

## Abstract

In the Internet, TCP is the primary transport protocol. It performs well in traditional networks where congestion is the primary cause of packets loss. However, wireless networks also suffer from significant losses due to bit errors, TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in the degraded end-to-end performance unnecessarily.

Firstly three wireless networks-satellite WAN、802.11 LAN and 802.15.4 PAN and the challenges TCP ment are introduced. Then the dissertation reviews acknowledgement mechanism and congestion avoidance mechanism in TCP protocol, introduces three classical version of TCP- TCP Tahoe、TCP Reno and TCP Vegas. Following the dissertation studies some ideas which improve the performace of TCP protocol in random error wireless links, an improved congestion control mechanism that is able to deal with random loss effectively. The improved mechanism differs from the conventional TCP in a major way: it monitors the congestion level using an subsection estimation algorithm and uses the knowledge to refine the congestion control algorithm of TCP attempting to stay longer in an operating region in which the bandwidth is fully utilized.

Secondly, network tested experiments have been conducted and they consistently show that improved mechanism can achieve significant performance improvement over Reno in many different scenarios. Throughput improvement of as high as 100% is typical in the satellite WAN seting with random loss at  $10^{-4}$ , throughput improvement of as high as 20% is typical in the 802.11 LAN setting with random loss at  $10^{-2}$ , throughput improvement of as high as 5% is typical in the 802.15.4 PAN setting with random loss at  $10^{-2}$ . The performance of improved mechanism has direct ratio with latance and bandwidth in wireless networks. It can coexist with concurrent TCP protocol and UDP protocol connections without degrading the performance of TCP or UDP connections. And it can be deployed easily with only minor modifications of the sender-side algorithm.

At last, upon mobile Ad hoc network, one transport protocol has been picked out. The component of protocol is analyzed, the application of this protocol in internet is discussed.

**Keywords:** wireless network TCP protocol congestion control AHTP

# 第1章 绪论

## 1.1 计算机网络与移动计算机网络

计算机网络<sup>[1]</sup>是通过各种计算通信设备和传输介质将处于不同位置的多台具有独立功能的计算机连接起来，并在相应的网络管理软件的管理下实现多台计算机之间信息传递和资源共享的系统。它涉及到通信与计算机两个领域，一方面，通信网络为不同计算机系统之间的数据传递和交换提供了必要的于段，另一方面，数字技术的发展融入到通信技术中，又提高了通信网络的性能。相对于电信通信的电路交换方式，计算机网络引入了分组交换的概念，使得通信链路的利用率大大提高，并成为网络发展的主导方向。

根据网络的规模和覆盖范围，计算机网络通常可以分为三大类：广域网、城域网和局域网<sup>[2]</sup>。

广域网（Wide Area Network, WAN）可以跨越辽阔的地理区域进行长距离的数据、语音、多媒体传输，它所包含的地理范围可以是一个国家、一个大陆甚至整个世界。

城域网（Metropolitan Area Network, MAN）是为了将网络覆盖范围延伸整个城市，它可以是一个有线电视网的单一网络，也可以是众多局域网连接而成的一个更大的网络。

局域网（Local Area Network, LAN）是用于连接一个办公室、一幢建筑物或者一所校园的专用网络。局域网常用于连接个人计算机和工作站，以便共享资源和交换信息。

局域网的配置安装灵活方便和传输速率高等特点使其成为应用最广泛的一种网络类型。局域网可以使用多种传输介质，比如双绞线、同轴电缆和光纤等。以太网是一种典型的局域网，传输速率达到 10 Mbit/s-1000Mbit/s，传输延迟一般为几十毫秒，并且误码率一般可达到  $10^{-6}$  以下，最新的万兆以太网<sup>[3]</sup>已经能够提供 10000Mbit/s（即 10Gb/s）的数据传输速率。

局域网可以有多种不同的拓扑结构，比较典型的有三种：总线型、环型和星型结构，如图 1-1 所示。

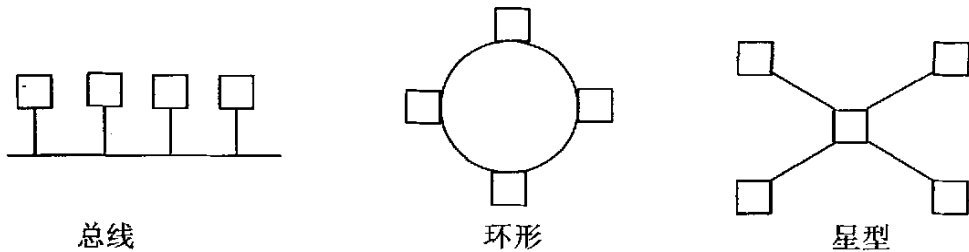


图 1-1 局域网的拓扑结构

计算机网络技术的发展，极大的满足了用户的需求，但由于办公室缆线泛滥和无法移动等缺点的存在，人们开始考虑计算机网络的移动性扩展，借助先进的无线通信技术，建立支持移动计算能力的移动计算机网络<sup>[4]</sup>。

所谓移动计算机网络，指的是支持主机移动性的网络。它具有如下的特征：

- 1、移动主机可以在网络中自由移动且对网络层及以上应用透明；
- 2、网络是基于无线 IP 技术的。

移动计算机网络通常由无线网络和固定网络组成。按照无线网络的规模和覆盖范围可以分为无线广域网、无线局域网和无线个域网等几类。

无线广域网直接利用现有的模拟或数字蜂窝电话网来传送电路交换计算机数据，也可以在现有的蜂窝电话网上建立数字分组交换网络实现分组数据传输，还有使用卫星链路传送分组数据。无线广域网包括电路交换无线网、分组交换无线网和卫星通信网三种类型。

现有模拟或数字蜂窝电话网都提供传送计算机数据的电路交换移动数据<sup>[5]</sup>业务，比如 GSM、PHS 网络。计算机通过调制解调器与手机相连接，手机把调制后的数据发往本小区的基站，再由移动交换中心访问网中的其它计算机。这种方法的特点是网络覆盖广，缺点是传输速率比较低，对 GSM 及 PHS 系统而言，一般只能提供 9.6kbps 的接入速率。

在现有的蜂窝电话网基础设施上叠加专用的数据分组交换<sup>[6]</sup>网可以实现分组数据的无线传输。通用分组无线业务（GPRS）以及进化型高速数据传送（EDGE）是叠加在 GSM 网络上的分组数据网络。GPRS 与 EDGE 利用现有的 GSM 网络中的移动通信设备，增加一些硬件设备和升级软件，形成一个新的网络逻辑实体，它以分组交换技术为基础，采用 IP 数据网络协议，使现有 GSM 网络的数据业务最高数据接入速率可达 171.2kbps 以上。

卫星通信系统<sup>[7]</sup>包括空间部分和地面部分。地面部分包括关口站、网络控制

中心、操作控制中心。其中网络控制中心和操作控制中心负责整个网络资源的管理、卫星的操作、轨道的控制。关口站作为各种外部网络和卫星通信网络的网络接口。空间部分一般为静止或非静止轨道卫星。宽带卫星通信系统<sup>[9]</sup>多数采用 ka 波段,甚至更高频段,ka 波段增大传输频谱,支持小型地面终端的宽带数据传输,基于标准的开放技术,成本低廉。卫星通信网能提供 2M 以上的带宽,延时一般在 250ms 左右(静止轨道卫星)。目前有多种系统提供接入服务,如休斯顿公司的 SpaceWay 系统以及 Motorola 公司的 Celestri 系统等<sup>[9]</sup>。

无线广域网的数据传输速率一般在 2 兆以下,网络中每个小区的半径较大。由于数据速率低,他们的应用也局限于电子邮件、数据查询及数据链路备份等方面。无线局域网<sup>[10]</sup>则实现一种高速率、能够支持实时多媒体业务的移动计算机网络,而且这种网络可由局部区域逐步扩展,形成以有线网络为骨干网的复合网络。

便携式消费类电子产品(比如家庭中使用的数码相机、MP3 播放器、PDA、数字电视机、打印机等设备)的飞速发展产生了设备之间互联实现信息共享的需要。这些设备之间的通信必须通过无线信道来完成,WPAN(Wireless Personal Area Network,无线个域网)<sup>[11]</sup>正适合了这种需求。无线个域网和无线局域网的主要区别在于其应用的范围不同,WPAN 主要在 POS(Personal Operating Space,个人操作区域)内工作,传输距离一般小于 10 米。

## 1.2 无线网络中 TCP 性能问题

传输控制协议<sup>[12]</sup>(Transmission Control Protocol, TCP)是因特网上最通用的传输层协议,它提供可靠的端到端的流服务,大多数的 Internet 应用如 WWW、FTP、TELNET 和 EMAIL 等都使用了 TCP 提供的这种可靠服务。因而用户所能感知到的网络服务的性能主要取决于 TCP 的性能,TCP 的性能也就成为设计和构建高性能 TCP/IP 网络的关键。

由于传统的有线网络是高速宽带网络,网络中数据包的丢失主要由于网络拥塞引起的,TCP 协议专门对这种情况进行了优化,假设所有的数据包丢失是由网络拥塞造成的,进行相应的拥塞控制<sup>[13]</sup>来降低网络数据传输速率,从而取得了较好的性能。

在网络包括无线链路的情况下,由于无线路由切换和无线链路特性的改变而导致数据包的丢失,传统的 TCP 协议对此不能进行有效的处理,导致网络吞吐量

和服务质量的下降。在无线网络中使用传统 TCP 协议提供数据包传送服务时，由于网络环境不同于固定网络，除了网络拥塞以外还有其他因素可能导致数据包的丢失，主要包括以下两类：

### 1、路由变化导致数据包丢失或者失序到达

路由中断和路由优化都可能导致多个数据包的丢失或者失序到达。在高移动性的无线网络环境中，移动节点经常会改变位置，为了保证可以使用最优的路由，路由协议需要频繁的更新路由，导致数据包的丢失或者失序到达。当路由协议检测到路由失效后，在寻找到新的路由之前的时间内可能导致多个数据包的丢失。由于传统的 TCP 协议缺乏识别数据包丢失原因的能力，导致不必要的调用拥塞控制和避免机制，进而引起 TCP 吞吐量下降和数据传输时延增加。

### 2、误码特性导致的数据包丢失

由于无线链路的特性引起数据包的随机误码和突发错误。随机误码可能导致拥塞窗口内的单个或者多个数据包的丢失，突发错误可能导致多个数据包的丢失。单个数据包的丢失引起快速重传和快速恢复<sup>[14]</sup>，多个数据包的丢失导致重传超时，因而 TCP 会调用慢启动机制，这两种情况都会引起 TCP 连接的吞吐量的下降。无线网络的介质是由多个移动台共享的，它们通过随机争用的方式发起数据传输，隐终端问题会导致接收信号的混淆，使得移动台不能正常通信。

## 1.3 无线 TCP 协议研究现状

在现有的各种 TCP 性能改善方案中，按照不同的协议层可划分为三种：TCP 本身的改进、链路层协议的改进和应用层协议的改进方法。

目前大多数的研究工作集中与对 TCP 本身的改进和 TCP 与链路层相结合的改进方法。这些方法主要是基于以下两个思路：发送端尽可能屏蔽与拥塞无关的数据包丢失，非拥塞的丢包由接受方和中间节点进行处理，发送端的 TCP 协议不需修改；发送方觉察，让 TCP 发送方区分数据包丢失的原因——拥塞或者非拥塞，当出现非拥塞的数据包丢失时，发送端不启动拥塞控制算法而是采用新的处理机制，发送端的 TCP 需作修改。

当前提出的传输层改进方法分五类。纯端到端方案：直接修改 TCP 连接两端的协议使之更好的应用于无线环境。TCP 分段连接方案：建议根据有线和无线链路的不同特征而采取不同的发送策略，将移动主机和通信对端间的 TCP 连接在中

间节点分为两段，有线连接段使用基本 TCP 协议，无线连接段使用适应无线环境的改进协议。TCP 缓存方案：保持了端到端的语义，在代理中采用了软状态方式，软状态的丢失虽会影响性能，但不会阻止 TCP 端到端的数据传送。在中间节点增加探察代理来缓存和探测发送移动主机的 TCP 数据段和返回确认，根据情况决定是否重发缓存中的数据段，它提供可选择性的重传，使中间节点可教准确的区分拥塞和无线传输产生的错误。交叉层解决方案：由链路层活网络层将链路环境状态反馈到 TCP 层，TCP 依此采取相应的方法来处理数据段丢失。链路层解决方案：链路层协议是克服链路高误码率特性的常用方案，采用本地丢失恢复的思想，使无线链路向上呈现非常低的丢失率，从而对发送端完全透明。

应用层的改进，应用程序的开发者可以通过许多方法（如有效的对数据进行缓存和分配）来改进应用程序在移动和无线环境中的性能，其重点是对与 TCP 和整体性能直接相关的应用层协议的改进，如建立一个应用层协议，使得所有的数据传送可以共用一个 TCP 连接，而不是多个连接。

## 1.4 论文的主要安排

本论文着重研究无线网络的 TCP 性能。在论文的前两章介绍了无线网络的技术基础--无线广域网、无线局域网、无线个人网。第一章首先介绍了计算机网络、移动计算网络和无线网络的概念，并引出常规路由协议和 TCP 协议在无线网络中出现的问题以及目前的研究现状；第二章详细介绍了无线网络中的卫星广域网、802.11 无线局域网与 802.15.4 无线个域网；第三章阐述了 TCP 的基本理论，重点是其拥塞控制机制，并讨论了无线网络中影响 TCP 协议性能的主要因素，总结了针对无线网络的一些改进方案；第四章提出基于拥塞控制提高无线网络中 TCP 性能的改进方案，并对卫星广域网、802.11 无线局域网与 802.15.4 无线个域网中 TCP 性能进行了仿真和分析；第五章提出了适合 Ad hoc 网络的传输协议，介绍了协议组成与应用办法；最后对全文作了总结，并指出需要进一步研究的方向。

## 第2章 无线网络简述

第一章绪论中简要介绍了无线广域网、无线局域网和无线个域网。下面就这三种不同网络的技术特点等分别详细叙述。

### 2.1 卫星广域网

卫星通信由于其具有天然的广播、多播和覆盖面广的特性而越来越成为一种重要的通信工具，在通常的通信途径难于实施或者由于多山等地理特点妨碍信号传输的情况下，卫星更是首选的通信工具。随着空间技术的发展，国防、航天和其他诸多领域对卫星的需求也越来越多。在陆地通信网已经构成宽带多媒体通信网的环境下，利用卫星建成宽带卫星接入（BSA, Broadband Satellite Access）系统<sup>[15]</sup>被认为是较好而切合实际的方案，既经济又可靠。

作为地面网络的补充，卫星通信对于地面网络不能到底的不发达地区（比如我国的边远地区、西部地区）来说是一种有效的通信方式，可以将 internet 延伸到更广阔的范围。另外作为通信网络骨干网的通信方式，可以绕开店面电路的瓶颈。宽带卫星通信发挥了卫星潜在的优越作用，组成宽带交互通信和较广泛的应用。特别对于一些离骨干网较远的小型企业单位和住家用户，推行远程医疗和远程教育等设施，解决他们的实际困难问题，宽带卫星通信是必需的。

#### 2.1.1 卫星接入通信方式

为了让地面上众多用户能够共同使用卫星通信，每一用户终端（UT, User Terminal）各自设置小型的地面站。用户与同一卫星所属的远地对方用户互相通信，在发送信息时可以通过用户终端和地面站连接卫星，构成上行链路。卫星发射连往对方用户地区的无线卫星信号，把始发用户的信号传给对方用户，构成下行链路。对方用户回发的信号，则经过同一卫星由相反方向到达原来用户，实现双方互通的卫星会话通信。

