha + 1} \right) = 2 \left(1 - \frac{2}{\alpha + 1} \right) \quad (6)$$

显然, ρ 是一个递增数列.

由以上分析可以得出: 误差率 ρ 随递减因子 α 的增加而增大; 反之, 递减因子越小, 测量误差就越小, 但此时包列速率的下降幅度会变小, 测量时间随之变长.

3.2.2 测量时间与测量负载

设负载包个数为 d , 负载包及探测包大小均为 S , 探测包列初始速率为 R_{init} , 测量结束时发送探测包为 n , 测量花费时为 t_c , 递减因子为 α , 可用带宽值为 $avbw$, 根据第 2.1 节的算法思想, 公式(7)关系成立.

$$\frac{(d+n) \times S}{t_c} = \frac{R_{init}}{\alpha^{n-1}} = avbw \tag{7}$$

故测量时间 t_c 可表示为公式(8):

$$t_c = (d+n) \times S \times \alpha^{n-1} / R_{init} \tag{8}$$

因为 d, S, R_{init}, α 在测量中为定值, 所以 $t_c = f(n)$; 又因为 $f'(n) > 0$, 所以 $f''(n) > 0, f(n)$ 为递增凹函数, 如图 6 所示.

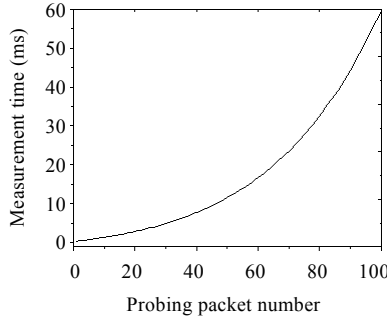


Fig.6 MT vs. number of probing packet
图 6 测量时间与探测包数量的关系

测量负载可表示为公式(9):

$$(d+n) \times S = t_c \times avbw \tag{9}$$

分析得出: 递减因子越小, 误差越小; 但测量时间增加, 测量负载变大; 在递减因子确定的情况下, 探测包初始速率 R_{init} 越接近实际可用带宽值, 测量速度越快, 测量负载也就越小.

3.3 算法参数具体取值

设置 NS-2^[21] 仿真拓扑环境: Path Persistent 背景流量(如图 7 所示)选用路径特征 $P = \{C_1, C_2, C_3, C_4\} = \{100, 100, 100, 100\}$. 另外, 本文中所有仿真链路时延均设为 10ms. 实验中的背景流由 CBR(恒定比特率)流与 VBR(变比特率)数据流复合而成, 通过调整 CBR 流量的开始/结束时间, 控制可用带宽总体水平, CBR 设定为 15Mb. VBR 背景流则由 10 条符合 Pareto 分布的流汇集而成, 以模拟突发性的背景流. Pareto 流的包大小设置与文献[2,11]相同: 包大小为 40/550/1500B 的数据包在负载中所占的比率分 50%, 40%, 10%, 包发送速率为 4Mb, Pareto 分布形状参数 $\alpha = 1.9$. 这是因为多个服从参数 $\alpha < 2$ 的 Pareto 分布的流聚合后表现出自相特征^[22].

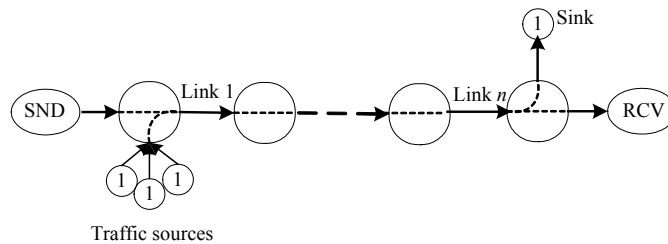


Fig.7 Path persistent cross traffic
图 7 Path persistent 背景流

为了验证第 3.2 节的论证, 直观地反映不同的 α 与 γ 取值对算法性能的影响, 按照图 7 拓扑设置背景流量, 递减因子 $\alpha = \{1.01, 1.015, 1.02, 1.025, 1.03, 1.035, 1.04\}$, $\gamma = \{1, 1.5, 2, 2.5, 3, 3.5, 4\}$. 测量时间、测量负载及测量精度取所有测量结果的平均值, 实验结果如图 8(a)~图 8(c)所示. 可以看出, α 越大, 算法测量时间就越短, 测量负载也就越小, 但测量误差会增大; γ 越大, 算法时间就越长, 测量负载随之增大; 当 $\gamma > 1$ 时, 测量误差变化幅度不大, 这与前面的分

