

强动态 Ad Hoc 网的拥塞控制:价格协作和滚动优化*

徐伟强^{1,2}, 吴铁军², 汪亚明¹, 张云华¹, 陈积明²⁺

¹(浙江理工大学 信息电子学院,浙江 杭州 310018)

²(浙江大学 信息科学与工程学院,浙江 杭州 310027)

Congestion Control for High Dynamic Ad Hoc Networks: Price Cooperation and Receding Optimization

XU Wei-Qiang^{1,2}, WU Tie-Jun², WANG Ya-Ming¹, ZHANG Yun-Hua¹, CHEN Ji-Ming²⁺

¹(College of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou 310018, China)

²(College of Information Science and Engineering, Zhejiang University, Hangzhou 310027, China)

+ Corresponding author: E-mail: jmchen@ieee.org

Xu WQ, Wu TJ, Wang YM, Zhang YH, Chen JM. Congestion control for high dynamic ad hoc networks: Price cooperation and receding optimization. *Journal of Software*, 2008,19(9):2389–2402. <http://www.jos.org.cn/1000-9825/19/2389.htm>

Abstract: In Ad Hoc networks, there exist two fundamental characteristics: Multi-Hop wireless-based transmission and node mobility. The former results in different contention relationship among flows from that in wireline networks, and the latter leads to the time-varying network situations. Firstly, on the basis of the link's interference set depicting the characteristics of the contention relationship, the congestion control problem for small time interval is formulated as a nonlinear optimization problem. Secondly, by using the dual decomposition theory, a price cooperation approach (PCA) is proposed to solve this optimization problem. In PCA, a price framework based on the link's interference set is built. Meanwhile, to implement PCA in realistic ad hoc environment, three deployment techniques are proposed: Queue Size Monitoring, Neighbor Set Approximation, and HELLO-based Message Piggybacking. Otherwise, the network status detection and receding optimization is introduced to deal with the uncertain changes of network situations, and an adaptive optimization strategy (AOS) is proposed correspondingly. The simulation results in MATLAB environment show that AOS has better performance of adaptation to time-varying network situations than PCA. The simulation result in NS2 environment show that PCA and PCA+AOS significantly outperform TCP, ATCP, and ATP in many important performances, including throughput, packet drop ratio, and fairness, under a variety of scenarios and mobility models.

Key words: ad hoc network; congestion control; dual decomposition; receding optimization; price cooperation

摘要: Ad Hoc 网络存在着无线多跳连接、节点移动这两个本质的特点.前者引起了与固定网络截然不同的

* Supported by the National Natural Science Foundation of China under Grant Nos.60702081, 60604029, 60473038 (国家自然科学基金); the Joint Funds of NSFC-Guangdong under Great No.U0735003 (国家自然科学基金委-广东省联合重点基金项目); the Natural Science Foundation of Zhejiang Province of China under Grant Nos.Y107309, Y106384 (浙江省自然科学基金)

Received 2007-01-27; Accepted 2007-10-12

信息流竞争新特点,后者导致了网络状态不断发生变化.首先,在采用链路干扰集描述信息流竞争特点的基础上,将小时间段内网络状态不变的拥塞控制问题表达成非线性优化问题;其次,运用基于对偶分解理论的价格协作法 PCA(price cooperation approach)求解该优化问题,构建了一个基于链路干扰集的价格框架.同时,运用队列长度监控、邻居集合近似和 HELLO 捎带信息这 3 种技术将 PCA 转化成在实际 Ad Hoc 网络环境中可实施的协议.另一方面,运用状态检测和滚动优化方法,有针对性地解决 Ad Hoc 网络状态不确定时变性带来的问题,相应地设计了自适应优化策略 AOS(adaptive optimization strategy).MATLAB 仿真结果表明,AOS 策略比 PCA 对网络状态的变化具有更好的自适应性能.NS 仿真实验结果表明了 PCA 和 PCA+AOS 在几乎所有的仿真场景和移动环境下,在重要的性能指标,包括吞吐量、丢包率、公平性等方面,都比 TCP,ATCP 和 ATP 有了明显的改进.

关键词: Ad Hoc 网络;拥塞控制;对偶分解;滚动优化;价格协作

中图法分类号: TP393

文献标识码: A

Ad Hoc 网络是由不确定数量的无线节点组成的自组织网络系统,具有无中心控制、节点移动、多跳无线连接的特点^[1].拥塞控制是解决网络资源在信息流之间有效公平分配的重要手段.在 Internet 网络上,TCP 拥塞控制算法已经成为保证网络稳定性的重要因素.然而,传统 TCP 并不适用于 Ad Hoc 网络,导致了严重的性能下降^[2].因此近年来,一些学者将 Internet 网络中基于优化理论的拥塞控制算法设计理念引入到了 Ad Hoc 网络中,提出了新的拥塞控制算法来提高 Ad Hoc 网络性能.Chiang 等人^[3]考虑了物理层的功率变化对拥塞控制的影响,提出了 TCP Vegas 和功率控制的联合优化算法,实现带宽供求的优化平衡,从而提高端到端吞吐率和能量使用效率.Chen 等人^[4]和 Wang 等人^[5]从不同角度考虑了 MAC 层对拥塞控制的影响,提出了拥塞控制和 MAC 层的联合优化算法,但没有将算法转化成相应的协议.进一步地,Chen 等人^[6]和 Eryilmaz 等人^[7]充分考虑了 MAC 层和路由协议对拥塞控制的影响,提出了 MAC 层、路由协议和拥塞控制的联合优化算法,但也同样没有将算法转化成相应的协议.Xue 等人^[8]针对 Ad Hoc 网络信息流竞争的新特点,提出了基于 maximal cliques 构建的拥塞价格框架,提高了信息流之间分配资源的公平性和有效性.Lin 等人^[9]和 Chiang 等人^[10]对基于优化理论的拥塞控制设计作了前瞻性综述.然而,上述这些方法都没有考虑到 Ad Hoc 网络存在无线多跳连接、节点移动的新特点.前者引起了与固定网络不同的信息流竞争新特点,后者导致了网络状态不断发生不可预测且不可忽略的变化.因而,所设计的算法和相关协议提高网络的性能受到限制.

本文将致力于这两个本质特点的研究:(1) 从信息流竞争新特点出发,引入链路干扰集的概念,构建一个基于链路干扰集的价格框架.在此框架中,某链路的拥塞信息用此链路干扰集的总价格来度量,并提出了相应的分布式并行算法——价格协作法 PCA(price cooperation approach)以求解在小时间段内网络状态不变的拥塞控制问题;(2) 针对 Ad Hoc 网络状态的时变性,设计了一种基于状态检测和滚动优化的自适应优化策略 AOS(adaptive optimization strategy).我们先前的工作^[11]利用 MATLAB 仿真实验证实了 PCA 有良好的收敛性,对网络状态的变化具有一定的自适应能力.为了考察自适应优化策略 AOS 的有效性,我们利用 MATLAB 仿真实验比较了 PCA 的自适应能力与 AOS 的自适应优化性能.仿真结果表明,在强动态网络环境情况下,AOS 的自适应优化性能要显著强于 PCA 本身的自适应能力.NS 仿真实验结果表明,PCA, PCA+AOS 在多个仿真场景和移动环境下,吞吐量、丢包率、公平性都比 TCP,ATCP^[12]和 ATP^[13]有了明显改进.需要指出的是,在静态仿真场景下,PCA+AOS 的吞吐量性能要略逊于 PCA,原因是实施 AOS 带来了额外开销.

