

基于拥塞预知的路由算法研究^①

段小龙^{②*} 郭承青^{**} 闫健恩^{③*} 迟乐军^{*}

(* 哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

(** 国家计算机网络应急技术处理协调中心 北京 100029)

摘 要 针对网络信息量的不断增大造成的网络拥塞已经成为制约网络性能的主要因素的问题,研究了现有路由算法的改进,以缓解可能出现的拥塞,并提出了一种拥塞预知路由算法(CPRA)。该算法通过周期性地检测队列缓冲区占用率(BOR)来判断链路是否有发生拥塞的趋势,当队列 BOR 达到一定阈值时,则认为链路有发生拥塞的趋势,并根据局部的拓扑结构与链路状态快速计算备用路由,当链路 BOR 达到一定阈值时启用备用路由转发数据包。在网络模拟器 NS3 上的仿真结果表明,相对于开放最短路径优先(OSPF)算法,CPRA 能够显著提高网络的性能:网络吞吐率提高约 50%,端到端平均时延减少 0.3s。

关键词 路由算法,拥塞预知,链路状态,备用路由

0 引 言

提高网络的性能包括提高网络的传输能力以及降低网络的传输时延,而网络拥塞往往是造成网络性能下降的主要原因,它越来越成为制约网络发展与应用的一个瓶颈,所以研究一种有效解决网络拥塞的方法具有重要的意义。研究表明,网络本身的拓扑结构及其采用的路由协议都会影响网络的拥塞程度^[1,2]。文献[3,4]从网络的拓扑结构方面分析了造成网络拥塞的原因,提出了一种通过增加或删除节点和链路来缓解拥塞的方法。该方法要在发生拥塞时改变网络拓扑结构,在真实网络中实施的可行性并不高,所以,相比于改变网络拓扑,改进现有的路由算法更具实际意义。开放最短路径优先(open shortest path first, OSPF)算法^[5]是一种基于最短路径优先(SPF)算法的内部网关路由协议,它通过链路状态通告来获得全网拓扑并计算路由,由于具有无环路、收敛速度快等特点,因而成为网络中广泛使用的自治系统路由协议。但是,由于该协议采用的选路算法总是计算源节点到目的节点的“最优路径”,因而导致数据包总是沿着该“最优路径”

进行转发。当网络中不同节点对的最优路径出现部分重叠时,就可能使得某些链路被频繁地使用和链路负载过大,进而导致局部网络拥塞。所以本文主要研究如何改进 OSPF 以缓解可能出现的拥塞问题。

在通过改进路由算法解决网络拥塞问题方面,文献[6]提出了一种基于无标度网络的全局动态路由算法,该算法以队列长度作为代价计算最短路径;文献[7]提出了一种基于局部信息的无标度网络路由算法,根据局部拓扑信息决定如何转发数据包;文献[8-10]采用了多径路由机制,路由表中存储多个到达某一目的网络的出口,当网络出现拥塞时路由器将目标网络相同的数据分组从不同的“下一跳”转发出去,使网络流量可以分散到不同的链路中去,因此可以缓解网络拥塞。上述算法虽然都能在一定程度上缓解网络拥塞和提高网络的性能,但是在计算链路代价时仅仅考虑了网络的静态特征或者动态特征,在选路时不能保证全局最优。对于多径路由,因为网络中只有小部分节点会发生拥塞,多数节点路由表项存储的多个出口并没有启用,造成了存储资源的浪费,所以可以考虑仅在发生拥塞的节点计算多个出口以分散流量。综合考虑各种因素,本文

① 国家科技支撑计划(2012BAH45B01),国家自然科学基金(61100189,61370215,61370211),国家信息安全 242 计划(2014A085)和山东省中青年科学家奖励基金(BS2011DX001)资助项目。
② 男,1990 年生,硕士生;研究方向:网络与信息安全;E-mail: pymongo@gmail.com
③ 通讯作者,E-mail: yje@hitwh.edu.cn
(收稿日期:2014-05-08)