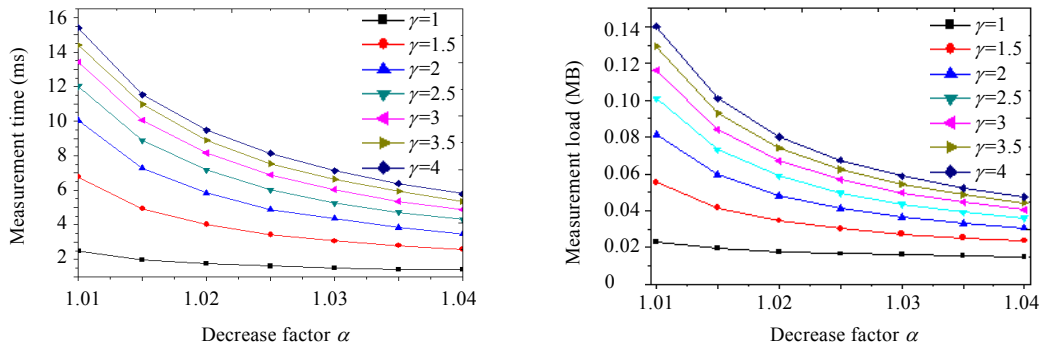
析相一致。

综合考虑测量负载、测量精度、测量时间的影响,本文在实验环境中选择 $\alpha=1.015, \gamma=2.5$ 。

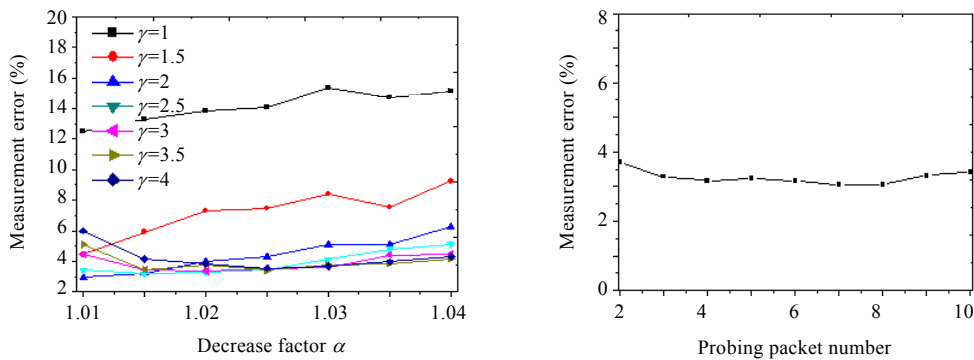
文献[23]研究表明,探测包越大,越容易避免排队时延噪声的影响,对接收端时戳粒度要求也越低,故 SLDRT 算法中设探测包大小 $S=1000B$ 。

初始速率 R_{init} 的设置对算法收敛速度有着直接的影响, R_{init} 越接近可用带宽值,算法收敛越快。但在实际测量中,却不能因此去假设路径中各链路容量或瓶颈链路容量已知,而将 R_{init} 直接设为瓶颈链路或紧链路的容量。因为路径最大容量是一个易于获取且常识性的数值,故将 SLDRT 包列初始速率 R_{init} 设为路径最大容量。本文的仿真及真实网络实验均为百兆网络,故设 $R_{init}=100Mbps$ 。第 4.3 节的实验结果表明,即使包列速度均从路径最大容量开始下降,且可用带宽与 R_{init} 相差较大,算法收敛速度仍然保持较理想状态。

负载包的设置是 SLDRT 算法的一个重要特点,其主要作用是,在出现可用带宽值与 R_{init} 接近这种极端情况下,探测包的 OWD 依然能呈现如图 2 所示的变化,保证了 SLDRT 在此种情况下的算法健壮性。SLDRT 算法的测量误差主要由其递减因子 α 决定(证明见第 2.2.2 节、第 3.2.1 节),而与负载包的数量无关。在图 7 的实验条件下,取 $d=\{2,3,4,5,6,7,8,9,10\}$,分析了负载包数量与测量误差 ME 之间关系,如图 8(d)所示。考虑到在真实网络中,主机的数据包处理能力不可能像在仿真条件下那样理想,适当地设置负载包的个数,可使 SLDRT 包列时延变化的趋势更为明显,有利于对算法是否收敛的判断。本文在实验中设负载包的个数为 5。



(a) MT(测量时间)与 α (递减因子)和 γ (初始速率因子)的关系 (b) MO(测量负载)与 α (递减因子)和 γ (初始速率因子)的关系



(c) ME(测量误差)与 α (递减因子)和 γ (初始速率因子)的关系 (d) ME(测量误差)与 d (负载包个数)的关系

Fig.8 SLDRT performance vs. decrease factor α , initial rate factor γ and loading packet number d

图 8 SLDRT 算法性能与递减因子 α 、初始速率因子 γ 及负载包数量 d 的关系

4 算法性能对比分析与验证

文献[24]对 spruce^[5], abing^[15], Pathload^[11]和 pathChirp^[12]算法进行了对比实验,并指出,Pathload 至少在文献

[24]所设定的实验环境中取得了最准确的测量结果.文献[25]通过实验也证明,总体而言,pathChirp 各种消耗最少,如需对网络链路进行持续测量,则优先选用 pathChirp 工具.因此,本文选择 Pathload 和 pathChirp 这两种算法作为对比对象.因为 Pathload 算法测量结果是可用带宽的范围区间,且测量时间长,采样次数有限,本文在给出的测量结果对比图中以选取代表性的点为主.

4.1 仿真实验

4.1.1 稳定背景流情况下的可用带宽

为了验证各算法对稳定背景流下的可用带宽测量性能以及分析算法测量的稳定性,实验采用 Path Persistent 背景流量,路径特征为 $P=\{C_1, C_2, C_3\}=\{100, 100, 100\}$. 不同时刻注入两条 CBR 背景流(目的是控制可用带宽在链路容量 30%~70%之间变化),大小均为 3.5Mb/s, CBR1 运行时间为 0~300s, CBR2 运行时间为 150s~300s. 实验中发现, pathChirp 算法为每个 chirp 设定不同的指数增长区间(*lowrate, highrate*), 测量结果会有很大差异. 为了取得 pathChirp 算法的最佳效果, 在多种不同的设定下取最优值, 本节实验中取 *lowrate*=2Mbps, *highrate*=10Mbps. 在稳定背景流下, 算法比较实验结果如图 9 所示.