1 基本问题及其分布式算法 PCA

Ad Hoc 网络的一个重要特点是节点移动导致其参数和拓扑结构一直在发生不可预测且其影响不可忽略的变化.这种不确定性时变的特点,使得在其他网络中行之有效的办法(如 Internet 网络上的拥塞控制方法)不能简单地移植到 Ad Hoc 网络中来.为了解决不确定性时变情况下 Ad Hoc 网络的拥塞控制问题,本文将整个问题的求解分为两部分:

- (1) 将网络的运行时间切割成一系列足够小的时间段,并假设在每一个时间段内网络的参数和拓扑结构均

不发生变化.在此基础上,将该时间段内的 Ad Hoc 网络拥塞控制问题表达成一个确定性的非线性优化问题(本文称为基本问题),并提出一种高效的分布式求解算法:价格协作法 PCA.

(2) 在上述基本问题求解的基础上,利用我们先前的工作^[14],设计一种基于状态检测和滚动优化的自适应优化策略 AOS 来适应 Ad Hoc 网络参数和拓扑结构的不确定时变性.

1.1 基本问题

设在 $t \in [t_i, t_i + \Delta t_i], i=1, 2, \dots$ (Δt_i 是足够小的时间段)上所考察的 Ad Hoc 网络参数和结构不变,以下所述结论均在该时间段上有效.为叙述方便,下文中所有时变参数均省略了时间标记 t .

设 Ad Hoc 网络拓扑结构可用无向图 $G(N, L)$ 来描述,其中 $N=\{1, 2, \dots, N\}$ 是无线节点集合, $L=\{1, 2, \dots, L\}$ 是无线链路集合.设无线链路 $l \in L$ 的传输容量为 $C_l, S=\{1, 2, \dots, S\}$ 是分享网络资源的信息流集合,每个信息流 $s \in S$ 从源节点 s 出发经过无线链路集合 $L(s) \in L$ 到达目标端.我们称这样的信息流为端到端的多跳信息流.相应地,称该信息流在某条无线链路上的传输为信息子流, $S(l)=\{s \in S | l \in L(s)\}$ 是经过无线链路 l 的信息子流集合.每个信息流 s 都有一个效用函数与之对应: $U_s(x_s), \mathcal{R}^+ \rightarrow \mathcal{R}$, 其中, x_s 是信息流 s 的传输速率. $U_s(x_s)$ 满足如下条件: (1) $U_s(x_s)$ 是定义在 $x_s^{\min} \leq x_s \leq x_s^{\max}$ 上的单调递增且严格凹的二阶连续可微函数,其中, x_s^{\min} 为信息流 s 允许的最小传输速率, x_s^{\max} 为信息流 s 允许的最大传输速率; (2) 在 $x_s^{\min} \leq x_s \leq x_s^{\max}$ 都满足 $-U_s''(x_s) \geq 1/\bar{\alpha}_s > 0$.

针对 Ad Hoc 网中信息子流在通过无线链路 l 时面临着时间域和空间域同时竞争的特点,我们在先前的工作^[11]中引入了无线链路干扰集的概念.无线链路 l 的干扰集 IS_l 是指那些干扰了信息子流在无线链路 l 上传输的无线链路集合.在这些相互竞争的信息子流中,在某个时刻只能有一个信息子流可以真正占用无线链路 l .因此,在干扰集的所有无线链路上传输的信息流的累加速率不能超过无线链路 l 的信道容量,有如下表达式:

$$\sum_{k \in IS_l} \sum_{s \in S(k)} x_s \leq C_l \quad (1)$$

根据信息流的竞争新特点所得到的公式(1),结合文献[15,16]中适用于 Internet 的效用最大化问题,将 Ad Hoc 网络中拥塞控制问题表达成如下形式的非线性优化问题:

$$\begin{aligned} & \max_{\mathbf{x}} \sum_s U_s(x_s) \\ \text{s.t.} & \sum_{k \in IS_l} \sum_{s \in S(k)} x_s \leq C_l, \quad \forall l \end{aligned} \quad (2)$$

其中, $\mathbf{x} \in \mathbf{X} = \{x_s | x_s^{\min} \leq x_s \leq x_s^{\max}, s \in S\}$.

1.2 价格协作法 PCA

由问题(2)可知,目标函数是关于源端传输速率 x_s 的连续的严格凹函数,而且不等式限制是凸和紧的.所以,基于优化理论,问题(2)存在着关于源端传输速率的唯一最优解.我们用先前工作^[11]中提出的基于对偶分解理论的价格协作法 PCA 来求解拥塞控制问题式(2).在 PCA 中,通过不断更新无线链路的拥塞价格(即拉格朗日乘子 λ_l)和源端 s (即信息流 s)的传输速率来求解拥塞控制问题式(2).具体来说,公式(3)和公式(4)构成了求解拥塞控制问题式(2)的分布式迭代算法 PCA.

无线链路 l 的拥塞价格更新方式为

$$\lambda_l(n+1) = \left[\lambda_l(n) + \gamma \frac{\partial D}{\partial \lambda_l}(n) \right]^+ = [\lambda_l(n) + \gamma(x^l(n) - C_l)]^+ \quad (3)$$

其中, n 是迭代步数, $\gamma > 0$ 是迭代步长, $[z]^+ = \max\{0, z\}$, z 是实数, $x^l(n) = \sum_{k \in IS_l} \sum_{s \in S(k)} x_s(n)$ 为无线链路 l 上的信息累积速率.

源端 s (即信息流 s)的传输速率的更新方式为

$$x_s(n+1) = [(U_s')^{-1}(\lambda^s(n))]_{x_s^{\min}}^{x_s^{\max}} \quad (4)$$

其中, $\lambda^s = \sum_{l \in L(s)} \sum_{j \in IS_l} \lambda_j$ 被称为路径总拥塞价格, $[z]_a^b = \min\{\max\{z, a\}, b\}$, $(U_s')^{-1}$ 是 U_s 导数的逆.

关于迭代算法 PCA 的具体算法流程,请参见文献[11].

该迭代算法 PCA 表现出了如下一些重要特性:

(1) 如果把 λ_l 解释为无线链路 l 每单位带宽的“纯”价格,那么, $\sum_{j \in IS_l \setminus l} \lambda_j$ 就是无线链路 l 每单位带宽的“附加”价格; $\sum_{j \in IS_l} \lambda_j$ 就是无线链路 l 每单位带宽的“总”价格.事实上,将基于链路干扰集的总价格作为链路 l 的拥塞程度度量,非常符合 Ad Hoc 网络的信息流竞争新特性.在 Ad Hoc 网络中,当在某无线链路上传输信息时,对此无线链路的干扰集内的所有无线链路是否发生拥塞或拥塞严重的程度有直接影响.因此,为 Ad Hoc 网络设计的拥塞控制算法必须是基于整个无线链路干扰集的拥塞信息,而不仅仅是基于某无线链路上的拥塞信息. $\lambda^s = \sum_{l \in L(s)} \sum_{j \in IS_l} \lambda_j$ 就是信息流 s 所经无线链路的拥塞程度的总度量.此外,信息流竞争新特性还体现在信息累积速率 $\sum_{k \in IS_l} \sum_{s \in S(k)} x_s$. 该式表明了无线链路 l 上传输的信息子流除了“真实经过”部分 $\sum_{s \in S(l)} x_s$ 以外,还存在“虚拟干扰”部分 $\sum_{k \in IS_l \setminus l} \sum_{s \in S(k)} x_s$.

(2) 从公式(4)可以看出,PCA 的源端算法主要实现反馈拥塞信息的收集,用反馈拥塞信息来计算信息传输速率.源端没有针对丢包、延迟变大等隐式拥塞指示来调整传输速率,而是按照显式的拥塞度量来改变传输速率,这就克服了 TCP 拥塞控制机制中基于数据包丢失的拥塞指示在 Ad Hoc 网络中性能的显著下降的缺点.而且,PCA 算法是非经验的有严格优化理论指导而设计的基于速率(而非窗口)拥塞控制算法,克服了传统的 TCP 拥塞控制协议中基于窗口传输机制导致 TCP 性能下降的缺陷^[2].