卫星内部设置适当的交换设备，可按发信人的意图，选一个通路从卫星传向对方用户的地面站，星上交换设备的主要作用是变换频率和相应放大，但没有调制解调作用。

### 2.1.2 多址通信和容量分配

在卫星通信网中,实现多址联接的技术基础是信号分割,即在发送端要进行恰当的信号设计,使系统中各地球站所发射的信号各有差别;而各地球站接收端则具有信号识别能力,能从混合着的信号中选择出本站所需的信号。一个无线电信号可以用若干个参量(广义的)来表征,最基本的是:信号的频率、信号出现的时间和信号所处的空间。信号的差别集中反映在上述信号参量的差别上。在卫星通信中,信号的分割和识别可以利用信号的任一种参量来实现。考虑到实际存在的噪声和其它因素的影响,最有效的分割和识别方法是设法利用信号所具有的正交性来实现多址联接。在卫星系统中常用的为时分多址与频分多址。

动态分配<sup>[16]</sup>是转发器频带根据需要(如不同的数据传输速率)实时地分配给网中各站,从而提高转发器的利用率。这种分配制度主要结合时分多址方式,可用于数字语音和数据传输。随机分配是指网中各站随机地占用卫星通道。因为数据通信一般间断而不是连续地使用通道,且数据组发送的时间也是随机的,因而如果仍使用固定预分配甚至按申请分配,则信道利用率就很低。采用随机占用方式则可大大提高信道的利用率,当然也有它的一些特殊问题,如碰撞等。

### 2.1.3 卫星系统的 IP/TCP 运用

卫星 IP 方式主要是采用协议网关来实现。协议网络既可以是单独的设备,也可以将功能集成到卫星调制解调器中。它截取来自客户机的 TCP 连接,将数据转换成适合卫星传输的卫星协议,然后在卫星线路的另一端将数据还原成 TCP,以达成与服务器的通信。整个过程中,协议网关将端到端的 TCP 连接分成三个独立的部分:一是客户机与网关间的远程 TCP 连接;二是两个网关间的卫星协议连接;三是服务器方网关与服务器间的 TCP 连接。这一结构采取分解端到端连接的方式,保持了对最终用户的全部透明。

在卫星 ATM 网络中,卫星被设计为能支持几千个地面终端。地面终端通过星上交换机建立虚通道,与另一个地面终端之间传输 ATM 信元。当路由选择 IP 业务进出 ATM 网时,地面终端成为 IP 与 ATM 间的路由器,将多个 IP 流聚集到单个虚通道中,还提供在 IP 和 ATM 网间拥塞控制的方法。星上 ATM 交换机在信元和虚通道级完成业务管理。



## 2.2 IEEE 802.11 无线局域网

为了组建和实现无线局域网，不同的公司或组织机构提出了各自的技术解决方案，其中比较流行的有 IEEE 802.11 系列和 Bluetooth 等，表 2-1 对它们进行了简要的比较。

表 2-1 几种无线局域网技术的比较

	802.11a	802.11b	802.11g	Bluetooth
频带	5 GHz	2.4 GHz	2.4 GHz	2.4 GHz
扩频方式	OFDM	DSSS	OFDM/PBCC	FHSS
数据速率	54 Mbit/s	11 Mbit/s	54 Mbit/s	<1 Mbit/s
传输范围	100 m	100 m	100 m	10 m
功率	高	中	高	很低
应用领域	数据	数据	数据	数据/语音

### 2.2.1 IEEE 802.11 技术与标准

IEEE 802.11<sup>[17]</sup>属于 IEEE 802 标准的一部分，是局域网在无线环境下的扩展。与有线局域网相比，无线局域网具有开发运营成本低、投资回报快、组网方式灵活等优点，有效的弥补了传统有线局域网的不足。

IEEE 802.11 的工作频段一般选择为无需获得许可的工业、科研和医疗频段。不同的国家和地区的无线电管理机构都设置了相应的 ISM 频段，例如美国的 ISM 频段由 902MHz-928MHz，2.4GHz-2.4835GHz，5.725GHz-5.850GHz 三个频段组成。IEEE 802.11 及其扩展 IEEE 802.11b 工作在 2.4GHz 频段，IEEE 802.11a 工作在 5GHz 频段，IEEE 802.11g 工作在 2.4GHz 频段。

### 2.2.2 IEEE 802.11 结构

无线局域网可覆盖的区域称为服务区域，可分为基本服务区域（Basic Service Area, BSA）和扩展服务区域（Extended Service Area, ESA）两种。基本服务区域指出无线局域网中主机的无线收发机及地理环境所确定的，包括一个基站和多个移动主机，所有的移动主机运行同样的 MAC 协议并以随机争用的方式共享无线传输媒体。在 IEEE 802.11 标准中，基本服务区域中的基站称为接入点（Access Point, AP）。

一个基本服务区域可以是孤立的，也可以通过 AP 连接到一个主干分配系统，然后再接入到另一个基本服务区，构成扩展服务区域。在扩展服务区域中，移动

主机可以在不同的基本服务区域中漫游，并实现和任何移动主机或者固定主机的通信，而且对移动主机的协议栈的 MAC 层以上是透明的。

IEEE 802.11 标准支持两种组网模式：一种是单区网络（Ad hoc）；一种是多区网络（Infrastructure）。单区网络的组网方式，允许无线终端在无线网络的覆盖区域内移动，并利用无线信道上的 CSMA/CA 机制来自动建立点到点的对等连接，这种网络中节点自主对等工作。而多区网络的组网方式是指无线网络规模扩充或无线和有线网络并存时的通信方式，这是 802.11 最常用的方式。此时，需要无线接入点的支持，AP 负责监管一个小区，并作为移动终端和主干网之间的桥接设备。当无线网络节点增多时，网络存取速度会随着范围扩大和节点的增加而变慢，此时添加接入点可以有效控制和管理频宽与频段。

### 2.2.3 IEEE 802.11 冲突避免机制

如图 2-1 所示，每个应用 IEEE 802.11 标准的主机所采用的协议栈结构包括一个单一 MAC 和多个 PHY 中的一个。

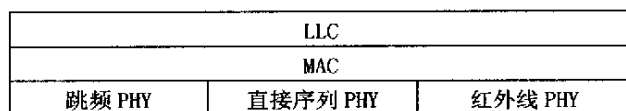


图 2-1 IEEE 802.11 协议栈结构

IEEE 802.11 标准覆盖了 OSI 模型的物理层和 MAC<sup>[18]</sup>层。它定义了单个 MAC 子层，与所有上述的 802.11 物理层联合使用。MAC 协议使用基于分布方式的无线介质访问控制的 CSMA/CD 协议，这种基于分布方式的无线介质访问控制也指分布式协调功能（DCF）协议，它仅提供尽最大努力的服务。802.11 通过利用免除争用的机制可以有选择性的支持对时间受限业务，即点式协调功能（PCF）。

在普通的异步传输流量的条件下，DCF 运行在物理层之上，当出现业务竞争的情况下，DCF 会采取相应的措施来实现优先权的控制。PCF 建在 DCF 之上，使用 DCF 提供的业务发送无竞争的业务流。IEEE 802.11 MAC 子层还提供鉴权和授予私有、加密以及功率节约的机制。

CSMA/CA 协议<sup>[19]</sup>的工作流程是：无线网络中一个工作站希望传送数据，如果探测到网络中没有数据正在传送，则等待随机时间，再选择一个时间片继续探测，如果无线网络中仍旧没有数据发送，则将数据发送出去。接收端如果收到完整的数据则回发一个 ACK 数据报，如果这个 ACK 数据报被发送端收到，则这个

数据发送过程完成，如果发送端没有收到 ACK 数据报，则或者发送的数据没有被完整地收到，或者 ACK 信号的发送失败，数据报则在发送端等待一段随机时间后被重传。CSMA/CA 通过这种方式来提供无线的共享访问，显式的 ACK 机制在处理无线问题时非常有效，但这种方式增加了额外的链路负担。

CTS/RTS (Clear to send/Request to send)，是两种特殊的管理数据帧，并不承载数据。每个 WLAN 节点，在获得了媒介使用权后发送数据之前，发送 CTS，宣告自己将要占用媒介一段时间  $x$ ，所有听到这个宣告的 WLAN 节点，无论媒介中是否真正有数据在发送，都会静候  $x$  时间，以避免冲突。当 AP 收到 CTS 的时候，会发送 RTS，再次通知其管辖范围内的所有节点，媒体将要被占用  $x$  时间。

采用 CTS/RTS 机制的好处在于避免 WLAN 中的冲突。因为冲突会导致数据包的丢失，丢失数据的节点需要重发数据包，这还可能引致下一轮冲突，从而严重影响到网络的运行效率。CTS/RTS 的另外一个好处是解决了隐藏节点的问题，从而进一步减小了发生冲突的可能性。CTS/RTS 在传统的 802.11 网络中，只是作为一个可选项，并且可以设定一个阈值，当数据包的大小超过这个阈值时，才使用 CTS/RTS。

802.11 的 MAC 子层提供了 CRC 校验和包拆分功能。在 802.11 协议中，每一个在无线网络中传输的数据包都被附加上了校验位以保证它在传送的时候没有出现错误，这和以太网中通过上层 TCP/IP 协议来对数据进行校验的机制有所不同。包拆分的功能允许大的数据包在传送的时候被拆分成较小的数据包分批传送。这在网络十分拥挤或者存在干扰的情况下（大数据包在这种环境下传送容易丢失或出错）是有效的。它减少了许多情况下数据包被重传的概率，从而提高了无线网络的整体性能。MAC 子层负责将收到的被拆分的大数据包进行重新组装，对于上层协议这个拆分的过程是完全透明的。

## 2.3 IEEE 802.15.4 无线个域网

低成本、低功耗、应用简单的 IEEE 802.15.4<sup>[20]</sup>诞生为无线传感网络及大量基于微控制应用提供了互联互通的国际标准，为也这些应用及相关产业的发展提供了有力的契机。

IEEE 802.15.4 符合传感器和低端面向控制等简单的应用专用标准，符合对功耗或网络性能较高要求应用。

### 2.3.1 IEEE 802.15.4 标准

2003年10月, 802.15.4/ ZigBee<sup>[21]</sup>为传感器网络 and 控制系统制定一个标准的解决方案。这一方案充分考虑了各种简单应用的需求, 制定了相应的以低功耗、低成本及实现简单为目标的协议标准。

IEEE 802.15.4 标准的 PHY 和 MAC 层其突出的特点是网络系统支持极低成本、易实现、可靠的数据传输、短距离操作、极低功耗等。IEEE 802.15.4 协议中明确定义了三种拓扑结构: 星型结构 (Star)、簇状结构 (Cluster tree) 和网状结构 (Mesh), 为了达到极低的设计成本和极低的功率消耗, 协议定义了两种相互配合使用的物理设备, 全功能设备和消减功能设备:

- 全功能设备 (Full function device, FFD), 可以支持任何一种拓扑结构, 可以作为网络协商者和普通协商者, 并且可以和任何一种设备进行通信;
- 消减功能设备 (Reduced function device, RFD), 只支持星型结构, 不能成为任何协商者, 可以和网络协商者进行通信。

IEEE 802.15.4 网络需要至少一个全功能设备作为网络协商者, 终端节点一般使用消减功能设备来降低系统成本和功耗, 提高电池使用寿命。寻址模式可以为网络加设备标识符的星型结构, 以及源和目标标识符的点到点结构两种。

### 2.3.2 IEEE 802.15.4 主要技术特点

IEEE 802.15.4 的物理层的使用可以支持欧洲的 868MHz 的频段、全球美洲和澳洲的 915MHz 的频段和现在已经被广泛使用的 2.4GHz 的频段, 以上频段基本为 ISM 频段, 这使得该协议具有更旺盛的生命力。

低功耗、实现简单。设备可以在使用电池的驱动下, 运行数月甚至数年, 低功耗意味着较高的可靠性和可维护性, 更适合体积小的大量日常应用中。

设备安装维护成本低。设备可以在标准电池供电的条件下, 而不需要任何更换电池或充电操作, 在内部自动可配置和网络设备的冗余等方面的简化降低维护费用, 另外电池供电可以使用设备的体积和面积都可以得到有效的降低, 从而以降低成本。

单个网络中可容纳高密度的节点。通过使用 IEEE 802.15.4 标准的物理层和 MAC 层支持几乎任意数目的设备, 这一点对于大规模传感器阵列和控制尤其重要。

### 2.3.3 IEEE 802.15.4 MAC 层

IEEE 802.15.4 帧结构的设计原则为在保证网络在有噪音的信道上以足够健壮性传输的基础上将网络的复杂性降到最低，MAC 层定义了 4 种基本帧结构：信令帧，提供协商使用；数据帧，承载所有的数据；响应帧，确认帧的顺利传送；MAC 命令帧，用来处理 MAC 对等实体之间的控制传送。

对于低延迟或者有特殊数据带宽要求的应用，无线个人网的协商者可以利用同步时隙<sup>[22]</sup> (Guaranteed Time Slot, GTS)，这些时隙是由信道无竞争周期 (Contention Free Period, CFP) 组成。CFP 一般出现在活动的超帧尾端，前面一般跟随着一些信道竞争访问周期 (Contention Access Period, CAP)。

在 802.15.4 中，有三种不同的数据转移：从器件到协调器；从协调器到器件；在对等网络中从一方到另一方。为了突出低功耗的特点，把数据传输分为以下三种方式：直接数据传输，这适用于以上所有三种数据转移，采用无槽载波检测多址与碰撞避免 (CSMA/CA) 或开槽 CSMA/CA 的数据传输方法，视使用非信标使能方式还是信标使能方式而定；间接数据传输，这仅适用于从协调器到器件的数据转移，数据帧由协调器保存在事务处理列表中，等待相应的器件来提取，通过检查来自协调器的信标帧，器件就能发现在事务处理列表中是否挂有一个属于它的数据分组，在非信标使能方式中也可能发生间接数据传输，在数据提取过程中也使用无槽 CSMA/CA 或开槽 CSMA/CA；有保证时隙 (GTS) 数据传输，这仅适用于器件与其协调器之间的数据转移，既可以从器件到协调器，也可以从协调器到器件，在 GTS 数据传输中不需要 CSMA/CA。

另外，IEEE 802.15.4 标准还支持可选的超帧结构。该超帧结构的格式是由协商者来定义，绑定了网络信令帧，而由协商者来使用的。超帧被划分为 16 个大小相等的时隙，信令帧在每一个超帧的第一个时隙中进行传输，如果协商者不希望使用超帧结构，可以关闭信令帧的传输，信令帧可以同步网络中的设备，识别无线个人网并且描述超帧结构。在冲突访问阶段，任何一个设备如果想进行通信，必须与其他设备使用 CSMA/CA 的机制，而且所有的事务必须在下一个网络信令帧到来前完成。

## 第3章 无线网络 TCP 协议

### 3.1 TCP 协议概述

在 Internet 中，路由器的处理速度、通信信道及缓冲空间通常既是网络中所有主机共享的资源，也是网络系统潜在的瓶颈。随着主机数和数据业务量的不断增加，瓶颈处就会发生资源竞争，从而导致网络拥塞。网络拥塞控制（Congestion Control）的基本思想是在共享资源管理的基础上，按一定的算法控制发送端的数据发送量，合理使用网络资源，保证网络的稳定性。

传输控制协议就是 TCP（Transmission Control Protocol）<sup>[23]</sup>。在[RFC 793]中给出了明确的描述。它是 internet 中的传输层协议，位于 OSI 参考模型网络层协议 IP 之上。图 3-1 为 OSI 的参考模型。

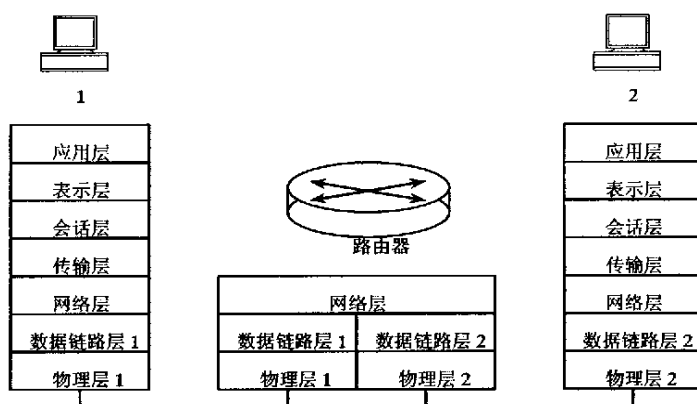


图 3-1 OSI 参考模型

TCP 是一种提供可靠端到端字节流服务的传输层协议。TCP 对用户数据进行透明分段和重组，并进行流量控制和拥塞控制<sup>[24]</sup>。在基于不可靠的 IP 协议之上，TCP 协议提供以下服务以保证可靠的面向连接服务：

**基本数据传输：**TCP 能够在用户之间传输连续字节流。一般由 TCP 协议自身算法决定发送数据的数量和速度，也可由用户使用推操作强制发送。

**拥塞控制：**发送方可以根据网络的拥塞状况自适应调整数据发送的速率，以缓解网络拥塞状况。

**流量控制：**发送方必须保证发送数据速率不超过接收方处理能力，避免因接收方处理能力有限造成的丢包。

**复用：**为了使得同一台机器中的多个应用程序可以复用 TCP 连接，引入了端

口的概念。每个主机分配一系列的端口，可以和不同应用程序绑定，以区分同个主机的不同数据流。

**可靠性：**TCP 必须保证用户提交的所有数据正确传输，处理破坏、丢失、重复和失序数据报的情况。TCP 通过为数据流分配序列号以及肯定确认等机制来保证可靠性。

**流量控制**<sup>[25]</sup>用来防止发送方超限额使用接收方的容量，避免因接受方处理能力有限造成的丢包，处理端到端的问题；**拥塞控制**用来防止过多的数据送往网络，造成路由交换和链路的负担过重，处理主机和网络交互的问题。

TCP 协议传输的数据单元称为报文段 (segment)<sup>[26]</sup>，通过报文段的交互来建立连接、传输数据、发出确认、通告窗口大小以及关闭连接。图 3-2 为 TCP 报文段在 IP 数据包中的封装，图 3-3 给出了 TCP 的报文段格式。



图 3-2 TCP 数据在 IP 数据报中的封装

源端口				目的端口				
序号								
确认号								
HLEN	保留	URG	ACK	PSH	RST	SYN	FIN	窗口
检验和				紧急指针				
选项 (若有)							填充	
数据								
.....								