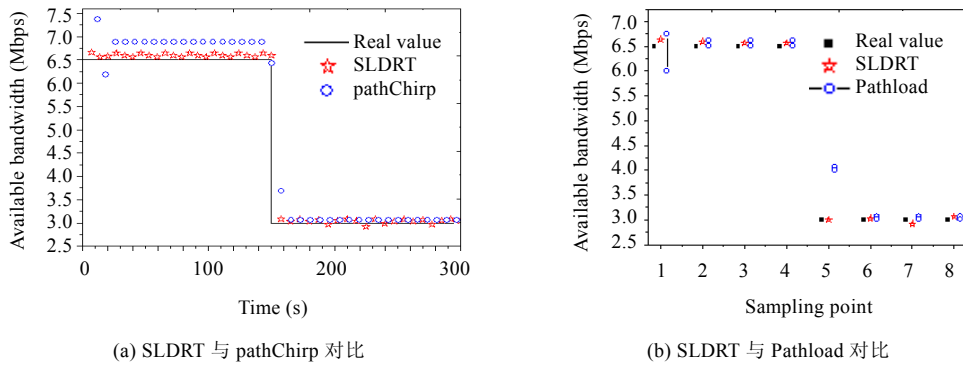


Fig.9 Contrast between SLDRT, pathChirp and Pathload in the condition of fluid cross traffic

图 9 SLDRT 与 pathChirp 和 Pathload 在稳定背景流情况下的对比

在稳定背景流情况下, 3 种算法的测量结果均可准确测得可用带宽值, 但 pathChirp 和 Pathload 都需要对多次测量结果求均值才可获得可用带宽值(pathChirp 的 Chirp 数在原作者提供的 NS-2 源码中默认为 $7^{[26]}$, Pathload 算法 fleet 中 stream 数量为 12)^[27]. 因 pathChirp 和 Pathload 测量周期长, 在流量大小发生突变时, 测量结果就会出现大的偏差(图 9(a)150s 处, 图 9(b)采样点 5 处); 而 SLDRT 算法则没有出现这种情况. 在测量误差统计时, 若包含这些异常点, 则 pathChirp 的 ME 为 7.36%, Pathload 的 ME 为 2.75%; 若排除这些点, 则 pathChirp 算法的 ME 为 5.77%, Pathload 算法的 ME 为 1.31%. 表 2 是算法性能的比较结果.

Table 2 Result of simulation validation with fluid cross traffic

表 2 在稳定背景流情况下验证实验结果

Methodology	MT (s)	MO (MB)	ME (%)
SLDRT	0.135	0.074	1.86
pathChirp	1.538	0.193	7.36
Pathload	12.76	2.114	2.75

SLDRT 的测量误差要略大于第 2.2.2 节对算法测量误差的理论分析, 这是因为 SLDRT 的测量时间在毫秒级, 甚至可达几毫秒, 此时, NS-2 仿真环境中流量真实值与设定值之间会产生波动, 根据设定值去计算, 测量误差则会偏大.

4.1.2 突发背景流下的可用带宽

1. 单瓶颈链路路径

设置路径 $P=\{C_1,C_2,C_3,C_4\}=\{100,100,10,100\}$, Pareto 流量以 One-Hop Persistent 方式发送(如图 10 所示), 整个链路注入两条 CBR 流量, 大小为 2Mb/s, CBR1 运行时间为 20~50s, CBR2 运行时间为 80s~100s; 将 Pareto 流量的速率改为 0.4Mbps, 其余参数与第 3.3 节相同. 实验部分结果如图 11 所示.