2 PCA 协议化的解决方案

由 PCA 的无线链路算法公式(3)可知,要将 PCA 算法转化成协议需要解决如下几个难点问题:

(1) 无线链路 l 在计算总的累积信息速率 $\sum_{k \in IS_l} \sum_{s \in S(k)} x_s$ 时,需要收集到干扰集内所有无线链路 $k \in IS_l$ 的关于信息传输速率 $\sum_{s \in S(k)} x_s$ 的信息;

(2) 无线链路 l 在计算总的拥塞价格 $\sum_{k \in IS_l} \lambda_k$ 时,需要收集到干扰集内所有无线链路 $k \in IS_l$ 的关于“纯”拥塞价格 λ_k 的信息;

(3) 无线链路 l 在计算无线链路的“纯”拥塞价格时,似乎需要精确知道链路的容量.

要解决难点(1)和难点(2),实际上等价于需要解决两个问题:

(1') 链路的干扰集如何构建?

(2') 通过何种方式将关于信息传输速率和拥塞价格的信息告知干扰集内的无线链路?如何将收集到的这些信息组合成所需的信息累积速率和总拥塞价格.

对于问题(1'),可以通过邻居集合近似法来解决.对于问题(2'),可以通过 HELLO 捎带信息法来解决.对于问题(3),可以通过队列长度监控法来解决.下面详细描述这 3 种方法.

2.1 队列长度监控法

处于不同地理位置的无线媒质,其容量是不同的.而且,真实的链路容量受多种因素的影响,包括所采用的 MAC 层协议机制,通常无法精确知道.事实上,根据公式(3),链路的拥塞价格是由信息累积速率和链路容量的差决定的,并不一定需要知道链路的容量.由此,本文提出了队列长度监控法.

设算法在链路 l 第 n 次迭代时刻为 T_n ,则 $\int_{t=T_n}^{T_{n+1}} C_l(t) dt$ 表示了链路 l 从第 n 次迭代时刻到第 $n+1$ 次迭代时刻能传输的信息总量,而 $\sum_{k \in IS_l} \left(\int_{t=T_n}^{T_{n+1}} \sum_{s \in S(k)} x_s(t) dt \right)$ 表示了从第 n 次迭代时刻到第 $n+1$ 次迭代时刻网络想在链路 l 上传输(包含了干扰传输部分)的信息总量.如果后者小于前者,则链路 $k, k \in IS_l$ 上的信息累积量就会减少,表现为链路 $k, k \in IS_l$ 的队列长度变短,甚至出现队列是空的;相反地,如果后者大于前者,则链路 $k, k \in IS_l$ 上的信息累积量就会增加,表现为链路的队列长度变长,甚至会引起数据包的丢失.因此有如下表达式:

$$\int_{t=T_n}^{T_{n+1}} \left(\sum_{k \in IS_l} \sum_{s \in S(k)} x_s(t) - C_l(t) \right) dt = \sum_{k \in IS_l} (q_k(n+1) - q_k(n))^+ \quad (5)$$

其中, $q_k(n)$ 为链路 k 第 n 次迭代时的队列长度.

相应地,无线链路 l 的拥塞价格更新方式(公式(3))改变为

$$\lambda_l(n+1) = (\lambda_l(n) + \gamma \sum_{k \in IS_l} (q_k(n+1) - q_k(n))^+)^+ \quad (6)$$

为了使链路的队列长度能够稳定在一个较低的值,借鉴了文献[17]的方法,将公式(6)转化为

$$\lambda_l(n+1) = (\lambda_l(n) + \gamma[\alpha(\sum_{k \in IS_l} q_k(n) - q_l^{th}) + \sum_{k \in IS_l} (q_k(n+1) - q_k(n))^+])^+ \quad (7)$$

其中, q_l^{th} 是无线链路 l 的目标队列长度, α 是大于 0 的小常数.

考虑到数据包的到达可能是突发的,会造成即时队列长度的振荡,本文通过平均队列长度来代替即时队列长度,消除突发流量的影响.因此,采用平均队列长度而不是即时队列长度来更新链路拥塞价格.本文通过指数加权滑动平均方法计算平均队列长度 $q_l^{ave}(n)$ 来代替公式(7)中的 $q_l(n)$.

$$q_l^{ave}(n+1) = wq_l^{ave}(n) + (1-w)q_l(n+1) \quad (8)$$

其中, w 为滤波系数,在本文的仿真中取 0.85.

由上述讨论可见,链路 l 上的队列长度变化能够很好地体现在一定时间段内,链路 l 上受不同流在该链路上竞争的影响,因此,公式(7)能够很好地反映链路的拥塞程度变化.队列长度监控法使得链路的容量不需要事先已知,从而解决了 PCA 算法协议化面临的难点(3).

2.2 邻居集合近似法

定义. 邻居节点集合:不失一般性,本文定义节点 N 的邻居节点集合包含了节点 N 本身.节点 N 的邻居节点集合定义为 $Neighbor(N) = \{N' | (N, N') \in L\}$. 节点 N 的 k 跳邻居节点集合可以通过递归定义为

$$Neighbor^{k+1}(N) = \begin{cases} \{N' | (N, N') \in L\}, & k = 0 \\ Neighbor^1(Neighbor^k(N)) \cup Neighbor^k(N), & k > 0 \end{cases} \quad (9)$$

精确的链路干扰集可以通过 SIR 模型获得,但实际上很难实行.在实际中,如果一个链路的接收节点或发送节点在另一个链路的接收节点或发送节点的载波侦听距离或称干扰距离之内,就可以认为两个链路是相互干扰的^[8].由此可知,如果假定干扰距离和通信距离相等,则无线链路的干扰集就是以此链路的接收节点或发送节点为直接邻居节点作为其接收节点或发送节点的那些链路集合.在网络拓扑图中表现为两个无线链路彼此相距不超过两跳距离.其形式化定义如下:

$$IS_l = \left\{ k \left[\begin{array}{l} (T_k \in Neighbor(T_l)) \cup (T_k \in Neighbor(R_l)) \\ \cup (R_k \in Neighbor(T_l)) \cup (R_k \in Neighbor(R_l)) \end{array} \right] \right\} \quad (10)$$

如果干扰距离是通信距离的 K 倍,则无线链路干扰集的形式化定义如下:

$$IS_l = \left\{ k \left[\begin{array}{l} (T_k \in Neighbor^{K+1}(T_l)) \cup (T_k \in Neighbor^{K+1}(R_l)) \\ \cup (R_k \in Neighbor^{K+1}(T_l)) \cup (R_k \in Neighbor^{K+1}(R_l)) \end{array} \right] \right\} \quad (11)$$

因此,本文提出的拥塞控制方案对于干扰距离大于通信距离的情况仍然适用.

2.3 HELLO捎带信息法

需要指出的是,在 Ad Hoc 网络中,不存在物理可见的无线链路,更不存在求解算法的执行机构.在实际的协议实现中,要标示一条链路的拥塞程度,一般都是从这条链路的发送节点处着手,每条链路的拥塞程度值可以独立地在发送节点上计算.所以,中间节点需要为它的每个输出链路记录关于其(平均)队列长度的信息,但是没有必要为每个经过此链路的信息流记录相关信息.显然,中间节点的输出链路(等价于 1 跳邻居的个数)一般是较少的,而网络中的信息流个数可能会很大.因此,基于输出链路的信息维持方式能够节省中间节点资源,使得 PCA 能够适合在线实施,并且可以升级到大规模的网络,有良好的可扩展性能.