提出了一种拥塞预知路由算法 (congestion perception routing algorithm, CPRA), 并给出了一种新的链路代价模型。该算法综合考虑了网络的静态特征与动态特征, 根据代价模型计算出全局最优路由, 当网络出现拥塞时在局部查找其它可行出口, 在所有可行出口间平衡网络流量, 缓解网络的拥塞程度。本研究在 NS3 网络模拟器^[11]上对 CPRA 进行了仿真, 并将仿真结果与开放最短路径优先 (OSPF) 算法的结果进行了对比。

1 BA 网络模型

Barabasi 等人的研究表明, 以因特网为代表的大型通信网络都具有小世界及无标度 (scale-free) 的连接特性^[12,13], 即节点与节点之间的连接分布遵循幂次定律: 大部分节点只有少数连接, 少数节点 (hub 节点) 则拥有大量的连接。为了不失一般性, 本文利用著名的 (Barabasi-Albert, BA) 模型构造一个无标度网络对算法进行测试。构造方法是: 首先构造 m_0 个初始结点, 每经过一个时间步长引入一个节点, 由该节点生成 m ($m < m_0$) 条边与原有的节点进行连接, 连接过程中优先考虑度数高的结点, 已有节点被连接的概率正比于该节点的度数, 即对于某个原有节点 s_i , 将其在原网络中的度数记作 d_i , 那么新节点与之相连的概率 p_i ^[7] 为

$$P_i = \frac{d_i}{\sum_{j=1}^n d_j} \quad (1)$$

2 拥塞感知队列模型

现有的动态路由协议在网络发生变化时通过洪泛路由表或者链路状态通告来实现路由的更新, 在重新计算路由的过程中大都以带宽、时延以及跳数等因素作为链路代价计算的依据。由于没有考虑链路当前的拥塞程度, 导致流量无法避开拥塞的链路。如果在路由选择之前通过拥塞检测预知到链路拥塞, 避开较繁忙的链路, 就能有效减少传输过程中拥塞的发生, 从而在根本上提高数据传输效率。

2.1 拥塞度量

衡量拥塞程度的方法一般分为两种: 缓冲区占用率 (buffer occupation rate, BOR) 和信道负载 (channel loading, CL)^[14]。使用缓冲区占用率来衡量链路拥塞程度的思想很简单: 定期检测接口链路的队列

缓冲区占用率, 如果缓冲区占用率达到事先设定的上限值, 则认为该链路有发生拥塞的可能性。由于缓冲区占用率仅表示某一时刻的占用率, 并不能代表缓冲区占用的变化趋势。所以, 为了更加准确地预测链路的拥塞程度, 本文使用缓冲区占用率和堆积率进行拥塞度量。

定义 1: 堆积率 (accumulation rate, AR) 为数据包到达节点的速率与流出节点速率的比率。如果堆积率大于或等于 1, 说明报文到达的速度大于或等于流出的速度, 缓冲区就会开始堆积, AR 值越大, 拥塞越严重。

定义 2: 拥塞度 (congestion degree, CD) 为链路发生拥塞可能性大小的度量, 其计算公式为

$$CD_{i,j}(t) = \mu \times BOR_{i,j}(t) + \gamma \times (1 - e^{-AR_{i,j}(t)}) \quad (2)$$

其中 μ, γ 为可调节参数, 分别表示缓冲区占用率 (BOR) 和堆积率 (AR) 在计算拥塞度 (CD) 时所占比, 且 $\mu + \gamma = 1$; $BOR_{i,j}$ 表示节点 i, j 之间接口对应的缓冲区占用率, $0 < BOR_{i,j} < 1$, 其值越小说明节点的缓存被占用的就越少, 通过此节点进行数据传输发生拥塞的可能性就越小; AR 表示该接口的堆积率, AR 大于 1 或是小于 1, 反映了缓存占用率的增、减趋势。综上可知, 拥塞度的大小与节点 i 的接口链路发生拥塞的可能性成正比, 并且 $CD \in [0, 1]$ 。