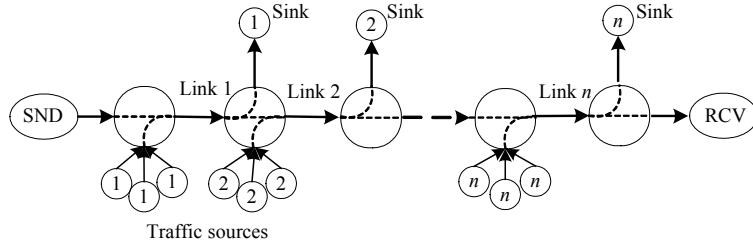


Fig.10 One-Hop persistent cross traffic

图 10 One-Hop persistent 背景流

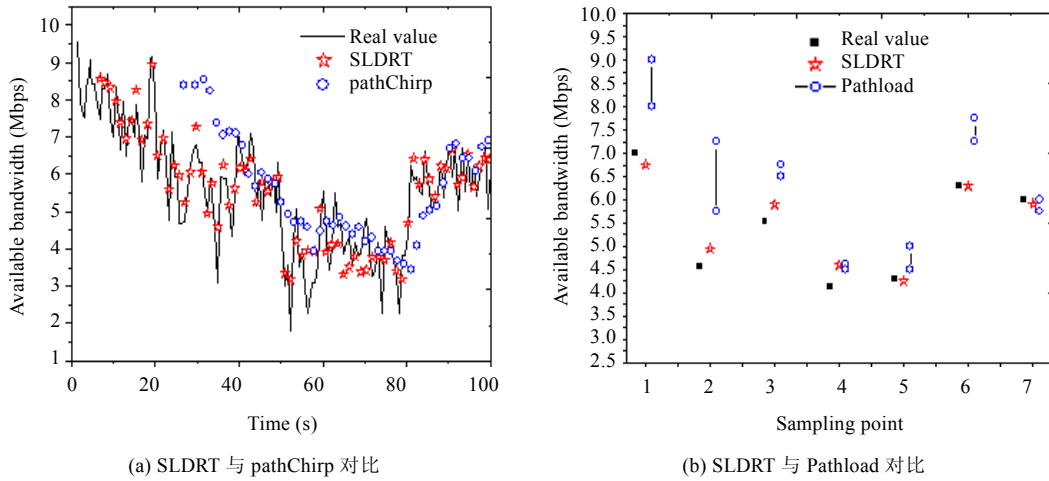


Fig.11 Contrast between SLDRT, pathChirp and Pathload under single-bottleneck path

图 11 单瓶颈链路路径中,SLDRT 与 pathChirp 和 Pathload 的对比

在突发背景流情况下, pathChirp ($lowrate=0.1Mbps, highrate=10Mbps$) 和 Pathload 均出现较大的测量偏差, Pathload 算法在 100s 的仿真时间内仅完成了 7 次测量. 表 3 是算法性能的比较结果.

Table 3 Result of simulation validation under single-bottleneck path

表 3 在单瓶颈链路路径中验证实验结果

Methodology	MT (s)	MO (MB)	ME (%)
SLDRT	0.104	0.074	12.61
pathChirp	1.601	0.193	20.35
Pathload	13.76	2.335	24.37

2. 多跳复杂路径

参考文献[2]中的路径模型, 本文设计网络拓扑环境: $P=\{C_1,C_2,C_3,C_4,C_5,C_6\}=\{100,100,10,100,10,100\}$, Pareto 流量如第 3.3 节(如图 12 所示), 其中也加入两个 CBR 流量, CBR1 贯穿 Link 3 和 CBR2 贯穿 Link 5, 流量均为

2M;CBR1 运行时间为 20s~80s,CBR2 运行时间为 40s~60s(目的是使紧链路发生切换).实验部分结果如图 13 所示.

Pathload 算法在 100s 的仿真时间内仅完成 4 次测量.表 4 是算法性能的比较结果.

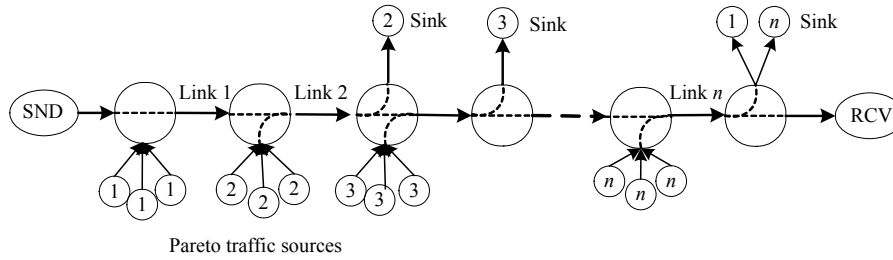


Fig.12 Multi-Hop test topology used in the NS2 simulations

图 12 NS2 中仿真多跳拓扑

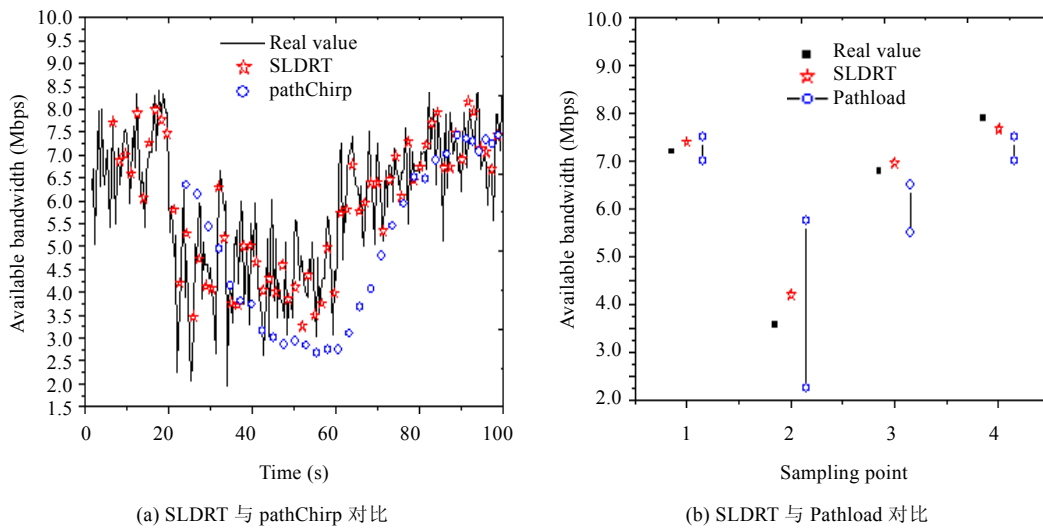


Fig.13 Contrast between SLDRT, pathChirp and Pathload under multi-bottleneck path
图 13 多瓶颈链路路径中,SLDRT 与 pathChirp 和 Pathload 的对比