为计算每个输出链路的总队列长度信息和总拥塞价格,每个节点需要维护一张广播消息表来记录自己产生的广播消息和需要通过它转发的广播消息.广播消息表的每个表项包含如下几个域:

- (1) 发送节点 ID 号和接收节点 ID 号:这两个字段能够唯一地标志一条链路.
- (2) K 跳邻居节点 ID 号:在链路 l 发送节点或接收节点的 $K+1$ 跳邻居的所有输出链路中,只有其接收节点是链路 l 的 K 跳邻居的无线链路才会对链路 l 产生干扰.因此,节点在为其输出链路计算总拥塞价格时,只有在其接收节点和广播消息包中的“ K 跳邻居节点 ID 号”域一致时,才累加此广播消息包所指示的链路拥塞价格和队列长度.
- (3) 消息序列号:根据干扰集的形式化定义可知,如果干扰距离是通信距离的 K 倍,此信息要发表给链路发送节点的 $K+1$ 跳邻居集合和链路接收节点的 $K+1$ 跳邻居集合.由于发送节点的邻居集合和接收节点的邻居集合存在一些公共节点,为防止信息的重复接收和计算,采用了“消息序列号”,以标志消息的新旧程度.
- (4) TTL:为保证消息只在干扰集传输,添加了 TTL 域,并设置其初始值为 $K+1$.
- (5) 消息类型:“消息类型”位如果是 1,表示消息是自己产生的,否则就是接收到的广播消息.
- (6) 广播标志:“广播标志”位表示了消息被处理的情况,1 表示消息已经被广播过;0 表示还未被广播.

下面描述了关于广播消息产生、转发和处理的过程.

消息初始化阶段:发送节点首先在字段“发送节点 ID 号”和“接收节点 ID 号”填入相应的信息,把最近更新周期得到的队列长度信息和链路拥塞信息填入到相应字段,将 TTL 初始化为 $K+1$,把它放到自己的广播消息表中,置“广播标志”位为 0.

消息发送或转发阶段:当广播周期一到,将广播消息表中所有“广播标志”位为 0 的表项广播出去.

消息接收阶段:当节点收到一个广播消息包时,由于一个广播消息包很可能含有多个表项,对每个表项按如下顺序操作:

- (1) 首先根据字段“发送节点 ID 号”和“接收节点 ID 号”来判断自己的广播消息表是否已经存在和新接收到的表项有同样字段值的表项.如果表中没有这样的表项,则节点把新接收到的表项添加到表中;如果表中已经有相应表项,但新接收到的表项“消息序列号”的值较大,则更新表格中的对应表项;否则,丢弃此新接收到的表项,并且转步骤(4).
- (2) 根据“接收节点 ID 号”域,判断是否与自己的 ID 号一致,如果一致,则表明新接收到的表项还需要通过它转发到其 $K+1$ 跳邻居集合.因此,TTL 不需要减 1,将“广播标志”位置 0,等待将其广播出去,并且转步骤(4);否则,转步骤(3).
- (3) 如果检查到 TTL 域大于 1,则 TTL 域减 1,将“广播标志”位置 0,并且转步骤(4).如果检查到 TTL 域等于 1,则将自己的 ID 号填入“ K 跳邻居节点 ID 号”域中,同时,TTL 域减 1,将“广播标志”位置 0,并且转步骤(4).检查到 TTL 等于 0,将“广播标志”位置 1,不再广播此表项,并且转步骤(4).
- (4) 如果还有表项要处理,则转步骤(1);否则,转步骤(5).
- (5) 等待接收新的广播消息包.

消息处理过程:节点在为其输出链路计算总拥塞价格时,除了“ K 跳邻居节点 ID 号”域和输出链路的接收节点 ID 号不一致的表项以外,累加广播消息表中所有其他表项所指示的链路“纯”拥塞价格,这样就能得到此链路的总拥塞价格.用同样方式得到总队列长度信息.

目前,尚无专门为 Ad Hoc 网络设计的广播协议.为发送广播消息表中的表项而独立设计新形式的广播协议,会增加额外的通信开销.而在基于链路状态的路由协议 AODV 中,节点需要周期性地广播 HELLO 消息,从而可以准确地建立路由或获取路由信息.为减少新广播协议产生的额外通信开销,采用了 AODV 的 HELLO 消息“捎带(piggyback)”表项的广播形式.即节点间通过定期地交换 HELLO 消息的方式收集 $K+1$ 跳邻居节点的拥塞信息.而且,当网络拓扑结构发生变化时,也可以通过这种方式来更新先前所记载的邻居节点信息.所以,尽管 Ad Hoc 网络的拓扑结构是变化的,但是每个节点还是可以维护正确的邻居节点相关信息.频繁的 HELLO 消息广播能得到更及时、更精确的信息,但会耗费更多的网络资源.如何选择其广播频率存在折衷.在本文的 NS 仿真中,HELLO 间隔设置为 500ms.

3 自适应优化策略 AOS

应该指出,当且仅当 Ad Hoc 网络的参数和拓扑结构不变时,第 1 节中基本优化问题的解才有效,因此,上述分布式算法 PCA 原则上只能适用于足够小时间段上的拥塞控制问题.而实际上,Ad Hoc 网络参数和拓扑结构的时变性是不可忽略且无法预测的,因此,在应用上述算法求解实际的 Ad Hoc 网络拥塞控制问题时,必须考虑这种时变性.利用我们先前的工作^[14]设计了一种基于状态检测和滚动优化^[18]的自适应优化策略 AOS 来解决网络状态的不确定性时变问题.由于文献[14]已经对这种自适应优化策略进行了详细的描述,下面仅描述该策略的实施过程.

在源端和接收端开始建立连接时,源端用探测包机制来获取网络允许的数据包传输速率,将以此速率作为初始传输速率,即作为基本问题求解算法 PCA 的初始解,开始迭代执行该算法 PCA.为了提高系统的实时响应性,并考虑到迭代的收敛性(即在网络状态不变的情况下,每次迭代获得的解总是比上一次迭代获得的解更趋向于全局最优解)和滚动优化机制,每一轮迭代获得的中间解都立即被用作各源端的实际传输速率.

在迭代执行算法 PCA 的过程中,如果没有发生“路由中断”,则不停止地迭代执行基本问题求解算法 PCA.即使在迭代执行算法的过程中,网络状态已经发生了变化,也继续迭代执行该算法,因为 PCA 对网络状态变化具有一定的自适应能力(具体描述参见文献[11]).

在迭代执行算法的过程中,如果有“路由中断”的报告,则源端停止发送数据包,立即发送探测包,将探测包指示的速率作为初始解开始执行新一轮的迭代.当网络的拓扑结构突变导致原有路由中断时,采取的策略是:(1) 在路由恢复之前,源端停止发送任何数据包,因为重发的数据仍会因为无法路由而被抛弃,这会浪费网络资源;(2) 立即以一定的时间间隔发送探测包,去探测是否已经有路由可用.一旦有路由可用,探测包中携带的关于新网络状态所允许的数据包传输速率被源端设置为在新网络状态的初始传输速率,即作为公式(4)的新初始解,开始执行其后的迭代.这样就能快速、及时地探测新路由的连接可用带宽.

总之,基本优化问题的求解都是滚动进行的:每当网络的状态发生变化时,无论原优化问题的迭代求解是否已经收敛,都将启动新的求解过程,这使得优化过程对于 Ad Hoc 网络的变化具有良好的适应性和实时响应性.

4 仿真研究

本文用两种仿真平台来验证所提出算法的性能:一种是 MATLAB 仿真平台;另一种是 NS2 仿真平台. MATLAB 仿真平台用来验证 AOS 的自适应优化性能,NS2 仿真平台用来考察 PCA、PCA+AOS 在多种网络环境中的重要性能指标,包括吞吐量、丢包率、公平性等.

4.1 MATLAB 仿真实验验证

我们先前的工作^[11]利用 MATLAB 仿真实验验证了 PCA 有良好的收敛性,对网络状态的变化具有一定的自适应能力.为考察自适应优化策略 AOS 的有效性,本文利用 MATLAB 仿真实验来比较 PCA 的自适应能力与 AOS 的自适应优化性能.