根据接口缓冲区的拥塞程度将数据包在链路中的传输过程分为 4 个阶段, 为了定量地描述链路的状态, 设置一个拥塞标志变量 CF (congestion flag), 分别对应链路所处的 4 个阶段:

正常传输 (normal translation, NT) 阶段: 根据路由表转发数据包;

拥塞避免 (congestion avoid, CA) 阶段: 根据路由表转发数据包, 并计算备用路由;

负载均衡 (load balance, LB) 阶段: 当某条链路负载达到一定阈值之后, 启用备用路由, 在备用路由和原路由之间平衡网络流量;

丢包 (packet dropping, PD) 阶段: 丢弃数据包。

本文为 BOR 设置了两个阈值即 α_1, α_2 , 为 AR 设置了一个阈值 β 。通过周期性检测 BOR 与 AR 的值确定 CF 的值, 其规则如下:

规则 1: 当 $BOR < \alpha_1$ 时, 链路处于 NT 阶段, 拥塞标志 CF 置为 0。

规则 2: 当 $BOR > \alpha_1$, $BOR < \alpha_2$ 和 $AR < \beta$ 时, 链路处于 CA 阶段, 拥塞标志 CF 置为 1。

规则 3: 当 $BOR > \alpha_1$, $BOR < \alpha_2$ 和 $AR > \beta$ 时, 链路处于 LB 阶段, 拥塞标志 CF 置为 2。

规则 4: 当 $BOR > \alpha_2$, 链路处于 PD 阶段, 拥塞标志 CF 置为 3。

2.2 链路代价模型

在实际网络中, 链路的状态是动态变化的, 所以在不同时刻选择某条链路的代价也是不同的。以开放最短路径优先 (OSPF) 为例, 当路由器之间定期更新路由表或者通过触发方式更新路由表时, 需要进行链路状态通告, 每一次通告中接口的度量 (metric) 应该是动态变化的, 所以定义:

$$w_{i,j}(t) = c_{i,j} + CF_{i,j}(t) \times K_j^{CD_{i,j}(t)} \quad (3)$$

其中, $w_{i,j}(t)$ 表示链路 ij 对应接口的度量, $c_{i,j}$ 表示链路 ij 的代价值, $CF_{i,j}(t)$ 表示 t 时刻链路 ij 对应的 CF 值, K_j 表示节点 j 的度数, $CD_{i,j}(t)$ 表示 t 时刻链路 ij 的拥塞度。在链路畅通的情况下 CF 值始终为 0, 所以度量始终为链路的代价值, 当链路有拥塞趋势时度量增大, 并且随着拥塞程度的增大度量增大得越快, 所以再次更新路由时即可绕开拥塞的链路。假设源节点到目的节点的路径表示为 $P(s \rightarrow d) = x_0, x_1, \dots, x_n$, 路径总代价为

$$\begin{aligned} T_{sd}(t) &= \sum_{i=0}^{n-1} w_{i,i+1}(t) \\ &= \sum_{i=0}^{n-1} c_{i,i+1} + \sum_{i=0}^{n-1} CF_{i,i+1}(t) \times K_{i+1}^{CD_{i,i+1}(t)} \end{aligned} \quad (4)$$

在所有 s 与 d 之间的路径中, 选择使得 T_{sd} 最小的路径作为“最短路径”进行路由。由公式 (4) 知, 在链路畅通的情况下公式的第二项为 0, 路由选择结果与 OSPF 相同, 当有拥塞出现时公式第二项会综合考虑网络的各种参数, 根据网络实时状态确定“最短路径”, 所以该模型能确保算法始终选择全局最优路径。