Table 4 Result of simulation validation under multi-bottleneck path

表 4 在多瓶颈链路路径中验证实验结果

Methodology	MT (s)	MO (MB)	ME (%)
SLDRT	0.099	0.073	11.15
pathChirp	2.406	0.227	22.02
Pathload	22.853	2.128	10.13

上述实验结果表明,SLDRT 算法在网络负载大致相同的条件下,拓扑和流量的变化对 SLDRT 的测量精度、测量时间和测量负载的影响不大.而随着网络路径跳数的增加、背景流突发性的增强,pathChirp 的测量时间、测量负载、测量误差都呈上升趋势;而 Pathload 算法的测量时间有所增加,但测量负载及测量误差的统计结果已无规律可循.这是因为在不同的突发背景流下,Pathload 算法的收敛条件及灰色区域大小的判定会有较大差异,且两组实验中(持续时间 100s),Pathload 仅可测得 7 组和 4 组数据,单个异常点的测量结果对整个测量数据的统计影响较大.

实验结果表明,在 3 种不同网络环境中,SLDRT 的测量精度均优于 pathChirp,Pathload,并缩短了测量时间,减少了测量开销,对比分析如图 14 所示.

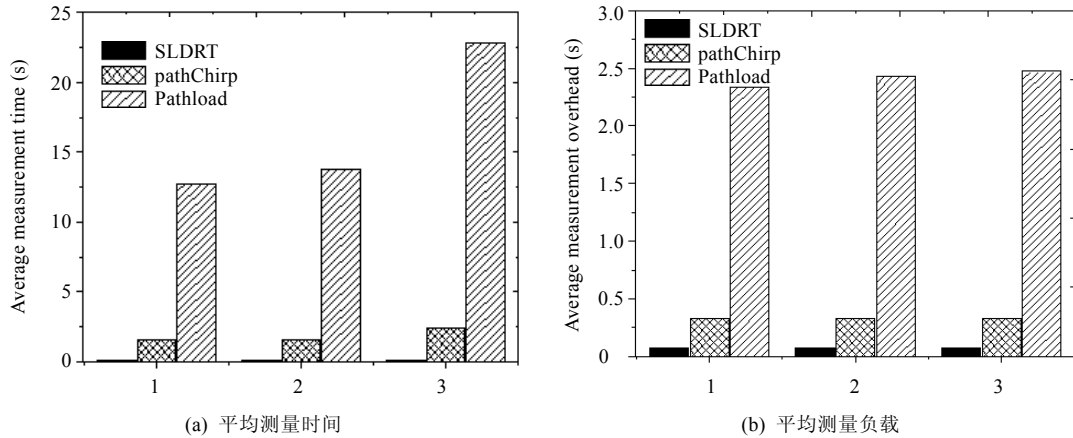


Fig.14 Comparison of average measurement time, measurement overhead with SLDRT, pathChirp, Pathload for three kinds of network model

图 14 3 种不同网络环境下,SLDRT,pathChirp,Pathload 算法的平均测量时间和测量负载比较

4.2 实验网验证

搭建一个实验网,拓扑结构类似于图 10,各中间节点都是锐捷路由器,和文献[1,7,24]一样,实验网的背景流为来自 WIDE 主干网的一条连接日本和美国的网络路径上实际采集的流^[28],流重放工具为 Tcpreplay^[29],同时运行 Net-snmp^[30],每隔 1s 访问实验网上路由器的 MIB 库,获得流过相应链路的流量强度,进而推算实际的可用带宽值.实验中,受主机性能波动的影响,不能确保每次实验的背景流量特性完全一致,而同时运行 3 种算法可能会产生相互干扰,因此在实验网中所得实验结果之间的可对比性不如在第 4.1 节所得数据.

实验中,pathchirp 算法指数增长区间为($lowrate=1Mbps,highrate=10Mbps$).为了检验 SLDRT 在 R_{init} 与可用带宽值相差较大情况下的算法性能,本节验证实验每次测量发起时,SLDRT 包列速率都从 $R_{init}=100Mbps$ 开始下降.实验部分结果如图 15 所示.表 5 各算法性能的比较结果.

实验结果表明,SLDRT 用远低于 Pathload 和 pathChirp 的测量时间,获得了较高的测量精度.