图 3-3 TCP 报文段格式

每个报文段包括两个部分：头部和数据。TCP 头部携带了所需的标识和控制信息。源端口和目的端口字段包含了连接双方对应用程序进行标识的 TCP 端口号，实现复用功能。序号字段指出了这个报文段在发送方的数据字节流中的位置，确认号字段则指出了本机希望接收下一个字节的序列号，提供精确的可靠性传输。由于选项字段长度会根据包含内容不同而有变化。

引入了头部长度的字段，它以 32 字节为单位，其中 20 字节为固定格式。6 比特的保留字段是为将来的应用而保留的。在 TCP 协议中使用 6 个比特来指示报文段的目的和内容，具体参加表 3-1。报文段有不同的类型，比如有些报文段是仅仅

携带确认信息，有的携带建立或者关闭连接的请求。

由于 TCP 使用了捎带确认技术，对于从 X 主机到 Y 主机的确认信息，尽管它是对从 Y 到 X 的数据确认，但可以放在从 X 到 Y 的数据中。

表 3-1 TCP 头部某些字节的含义

字节名称	该字节置 1 时含义
URG	紧急指针字段有效
ACK	确认字段有效
PSH	本报文请求 PUSH 操作
RST	连接复位
SYN	序号同步
FIN	发送方字节流结束

## 3.2 TCP 协议工作机制

### 3.2.1 TCP 连接建立与终止

TCP 连接过程分为三步。接收方 TCP 处于 LISTEN 状态，等待发送方发送连接请求。发送端发送一个带 SYN 位的请求，向接收端表示需要连接，假设发送包请求序号为 0，则为：SYN=0，ACK=0，然后等待接收端的响应。接收端收到请求后，查看是否在 LISTEN 的指定的端口，若不在指定端口发送 RST=1 应答，拒绝建立连接。如果接收连接，那么接收端发送确认，SYN 为接收端的一个内码，假设为 10，ACK 位则是发送端的请求序号加 1，本例中发送的数据是：SYN=10，ACK=1，用这样的数据发送给发送端，向发送端表示，接收端连接已经准备好了，等待发送端的确认。发送端接收到消息后，分析得到的信息，准备发送确认连接信号到接收端。发送端发送确认建立连接的消息给接收端。确认信息的 SYN 位是接收端发送的 ACK 位，ACK 位是接收端发送的 SYN 位加 1，即：SYN=1，ACK=11。此时，连接已经建立起来，然后发送数据。这是一个基本的请求和连接过程。

断开连接有两种方式：一是主动断开连接；另一是被动断开连接。TCP 通过发送控制位 FIN=1 的数据片来关闭本方数据流，但还可以继续接收数据，直到对方关闭那个方向的数据流，连接关闭。由于 TCP 的半关闭性，协议使用修改的三次握手协议来关闭连接。关闭的原则就是当一方完成它的数据发送任务后就能发送一个 FIN 来终止这个方向连接。当一端收到一个 FIN，它必须通知应用层另一端已经终止了那个方向的数据传送。发送 FIN 通常是应用层进行关闭的结果。



### 3.2.2 TCP 流量控制机制

#### 3.2.2.1 确认机制

TCP 协议使用带重传的肯定确认机制来提供可靠性，接收方收到数据以后向源站发送 ACK 报文。发送方对发出的每个分组都保存记录，并启动一个定时器，在发送下一个分组之前等待确认信息，如果定时器超时没有收到确认信息则重发分组。图 3-4 分别给出了数据发送成功和未成功两种情况下的事件序列。

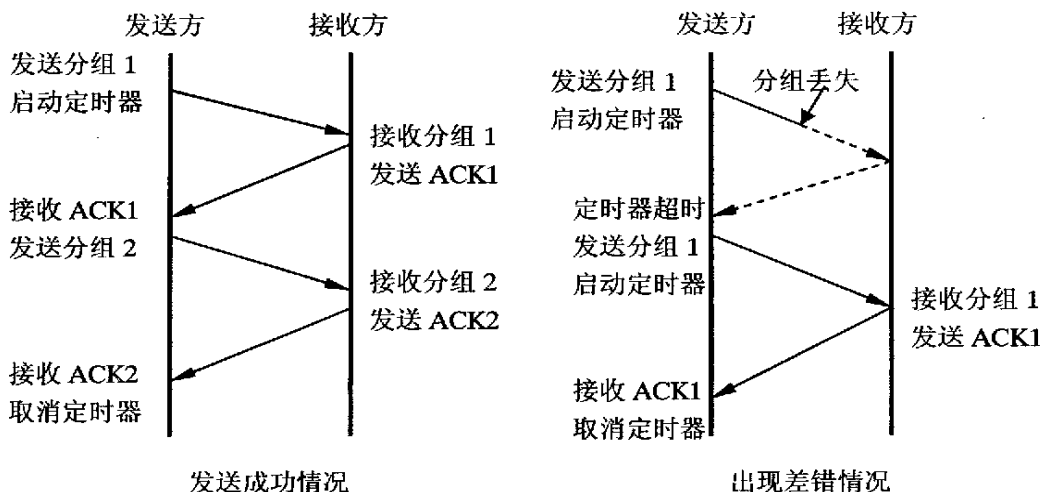


图 3-4 “带重传的肯定确认”机制

可靠性还需解决分组重复问题，由于网络时延较长，导致发送方误以为分组丢失而重发分组，因此分组和确认信息都可能会重复。TCP 协议通过给每个分组指定序号并要求接收方记住接收分组的序号来检测重复现象。为了避免迟到确认和重复确认带来的混乱，肯定确认协议在确认信息中携带一个序号，这样接收方就能正确的把分组与确认关联起来。

#### 3.2.2.2 滑动窗口

发送方送出一个分组以后，等待相应确认信息而不能发送下一个分组，在一段时间内，数据只在机器之间单向传输，且在机器延迟响应（如计算路由或者检测校验和等）时间段中网络完全是空闲的，浪费了大量宝贵的网络带宽。

滑动窗口机制是对带重传的肯定确认机制的改进，可以提高数据流传输效率以更充分利用网络带宽。它允许发送方在等待确认之前发送多个分组，可以发送的分组个数取决于窗口大小，当发送方收到对窗口内第一个分组的确认信息后，窗口可以向后滑动并发送下一个分组。随着确认的不断到达，窗口也在不断的向后滑动。滑动窗口协议的效率与窗口大小和网络接收分组的的速度有关。图 3-5 是

一个发送 3 个分组的滑动窗口协议操作示意图，在该图中，发送方在接收到确认之前就发送了全部 3 个分组。

滑动窗口还需记录已经确认的分组，并为每一个未确认分组设定定时器。如果某个分组丢失，对应的定时器超时之后发送方就会重新传输该分组。在接收端也有类似的窗口，在分组到达之后送回确认信息。

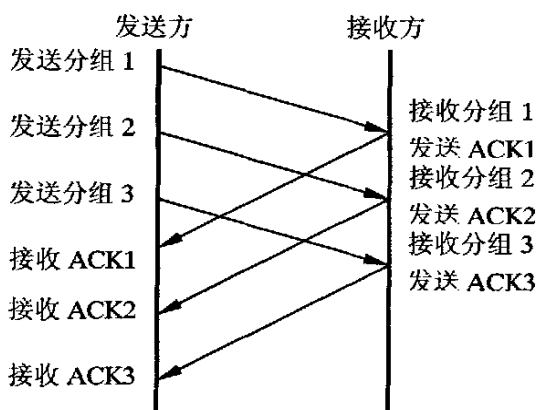


图 3-5 使用滑动窗口协议传输分组

### 3.2.2.3 累积确认

TCP 协议使用累积确认机制，当前数据包的确认也隐含了对该数据包之前到达的所有数据包的确认。发送方对每个 TCP 连接保留三个指针，如图 3-6 所示，最左边的指针区分已经发送并得到确认的字节序列与尚未得到确认的字节序列，最右边的指针标出了窗口的右边界，指出序列中在未得到确认的情况下可以发送的最高字节序列号，位于窗口中间的指针划分出已经发送的字节序列和尚未发送的字节序列的边界。即滑动窗口将数据流分为 4 个部分：已发送并得到确认部分、已发送但等待确认部分、可以发送还未发送部分和窗口外不可发送部分。接收到有效的确认后，窗口将向前滑动，以允许新数据发送。接收方也要维持一个类似窗口把接收到的字节序列重新拼装起来。

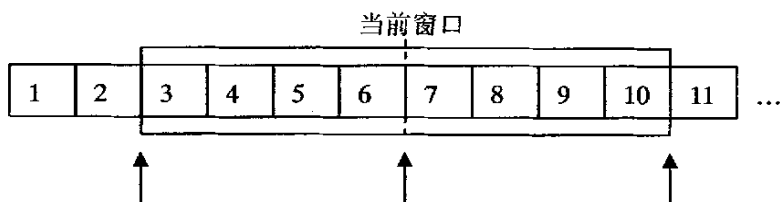


图 3-6 TCP 的滑动窗口

TCP 滑动窗口协议和简单的滑动窗口协议不同之处在于 TCP 根据一定算法改变窗口大小，以适应网络需求。

### 3.2.3 TCP 拥塞控制机制

拥塞控制机制<sup>[27]</sup>是 TCP 协议的核心。拥塞是由路由器流量过载引起的,即流量超过了路由器的缓冲处理能力时,路由器丢弃数据包,因而拥塞会引起数据包丢失和高延时。由于大多数的 internet 数据流是由高速可靠的有线链路来传输的,因而现有 TCP 的实现针对网络拥塞状况加以优化,即假设数据包的丢失是由于拥塞引起的并进行相应拥塞控制。检测到数据包丢失后,除了重传丢失的数据包以外, TCP 还降低传输速率,以减少网络中的数据流量,从而使路由器有足够时间来转发数据包。然后 TCP 逐渐增加传输速率同时探测网络容量,以避免拥塞再次发生。

#### 3.2.3.1 TCP 拥塞检测

TCP 协议有两种机制来检测数据包丢失或者网络拥塞:重传超时 (Retransmission Timeout, RTO)<sup>[28]</sup>和重复确认。TCP 使用超时机制检测错误,即在规定时间内不能接收到有效确认则认为数据包丢失了。在任意时刻, TCP 的发送方都为报文段设置重传定时器,定时间隔为 RTO。一般使用指数退避规则动态更新 RTO,如果经过一个 RTO 后没有接收到报文段的确认,除了重传报文段以外还要将 RTO 的大小设置为原始值的两倍,即  $RTO_{new} = 2 * RTO_{old}$ 。如果数据包非连续到达将导致接收方重复确认,发送方检测到多个重复确认后,一般是 3 个,将认为数据包丢失。

#### 3.2.3.2 慢启动

慢启动 (slow start) 机制观察新报文段进入网络的速率和接收方接收报文段的速率,并设法使其相等。

TCP 连接建立以后,拥塞窗口 cwnd 大小初始化为 1 个报文段(段的大小为接收方的通告值或者默认值,典型的默认值是 536 或者 512),每接收到一个确认信息,拥塞窗口增加一个报文段。每经过一个往返时间 (Round Trip Time, RTT),拥塞窗口大小就会增加一倍。拥塞窗口大小近似指数增长(接收方可能延迟发送确认信息)。慢启动经过一定的时间后可能达到网络容量极限,中间路由器发生丢包,通知发送方拥塞窗口过大,须对其进行调整。

#### 3.2.3.3 拥塞避免

在拥塞避免阶段,假设当前拥塞窗口大小为 cwnd 个字节,则可以发送的报

文段数目为  $cwnd/segsz$  个 ( $segsz$  表示报文段大小), 如果接收到全部确认信息, 则  $cwnd$  可以增加一个报文段的大小。

慢启动机制和拥塞避免机制是相互独立的算法, 但是当拥塞发生以后, TCP 先减慢数据包传输速率, 然后调用慢启动机制来恢复数据传输。在实际的实现中将两者结合起来, 并为每个 TCP 连接维护两个变量: 拥塞窗口 ( $cwnd$ ) 和慢启动门限 ( $ssth$ )。完成三次握手信号建立连接以后, 两者联合操作的算法可描述如下: 1、初始化: 将  $cwnd$  初始化为 1 个报文段,  $ssth$  初始化为 65536 字节; 2、慢启动: 可发送的报文段数目为  $cwnd$  和通告窗口中的最小值, 每接收到一个成功的 ACK 后将  $cwnd$  增加一个报文段; 3、拥塞控制: 当拥塞发生时, 将  $ssth$  设置为  $cwnd$  当前值的一半, 如果拥塞是由于超时引起的, 则将  $cwnd$  复位为 1 个报文段并转到慢启动状态, 否则调用拥塞避免机制; 4、拥塞避免: 每接收到一个确认信息后,  $cwnd$  增加  $segsz * segsz / cwnd$  个字节, 在一个 RTT 中成功接收了所有报文段的确认信息  $cwnd$  增加一个报文段的大小。

### 3.2.4 效率和公平性

衡量协议算法优劣的指标主要集中在资源使用效率和协议间公平性上。

网络资源的使用效率是指发送端要求的总资源与网络所能提供的资源之间的关系。如果二者刚好相等或者很接近, 那么这种算法的效率就是高的, 否则都是效率不高的表现。

公平性是指在网络发生拥塞时各连接能公平地共享网络资源。公平性主要为两方面: 1、TCP 流和 UDP 流之间资源享用的公平性; 2、TCP 流之间资源享用的公平性。

### 3.2.5 TCP 协议的发展过程

现在的 TCP 实现基本上都是由伯克利软件分发 (Berkeley Software Distribution, BSD) 网络发展而来, 在 BSD 网络上先后实现了 Tahoe、Reno 和 Vegas 等几个不同版本的 TCP 协议。