仿真的 Ad Hoc 网络场景是:在 $[0,1] \times [0,1]$ 的单位平面内随机分布 50 个节点.这些节点是随机移动的:随机选择一个方向,并以在 $[0, V_{\max}]$ 单位距离(V_{\max} 是节点的最大允许速率)内随机选择一个速率向目的地移动.假定在节点之间的距离小于 0.25 时,存在无线链路可以直接通信.在仿真中,链路 l 干扰集是那些其中点和链路 l 的中点之间距离小于 0.25 的所有链路.虽然这种做法简化了干扰集的构成,但也能够充分考察算法的性能.随机选择 5 对节点分别作为 5 个信息流的源端和接收端,用最短路径方法选择源端和接收端之间的路由.按比例公平模型,将总效用函数设定为

$$U(\mathbf{x}) = \sum_{s=1}^5 \log x_s.$$

每次仿真的运行时间设定为 200 次迭代.在上述网络场景分别考察 PCA 的自适应能力与 AOS 的自适应优化性能,其结果见表 1.该结果是相同仿真条件下运行 10 次获得的平均值.

表 1 列出了在节点的最大允许速率 V_{\max} 分别取值为 0.005, 0.05 和 0.2 单位/迭代时, AOS 和 PCA 的最大

总效用函数值(maximum of total utility,简称 MTU).而且,这些结果与全局最优解(通过 MATLAB 优化工具箱提供的优化函数来集中式离线求解优化问题而求得的最优解)进行了比较,得到了相对误差的情况.从表 1 可知,AOS 在 V_{max} 取不同值时的最大总效用函数值都非常接近理论上的全局最优解,即使在强动态网络环境下(即 V_{max} 取值较大),例如 $V_{max}=0.2$ 单位距离/迭代,其相对误差也仅为 5.6722%.而 PCA 本身的自适应能力在强动态网络环境下(如 $V_{max}=0.2$ 单位距离/迭代),其最大总效用函数值和理论上的全局最优解有较大的误差,其相对误差达 20.3685%.并且, V_{max} 越大,其最大总效用函数值越小.这表明了在强动态网络环境下,PCA 不能以较快的速度渐近收敛于最优目标值.因此,AOS 的自适应优化性能要显著地强于 PCA 本身的自适应能力.由此可见,在强动态网络环境下,PCA 本身的自适应能力不足以解决网络状态的强时变性,必须采用 AOS 来解决网络状态的强时变性.

Table 1 Maximum of total utility and relative error

表 1 最大总效用函数值及相对误差

Solution	Network state		$V_{max}=0.005$		$V_{max}=0.05$		$V_{max}=0.2$	
	MTU	Relative error (%)	MTU	Relative error (%)	MTU	Relative error (%)	MTU	Relative error (%)
Optimum	23.295 1	0	24.654 7	0	25.367 2	0		
AOS	22.977 1	1.365 1	23.952 1	2.846 3	23.928 3	5.672 2		
Adaptation of PCA	22.973 7	1.379 2	22.772 8	7.632 9	20.195 7	20.368 5		

为了进一步考察这个性能比较结果是否存在普遍性,我们仿真了在 V_{max} 更多取值时,AOS 的自适应优化性能和 PCA 本身的自适应能力之间的性能比较,其结果如图 1 所示.从图 1 可知,AOS 和 PCA 中最大总效用函数值的相对误差都是随着 V_{max} 的增大而增大的.但是,PCA 中相对误差的增加幅度要远大于 AOS 中相对误差的增加幅度.而且, V_{max} 越大,PCA 中相对误差的增加幅度越明显于 AOS 中相对误差的增加幅度.从图 1 我们还可以看出,AOS 中相对误差随 V_{max} 增加而缓慢增加,即使在 $V_{max}=0.2$ 单位距离/迭代时,其相对误差也仅为 5.6722%.这充分说明了 AOS 对网络的动态变化有令人满意的自适应性.再一次表明了,在强动态网络环境下,采用 AOS 来解决网络状态的强时变性是提高网络性能的有效手段.

需要指出的是,选取不同的效用函数会导致不同的系统公平模型.资源的利用率和分配的公平性在选择效用函数时需要综合考虑,单纯以吞吐量最大、带宽利用率最高为分配目标会导致对某些用户数据流严重的不公平.因此,通常衡量数据流共享网络资源的公平性准则主要有两种:最大最小公平性和比例公平性.最大最小公平模型追求的是一种绝对意义上的公平,即在链路带宽可用范围内,各数据流尽可能均分链路带宽.但是,最大最小公平模型力求使低速率用户保持尽可能高的速率,这可能会导致在某些链路上带宽利用率较低.与之不同的按比例公平模型是根据业务流所占网络资源的多少进行带宽分配.就网络资源利用率和传输效率而言,比例公平性是更好的选择.因此,在本文的仿真实验中均采用了比例公平性准则,即将每个源端的效用函数设定为 $\log x_s$ ^[16].

4.2 NS仿真实验证

本节用 NS-2 仿真平台来比较 PCA,PCA+AOS,TCP-Reno,ATCP 以及 ATP 的性能.仿真的环境是:各节点使用全向天线,无线传播模型采用 TwoWayGround 模型.信道传输速率为 2Mbps.MAC 协议按文中修改后的 IEEE 802.11 协议,有效传输距离为 250m,侦听距离为 550m;路由采用按文中修改后的 AODV 协议;每个节点的缓存大小为 50.数据包大小等于 1 000byte,传输的业务为 FTP 数据.所有仿真实验的运行时间都为 200s.分别考察了对网络的拓扑模式,静态网络采用链式拓扑,动态网络采用随机拓扑.

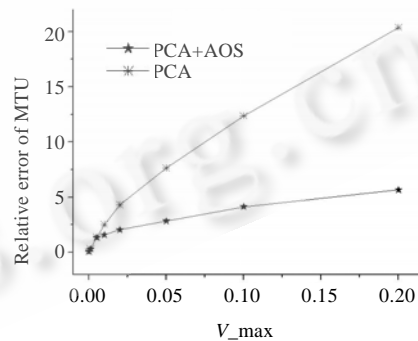


Fig.1 Maximum of total utility and relative error under the different speed of node mobility

图 1 在不同节点移动速度下,最大总效用函数值的相对误差

网络拓扑模式包含了静态拓扑和动态拓扑两种类型.首先,我们研究了 PCA 在静态拓扑中的性能,包括链式拓扑、交叉拓扑和网格拓扑 3 种情况.其次,我们研究了 PCA+AOS 在动态拓扑中的性能,考察了 3 种移动模型,即 RWP(random way point)模型、FW(freeway mobility)模型和 MH(Manhattan mobility)模型^[19].

4.2.1 静态网络性能分析

研究静态拓扑的网络性能能够得到一些本质性的结论.对于链式拓扑,将考察协议 PCA 在平均吞吐量、吞吐量稳定性、丢包率方面的性能.对于交叉拓扑和网格拓扑,将考察协议 PCA 在公平性方面的性能.

对于如图 2 所示的链式拓扑,任何两个相邻节点间的距离为 200m,使得每个节点只能与它的邻居节点进行通信.网络中只有一个连接,该连接的源端为节点 1,接收端根据连接跳数的不同而不同.从不同跳数下的平均吞吐量、吞吐量稳定性、丢包率来考察 PCA 和 TCP-Reno,ATCP,ATP 的性能.