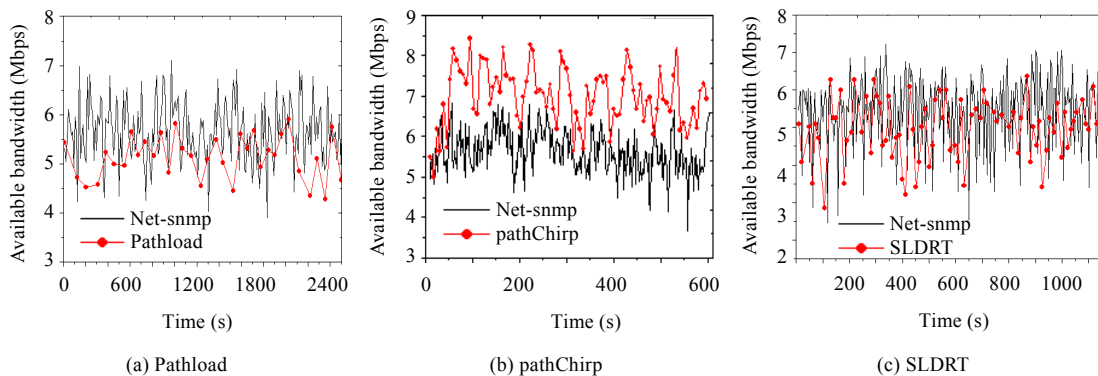


Fig.15 Contrast between SLDRT, pathChirp and Pathload under experimental network

图 15 实验网络条件下,SLDRT 与 pathChirp 和 Pathload 的对比

Table 5 Result of simulation validation under experimental network**表 5** 在实验网络条件下验证实验结果

Methodology	MT (s)	MO (MB)	ME (%)
SLDRT	0.344	0.211	11.96
pathChirp	7.088	0.253	27.26
Pathload	65.23	1.92	11.29

4.3 算法入侵度分析

PRM 算法虽然不需要瓶颈链路和紧链路为同一链路的假设条件,但都不可避免地会引起网络拥塞,进而对背景流量产生负面影响.文献[25]分析了多种可用带宽测量算法性能后指出,算法的收敛速度越快,对背景流的影响就越小.从表 2~表 5 的实验结果可知,SLDRT 在测量时间上与其他算法相比已相对较短.我们认为,PRM 算法对背景流的负面影响主要发生在测量包列速率高于实际可用带宽值,进而造成网络拥塞的阶段.那么,在此阶段测量包列造成的持续拥塞时间越短、注入探测包的数量越少,该算法对背景流的影响就越小.

在类似于图 8 的 NS-2 仿真环境下,通过分析 trace 文件,本文对比了 SLDRT,Pathload 算法在不同可用带宽条件下,单次测量对网络产生的持续拥塞时间.统计过程中:以 SLDRT($R_{mir}=100\text{Mbps}$)的测量时间作为其拥塞时间;将 Pathload 算法中 fleet 速率超过可用带宽值,且其中 12 个 streams 的单向时延值都递增趋势时所占的时间记为该算法的拥塞时间.实验结果见表 6.

Table 6 Comparison of congestion time between SLDRT and Pathload**表 6** SLDRT 和 Pathload 算法拥塞时间对比

Avbw (Mbps)	Congestion time (ms)		
	SLDRT ($R_{mir}=100\text{Mbps}$)		Pathload
	Decrease factor $\alpha=1.1$	Decrease factor $\alpha=1.015$	
5	64.18	213.35	897.6
10	26.87	99.82	1 153.2
20	9.30	35.71	960
30	5.25	17.74	720
40	3.32	10.75	1 080
50	2.40	6.56	960
60	1.70	4.28	840
70	1.17	2.74	720
80	0.96	1.83	840
90	0.77	1.09	360

从实验结果可以看出,在可用带宽值越接近 R_{mir} ,SLDRT 算法造成的拥塞时间越短;且在低可用带宽情况下,也不会造成长时间的网络拥塞.在算法的具体应用中,可针对背景的流特性,通过改变递减因子来调整算法入侵度.如背景流是实时流媒体,则加大递减因子,进一步加快算法收敛,减小拥塞时间.

此外,实验中也证明了文献[25]的结论:pathChirp 对网络造成的拥塞最小.PathChirp 算法中,每一个 chirp 造成的网络拥塞时间仅为 1ms~2ms.

4.4 算法设计与性能对比分析的几点说明

SLDRT 是一种采用指数流技术的可用带宽测量算法,这是因为指数流技术抛弃了过去可用带宽测量中以恒定比特率(CBR)发送探测包的方式,而是以指数的方式改变一条包列内的发送速率,以达到单条包列即可测量出可用带宽的效果,具有高速、入侵度低的特点.采用指数流技术的代表工具为 pathChirp.它首先需要定义偏离段:将报文 i 的队列延迟时间记为 q_i ,当 $q_i > q_{i+1}$ 时,报文 i 就是一个潜在的偏离段的开始.如果在报文 j 是第 1 个满足:

$$q(j) - q(i) < \frac{\max_{i < k < j} [q(k) - q(i)]}{F} \quad (10)$$

那么报文 j 是这个偏离段的结束.如果整个过程都不存在满足条件的 j ,则认为该偏离段没有结束.公式(10)中的 F 是被称为减少因子的参数.利用对延迟图形的分析,计算每个探测报文的瞬间可用带宽,将报文 i 的瞬间可用

带宽记为 $E_m^{(k)}$,再通过一个加权公式来计算这个探测链测得的可用带宽,记为 $D^{(m)}$.其计算公式如下:

$$D^{(m)} = \frac{\sum_{k=1}^{N-1} E_k^{(m)} A_k}{\sum_{k=1}^{N-1} A_k} \quad (11)$$

最后,再将几次探测测得的可用带宽加以平均,便得到最终结果.