最初的 TCP 协议只有基于窗口的流量控制机制而没有拥塞控制机制。流量控制是一种局部控制机制, 其参与者仅仅是发送方和接收方, 它只考虑了接收端的接收能力, 而没有考虑到网络的传输能力; 而拥塞控制则注重于整体, 其考虑的是整个网络的传输能力, 是一种全局控制机制。正因为流控制的这种局限性, 从

而导致了拥塞崩溃现象的发生。1986年初, Jacobson 开发了现在在 TCP 应用中的拥塞控制机制。1988年 Jacobson 针对 TCP 在控制网络拥塞方面的不足, 提出了慢启动 (Slow Start) 和拥塞避免 (Congestion Avoidance) 算法。1990年出现的 TCP Reno 版本增加了快速重传 (Fast Retransmit), 快速恢复 (Fast Recovery) 算法, 避免了网络拥塞不严重时采用慢启动算法而造成过大地减小发送窗口尺寸的现象, 这样 TCP 的拥塞控制就由这 4 个核心部分组成。正是这些拥塞控制机制有效的防止了今天网络的拥塞崩溃。

Brakmo, O'Malley 和 Peterson 共同提出了 TCP Vegas<sup>[29]</sup>。Vegas 修改了 Reno 中的重传机制<sup>[30]</sup> 与拥塞控制机制<sup>[31]</sup>。Vegas 的与拥塞控制机制比 Reno 中的拥塞控制机制主动。Vegas 将期望的吞吐量和实际测量吞吐量进行比较, 认为实际吞吐量和期望吞吐量之间的差距越大则表明网络的拥塞状况越严重, 因而减小发送速率。

Vegas 的改进的慢启动机制引入了拥塞检测机制<sup>[32]</sup>。Vegas 允许以 2 个 RTT 为单位对拥塞窗口进行指数增长。通过维持拥塞窗口的相对稳定, 比较期望吞吐量和实际吞吐量, 当到达某一个特定的门限时, 拥塞窗口将从指数增长模式切换到线性增长模式。

Tahoe, Reno 和 Vegas 二种不同 TCP 协议版本的简单比较参见表 3-2。

表 3-2 三种不同 TCP 协议拥塞控制机制比较

	慢启动	拥塞避免	快速重传	快速恢复
Tahoe	有	有	有	无
Reno	有	有	有	有
Vegas	改进	改进	改进	有

### 3.3 无线环境对 TCP 的要求

由于无线网络存在的环境不可靠, 所以不能保证环境总是适应于数据的传输。传统 TCP 的差错控制针对于拥塞丢包, 而非无线网络中典型的切换和随机误码所造成的瞬间随机错误或暂时的中断。拥塞窗口调节的处理方法目的是避免发生拥塞崩溃, 它不是无线网络中 TCP 连接错误恢复的最佳策略。

#### 3.3.1 无线信道的特点

在无线环境中, 信道由于容易受到各种环境因素的影响而变得非常复杂。主

主要表现在：

#### BER 高

由于信号衰落、反向多径传播信道、来自其它发射机或周围环境的干扰和无线网络所处环境的变化，使无线网络的 BER 明显高于有线网络，即使同一会话上的 BER 都会发生数量级的变化。特别地，突发性错误造成连续多个分组丢失，使拥塞窗口连续减半，发送端发送能力大幅下降；当拥塞窗口低于 4 个报文时，TCP 就无法激活快速重传/恢复算法，发送端只能等待超时重传，TCP 的性能进一步降低。

#### 带宽有限

无线信道提供的带宽一般为 1-11Mbps，远远小于有线网络所提供的 10-1000Mbps 带宽。带宽对无线网络来说是一种宝贵的资源，因此如何高效地使用带宽成为无线网络协议设计的重要问题。

#### 移动（如切换）问题

移动设备在移动过程中，即从一个区域移动到另一个区域时，会产生暂时性的连接中断。频繁的移动通常会导致丢包或延迟，而 TCP 可能错误地解释为网络的拥塞，从而激活拥塞控制机制，引起吞吐量的降低。

#### 能量有限

无线网络节所消耗的能量可以从 2 个因素考虑：第一，接收/处理能量，即备用模式下或接收数据时消耗的能量；第二，发送能量，即通过无线链路发送数据的能量。通常当网络半径超过 100 米时，能量开销主要指传输能量。移动设备通常使用电池提供有限的能量，从而使其通信时间受到限制。频繁的分组丢失引起的重传致使 TCP 连接的时间较长，能量损耗高。

### 3.3.2 无线环境下 TCP 的挑战

无线环境的这些特点使得传统 TCP 在无线环境下表现出许多不尽人意的地方，究其原因，最关键的是其缺乏全面的错误控制能力。目前的 TCP/IP 协议是为有线网络、固定主机设计的，TCP 的差错控制是以网络拥塞丢包为中心，而忽略链路传输错误等其它问题，这在有线网络上成立的。但在无线移动环境下链路错误 BER、越区过程中的丢包问题等典型问题，TCP 缺乏处理这些错误类型的的能力，因此必须进行改进。对 TCP 性能的改进主要表现在以下几个方面：

更可靠、精确的拥塞检测能力，包括发送路径上和 ACK 分组返回路径上的拥塞。增强拥塞检测能力不但可以解决上述因 ACK 丢包引起的传输能力下降问题，也就是所谓链路的非对称性问题，而且由于超时是基于对 RTT 的测量，因此还能减小超时对协议性能的影响。

检测出导致丢包的出错性质（包括错误发生频率、持续时间等），基于不同的错误性质采用不同的错误恢复策略。如链路上持续发生错误，则暂时停止数据发送，避免数据分组的进一步丢失；在拥塞期间采用渐变的窗口调整；在切换期间暂停数据传输并在其后实施积极的快速重传。

吞吐率和能量消耗之间的权衡，也就是吞吐量、能量的效率问题。虽然很多人认为能量消耗越多，所获得的吞吐量也越大。但在很多情况下，特别是在无线移动环境下并不成立。在暂时突发错误存在下，TCP 的退避机制以大量减少吞吐量为代价节约能量，可应用于移动的、以电池为电源的设备中，如 802.15.4 传感器网络。

### 3.4 无线网络 TCP 协议改进

TCP 协议是为包含低差错率链路和固定主机的传统网络设计的，然而，无线网络却不符合这些特征。现有 TCP 实现中将数据包丢失作为网络拥塞的征兆，而在无线网络中数据包丢失极有可能是由无线链路差错引起的。

TCP 在无线链路上存在的主要问题是不能区别数据包丢失的原因是否由网络拥塞引起，从而将链路传输错误、越区切换等引起的数据包丢失误认为网络拥塞，不必要拥塞控制使得网络吞吐量下降和时延增大。另外，由于现有 TCP 协议通过使用重传定时器来恢复链路层错误，为了适应于各种网络，定时器定时精度一般都在秒数量级，这将导致无线信道上错误恢复时间的增加。针对无线链路上 TCP 存在的性能问题提出了各种不同的改进方案。

#### 3.4.1 纯端到端方案

直接修改 TCP 连接两端的协议使之更好的应用于无线环境，如 TCP-SACK、TCP-HACK 等。

选择性重传是一种纯端到端的方案。在同一个窗口中多个数据报文段丢失时，TCP 基本拥塞控制机制将导致较差性能。由于 TCP 使用累积确认机制，发送方在一个 RTT 内只能知道一个数据报文段丢失。即使用累积确认不能够提供足够的信

息来对一个传输窗口中发生的多个错误进行快速恢复，选择性重传改进了传统 TCP 的累积确认机制的这个缺陷。有两种选择性重传机制：TCP SACK<sup>[33]</sup>和 SMART<sup>[34]</sup>。TCP SACK 通过给 TCP 增加 SACK 选项使每次确认最多可以包含 3 个成功接收的不连续数据块，每一数据块通过其起始和结束序列号来定界。SMART 在进行累积确认的同时将引起该确认的数据包序列号传送给发送方，接收方根据这个信息创建一个掩码来表示成功确认的数据包。一旦发送方检测到该掩码中有空隙就认为数据包丢失了。

TCP-HACK<sup>[37]</sup>通过头校验和选项来提高衰落链路上的 TCP 性能，其出发点是当数据包在无线链路上被破坏时，很可能发生错误的是数据包中的数据而不是 TCP 头，这是因为在典型的 MTU (Maximum Transfer Unit, 最大传输单元)下数据包的数据部分通常比数据包头要长的多，因而通过让 TCP 恢复这些被破坏的数据包头，可以判定网络上是否发生了拥塞。该方案增加了两个选项，第一个选项是对数据包的校验和，第二个选项是头校验和确认选项，包含被破坏的 TCP 报文段的序列号，只使用在接收到错误数据包时发送的确认信息中。通常，TCP 只携带一个校验和，它对整个数据包进行校验，如果由于数据包在无线链路上发生了错误，校验和就会失效，此时要将整个数据包丢弃。多数情况下，由于数据包错误可能发生在数据部分，因而可以将数据包头恢复出来。因而通过增加对 TCP 头的校验和，可以校验 TCP 头的完整性，通过恢复出 TCP 头以后，接收者可以发送一个特殊的确认给发送者指示发生了数据错误。这个确认将包含发生错误的数据包序列号，除了附加选项以外该确认和普通确认一样。

### 3.4.2 分段连接方案

建议根据有线和无线链路的不同特征而采取不同的发送策略，将移动主机和通信对端间的 TCP 连接在中间节点分为两端，有线连接段使用基本 TCP 协议，无线连接段使用是与无线环境的改进协议，无线链路的数据丢失对发送端时屏蔽的，中间节点保留了数据的硬状态。

有线链路传输速度快且误码率低，而无线链路速度依然很慢（相对于有线链路而言），并且容易受到噪声干扰和信道衰落的影响。在同时包含有线和无线两种链路的网络中对这两种性能相差很多的链路使用同样的流量控制和拥塞控制，无线链路显然会成为系统性能的瓶颈，将其分开进行处理有利于提高链路的



利用率。为此提出了多种分段连接的方案如 I-TCP<sup>[35]</sup>和 WTCP<sup>[36]</sup>等。

I-TCP 中涉及到二个节点实体：移动主机、固定主机和移动支持路由器，移动支持路由器和有线网络直接相连。I-TCP 将发送者和接收者之间的 TCP 连接在基站处分为两个部分：

发送者到基站的有线连接和基站到移动主机的无线连接。值得注意的是，分段连接的第二部分是只有一跳之隔的无线链路，在该链路上不一定要使用 TCP，可以采用其他适合无线链路特性的传输机制（或者对现有 TCP 加以改进）来提高网络性能。

I-TCP 对固定主机来说是不可见的，因而可以与现有的有线网络后向兼容，而且该模型使得基站可以来处理移动节点的大部分通信开销，甚至可以只在移动节点上运行简单的无线协议和基站进行通信实现和固定网络的连接功能，享受固定网络的服务。

### 3.4.3 缓存方案

分段连接方案改变了 TCP 端到端传输的基本语义，因此出现了 TCP 缓存方案，它在代理中采用了软状态方式，软状态的丢失虽会影响性能，但不会阻止 TCP 端到端的数据传送。在中间节点增加探察代理来缓存和探测发送移动主机的 TCP 数据段和返回确认，根据情况决定是否重发缓存中的数据段，它提供可选择性的重传，使中间节点可教准确的区分拥塞和无线传输产生的错误。

Snoop 协议监视通过 BS 的每一个数据包，包括确认包。在数据传输从 FH 到 MH 方向上，Snoop 维持了一个 TCP 包的缓冲（cache），这些包是由 FH 发送而来，但还没有被 MH 确认。同时它跟踪每个从 MH 而来的 ACK 包，通过到达的重复 ACK 包或局部超时来检测包的丢失。当丢包发生时，如果 Snoop 已经缓存此包，则进行局部重传，并且重复的 ACK 包被丢弃以避免导致发送端进入快速重传。使用局部重传，可以使基站不必传送重复的 ACK，对固定的发送端主机屏蔽在无线链路上丢失的数据包。在数据传输从 MH 到 FH 方向上，当指示无线分组丢失的重复 ACK 到达 BS 时，Snoop 通过设置 ELN（Explicit Loss Notification）位，来为 MH 端提供明确的无线分组丢失信息，进而保障为发送端提供更为有效的错误恢复机制。Snoop 设计简单高效，无需固定主机端协议栈有任何改动。

### 3.4.4 交叉层方案

由链路层活网络层将链路环境状态反馈到 TCP 层, TCP 依此采取相应的方法来处理数据段丢失。根据低层通知方式和 TCP 处理措施不同, 可细分为三类: 重传机制改进方案、精确状态通知法、超时冻结机制。

显式拥塞指示 (ECN): 要求所有路由器在拥塞时都向发送端发送 ECN。显式丢失指示 (ELN): 要求网络节点或接收端在随机丢失报文时向发送端发送 ELN。这两种方案目前基本不可行, 只能作为理论性的研究。

显式基站指示 (EBSN) 方案要求 BS 具备链路重传和缓冲能力, 在链路不好时, 缓冲并向发送端发送 EBSN 指示, 使得发送端将 TCP 计时器复位, 而其他参数不变。缺点在于 BS 必须具备链路层重传功能。而且不能解决返回的 ACK 在链路中重传造成 TCP 计时器超时的情况。此外链路层完全保证数据的传输是不可能的, 总会有数据丢失, 重传次数也有限。

### 3.4.5 链路层方案

链路层协议是克服链路高误码率特性的常用方案, 采用本地丢失恢复的思想, 使无线链路向上呈现非常低的丢失率, 从而对发送端完全透明, 传统的方法包括前向纠错 FEC 和自动重传请求 ARQ 以及这两种方法的混合。

AIRMAIL 协议是同时采用 FEC 和 ARQ 的链路层机制, 通过纠正不可靠无线信道上的错误而不是仅靠端到端重传来得到较高吞吐量和较小时延。该方法还对处理开销进行了不对称处理, 把大量智能化模块放到基站侧去处理, 这样可以减少移动主机侧的处理开销, 以适应移动主机的功率和处理能力受限的状况。

## 第4章 TCP 协议改进算法及性能仿真

本章节首先介绍了无线 TCP 协议的改进算法—Westwood 协议及基于 Westwood 协议的改进，并通过仿真表明改进的算法提高了网络的吞吐量，同时，证明了该算法可以和 Reno 和谐共存在同一个网络中，而不降低其性能，保持了很好的公平性。

### 4.1 Westwood 及 Westwood-BD 协议

Westwood<sup>[38]</sup>是最近才提出的一种基于测量技术的通过可用带宽估测实现网络流量控制的新方法。Westwood 利用连续地监测接收端返回的 ACK 速率，得到单位时间发送的分组数目与分组的大小相乘，即可获得发送端实际的传送带宽。在获得链路的情况后，当拥塞发生后，利用这些测量值来计算新的 *cwnd* 和慢启动的门限阈值 *ssth*。具体算法如下：当拥塞出现时，若发送端连续接收 3 个重复 ACK，则  $ssth = (B * RTT_{min}) / seg\_size$ ,  $cwnd = ssth$ ；若传输超时，则  $ssth = (B * RTT_{min}) / seg\_size$ ,  $cwnd = 1$ ；在正常传输时，ACK 正常返回时，*cwnd* 变化遵循 TCP Reno 的窗口变化规律。Westwood 在 TCP Reno 的基础上，通过测量估算出网络的可用带宽，对拥塞窗 *cwnd* 进行适当调整，实现更快速的恢复，这种机制尤其在无线网中非常有效。与 TCP Reno 相比吞吐量成倍提高。根据上面分析，Reno 窗口变化关系如式(1)，Westwood 中 *ssth* 与 *cwnd* 算法如式(2)：

$$cwnd(t + \Delta t) = \left\{ \begin{array}{l} \text{slow start phase:} \\ cwnd(t) + 1, \text{ if } cwnd(t) < ssth \\ \text{congestion avoidance phase:} \\ cwnd(t) + \frac{1}{cwnd(t)}, \text{ if } cwnd(t) \geq ssth \end{array} \right\}$$

$$ssth = cwnd(t) / 2, cwnd(t) = ssth, \text{ if receive 3dupACKs}$$

$$ssth = cwnd(t) / 2, cwnd(t) = 1, \text{ if } RTT > RTO \quad (1)$$

$$ssth = \frac{B * RTT_{min}}{seg\_size}, cwnd(t) = ssth, \text{ if receive 3dupACKs}$$

$$ssth = \frac{B * RTT_{min}}{seg\_size}, cwnd(t) = 1, \text{ if } RTT > RTO \quad (2)$$