Fig.2 Chain topology
图 2 链式拓扑

(1) 吞吐量

1~19 跳的网络在 200s 内的平均吞吐量对比结果如图 3 所示.从图 3 的仿真结果可以看出,PCA 和 TCP-Reno,ATCP,ATP 的吞吐量随着跳数的增加而显著减小.在 4 跳之后,吞吐量变化得非常缓慢,逐渐趋于稳定.与 TCP-Reno 和 ATCP 相比,在各个跳数下,PCA 能够获得较大的吞吐量增益,而且跳数越高,提高的幅度越大.原因是此时相对于 TCP 源端(节点 1)来说,网络中存在隐藏节点.但是,IEEE 802.11 MAC 协议不能解决由隐藏节点带来的共享信道竞争问题,同时,因为 TCP-Reno 和 ATCP 的接收端不能设置正确的最大窗口大小,使得 TCP 的拥塞控制机制也不能解决这个问题.但是,PCA 通过显式拥塞度量并在干扰集之间相互交换拥塞信息,能够很好地缓解信道竞争的程度,提高了吞吐量.ATP 通过显式的速率指示也能缓解信道竞争程度,因此能够取得和 PCA 差不多的吞吐量.

(2) 稳定性

以 14 跳为例分析吞吐量的稳定性,考察网络的吞吐量随时间变化的趋势,结果如图 4 所示.TCP 传输出现了 49 个零点,而 PCA 中只出现 10 个零点.从曲线上可以很直观地看出,PCA 的吞吐量稳定性有了明显改善.另外,在同样的仿真环境下,ATCP 传输出现了 29 个零点,ATP 传输出现了 12 个零点,它们的性能介于 PCA 和 TCP 之间.

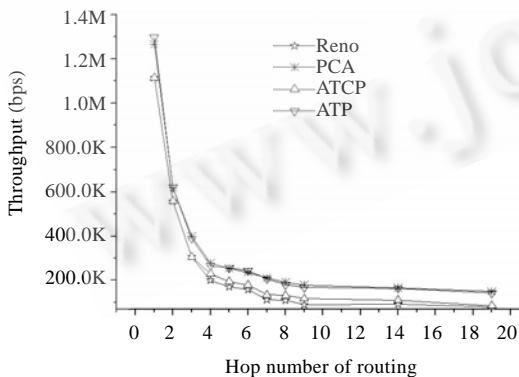


Fig.3 Throughput in the different number of hops
图 3 不同跳数下的吞吐量比较

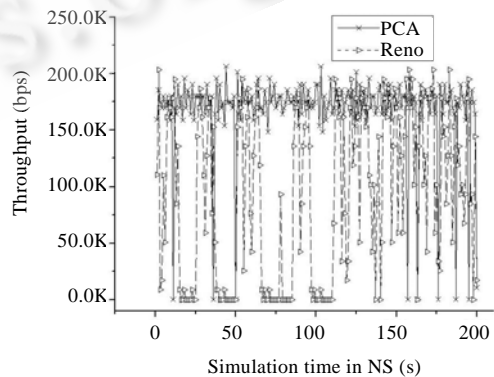


Fig.4 Stability of throughput in 14 hops
图 4 吞吐量稳定性(14 跳)

同时,我们考察了网络吞吐量的稳定性随路由长度变化的趋势,图 5 显示了 PCA 和 TCP-Reno,ATCP,ATP 在 1~19 的跳数下 200s 的仿真时间内吞吐量等于 0 的次数(秒数).可见,当跳数减小时,吞吐量的稳定性得到改善,

同时,PCA 的稳定性大大好于 TCP-Reno 及其 ATCP,也略好于 ATP.这也是由于 PCA 采用了显式拥塞价格并在干扰集之间相互交换拥塞信息所致.ATP 通过显式的速率指示也能部分地改进吞吐量的稳定性.

(3) 丢包率

由图 6 可见,当跳数增加到 5 跳以上时,TCP-Reno 的数据包丢失率会高达 15%~20%,也就是说,平均每发送 5~6 个数据包就会丢失 1 个,如此大的丢失率是不能容忍的.尽管 ATCP 能在一定程度上缓解这种缺陷,但是没有从根本上解决,其原因是它们采用的都是基于窗口的传输策略,导致了严重的“突发性”数据包传输,从而增大了数据包丢失的概率.ATP 和 PCA 都采用了基于速率的传输策略,能够避免“突发性”数据包传输,因此,在静态多跳网络中极其有效地降低了数据包的丢失率.

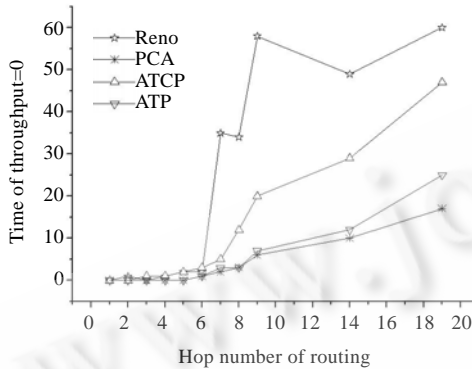


Fig.5 Time of throughput=0 in different number of hops

图 5 不同跳数时,吞吐量等于 0 的秒数

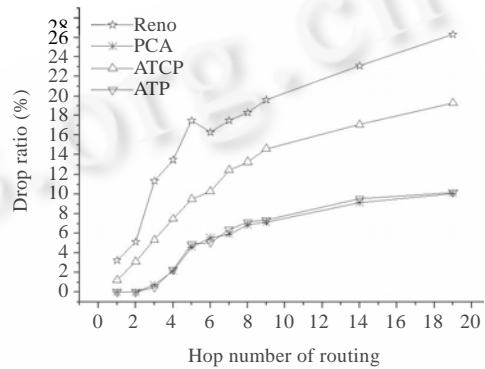


Fig.6 Packet drop ratio

图 6 数据包丢失率

(4) 公平性

对于如图 7 所示的网络拓扑,有 6 个 FTP 在传输信息.分别考察了 PCA,TCP-Reno,ATCP 和 ATP 在这样的场景下的信息流的公平性,其结果如图 8 所示.从图 8 可以看出,PCA 在公平性上要显著好于 TCP-Reno 及其 ATCP,其主要原因是各个节点通过下行链路的拥塞信息进行协调,当下行链路出现拥塞时,其拥塞的情况马上会广播给它的干扰集,让其干扰集不要过分地“霸占”它的信道资源.此外,PCA 利用链路的拥塞价格来提高 MAC 层接入的公平性,也在一定程度上改进了端到端的信息流之间的公平性.

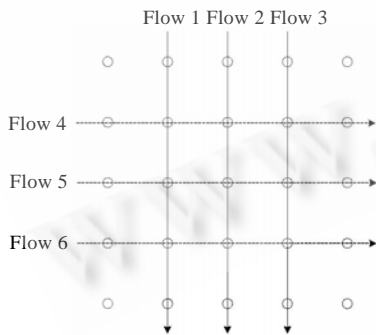


Fig.7 Grid topology

图 7 网络拓扑

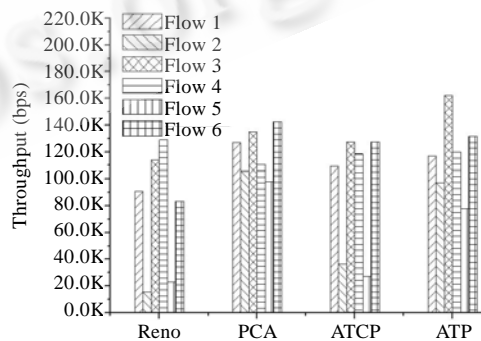


Fig.8 Fairness among end-to-end flows

图 8 端到端信息流的公平性

4.2.2 动态网络性能分析

在本节中,采用了 3 种动态网络模型,包括 RWP 模型、FW 模型和 MH 模型.在这些移动模型中,主要考察 PCA+AOS,TCP-Reno,ATCP 和 ATP 在这些移动场景下的吞吐量性能.分析这些协议在网络动态变化的情况下,

尤其是在强动态网络环境下其性能下降情况.在仿真中,节点总数设置为 50 个,各节点在 2000m×2000m 的平坦区域内按照这 3 种动态网络模型指定的方式运动.在它们之间形成了 20 个 FTP 连接.每个 FTP 连接的源端、接收端和所经的路由都是随机选择的.由于仿真场景具有随机性,所以每次仿真都会重复进行 10 次,然后对结果求平均值.