3 局部路由策略

当网络中局部发生拥塞时, 为了保证网络业务不受影响, 要及时改变当前路由以避免拥塞加剧。该过程同上下班高峰开车类似, 开始时按最短路径行驶, 当看到前面路口发生堵塞时, 就近选择一条比较畅通的道路, 从而避开这个拥塞的路口。虽然该路径不是最短路径, 但比继续堵在拥塞的路口顺畅得多。在无标度网络中, 度数较大的节点 (hub 节点) 往往是网络的枢纽, 此类节点更容易发生拥塞,

所以当该节点的某条链路流量过大时可以将流量分散到其它链路中。

图 1 表示网络的某个局部拓扑结构, 箭头代表流量的流向。当链路处于 CA 阶段时, 即链路 ig 有拥塞的趋势时, 在节点 i 的邻居节点中寻找可行的下一跳以分担链路 ig 上的流量。定义下一跳节点满意度函数 $f(i, j)$, 表示节点 j 作为 i 的下一跳的

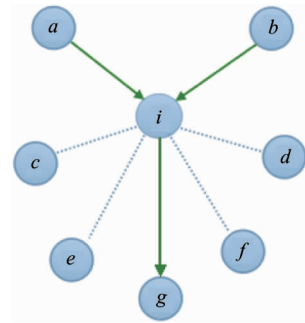


图 1 hub 节点局部拓扑图

优先权的大小。以 p 表示数据包, $PH(p)$ 表示数据包 p 的上一跳节点, $DST(p)$ 表示数据包的目的地址, $NH(i, DST(p))$ 表示节点 i 到达目的地 $DST(p)$ 的下一跳节点, $N(i)$ 表示 i 的邻居节点的集合。则满意度函数 $f(i, j)$ 表示如下:

$$f(i, j) = (k_j e^{-CD_{i,j}})^\alpha \quad (5)$$

约束条件为:

$$\begin{cases} j \in N(i) \\ j \neq PH(p) \\ NH(j, DST(p)) \neq i \end{cases}$$

其中, α 为可调节参数, k_j 为节点 j 的度数, $CD_{i,j}$ 代表节点 ij 间链路的拥塞度。公式 (5) 表明, α 大于零时优先选择度数大的邻居节点作为下一跳节点, α 小于零时优先选择度数小的邻居节点作为下一跳节点。在无标度网络中度数大的节点会承担较多的网络流量, 所以当局部拥塞时选择度数较小的节点转发数据包能够避开拥塞的链路。因此满足 $\min(f(i, j))$ 的节点即为备用下一跳节点。当节点链路处于阶段 LB 时, 若下一跳节点 g 的备用下一跳为 c , 则利用链路 ic 分担 ig 原有的流量。

4 拥塞预知路由算法 (CPRA)

本节给出 CPRA 算法的实现过程。算法综合考虑了网络的静态特征 (拓扑结构) 和动态特征 (队列拥塞程度), 利用第 2 节提出的链路代价模型计算

全局路由,当网络拥塞时利用第 3 节最后给出的一种局部路由查找策略,在链路拥塞时能够快速地计算备用路由。CPRA 算法如下:

```
输入:节点集合 nodes、权值  $\mu$ 、 $\gamma$ 、 $\alpha_1$ 、 $\alpha_2$ 、 $\alpha$ 、 $\beta$ 
BEGIN
  初始化  $\mu$ 、 $\gamma$ 、 $\alpha_1$ 、 $\alpha_2$ 、 $\alpha$ 、 $\beta$ 
  for 节点 n in nodes
    for 节点 n 的每一个接口 do
      利用式(3)计算接口的度量
    end for
  end for
  利用 dijkstra 算法计算全局路由表
  while 缓冲区队列非空 do
    出队一个数据包 Packet
    根据路由表查询 Packet 的下一跳出口 d
    switch (CF)  $\leftarrow$  d 对应链路的拥塞标志变量,
      见第 2 节
    case 0:从出口 d 转发 Packet
    case 1:利用式(5)计算备用路由
    case 2:利用备用路由与出口 d 同时转发 Packet
  end while
END
```