Westwood 采用了客观的带宽测量技术，通过计算采样时间间隔内的发送端成

功发送的数据量，作为链路的可用带宽， $R_t = D_t / \Delta_t$ ，其中  $\Delta_t$  为取样时间差， $D_t$  为  $\Delta_t$  时间段传送的数据量。为了获得有效的链路的带宽，Westwood 对可用带宽估价的方法做了改进，引入低通滤波器，如式(3)所述，

$$R_t = \alpha_t R_{t-1} + (1 - \alpha_t)(R_t + R_{t-1})/2 \quad (3)$$

其中， $\alpha_t = (2\tau - \Delta_t)/(2\tau + \Delta_t)$ ，低通滤波器的周期为  $\tau$ ， $\Delta_t$  为取样时间差，平滑参数  $\alpha_t$  由取样时间间隔和低通滤波器周期决定。

与分段连接、FEC、ARQ 等改进机制相比，westwood 仅需修改发送端的 TCP 协议栈，而无需修改中间节点（路由器、Gateway 等）的 TCP 协议栈，对于端到端链路的修改内容小，符合 TCP 端到端的语义，适合目前中间网络结构复杂、中间节点数量多特点的 internet。

Westwood-BD (Bandwidth Division) 是在 Westwood 协议基础上，借鉴 Vegas 对于网络的估计，对 Westwood 协议的网络状态估计部分进行修改。原有 Westwood 协议中网络状态估计主要为对网络的带宽估计，而且在整个传输过程中带宽的计算方式固定，由于拥塞窗口和慢启动机制窗口的确定与带宽估计值有固定关系，由此造成网络的部分带宽没有得到更好的利用，特别在网络传输性能优良时带宽被浪费了。

改进的基本思路为引入状态衡量值  $\lambda$ 、即时传输带宽  $BW_{now}$  与以往传输带宽  $BW_{pri}$ ，协议保存了第  $n-1$  个数据包的即时传输带宽计为  $BW_{pri}$ ，第  $n$  个数据包的即时传输带宽估计为  $BW_{now}$ ， $\lambda$  的定义见式 (4)。

$$\lambda = \begin{cases} 1, & \text{if } (BW_{now} / BW_{pri} \geq 0.8) \\ 0, & \text{if } (BW_{now} / BW_{pri} < 0.8) \end{cases} \quad (4)$$

当  $\lambda = 1$  时，采用激进式带宽估计算法，通过多次仿真调整带宽估计中  $\alpha_t$ ，使其达到最优化，并修改拥塞窗口的增长方式，每接收到一个新的 ACK， $cwnd = cwnd + 2$ ，使其充分利用网络的可用带宽。当  $\lambda = 0$  时，采用保守的带宽估计算法，即原有 Westwood 协议带宽估计算法不变。

当网络状态良好，在协议中体现即为网络的即时传输速率仍保持较高幅度，适当提高网络的吞吐量，整体上提高了网络利用率。

其次，当检测到逼近拥塞时新机制继续以比 Westwood 和 Reno 低的速率增加

窗口，显然和 Westwood 和 Reno 一致，确保了新的机制与 Westwood 和 Reno 的公平性、友好性。

最后，改进协议每次在进入拥塞避免时都将窗口设置为可利用带宽和时延乘积，由于它对链路可利用带宽的估计准确，所以改进协议在拥塞避免的起始就能充分地利用网络资源，即使后来用估计网络拥塞存在误差，也不会延长其到达平衡状态的时间，对其性能产生负面影响。

## 4.2 网络仿真环境

### 4.2.1 仿真试验软件

本文的仿真环境采用 Network Simulator 2.28 (NS2.28)<sup>[39]</sup> 版本，系统平台为 RedHat Linux 9.0。

#### 4.2.1.1 NS2 概述

NS2 是用 C++ 编写的面向对象的网络模拟仿真试验环境，用户使用 OTcl 编写试验脚本操纵网络组件进行试验。它起源于 1989 年的 Real Network Simulator 项目，后于 1995 由 Lucent Bell Lab, Xerox PARC, UCB 和 USC/ISI 所进行的 DARPA VINT 项目继续研究。现在，NS2 由美国 DARPA SAMAN 项目和 NS FCONSER (扩展 Ns Nam) 项目共同支撑，另外还有很多不同的公司和大学提供模块，包括 CMU<sup>[40]</sup> 的几个 mobile 对等网的路由和节点模型，以及 SUN 的 Mobile IP 模块等。

NS2 已经成为了进行网络模拟仿真试验的标准环境，在全世界有大约 1000 个研究所和超过 1 万用户使用，目前 NS2 为 ns-2.28 版本。NS2 可以运行在 FreeBSD, Linux, Solaris 和 Windows 以及 Mac 操作系统上。绝大多数的网络研究人员都使用 NS2 进行模拟仿真试验和结果比较。图 4-1 为 NS2 的一般使用方法。

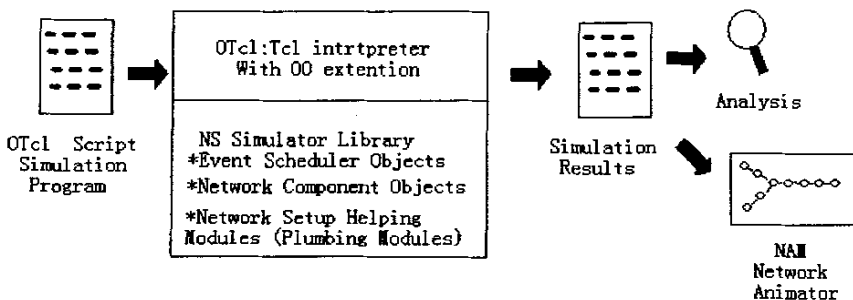


图 4-1 NS2 使用

4.2.1.2 NS2 功能与使用

NS 的软件实现采用了分离语言程序设计技术。将整个实现分成数据路径实现和控制路径实现两部分。数据路径部分是 NS 的核心，包括与数据处理相关的所有类的实现：如代理 (Agent)、队列、节点、链路、事件调度器等。为了执行效率，数据路径类都由 C++实现，在 NS 中由 C++实现的类统称为编译类。对应的控制路径是指在仿真过程中构建网络拓扑、网络参数等与配置相关的部分，为了易用和灵活性，这部分由 oTcl 解释语言实现，由 oTcl 实现的类称为解释类。这样 NS 即兼顾了配置的灵活性又提高了程序的执行效率，但需要掌握两种编程语言。

在 NS 环境下的一个仿真过程就可简单描述为：编写 oTcl 脚本程序，利用 NS 提供的仿真器类定义网络拓扑、配置节点、建立链路、收集仿真数据、最后分析结果的过程。如果测试修改网络协议和算法，就需了解 NS 的类层次结构以及 C++类与 oTcl 类之间的链接机制，用修改的算法对 NS 进行扩充。

NS2 网络对象组件库里面已经有：

1. 有线部分，支持 RoutingDV, LS, PIM-SM;
2. 传输层支持 TCP 和 UDP，包括 TCP Reno, TCP Vegas 等等;
3. 信息源支持 WEB, FTP, TELNET 和 CBR;
4. 调度：drop-tail, RED, FQ, SFQ 和 DRR;
5. 质量保证：IntServ 和 DiffServ;

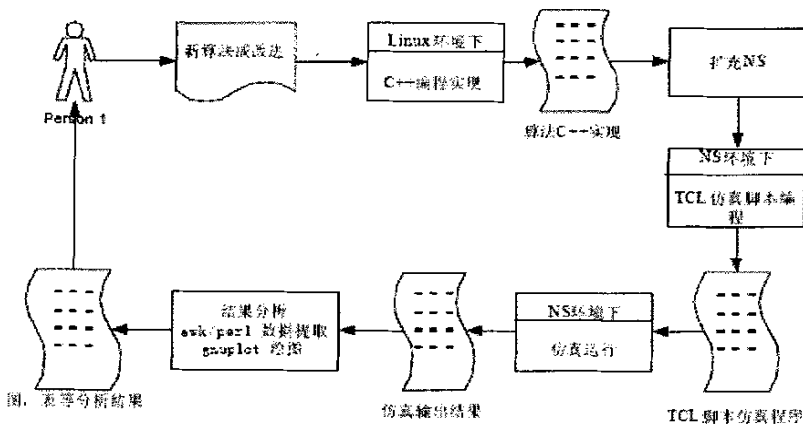
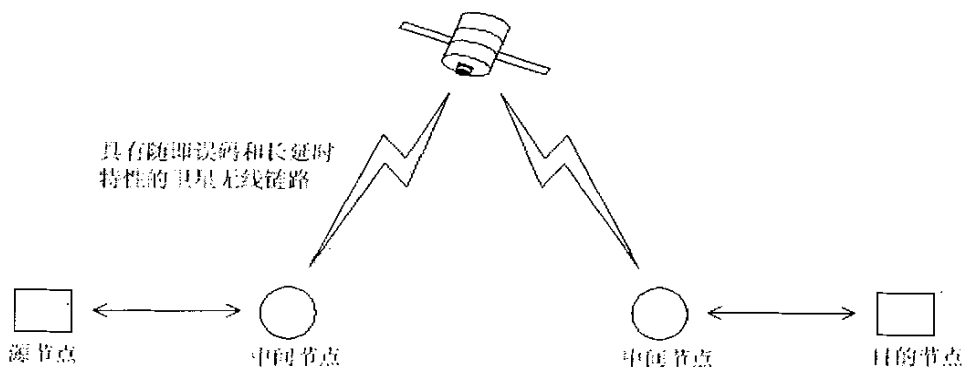


图 4-2 使用 NS2 仿真改进算法

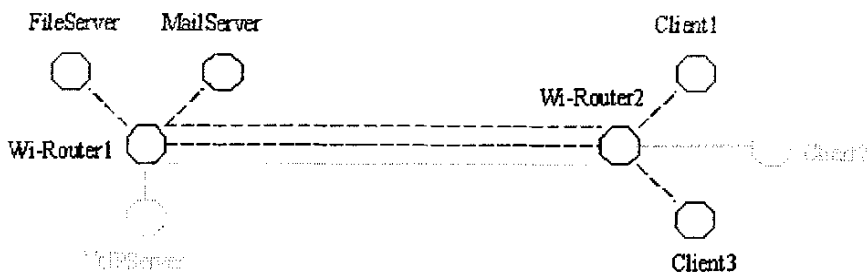
图 4-2 显示了用户使用 NS2 仿真改进算法的过程。这个过程是一个简化的流程，实际中，向 NS2 添加新算法，需详细了解 NS2 的结构，以及 NS2 中相关算

法的具体实现，构建 C++算法和 Tcl 之间的互通，如同修改 Linux 的系统一样，最后还需重新编译整个 NS2，这些都需要大量的工作。

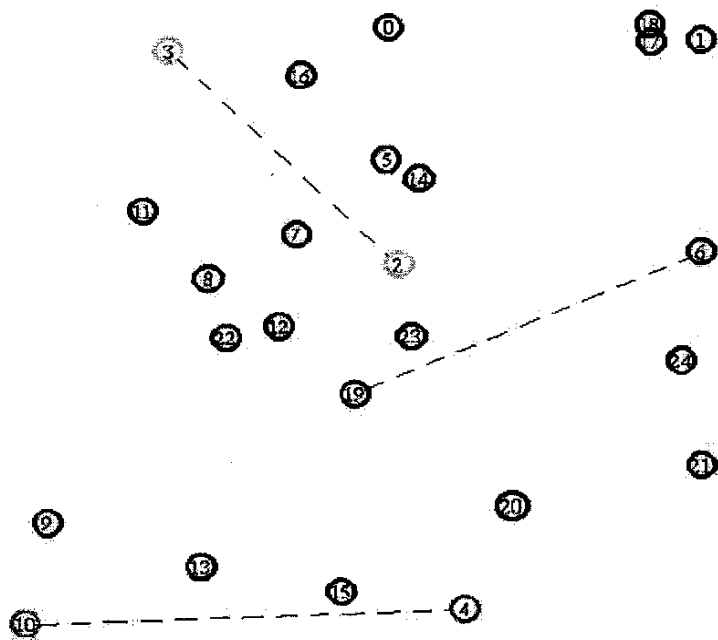
### 4.2.2 仿真拓扑



a. 卫星广域网



b. 802.11 无线局域网



c. 802.15.4 无线个域网

图 4-3 仿真试验网络拓扑结构

仿真试验中的网络拓扑结构如图 4-3 所示。图中 a 模拟了一个使用卫星链路连接两个接入点的典型拓扑，源节点为模拟了位于杭州的一个地面卫星站（北纬 30.1,东经 120.1），目的节点模拟了位于北京的一个地面卫星站（北纬 39.5,东经 116.3），其中的卫星链路带宽为 2Mbps，具有长延时和随机误码特性，链路延时为 220ms，网络 RTT 为 450ms。源节点与中间节点链路带宽 10Mbps，延时 2ms，目的节点与中间节点链路带宽 10Mbps，延时 2ms。在源节点模拟设置 FTP 服务器，目标节点使用 ftp 下载源节点一个 10MB 大小的文件。针对不同的卫星链路误码率模拟了 Reno、Sack、Vegas 与 Westwood 四种协议的应用。

图 b 模拟了一个有 8 个节点的 802.11 局域网。服务器区有 3 个服务器节点，Mail 服务器、File 服务器与 VoIP 服务器，客户区有 3 个客户节点，Client1、Client2 与 Client3，两个无线路由器 Wi-Router1 与 Wi-Router2，服务器区节点与客户区节点无法直接通过无线通信，需经过两个无线路由器建立多跳的无线连接。整个无线网络使用 IEEE802.11b 无线网标准，设置节点无线带宽为 2Mbps，两个无线路由器缓存为 50 个包，RED 队列管理方式。仿真中模拟了四个连接，连接一：FileServer 与 Client1 的 TCP 连接，承载大文件的传送，即 TCP 发送端一直有包等待发送；连接二：MailServer 与 Client3 的 TCP 连接，该连接模仿 Foxmail 的连接方式，间断性有数据传送；连接三、连接四：VoIPServer 与 Client2 之间持续的 UDP 连接。针对不同的无线信道误码率模拟了 Reno、Sack、Vegas 与 Westwood 四种协议的应用。

图 c 模拟了一个有 25 个节点的 802.15.4 无线传感器网络。各个节点符合 IEEE 802.15.4 的物理层、MAC 层标准，使用 CSMA/CA 拥塞管理协议，无线网络带宽为 136Kbps，节点位置由 setdest 命令生成，设置节点数量为 25，时间间隔为 2 秒，最大速度为 2m/s，区域限制为 50×50。网络内有 3 个多跳数据流，3-2，10-4，19-6 之间不间断有数据传输。针对不同的无线信道误码率模拟了 Reno、Sack、Vegas 与 Westwood 四种协议的应用。



表 4-1 仿真参数表

	误码率	Reno	Vegas	Sack	Westwood
卫星广域网	$10^{-5}$ - $10^{-1}$	✓	✓	✓	✓
802.11 无线局域网	$10^{-6}$ - $10^{-2}$	✓	✓	✓	✓
802.15.4 个域网	$10^{-5}$ - $10^{-1}$	✓	✓	✓	✓

### 4.3 网络仿真结果与分析

按上节设定的网络模型，生成 TCL 脚本仿真程序，在 NS 中仿真产生 tr 目标文件与 nam 文件。tr 目标文件包含了所有仿真结果数据，通过提取分析 tr 文件可以得到所需数据，并用图表的方式表示，方便分析。