1. RWP 模型

RWP 模型是研究中最常用的移动模型.其实现方式是:在每个时刻,节点随机选择一个目的地,并以在 [0,V_max](V_max 是节点的最大允许速率)随机选择的一个速率向目的地移动.到达目的地后,节点在该目的地停顿时间 pause time 指定的时间后,又开始随机选择新的目的地,再次重复该过程,直到仿真结束.在仿真中,节点以如下几种最大移动速度 V_max 在网络中漫游:0m/s,1m/s,5m/s,10m/s,15m/s,20m/s,25m/s,30m/s 和 35m/s.通过设置停顿时间为 0s 来体现网络拓扑结构的不停变化.

图 9 显示了在不同的移动速度下整个网络获得的吞吐量(吞吐量均为 10 个仿真场景的平均值,下同).由此可知,PCA+AOS 的吞吐量在大部分的移动环境下都是最高的,除了 5m/s 时稍微比 ATP 差些以外.与 TCP-Reno 相比,PCA+AOS 能够获得 52.8%的平均吞吐量增益.图 9 还表现出了一个奇怪的现象,即各协议在网络节点的移动从显示 0m/s 到 1m/s 和 5m/s 时,吞吐量是明显下降的,可是在 10m/s 以上的速率时,吞吐量反而上升了.对这种离奇现象的解释是,节点快速移动使得节点可以选择的路由大为增加,随时可以无缝切换到新路由,避免了某些瓶颈路由的限制,从而使得吞吐量能够离奇地增加.这种现象也在 MH 模型中有所体现.从图 9 中可以看出,PCA+AOS 在不同的移动速度下,其吞吐量的变化趋势相对于 TCP-Reno,ATCP 和 ATP 要平缓得多,其原因是 PCA+AOS 采用了自适应优化策略.再一次表明了强动态网络环境下,采用 AOS 来解决网络状态的强时变性是提高网络性能的有效手段.

2. FW 模型

FW 模型模拟了高速公路上的车辆移动行为,这在交通流量状况检测、车辆的运动轨迹追踪很有用处.FW 模型和 RWP 模型的差别在于:(1) 每个节点的运动被限制在高速公路的车道上;(2) 某时刻的节点运动速度与前一时刻的运动速度相关,即 $v_i(t+1)=v_i(t)+random \times a_i(t)$.其中, $a_i(t)$ 是节点的运动加速度.

本文在仿真时,关键参数取值如下:

- (1) 确定地图,即确定高速公路和车道情况,在仿真中采用了 1 条高速公路,4 个车道.
- (2) 确定节点个数,在仿真中采用了 50.
- (3) 确定加速度的取值,在仿真中采用了从 0~1,间隔为 0.1 的 11 种情况.

图 10 显示了在不同加速度下整个网络获得的吞吐量.由此可知,PCA+AOS 的吞吐量在大部分的移动环境下都是最高的.与 TCP-Reno 相比,PCA+AOS 能够获得 37.1%的平均吞吐量增益.

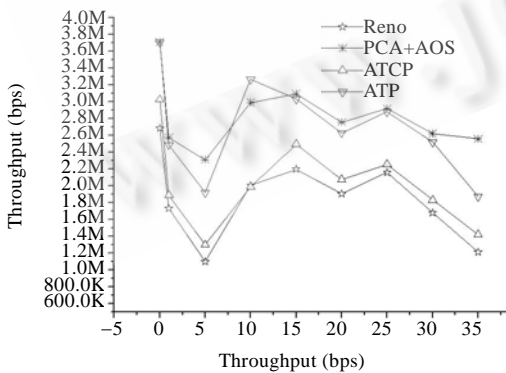


Fig.9 Total throughput under the different speed of node mobility

图 9 不同移动速度下的整个网络吞吐量

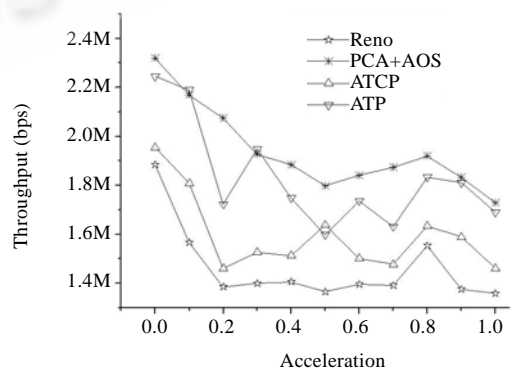


Fig.10 Total throughput under different acceleration

图 10 不同加速度下的整个网络吞吐量

3. MH 模型

MH 模型用来模拟移动节点在街道上的运动模式.该模式中的地图是由若干条纵横交错的街道构成,移动节点只能在这些纵横交错的街道网格中运动,在每个交叉十字路口,移动节点以一定的概率向左、向右或向前运动.本文在仿真时,关键参数取值如下:

- (1) 确定地图,即确定纵横交错的街道情况,在仿真中采用了三横三纵的地图.
- (2) 确定节点个数,在仿真中采用了 50.
- (3) 确定节点的最大最小允许的运动速率,在仿真中采用的最小允许运动速率均为 0,最大允许的运动速率分别为 0m/s,1m/s,5m/s,10m/s,15m/s,20m/s,25m/s,30m/s 和 35m/s.
- (4) 确定加速度的取值,在仿真中均采用了 0.

图 11 显示了在不同的移动速度下整个网络获得的吞吐量.由此可知,PCA+AOS 的吞吐量在大部分的移动环境下都是最高的,除了 10m/s 时稍微比 ATP 差些以外.与 TCP-Reno 相比,PCA+AOS 能够获得 35.5% 的平均吞吐量增益.

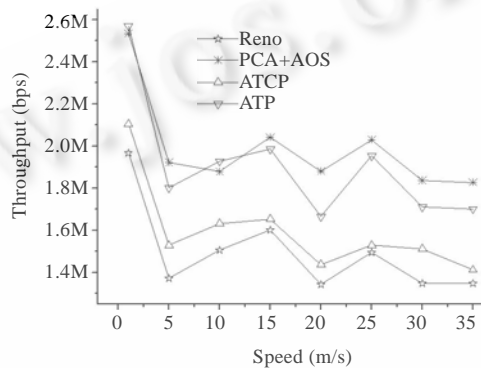


Fig.11 Total throughput under the different speed of node mobility

图 11 不同移动速度下的整个网络吞吐量

4.2.3 仿真结果讨论

从上面仿真结果可知,在静态网络环境下,PCA 在吞吐量、吞吐量稳定性、丢包率、公平性方面都比 TCP, ATCP 和 ATP 有所改进.在动态网络环境下,尤其是在强动态网络环境下,PCA+AOS 在吞吐量方面优于 TCP, ATCP 和 ATP,充分说明了 AOS 对网络的动态变化有着令人满意的自适应性.

需要指出的是,我们也考察了 PCA+AOS 在静态网络环境下的性能,发现 PCA+AOS 的吞吐量性能要略逊于 PCA,原因是实施 AOS 带来了额外开销.实际上,PCA 自身的自适应性能够解决静态网络环境下的微小变化(主要是无线链路的时变性).因此,AOS 在静态网络环境下没有必要采用.

我们也考察了 PCA 在动态网络环境下的性能,发现 PCA+AOS 的性能要优于 PCA.尤其是在强动态网络环境下,PCA+AOS 的性能要显著地优于 PCA 的性能.因此,AOS 在动态网络环境下是非常有必要采用的.它在很大程度上解决了网络状态的强时变性带来的问题,有效地提高了动态网络环境下的网络性能.