上述算法在转发数据包前会判断 CF 的值,当队列缓冲区占用率达到阈值 α_2 时,启用备用路由,缓解了链路的拥塞程度。

5 实验结果

本小节根据无标度网络模型构造了一个如图 2 所示的网络拓扑,在 NS3 网络模拟器上生成该拓扑。

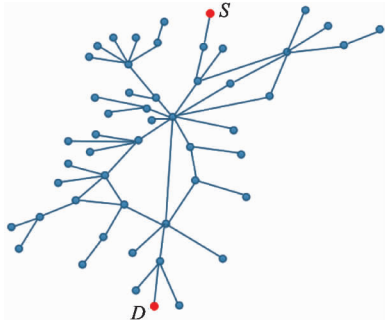


图 2 网络拓扑结构

在网络中随机选择一对节点(源节点 S 和目的节点 D)进行点对点通信,同时,在网络中选择多个

节点以不同的发包频率向目的节点发送用户数据报协议(UDP)数据包作为背景流量,并且在 60s 时更新路由表,同时节点 S 以频率 100 向节点 D 发送 UDP 数据包。

图 3 表示吞吐率随通信速率变化的曲线,为了方便统计,以单位时间内目的节点成功接收的数据包数量的最大值代表吞吐率。可以看出,当发包速率较低时两种路由策略吞吐率基本相同。随着发包速率的增加(增加到每秒发送 350 个数据包),由于节点处理的处理能力有限,链路将发生拥塞并不断丢包,网络吞吐率也将达到上限,但是同 OSPF 算法相比,CPRA 在拥塞时能够及时分流、缓解网络拥塞,所以吞吐率提高了约 50%。

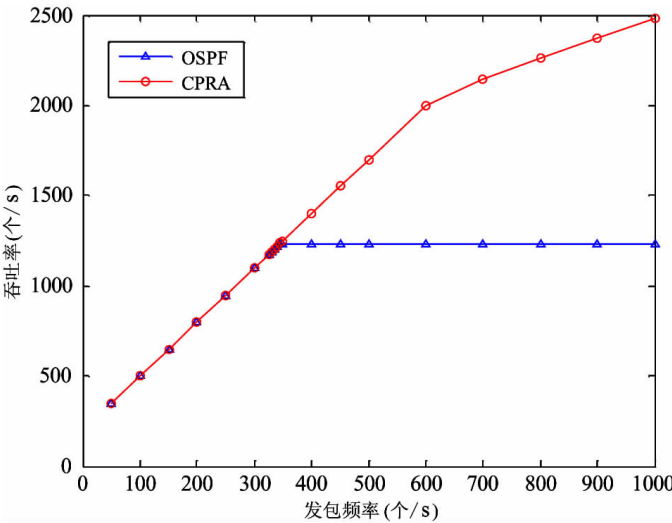


图 3 吞吐率随发包频率变化曲线

表 1 统计了部分链路在不同时刻的链路代价。表中数据,以 1/8.25 为例,代表使用 OSPF 算法时链路代价为 1,使用 CPRA 时链路代价为 8.25。表 2 统计了发包频率为 500 时,数据包由节点 A 到节点 D 的转发路径上各个节点的收包情况,表中数据,以 10190/14454 为例,代表使用 OSPF 算法时收包数量为 10190,使用 CPRA 时收包数量为 14454。

表 1 各链路代价统计表

时间(s)	链路 AB	链路 ACEB	链路 AGHIB
0	1/1	3/3	4/4
30	1/8.25	3/6.12	4/4
60	1/8.05	3/6.20	4/4
90	1/1	3/3	4/4