#### 4.3.1 吞吐量分析

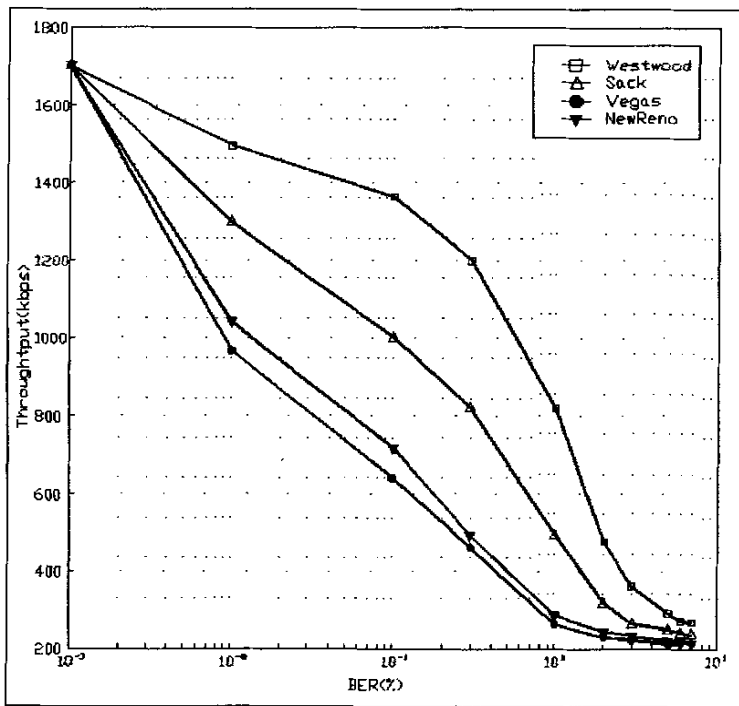


图 4-4 卫星广域网 TCP 有效吞吐量

图 4-4 比较了卫星广域网中四种不同 TCP 算法有效吞吐量的变化。当误码率在  $10^{-5}$  时，各 TCP 的吞吐量性能差别不大，有效平均吞吐量在 1.7Mbps-1.73Mbps 之间，在低误码率时，影响 TCP 协议性能的主要问题不在于 TCP 协议本身的拥塞控制，而是网络自身的延时、路由器的缓存等因素。由于卫星广域网的延迟时

间大，一般典型 RTT 为 450ms，因此卫星网的带宽利用率只能达到 85%。当误码率达到  $10^{-4}$  时，使用不同 TCP 协议的平均吞吐量有明显的不同，Reno 协议平均吞吐量为 1.05Mbps；Vegas 协议仅为 0.98Mbps，Vegas 协议自身的特性导致了这个结果，由于误码率的原因，有部分包被丢失了，Vegas 协议的实际吞吐量低于其预期的吞吐量，Vegas 减小了拥塞窗口，进而使得吞吐量更小，仿真结果符合对协议的理论分析；Sack 协议吞吐量为 1.3Mbps，较 Reno 高 30%；Westwood 协议平均吞吐量达到 1.53Mbps，远高于 Reno 协议，Westwood 协议有效的区分了网络中丢包的原因，即区分误码丢包与拥塞丢包，采用了不同的拥塞控制机制，因此协议在性能上有较大幅度提升，特别当误码率达到  $10^{-3}$  时，Westwood 吞吐量为 1.4Mbps，而 Reno 吞吐量仅为 0.7Mbps，性能提升达到 100%。当误码率继续增大时，各 TCP 协议的吞吐量均大幅下降，失去比较意义。从图 4-4 可以看出，误码率在  $10^{-5}$ - $10^{-2}$  之间 Westwood 的性能下降曲线较其他协议平滑的多。

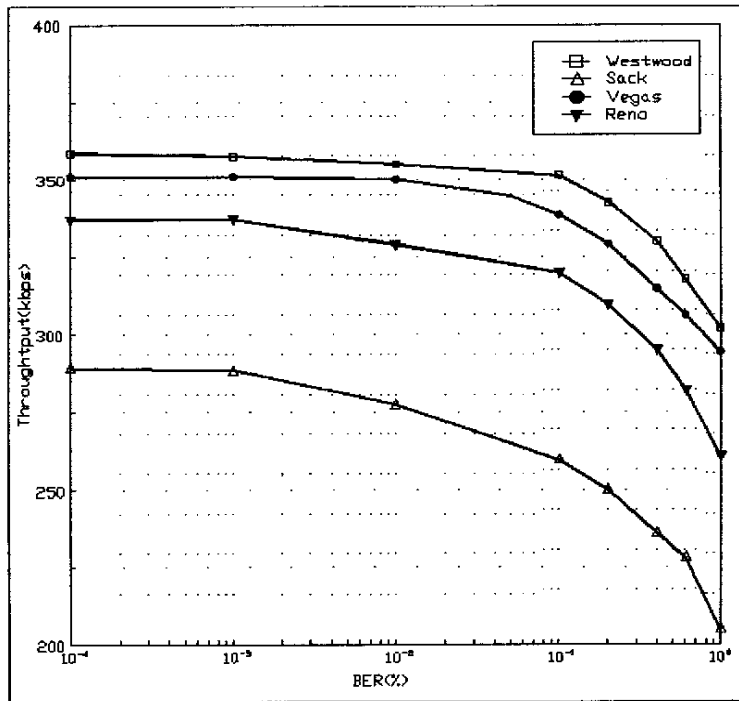


图 4-5 802.11 局域网 TCP 有效吞吐量

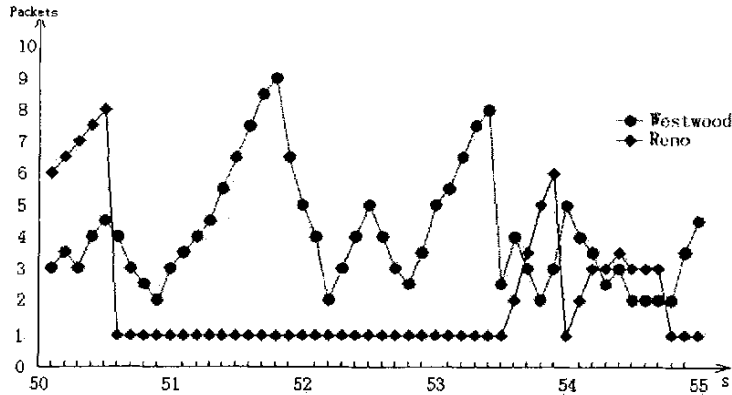


图 4-6 802.11 局域网 TCP 拥塞窗口

图 4-5 比较了 802.11 无线局域网中不同 TCP 协议有效每秒平均吞吐量的变化。由于 802.11 无线局域网的 MAC 层使用了 RTS/CTS 做为冲突控制机制，造成 802.11 网络有效载荷降低，网络的利用率不及卫星广域网高，仿真结果与理论分析一致。当链路的误码率为  $10^{-6}$  时，Reno 协议的平均每秒吞吐量为 351kbps，Vegas 协议的平均每秒吞吐量为 338kbps，Sack 协议的平均每秒吞吐量为 282kbps，Westwood 协议的平均每秒吞吐量为 356kbps，整个网络的带宽利用率较低，Westwood 协议的性能较 Reno 仅提高了 1.4%，这是由于在低链路误码率情况下，整个传输过程由于链路误码引发的丢包仅为 2-3 个，Westwood 协议拥塞控制机制的优点未得到充分发挥，Sack 协议的提出主要解决网络中出现一个数据段出现多个数据包丢失的情况，但在低误码情况时，一个数据段出现多个数据包丢失的几率非常小，Sack 协议性能反较 Reno 有降低。当链路误码率在  $10^{-2}$  以下时，Westwood 协议性能提高不明显，当误码率继续升高，达到 3% 时，Reno 协议每秒平均吞吐量为 152kbps，而 Westwood 协议可保持在 191kbps，性能较 Reno 提高达到 25%，误码率达到 5% 时，Reno 协议每秒平均吞吐量为 39kbps，而 Westwood 协议为 101kbps，性能较 Reno 提高达到 158%，可见在 802.11 无线局域网中，Westwood 协议适用于误码率较高的情况，而实际中，802.11 无线局域网受到的链路干扰多，如遮挡、冲激干扰等，误码率也相对较高。比较图 4-5 中 4 个协议性能随误码变化趋势，Westwood 协议在误码率为  $10^{-6}$ - $10^{-3}$  之间时，带宽保持稳定，未有大幅下降。

图 4-6 为误码率为 3% 时，Reno 协议与 Westwood 协议中 cwnd 的变化（截取了 50—55 秒的片断），在图中可明显看到两个协议丢包时的处理机制。Reno 协议在 50.6 秒时丢包发生后 cwnd 降为 1，后一段时间（50.6 秒—53.4 秒段）cwnd 一

直为 1，这是 Reno 协议自身的特征，Westwood 协议在整个 50 秒—55 秒时间段内 cwnd 一直保持 2 以上，锯齿型比较明细，对于快速恢复高吞吐率有利。

图 4-7 比较了 802.15.4 无线个域网中不同 TCP 协议有效每秒平均吞吐量的变化，由于误码率在  $10^{-3}$  以下时，各协议吞吐量变化很小，图 4-8 放大了误码率为  $10^{-3}$ — $10^{-1}$  段。当链路的误码率为  $10^{-5}$  时，Reno 协议的平均每秒吞吐量为 126kbps，Vegas 协议的平均每秒吞吐量为 128kbps，Sack 协议的平均每秒吞吐量为 126kbps，Westwood 协议的平均每秒吞吐量为 128kbps，与卫星网、802.11 无线局域网相比较，802.15.4 网络的带宽相对较低，Westwood 协议的性能较 Reno 仅提高了 2.4%，这是由于在低链路误码率情况下，整个传输过程由于链路误码引发的丢包仅为 2-3 个，Westwood 协议拥塞控制机制的优点未得到充分发挥。当链路误码率在  $10^{-2}$  以下时，Westwood 协议性能提高不明显，当误码率继续升高，达到 3% 时，Reno 协议每秒平均吞吐量为 108kbps，而 Westwood 协议可保持在 110kbps，性能较 Reno 提高达到 1.8%；误码率达到 5% 时，Reno 协议每秒平均吞吐量为 91kbps，而 Westwood 协议为 96kbps，性能较 Reno 提高达到 5.5%；误码率达到 10% 时，Reno 协议每秒平均吞吐量为 60kbps，而 Westwood 协议为 70kbps，性能较 Reno 提高达到 17%。可见 Westwood 协议在误码率高的 802.15.4 个域网比较适应。

横向分析各 TCP 协议在无线网络中的性能，发现改进协议性能提高与所在网络的带宽延时有很大的关系。在卫星广域网中，Westwood 协议可提高 TCP 协议性能达到 100% 以上，而在 802.11 无线局域网中可达到 20%，而在 802.15.4 无线个域网仅能达到 5%，与网络的带宽延时乘积成正比。表 4-2 为各网络的带宽、延时及带宽延时乘积。

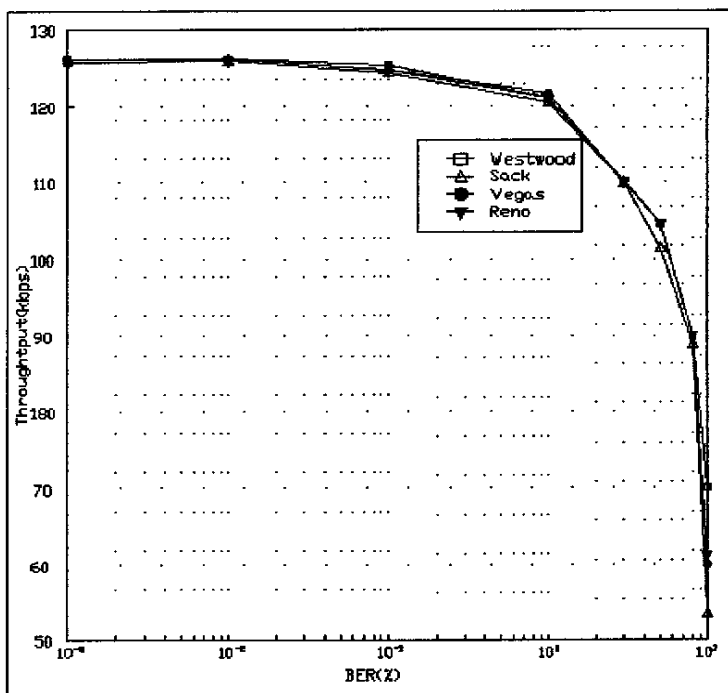


图 4-7 802.15.4 个域网 TCP 有效吞吐量

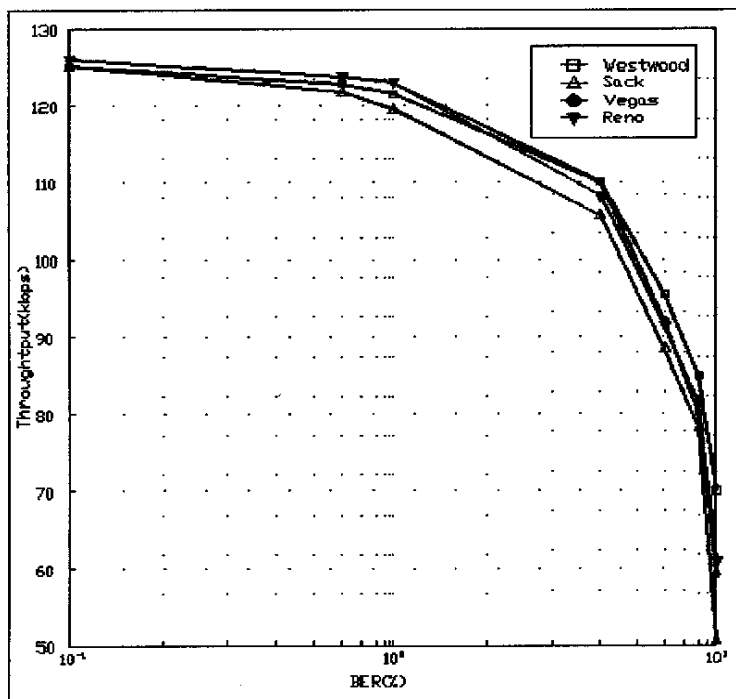


图 4-8 BER=10<sup>-3</sup>-10<sup>-1</sup> 段

表 4-2 各网络带宽延时

	带宽	延时	带宽延时乘积
卫星广域网	2Mbps	250ms	500Kbs
802.11 无线局域网	2Mbps	4.64ms	9280bs
802.15.4 无线个域网	136Kbps	3.52ms	480bs

### 4.3.2 公平性分析

公平性分析主要分为两个部分——研究 Westwood 协议与 Reno 协议之间的公平性；研究 Westwood 做为 TCP 协议与 UDP 协议之间的公平性。

Westwood 在 Reno 的基础上改进而来，通过对 Westwood 算法中窗口大小随时间变化的关系的数学分析，可得到式 (1) 的带宽分配与 RTT 关系。Westwood 的带宽分配具有与往返时间 RTT 成反比关系，与 Reno 基本一致。

$$B = \frac{1}{\sqrt{RTT * (RTT - base\_RTT)}} * \sqrt{\frac{1-p}{p}} \quad (1)$$

设定了如图 4-3b 的 802.11 无线局域网拓扑结构，有两个数据流，一个为 0s 开始的 Westwood 流，一个为 10s 开始的 Reno 流，总仿真时间为 60 秒，除开始时间外，两个数据流的 TCP 窗口大小、数据包大小等参数相同。仿真得到如图 4-9 的两个 TCP 数据流吞吐量。