同时还需要指出的是,不同应用场合的 Ad Hoc 网络对传输协议(拥塞控制)的性能要求不同.本文仅按照研究中最常采用的比例公平模型^[16],将每个源端的效用函数设定为 $U_s(x_s)=\log x_s$,并基于多种网络场景进行了仿真实验,验证了 PCA 和 PCA+AOS 在重要网络性能指标上取得了明显改进.但是,本文没有考察 PCA 和 PCA+AOS 在对传输协议不同设计要求的其他典型应用中的性能,这是值得进一步研究的内容.尽管本文没有在这方面作进一步的深入研究,但是我们认为,PCA 和 PCA+AOS 也能作为其他典型应用环境下的传输协议(拥塞控制),而且也能取得网络性能的改善.理由有如下两点:

- (1) 在构成拥塞控制问题(2)的目标函数中,效用函数是一般形式,即 $U_s(\cdot)$.而在构成本文设计的 PCA 算法之一的公式(4)中,效用函数导数的逆也是一般形式,即 $(U'_s)^{-1}(\cdot)$.由此可见,本文设计的 PCA 算法适用于任何满足

效用函数条件的任何函数形式。Shenker 等人^[20]已经证明,使凹效用函数最大化等价于实现某些全局范围内的公平。可见,一个凹效用函数对应于一种公平模型,能够决定网络资源分配的公平性准则。因此在 Ad Hoc 网络中,任何公平模型的建立都等价于确定的效用函数。由此可见,本文设计的 PCA 和 PCA+AOS 能够适用于其他典型应用下的传输协议设计。

(2) 本文设计的 PCA 和 PCA+AOS 是从优化论的角度出发的,运用优化理论方法为指导来设计,其优化解使得网络处于源端的效用尽量大、中间节点的拥塞尽量小的最佳状态,从而实现网络资源的合理分配,避免和控制网络拥塞现象。因此我们有理由推断,用这种优化思想设计出的拥塞控制算法 PCA 和 PCA+AOS 应用在其他设计要求的传输协议中时也能提高网络性能。

5 结 论

Ad Hoc 网络的节点移动和无线多跳连接的特点导致了网络状态不断变化,信息流竞争与 Internet 网络不同。本文首先从信息流竞争新特点出发,提出了一种适合在小时间段内网络状态不变的分布式求解算法 PCA;详细描述了 PCA 的协议化方案;针对网络状态的不确定时变性,设计了一种基于状态检测和滚动优化的自适应优化策略 AOS。MATLAB 仿真实验考察了 AOS 的自适应性,仿真结果表明,AOS 策略比 PCA 对网络状态的变化具有更好的自适应性。NS 仿真实验结果表明,在几乎所有的仿真场景和移动环境下,PCA 和 PCA+AOS 取得的性能比 TCP、ATCP 和 ATP 都要好。需要指出的是,AOS 在静态网络环境下是没有必要采用的。但是,AOS 在动态网络环境下非常有必要采用,它有效地提高了强动态网络环境下的网络性能。

为了进一步优化 PCA 和 PCA+AOS 的性能,如下几个方面的内容值得进一步深入研究:

- (1) 仿真分析并从理论上证明在信息反馈有时延时 PCA 算法收敛性;
- (2) 进一步优化 AOS 策略;
- (3) 考察 PCA 和 PCA+AOS 在对传输协议不同设计要求的其他典型应用中的性能;
- (4) 考察 HELLO 间隔的取值对 PCA 收敛性的影响;
- (5) 考察干扰距离和通信距离的比值对 PCA 收敛性的影响。

致谢 感谢审稿人对本文提出了许多宝贵的意见和建议。

References:

- [1] Chlamtac I, Conti M, Liu J. Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Networks*, 2003,1(1):13-64.
- [2] Xu W, Wu T. TCP issues in mobile ad hoc networks: Challenges and solutions. *Journal of Computer Science and Technology*, 2006, 21(1):72-81.
- [3] Chiang M. Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control. *IEEE Journal Selected Area Communications*, 2005,23(1):104-116.
- [4] Chen L, Low S, Doyle J. Joint congestion control and media access control design for ad hoc wireless networks. In: *Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2005)*. Helsinki: IEEE Press, 2005. 2212-2222.
- [5] Wang X, Kar K. Cross-Layer rate optimization for proportional fairness in multihop wireless networks with random access. *IEEE Journal on Selected Areas in Communications*, 2006,24(8):1548-1559.
- [6] Chen L, Low S, Doyle J. Cross-Layer congestion control, routing and scheduling design in ad hoc wireless networks. In: *Proc. of the 25th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2006)*. Helsinki: IEEE Press, 2006. 1-13.
- [7] Eryilmaz A, Srikant R. Joint congestion control, routing, and MAC for stability and fairness in wireless networks. *IEEE Journal on Selected Areas in Communications*, 2006,24(3):1514-1524.
- [8] Xue Y, Li B, Nahrstedt K. Optimal resource allocation in wireless ad hoc networks: A price-based approach. *IEEE Trans. on Mobile Computing*, 2006,5(4):347-364.

- [9] Lin X, Shroff NB, Srikant R. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 2006,24(3):1452–1463.
- [10] Chiang M, Low SH, Calderbank AR, Doyle JC. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. of the IEEE*, 2007,95(1):255–312.
- [11] Bao X, Xu W, Wu T. Optimal congestion control algorithm for ad hoc networks. *Journal of University of Electronic Science and Technology of China*, 2007,36(2):250–253 (in Chinese with English abstract).
- [12] Liu J, Singh S. ATCP: TCP for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 2001,19(7):1300–1315.
- [13] Sundaresan K, Anantharaman V, Hung YH, Sivakumar AR. ATP: A reliable transport protocol for ad hoc networks. *IEEE Trans. on Mobile Computing*, 2005,4(6):588–603.
- [14] Xu WQ, Wu TJ, Wang YM, Zhang YH, Chen JM. Adaptive congestion control strategy for multirate multicast sessions in ad hoc networks. *Journal of Software*, 2008,19(3):769–778 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/769.htm>
- [15] Low SH, Lapsley DE. Optimal flow control I: Basic algorithm and convergence. *IEEE/ACM Trans. on networking*, 1999,7(4):861–874.
- [16] Kelly FP, Maulloo A, Tan D. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 1998,49(3):237–252.
- [17] Athuraliya S, Li VH, Low SH, Yin Q. REM: Active queue management. *IEEE Trans. on Networking*, 2001,15(3):48–53.
- [18] Rawlings JB. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 2000,20(3):38–52.
- [19] Bai F, Sadagopan N, Helmy A. The important framework for analyzing the impact of mobility on performance of routing for ad hoc networks. *Ad Hoc Networks*, 2003,1(4):383–403.
- [20] Shenker S. Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications*, 1995,13(7):1176–1188.

附中中文参考文献:

- [11] 包晓安,徐伟强,吴铁军.一种用于 Ad hoc 网络的优化拥塞控制算法. *电子科技大学学报*,2007,36(2):250–253.
- [14] 徐伟强,吴铁军,汪亚明,张云华,陈积明.用于 Ad Hoc 网络的自适应多速率多播拥塞控制策略. *软件学报*,2008,19(3):769–778. <http://www.jos.org.cn/1000-9825/19/769.htm>



徐伟强(1975—),男,浙江杭州人,博士,副教授.主要研究领域为 Ad Hoc 网络,无线传感器网络,拥塞控制,网络资源优化分配,跨层优化.



张云华(1965—),男,博士,副研究员,主要研究领域为计算机网络,软件工程,计算机控制.



吴铁军(1950—),男,博士,教授,博士生导师,主要研究领域为大系统智能控制与优化,人机协同控制与决策,混杂系统智能控制,多智能体分散控制.



陈积明(1978—),男,博士,副教授,主要研究领域为无线传感器/执行器网络,弱硬实时调度.



汪亚明(1972—),男,博士,教授,博士生导师,主要研究领域为计算机图像,运动估计,计算机网络,智能算法.