然而,pathChirp 根据的实质是,根据包列中相邻包对的速率是否大于可用带宽来进行测量.这种方法的测量结果与包列速率无关,仅与第 1 对遇到增长排队延迟的包对速率有关.因此,这种方法受带宽突发性的影响,在多跳突发背景流下的测量误差较大.为此,pathchirp 采用发用多个 Chirp 求平均的方法来减小测量误差.在 pathChirp 作者提供的 NS-2 仿真环境代码中(version:pathchirp_ns_2.27),Chirp 数为 7;在实际网络代码中,Chirp 设置为 11(version:pathchirp-2.4.1).

本文在实验过程中发现,pathChirp 算法中指数增长区间(*lowrate,highrate*)的设置对测量时间、测量精度有着很大的影响,适用于某一拓扑和背景流量的指数增长区间并不一定适用于其他拓扑和背景流量,甚至会出现较大偏差.但如何针对特定的拓扑和背景流量特性找到 pathChirp 最佳的指数增长区间,不是本文的研究内容.因此,在所有对比实验中,我们都从多次实验过程中选出性能较好指数增长区间组合,并具体列出.

SLDRT 算法是依据整个包列速率去推测可用带宽,这与 pathChirp 算法本质上不相同.正如文献[6]中指出的:由于网络流量的不确定性,完全精确的可用带宽测量无法达成,但是较长的包列能够显著削弱误差.理论及实验结果表明,SLDRT 算法具备了单次采样、准确测量的特性.

尽管在本文实验中设定的递减因子取值较小,但即使在初始速率与可用带宽值相差约 20 倍的情况下,SLDRT 完成一次测量也仅需 200 多 ms.如果可以从网络服务供应商(ISP)那里获得,或者是利用某些软件得到瓶颈链路的大致容量,有针对性地设置 SLDRT 算法初始速率,则算法的收敛速率会更快.

5 结 论

本文提出一种采用降速率包列(SLDRT)的可用带宽测量方法.与目前的 PRM 算法相比,SLDRT 不需要背景流为流体模型的假设.本文从理论上证明了流体/非流体模型下 SLDRT 算法的正确性,表明 SLDRT 具备单次采样、准确测量的特性.其次,本文通过理论证明和实验验证的方法研究了 SLDRT 算法各个参数的取值情况,并在 NS-2 仿真实验平台和实验网上与 PRM 算法中最具代表性的 pathChirp 和 Pathload 算法从测量时间、测量精度、测量负载及入侵度等方面进行了对比,验证其具有准确度高、测量速度快的优点,并且在多跳、突发性背景流下具有较强的健壮性.

考虑测量包丢失的情况以及根据网络容量及负载状况动态选取 SLDRT 算法的参数,是未来算法改进及应用的重要方向.

致谢 文献[19]的作者提供了 Jain^[11]写的 Pathload 算法在 NS-2 仿真环境下的源码,使本文在对比分析实验中所用的程序均来源于算法原作者,在此表示感谢.

References:

- [1] He L, Yu SZ. Methodology for measuring available bandwidth on arbitrary links. Journal of Software, 2009,20(4):997-1013 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3306.htm> [doi: 10.3724/SP.J.1001.2009.03306]
- [2] Liu M, Li ZC, Guo XB, Deng H. An end-to-end available bandwidth estimation methodology. Journal of Software, 2006,17(1): 108-116 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/17/108.htm> [doi: 10.1360/jos170108]
- [3] Zhou AF, Liu M, Li ZC, Xie GG. Self adaptive method for end-to-end available bandwidth estimation. Journal on Communications, 2008,29(12):37-45(in Chinese with English abstract). [doi: CNKI:SUN:TXXB.0.2008-12-008]
- [4] Huang YC, Lu CS, Wu HK. JitterPath: Probing noise resilient one-way delay jitter-based available bandwidth estimation. IEEE Trans. on Multimedia, 2007,9(4):798-812. [doi: 10.1109/TMM.2007.893343]