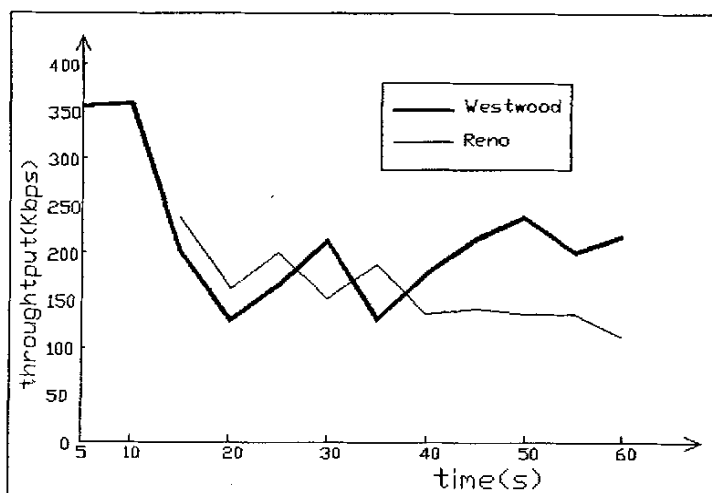


图 4-9 两个 TCP 数据流吞吐量

在 0-10s，Westwood 协议的吞吐量一直保持在 350Kbps 以上，10s 后，Reno 协议数据流开始，Westwood 协议吞吐量下降到 200Kbps，Reno 协议的吞吐量则为 240Kbps，之后，10-60s 之间，两个数据流保持了相对稳定的数据流量，Westwood

协议的吞吐量略高于 Reno 协议。由此分析可得出 Westwood 与 Reno 协议可以和谐共存在一个网络中，保持良好的公平性。

设定了如图 4-3b 的 802.11 无线局域网拓扑结构，有两个数据流，一个为 0s 开始的 Westwood 流，一个为 10s 开始的 UDP 数据流，总仿真时间为 60 秒，仿真得到如图 4-10 的两个不同类型协议数据流吞吐量。

在 0-10s，Westwood 协议的吞吐量一直保持在 350Kbps 以上，10s 后，Reno 协议数据流开始，Westwood 协议吞吐量下降到 210Kbps，UDP 协议的吞吐量则

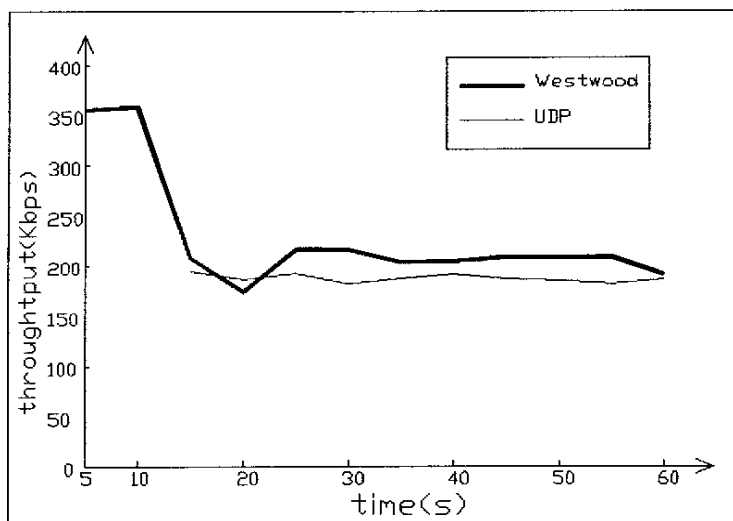


图 4-10 TCP、UDP 协议数据流吞吐量

为 190Kbps，之后，10-60s 之间，两个数据流保持了相对稳定的数据流量，Westwood 协议的吞吐量略高于 Reno 协议。由此分析可得出 Westwood 与 UDP 协议可以和谐共存在一个网络中，保持良好的公平性。

### 4.3.3 Westwood-BD 协议性能分析

按第 4.1 章节的思路修改 Westwood 协议，并将其添加到 NS2 的 TCP 协议簇中，重新编译 NS2。设定如图 4-3b 拓扑结构的 802.11 无线局域网，网络场景与前章节中设置相同，比特误码率设置为从  $10^{-5}$ - $10^{-1}$ ，分析 tr 目标文件，得到 Westwood-BD 协议的每秒平均吞吐量，见图 4-11。

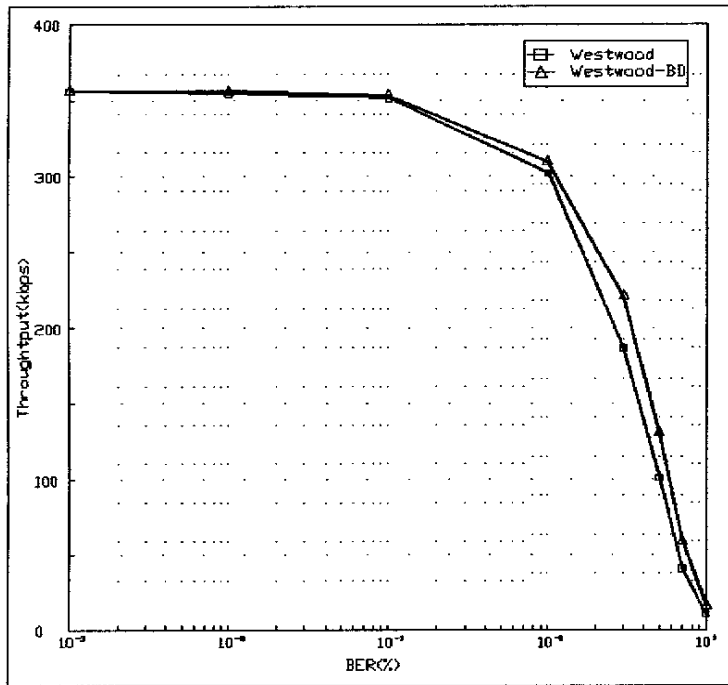


图 4-11 Westwood-BD 协议数据流吞吐量

由图 4-11 可以发现，误码率为  $10^{-5}$ - $10^{-3}$  时，Westwood-BD 协议每秒平均吞吐量与 Westwood 协议基本相同，当误码率上升到  $10^{-2}$  时，Westwood-BD 协议每秒平均吞吐量为 310Kbps，Westwood 协议为 302Kbps，Westwood-BD 协议性能较 Westwood 协议有一定程度提高，误码率继续上升时，改进的协议性能提高更多，特别是当误码率在 0.03 时，Westwood-BD 协议每秒平均吞吐量为 212Kbps，Westwood 协议为 190Kbps，性能可提高 11.6%，这因为此时网络中的丢包主要是由于误码引起的，网络拥塞引起的丢包仅占了很小的一部分，而且仿真过程中网络输出均保持相对较高水平，因此，改进的 Westwood-BD 算法在仿真中采用了更激进的拥塞控制机制，在丢包发生时，保持了较大的 cwnd 和 ssth，有效利用了网络的带宽，在数据传输中保持相对稳定，从而更利于改进协议的性能发挥。误码率继续升高，Westwood-BD 的协议性能可较 Westwood 协议提高 40% 以上（误码率为 0.07），但此时整个网络吞吐量仅为 60Kbps，带宽利用率为 3%，整个网络的传输能力不可接受，需考虑降低网络的误码率。



## 第5章 Ad hoc 网络中无线 TCP 协议

Ad hoc 网络是自组织, 对等式无线网络。移动节点间的通信可能要经过多个中间节点的转发, Ad hoc 网络也可通过网关节点与有线网络互联互通。

### 5.1 Ad hoc 网络对 TCP 协议的影响

Ad hoc 网络的网络环境比较特殊, 不但要考虑无线链路高误码问题, 而且由于节点间的连接是时时变化的, 则可能需要重新计算到目的节点的路由, 有时还可能引起 Ad hoc 网络暂时性分离。

#### 5.1.1 Ad hoc 路由影响

##### 5.1.1.1 路由变化的影响

当一个路由不可用时, 发送方网络层将试图寻找新路由。在寻找到新路由之前的时间内可能导致多个数据包丢失。路由优化和路由修复都可能导致多个数据包丢失或者失序到达。在高移动性的 Ad hoc 网络环境中, 移动节点经常会改变位置, 为了保证可以使用最优路由, 路由协议需要频繁更新路由, 可能导致多个数据包丢失或者失序到达。

另外, 寻找新路由的时间可能会超过 **RTO**, 发送方重传数据报文并进行拥塞控制, 由于没有可用路由, 重传数据报文还会丢失。新路由建立后 **TCP** 要使用慢启动和拥塞避免来缓慢恢复数据传输, 因而频繁的路由变化会导致 **TCP** 吞吐量急剧下降。

##### 5.1.1.2 多路由的影响

为了尽量减少频繁的路由重新计算, 某些路由协议在通信节点之间维护多条路由信息。同一个连接的数据流可能经过不同的路由传送, 这可能会导致数据报文段失序到达目的节点。于是接收方发出重复确认信息, 引起发送方拥塞控制。

#### 5.1.2 Ad hoc 网络分割的影响

移动 Ad hoc 网络可能会经常性的分割为几个小的子网络。如果 **TCP** 连接的发送方和接收方被分割到不同子网络中, 由于没有可用路由, 发送方发送的所有数据包都被丢弃, 导致发送方拥塞控制。如果网络分割状况持续较长时间(比如是 **RTO** 数倍的时间), 还将导致更为严重的“连续超时”现象, 如图 5-1 所示。

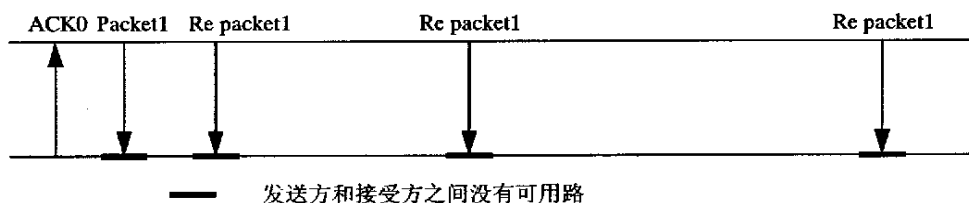


图 5-1 “连续超时”现象

“连续超时”是指对同一个数据报文段多次重传都发生在接收方和发送方之间没有可用路由的情况下，因而所有重传数据报文段都丢失的现象。每次重传失败后，发送方重传定时器定时间隔就加倍（直到达到 64s）。即使后来有了可用路由，TCP 也必须等待定时器溢出而不能传输数据，浪费了宝贵的网络资源。

## 5.2 Ad hoc 网络 TCP 协议改进思路

现有的改善 Ad hoc 网络 TCP 性能的各种方法的基本思路为：首先对网络状态进行分类，而后再根据不同状态进行相应的控制策略。通常的解决方案是针对 TCP 不恰当地启动拥塞控制，是否是反馈机制，大致可分为开环（open loop）和闭环（closed loop）两种类型：

### 5.2.1 开环方案

开环方案即非反馈方案，方案中路由的变化往往不需要网络中其它协议的支持，而仅仅依靠观察网络性能的定量指标来判断此时网络的状态。

在 TCP Door<sup>[41]</sup>方案中，发送端通过检测乱序的分组来判断路由的改变，可通过避免不必要的拥塞控制机制来提高 TCP 的性能，此方法与传统 TCP 相比可获得 50%的性能提高。

ADTCP<sup>[42]</sup>是采用非反馈机制的一综合性方案，方案中综合考虑了网络处于拥塞、链路断开、信道差错、路由变化、正常等不同状态的情况。整个方案是从控制系统的观点来看 TCP 的性能问题，整个系统分为两部分，网络状态标志部分和控制部分，这两部分相互作用组成一自适应系统。

固定 RTO<sup>[43]</sup>（Retransmission TimeOut）策略采用试探性方法来区分路由失败和拥塞。当连续发生超时，发送端就认为路由失败而并非网络拥塞，就会重发未确认的分组，RTO 保持不变，直到新路由建立，重传的分组得到确认。虽此方法得到的性能改善有限，但可以作为其它方法的补充。

### 5.2.2 闭环方案

通常，闭环方案需要网络或下层协议提供反馈，发送端通过反馈信息来对网络状态进行判定，是一种有反馈机制的方案。在以前的工作中，大部分的方案都是基于反馈思想的，不同解决方案的区别只是在于如何获得反馈信息和如何对反馈信息进行响应。该类型协议主要有 TCPF，TCP Bus 和 ATCP 等。

TCPF<sup>[44]</sup>利用网络层的反馈信息，由 TCP 连接中的中转节点来检测路由的失败。发送端处于激活态或冻结态。在激活态时，传输层由传统 TCP 进行控制。只要中间节点检测到路由失败，则向发送端发送一个路由失败分组，并记录此事件。发送端收到路由失败分组后，进入冻结态，停止发送分组并冻结 TCP 的状态值。发送端会一直处于冻结态，直到收到来自原中间节点的路由恢复分组，这时发送端又恢复到激活态，继续中断的数据传输。

TCP Bus<sup>[45]</sup>是另外一种基于反馈的方法，充分地利用了下层路由协议所提供的信息。TCP Bus 协议中，除了由中转节点发送显式路由中断消息和显式路由恢复消息，通知发送端路由的失败和恢复外，沿路各节点所缓存的分组超时值延长，从而减少了链路中不必要的重传，此外，收端采用选择性重传机制，以提高效率。同时 TCP Bus 采用定时器和探测帧保证控制信息的可靠传输。

ATCP<sup>[46]</sup>是利用网络层反馈的一种综合解决方案。方案中处理了路由失败与高误码率问题，根据不同的网络状态触发不同处理机制，解决 TCP 盲目进行拥塞控制所带来的问题。在实现中对传统 TCP 协议栈不进行修改，通过显式拥塞消息和 ICMP 消息来监听网络状态，使 TCP 进入恰当的模式。当发送端收到 ICMP 信息时，就认为 TCP 连接中断而进入保持模式；通过显式拥塞消息来通知网络拥塞，当发送端收到显式拥塞消息就直接激活拥塞控制机制；如果分组丢失而未收到显式拥塞消息信息，假设是由于无线链路的高误码率带来数据丢失，不激活拥塞控制，而重传丢失的分组。

### 5.2.3 两种方案比较

闭环方案对网络状态的判断较为准确，这是因为反馈机制的信息是直接来自于网络，其准确性高，但实现有一定难度，且网络开销也会很大。而开环方案，一般是纯端到端的机制，只需终端节点通过一定的计算来判断网络状态，其优点在于适用于较复杂的网络环境，如 Ad hoc 网和固定网络相结合的网络环境中，但

其对网络状态的判断不如闭环方案准确，还有开环方案一般通过分组的传送来判断网络状态，而数据的传送是存在一定时延，其准确度会更进一步降低。

从各种方法的比较还可以看出，对网络状态判断准确率的提高都是以算法的复杂度为代价的，算法的复杂带来协议的复杂而必然对整个网络数据传送的效率有影响，于是应该在准确度和复杂度间取一个平衡点。

学术界还有一种新的方案是修改 TCP 协议包传输机制，比如 TPA 协议<sup>[47]</sup>，其大致思路为将 TCP 原有以包为单位变为以块为单位传输，改进块的传输机制。

以下是各种应用在 Ad hoc 网络中的 TCP 方案的性能比较。

	TCP Door	ADTCP	TCP Bus	TCPF	ATCP
处理高 BER	否	可	否	否	可
处理路由变化、网络断开	可	可	可	可	可
调整 Cwnd	否	可	否	否	可
需要低层协议的支持	否	否	是	是	是
保持端对端的语义	是	是	是	是	是

### 5.3 Ad hoc 传输协议

为了更好的适应 Ad hoc 网络的特性，在 I-TCP、TCPF 及 TPA 几种改进协议基础上提出一种传输协议——Ad hoc 传输协议 (AHTP)，力求将发送或重发包的数量最小化从而减少能量消耗。Ad hoc 传输协议是一种基于窗口可调整的机制，窗口大小随流量控制和拥塞控制算法动态变化。