表 2 各节点累计收包数量统计表

时间(s)	节点 A	节点 B	节点 C	节点 E	节点 G
10	14907/14907	10190/14454	0/3845	0/3830	0/0
20	29907/29907	20326/29455	0/8162	0/8147	0/0
30	44907/44907	30462/44455	0/12456	0/12441	0/0
40	59907/59907	40598/59460	0/16783	0/16775	0/0
50	74907/74907	50734/74444	0/21073	0/21058	0/0
60	89907/89907	60870/89455	0/25403	0/25388	0/0
70	90924/90924	61801/90472	0/25403	0/25388	0/934
80	91924/91924	62801/91470	0/25403	0/25388	0/1934
90	92924/92924	63801/92470	0/25403	0/25388	0/2934
90.1	93000/93000	64801/93000	0/25403	0/25388	0/3000

使用 OSPF 算法时,链路 AB 的代价最低,所以数据包通过链路 ABFD(图 4(a))进行转发。整个转发过程中节点 C 和 E 的收包数量始终为 0,说明数据包仅仅通过路径 ABFD 进行转发。同时节点收包数量由节点 A 的 90000 个降到节点 B 的 60881 个,说明大量数据包因拥塞而被丢掉;而使用 CPRA 时,有一部分数据包通过路径 ACEBFD(图 4(b))进

行转发,说明 CPRA 在链路发生拥塞时会根据局部路由策略分散拥塞链路的流量,同时转发过程中几乎没有丢包现象。在 60s 之后,数据包沿着 AGHIB 进行转发,并且链路 AGHIB 的代价最低,所以改进后链路代价模型能够保证选路的全局最优。