- [5] Strauss J, Katabi D, Kaashoek F. A measurement study of available bandwidth estimation tools. In: Proc. of the ACM SIGCOMM Conf. on Internet Measurement. New York: ACM, 2003. 39–44. [doi: 10.1145/948205.948211]
- [6] Liu X, Ravindran K, Loguinov D. A stochastic foundation of available bandwidth estimation: Multi-Hop analysis. *IEEE/ACM Trans. on Networks*, 2008,16(1):130–143. [doi: 10.1109/TNET.2007.899014]
- [7] Liu XC, He L, Yu SZ. Algorithms of accurate network available bandwidth measurement. *Acta Electronica Sinica*, 2007,35(1): 68–72 (in Chinese with English abstract). [doi: CNKI:ISSN:0372-2112.0.2007-01-014]
- [8] Leland WE, Taqu MS, Willinger W, Wilson DV. On the self-similarity nature of Ethernet traffic. *IEEE/ATM Trans. on Networking*, 1994,2(1):1–15. [doi: 10.1109/90.282603]
- [9] Loa L, Dovrolis C, Sanadidi M. The probe gap model can underestimate the available bandwidth of multihop paths. *ACM SIGCOMM Computer Communication Review*, 2006,36(5):29–34. [doi: 10.1145/1163593.1163599]
- [10] Hu NN, Steenkiste P. Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications*, 2003,21(6):879–894. [doi: 10.1109/JSAC.2003.814505]
- [11] Jain M, Dovrolis C. End-to-End available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. *IEEE/ACM Trans. on Networking*, 2003,11(4):537–549. [doi: 10.1109/TNET.2003.815304]
- [12] Ribeiro VJ, Riedi RH, Baraniuk RG, Navrati J, Cottrell L. PathChirp: Efficient available bandwidth estimation for network paths. In: Proc. of the Passive and Active Measurement Workshop. La Jolla, 2003. 200–211.
- [13] Zhang DL, Huang WL, Lin C. Locating the tightest link of a network path. In: Proc. of the ACM SIGMETRICS Performance Evaluation Review. New York, 2004. 402–403. [doi: 10.1145/1012888.1005738]
- [14] Ribeiro VJ, Riedi RH, Baraniuk RG. Locating available bandwidth bottlenecks. *IEEE Internet Computing Magazine*, 2004,8(5): 34–41. [doi: 10.1109/MIC.2004.45]
- [15] Navrati J, Cottrell RL. ABwE: A practical approach to available bandwidth estimation. In: Proc. of the Passive and Active Measurement Workshop. La Jolla, 2003.
- [16] Ribeiro VJ, Coates MJ, Riedi RH, Sarvotham S, Hendricks B, Baraniuk RG. Multifractal cross traffic estimation. In: Proc. of the ITC Specialist Seminar on IP Trac Measurement, Modeling, and Management. Monterey, 2000.
- [17] Liu Q, Hwang JN. End-to-End available bandwidth estimation and time measurement adjustment for multimedia QoS. In: Proc. of the IEEE Int'l Conf. on Multimedia and Expo. Baltimore, 2003. 373–376. [doi: 10.1109/ICME.2003.1221326]
- [18] Melander B, Bjorkman M, Gunningber P. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In: Proc. of the IEEE Global Internet Symp. San Francisco, 2000. 415–420. [doi: 10.1109/GLOCOM.2000.892039]
- [19] Hei XJ, Bensaou B, Tsang DHK. Model-Based end-to-end available bandwidth inference using queueing analysis. *Computer Networks*, 2006,50(12):1916–1937. [doi: 10.1016/j.comnet.2005.09.029]
- [20] Li WW. The researches on monitoring techniques for end to end Internet performance [Ph.D. Thesis]. Changsha: Hu'nan University, 2006 (in Chinese with English abstract). [doi: CNKI:CDMD:1.2007.020399]
- [21] Ns2: Network simulator. <http://nslam.isi.edu/nsnam/>
- [22] Taqu MS, Willinger W, Sherman R. Proof of a fundamental result in self-similar traffic modeling. *ACM SIGCOMM Computer Communications Review*, 1997,27(2):5–23 [doi: 10.1145/263876.263879]
- [23] Dovrolis C, Ramanathan P, Moore D. What do packet dispersion techniques measure? In: Proc. of the INFOCOM 2001. Anchorage, 2001. 905–914. [doi: 10.1109/INFCOM.2001.916282]
- [24] Shriram A, Murray M, Hyun Y, Brownlee N, Broido A, Fomenkov M, Claffy K. Comparison of public end-to-end bandwidth estimation tools on high-speed links. In: Proc. of the Passive and Active Measurement Workshop. Boston, 2005. 306–320. [doi: 10.1007/978-3-540-31966-5_24]
- [25] Shriram A, Kaur J. Empirical evaluation of techniques for measuring available bandwidth. In: Proc. of the INFOCOM 2007. Anchorage, 2007. 2162–2170. [doi: 10.1109/INFCOM.2007.250]
- [26] PathChirp software. <http://www.spin.rice.edu/Software/pathChirp>
- [27] Pathload software. <http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/bw-est/pathload.html>
- [28] MAWI. <http://tracer.csl.sony.co.jp/mawi/>
- [29] Tcpreplay. <http://tcpreplay.synfin.net/trac/wiki/>

[30] Net-Snmp. <http://net-snmp.sourceforge.net/>

附中文参考文献:

- [1] 何莉,余顺争.一种测量任意链路可用带宽的方法.软件学报,2009,20(4):997-1013. <http://www.jos.org.cn/1000-9825/3306.htm> [doi: 10.3724/SP.J.1001.2009.03306]
- [2] 刘敏,李忠诚,过晓冰,邓辉.端到端的可用带宽测量方法.软件学报,2006,17(1):108-116. <http://www.jos.org.cn/1000-9825/17/108.htm> [doi: 10.1360/jos170108]
- [3] 周安福,刘敏,李忠诚,谢高岗.自适应的端到端可用带宽测量方法.通信学报,2008,29(12):37-45. [doi: CNKI:SUN:TXXB.0.2008-12-008]
- [7] 刘星成,何莉,余顺争.网络可用带宽的高精度测量算法.电子学报,2007,35(1):68-72. [doi: CNKI:ISSN:0372-2112.0.2007-01-014]
- [20] 黎文伟.端到端互联网性能监测技术研究[博士学位论文].长沙:湖南大学,2006. [doi: CNKI:CDMD:1.2007.020399]



张大陆(1949—),男,山东莱阳人,教授,博士生导师,主要研究领域为计算机网络,分布式系统.



朱安奇(1984—),男,硕士,主要研究领域为计算机网络,分布式系统.



胡治国(1977—),男,博士生,主要研究领域为计算机网络,分布式系统.



张俊生(1984—),男,硕士,主要研究领域为计算机网络,分布式系统.