图 5 代表不同发包频率(rate)下节点缓冲区数据包个数随时间变化的曲线。可以看出,当通信频

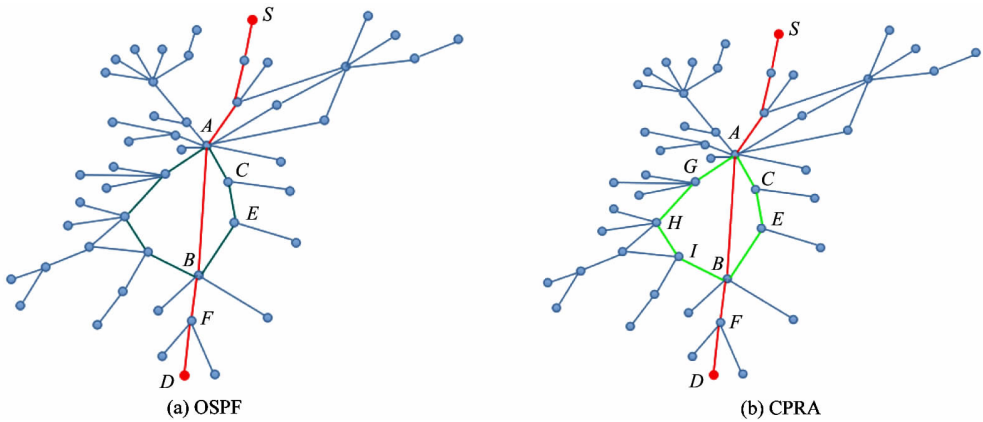


图 4 采用 OSPF(a)和 CPRA(b)时数据包转发路径

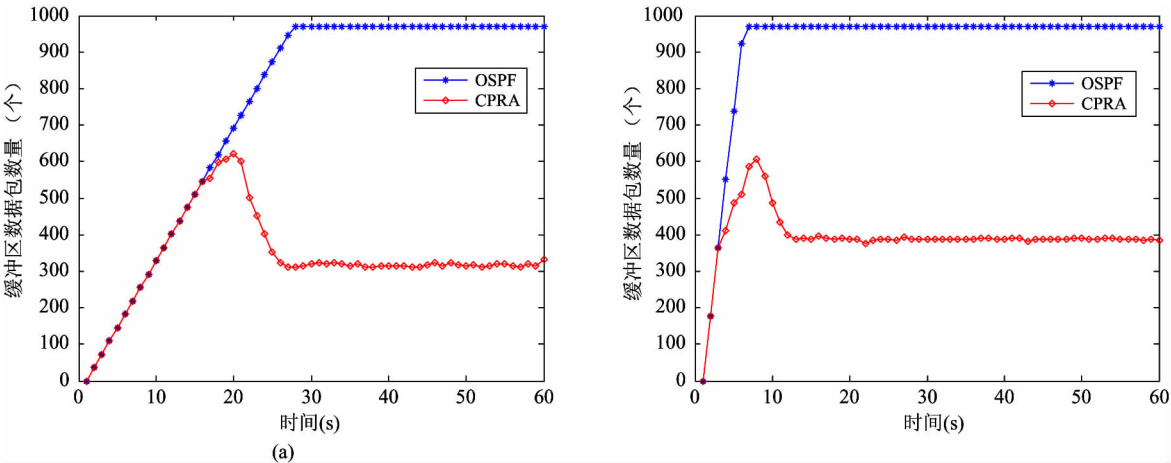


图 5 通信频率为 350(a)和 500(b)时缓冲区数据包数量变化

率逐渐增大到使网络发生拥塞时,若采用传统的队列模型缓冲区将很快被充满,采用本文的拥塞预知队列模型以及局部路由策略则能够及时地分散网络流量,使得节点缓冲区占用率维持在一个相对较低的水平。

表3统计了采用CPRA以频率350发包时,源节点发送数据包的数量和目的节点接收数据包的数量,不难看出,目的节点接收的数据包数量等于源节点发送的数据包数量,证明图5所示的缓冲区占用率下降是由于分流而非队列丢包所致。

表3 源节点累计发包与目的节点累计收包数量统计表

时间(s)	源节点累计发包数量	目的节点累计收包数量
10	11808	11784
20	23944	23920
30	36080	36056
40	48216	48192
50	60352	60328
60	72488	72464
61	73987	73677
62	73987	73987

图6代表端到端时延随通信速率变化的曲线,由于CPRA在网络局部发生拥塞时采用多条路径转发,减少了排队产生的时延,所以大大降低了端到端时延。

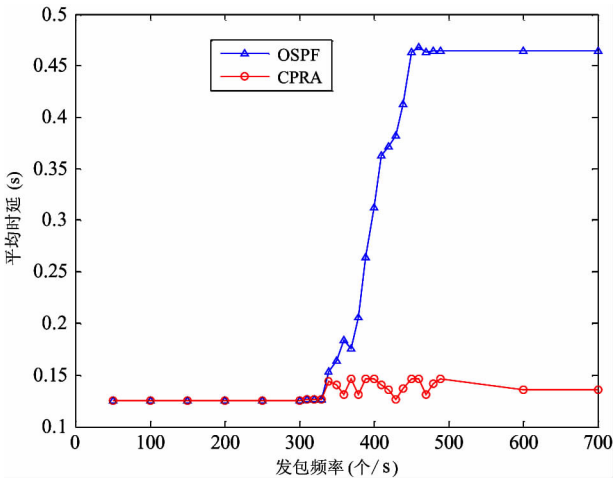


图6 端到端时延随通信频率变化曲线

6 结 论

本文从队列模型和路由策略方面进行研究,分析了网络的静态特征与动态特性对网络性能的影响,

针对网络拥塞问题,提出了一种拥塞预知路由算法CPRA。该算法采用拥塞预知队列模型,周期性地检测队列的拥塞程度,并将结果反馈给路由模块。路由模块根据链路队列缓冲区的拥塞情况,采用全局路由表转发与局部路由相结合的策略进行路由。另外,此算法改进了OSPF算法的链路代价模型,在确定链路代价时充分考虑了网络的拓扑结构以及拥塞程度等因素,保证了选路的全局最优。仿真结果表明,在理想情况下同传统的OSPF算法相比,采用CPRA使得网络吞吐率提高了约50%,端到端平均时延减少了0.3s。本文提出的拥塞预知路由算法提供了一种拥塞控制的思路,同时为后期研究新的路由协议奠定了基础。

本文提出的算法的实现还比较粗糙,没有过多考虑到效率问题,今后将对算法的效率进行分析并改进。另外,算法仅仅在仿真环境下进行了测试,而在真实环境中算法的性能会受到很多因素的制约,所以,在真实环境中对算法进行测试将是下一步研究的重点。

参考文献

[1] Xia Y, Tse C K, Lau F C M, et al. Traffic congestion analysis in complex networks. In: Proceedings of IEEE International Symposium on Circuits and Systems, Kos, Greece, 2006. 2625-2628

[2] Zhuo Y, Peng Y, Liu C, et al. Traffic dynamics on layered complex networks. *Physica A: Statistical Mechanics and its Applications*, 2011, 390(12): 2401-2407

[3] Huang W, Chow T W S. An efficient strategy for enhancing traffic capacity by removing links in scale-free networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2010, 2010(01): P01016

[4] Huang W, Chow T W S. Effective strategy of adding nodes and links for maximizing the traffic capacity of scale-free network. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2010, 20(3): 033123

[5] 刘惠山, 徐明伟, 徐格等. 因特网路由协议研究综述. *电信科学*, 2003, 19(10): 28-32

[6] Ling X, Hu M B, Jiang R, et al. Global dynamic routing for scale-free networks. *Physical Review E*, 2010, 81(1): 016113

[7] Yin C Y, Wang B H, Wang W X, et al. Efficient routing on scale-free networks based on local information. *Physics letters A*, 2006, 351(4): 220-224

[8] Cevher S, Ulutas M, Hokelek I. Performance evaluation of multiple routing configurations. In: Proceedings of the

- 21st Signal Processing and Communications Applications Conference (SIU), North Cyprus, Turkey, 2013. 1-4
- [9] Chen L, Ji H, Li Y, et al. Multi-path routing based on load-balance for cognitive packet networks. *The Journal of China Universities of Posts and Telecommunications*, 2011, 18(5): 71-75
- [10] Chen K C, Lin S Y, Hung H S, et al. Traffic-balanced topology-aware multiple routing adjustment for throttled 3D NoC systems. In: *Proceedings of IEEE Workshop on Signal Processing Systems*, Que' bec City, Canada, 2012. 120-124
- [11] 张登银, 张保峰. 新型网络模拟器 NS-3 研究. *计算机技术与发展*, 2009, 19(1): 1
- [12] Barabási A L, Albert R. Emergence of scaling in random networks. *Science*, 1999, 286(5439): 509-512
- [13] Cui L Y, Kumara S, Albert R. Complex networks: an engineering view. *Circuits and Systems Magazine, IEEE*, 2010, 10(3): 10-25
- [14] 李姗姗, 廖湘科, 朱培栋等. 传感器网络中一种拥塞避免, 检测与缓解策略. *计算机研究与发展*, 2007, 44(8): 1348-1356

Research on a congestion perception routing algorithm

Duan Xiaolong^{*}, Guo Chengqing^{**}, Yan Jianen^{*}, Chi Lejun^{*}

(^{*} School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

(^{**} The National Computer Network Emergency Response Technical Team Coordination Center of China, Beijing 100029)

Abstract

Aiming at the problem that the network congestion caused by the increasing network information now becomes the main factor restricting the performance of networks, the improvement of existing routing algorithms was conducted to reduce the possible congestion, and a congestion perception routing algorithm (CPRA) was presented. This algorithm performed the periodic testing of queue buffer occupancy rate (BOR) to predict whether the link is going to be congested or not. When the queue buffer occupancy rate reached a certain threshold, the link was considered more likely to be congested. A fast calculation of alternate routing was performed according to the local topology and link state, and the alternate routing was used to forward packets when the queue buffer occupancy rate reached a certain threshold. The results of the extensive simulation on the simulator NS3 showed that, compared with open shortest path first (OSPF) algorithm, The CPRA greatly improved the network performance: raising the network throughput by about 50% and decreasing the average end-to-end delay by about 0.3s.

Key words: routing algorithm, congestion perception, link state, alternate routes