#### 5.3.1 数据传输

将需要传送的包按块管理，一个块包含 N 个包。AHTP 的发送端将发送缓存中的包封装为包含 N 个包的 AHTP 块，并将其可靠的传输到接收端。只有一个块内的所有包被确认，AHTP 才处理下一个块。每个包头包含了确定包归属的序号域，N 位的数据位图域，标志包在块中的位置。AHTP 头还包含了两个与 ACK 相关的域，ACK 序号域和 ACK 位图域。ACK 序号确定块中包含的需要确认的包，ACK 位图告知块中相应的包被接收端成功接收。当然，也可以通过在位图域中设定相应的位告知发送端多个包被接收端确认（一个 ACK 可以包含一个块中所有包的确认信息）。

包的传输有如下几个要点。当包发送时，发送端 AHTP 设定一个定时器，等待接收端发送相应 ACK。发送端 AHTP 接收到一个醒目包（相当于 TCP 协议 3

次握手连接的第一个 TCP 包) 的 ACK 后, 开始以下步骤: 1. 依据拥塞和流量控制算法设定新的窗口大小; 2. 计算在新窗口大小下有多少包可以发送; 3. 发送块中的下一个包。此外, 如果目前窗口中的包发生超时, 发送端 AHTP 记录超时包, 并将超时包汇入超时流, 继续前述的 1-3 步骤, 如同这个包已被确认一样。当 AHTP 完成一个块的发送, 但没有重新发送超时包, 发送端开始发送超时包, 定义为超时重发流。同样完成前面介绍的 1-3 步骤, 这个过程一直重复到块中的所有包被确认。如果接收到一个属于超时重发流中包的 ACK 时, 确认该包已经成功接收, 在超时重发流中丢弃该包。图 5-2 为发送流程图, 图 5-3 为超时重发流产生图。

AHTP 协议在数据传输上的优点在于: 1. 传输过程中, 一些由于超时发生前未收到 ACK 而进行无用传输的可能性降低了, 丢包后不是马上重发而是进入超时流, 下一轮继续发送。这点在移动 Ad hoc 网络中尤为重要, 由于节点的高移动性, 超时定时器值并不能真正反映连接的往返时间; 2. AHTP 协议中设置等待时间并没有降低吞吐量, 等待期间发送端可以发送另一个包; 3. AHTP 协议可以良好的处理 ACK 包丢失, 传输中一个 ACK 就可以充分得通知发送端当前块中所有丢失的包; 4. 发送端没有受到乱序得干扰, AHTP 协议可以在多路径移动 Ad hoc 网络中有效应用。

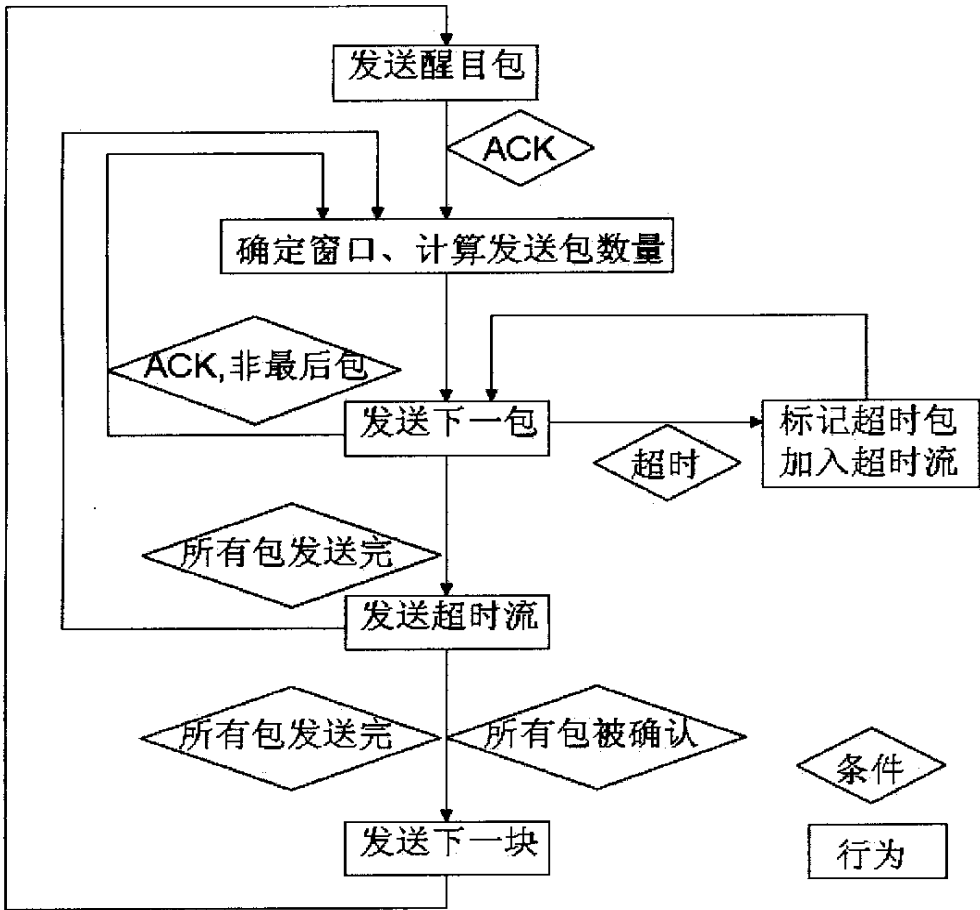


图 5-2 发送流程图

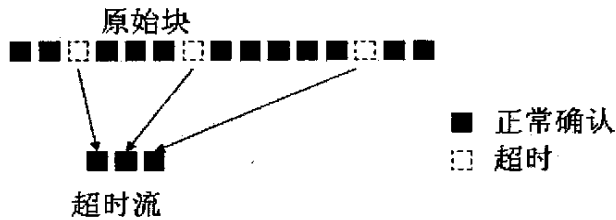


图 5-3 超时重发流产生图

### 5.3.2 流量控制机制

与 TCP 类似，AHTP 使用滑动窗口机制实现流量控制。允许发送方在等待确认之前发送多个分组，可以发送的分组个数由拥塞窗口大小确定，当发送方收到对窗口内第一个分组的确认信息后，窗口可以向后滑动并发送下一个分组。随着确认的不断到达，窗口也在不断的向后滑动。窗口值大小由拥塞控制机制确定，将在 5.3.5 中详细叙述。

### 5.3.3 路由中断处理

若网络层在探测到路由中断后提供显式链路失败通知 (Explicit Link Failure Notification, ELFN), 当接收到一个 ELFN, 发送端 AHTP 进入冻结状态, 传输窗口限制为 1 个包大小, 限制发送包的数量, 同时, 重传定时器设置为双倍时间。如果网络层没有提供 ELFN 服务, 发送端 AHTP 仍可以通过探测到多个连续超时得知路由中断。特别的, 发送端 AHTP 设定当探测到  $RF_n$  个连续的超时, 判断为路由中断, 此时, 进入冻结状态。 $RF_n$  是一个需要合理设定的协议参数。

假设网络层也不提供网络恢复消息, 当 AHTP 接收到一个最后发送包的 ACK 时就得知路由恢复, AHTP 做如下操作: AHTP 结束冻结状态; 设定拥塞窗口为最大值  $CWND_{max}$ ; 继续发送数据。如果网络层可以提供路由恢复消息服务, AHTP 的处理方式可以得到优化, 在冻结状态, AHTP 可以不发送任何数据包, 仅等待路由恢复消息。

### 5.3.4 路由变化处理

AHTP 估计连接的往返时间 (RTT), 并依此设定重传超时定时器。计算方式与 TCP 协议类似如式 (1)。

$$\begin{aligned} ERTT(n) &= \alpha * RTT(n) + (1 - \alpha) * ERTT(n - 1) \\ DEV(n) &= \beta * RTT(n) - ERTT(n) * (1 - \beta) * DEV(n - 1) \\ RTO(n) &= ERTT(n) + 4 * DEV(n) \end{aligned} \quad (1)$$

$ERTT(n)$ 和  $DEV(n)$ 都是平均值,  $\alpha$ 、 $\beta$  均为(0,1)。当发生路由变化时, 新的路径与前一个路径的跳数不同, 这意味着 RTT 与 RTO 将不适合新的路由, 为了防止可能的重传, AHTP 协议必须在路由改变后尽快探测到, 改变 RTT 估计方法以得到可靠的新 RTT 估计。AHTP 探测路由改变的途径有: 当路由中断后新的路由变为可用; 当  $RC_n$  个连续的 RTT 样本值在  $[ERTT - DEV, ERTT + DEV]$  之外。通过检测路由改变, AHTP 增大 ERTT 与 DEV 估计算法中的  $\alpha$ 、 $\beta$  值, 因此新的 RTT 估计受到新 RTT 样本值的影响因素变大, 使得 AHTP 可以快速的适应路由的变化, 最后当多个更新的估计 RTT 后,  $\alpha$ 、 $\beta$  参数回到原先的设置值。

### 5.3.5 拥塞控制机制

拥塞在链路层冲突中到传输层有两种不同的方式。一个中间节点可能传送数据包给邻节点时失败, 发出 ELFN 给发送端节点 (需要网络层支持)。这种情况下,

数据抑制，发送端 AHTP 不能区分真正路由中断。另外，中间节点在传送 ACK 时可能失败，ACK 抑制，接收端接收到 ELFN 消息，发送端仅得知连续超时。当发送端 AHTP 探测到  $CT_n$  个连续超时假设 ACK 抑制发生，进入冲突状态。发送端 AHTP 接收到  $CA_n$  个连续的 ACK 时离开冲突状态。

如果网络层不支持 ELFN 服务，通过在发送端探测连续的超时可以探测到数据与 ACK 抑制，但拥塞与路由中断是无法区分的。因此拥塞控制与路由中断处理机制是相同的。

AHTP 拥塞控制机制与 TCP 协议相同，都是基于窗口控制。不同的是 AHTP 的最大拥塞窗口值 ( $CWND_{max}$ ) 比较小，通常为 3 个 AHTP 包大小，因此拥塞窗口最小值与最大值非常接近，这使得 AHTP 的拥塞控制算法极为简单。在非拥塞状态下，AHTP 设置拥塞窗口为最大值，当 AHTP 进入拥塞状态，AHTP 将拥塞窗口大小减少到 1，直到拥塞消失后逐渐增大。

### 5.3.6 协议应用

AHTP 协议是针对移动 Ad hoc 网络提出的专有协议，是一个独立的传输层协议，协议与网络层、会话层之间的接口与 TCP 协议相同，符合 OSI 对于传输层功能与接口的定义，理论分析上在 Ad hoc 网络中应用是可行的。

正因为 AHTP 协议为专有协议，在 internet 的应用中出现不能互连的问题。使用 AHTP 协议的无线节点与一个使用 TCP 的固定节点通信时，按 TCP 协议的要求有 3 次握手过程，固然 AHTP 协议也可以设定初始的 3 次握手建立连接过程，但两种协议的包格式不同，TCP 接收端收到一个 AHTP 数据包后，经过网络层处理后交由传输层，TCP 传输层会认为 AHTP 包是个错误包而丢弃，要求发送端重发，此时传输层的连接不能建立。

如何使 AHTP 节点与普通的 TCP 节点能正常通信，成为解决方案的一个重要问题。借鉴网络地址转换 (NAT) 的思路，建立一个中转代理，将 AHTP 节点与 TCP 节点的通信划分为三个部分：AHTP 与 AHTP 的连接；TCP 与 TCP 的连接；代理完成 AHTP 与 TCP 之间的转换。无线节点与外部网络的通信必定要经过一个同时连接无线网络和外部网络的中间节点，且这个节点一般为可控节点，这个中间节点担任通信代理就比较合适了。图 5-4 为 MH 与 FH 通信模型。



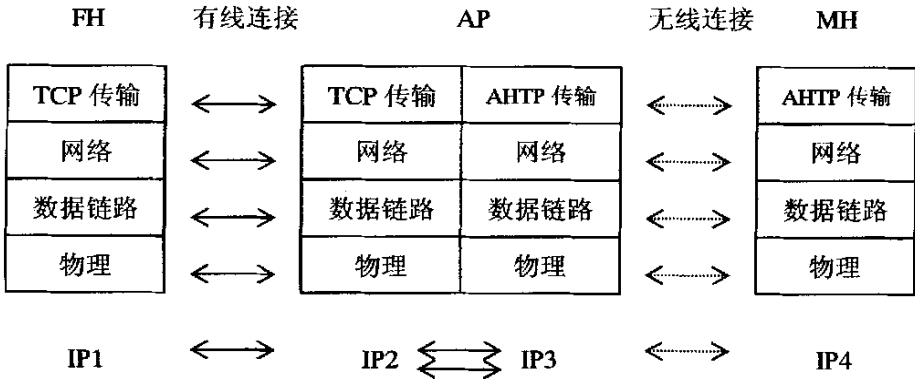


图 5-4 MH 与 FH 通信模型

## 结束语

因特网提供的信息浏览、电子邮件、文件传输等主要服务均建立在传输控制协议 TCP 之上。在有线网络和固定主机构成的环境中，由于链路误码率非常低，TCP 将数据包丢失归为网络拥塞和意外时延，从而启动拥塞控制机制，一方面通过重传丢失的数据包以提供可靠的端到端连接；另一方面通过降低发送速率来保证网络的稳定性。但是在无线网络中，数据包丢失主要是由于无线信道高的比特差错率 BER 和移动切换引起的。TCP 将这种数据包丢失误归于网络发生拥塞而采取拥塞控制措施，不必要地降低了端到端的吞吐率，导致自身性能的下降。研究和提高无线网络中 TCP 的性能，不仅可以创造一个无处不在的分布式计算环境，而且能进一步促进无线因特网的发展和部署。

本文针对无线网络环境，分析了 TCP 的两个经典版本 Reno 和 Vegas 的优缺点，简述了一些针对无线网络 TCP 的改进思路。详细阐述了一种新的改进机制，新的改进机制将 Vegas 的网络状态估计思路引入 Reno，当检测到数据包丢失时，通过评估网络性能来动态调整慢启动阈值，改进了 Reno 的指数衰减算法；同时，改进了 Reno 的线性增加算法，使得拥塞窗口调整变得精细化。在实现上，只需对 Reno 算法在发送端做少许修改即可。

在网络模拟仿真环境 NS2 上做的大量试验表明，与 Reno、Vegas 和 Sack 相比，改进的算法在多种无线网络环境下取得了较大的进步。在多种无线网络中，明显地提高了网络的吞吐量；在与其它 TCP 协议数据流竞争带宽或与 UDP 协议数据流竞争带宽时，具有良好的公平性。

本文的研究发现改进机制对于有大带宽时延乘积的网络中性能提高较好，但对于低带宽时延乘积的网络有欠缺。今后的网络则向两方面发展：高性能广域骨干网与多应用的无线网络，虽然改进协议在无线局域网中性能发挥良好，但无线个域网是典型的低带宽时延网络，如何适应无线个域网的特点，更好的提高 TCP 协议性能成为今后的一个研究方向。

同时，本文也存在着不足。首先，在改进 TCP 拥塞控制算法时，仿真的多个网络均为固定无线网络，只考虑了随机丢包对性能的影响，而没有考虑无线网络中的移动切换的影响，特别是在移动 IP 网络中，网络切换有可能为秒级别的，由于网络切换而引起数据包丢失几率更高。因此，如果应用于移动无线网络中，该

改进算法还可能存在的问题。同时本文第五章节提出应用于 Ad hoc 网络的 AHTP 协议，仅为理论的分析，没有通过实验仿真论证，另外，由于本人理论水平的局限，对协议的性能分析还不够完善，这些都有待于以后进一步的研究。

## 参考文献

- [1] 熊桂喜, 王小虎译, 《计算机网络》(第三版), 清华大学出版社, 1998
- [2] 谢希仁, 《计算机网络》(第二版), 大连理工大学出版社, 2000
- [3] <http://grouper.ieee.org/groups/802/3/ae/index.html>
- [4] Jim Geier, 王群, 李馥娟, 叶清扬译, 《无线局域网》, 人民邮电出版社, 2001
- [5] 孙儒石, 《GSM 数字移动通信工程》, 人民邮电出版社, 2002
- [6] R.J.Bates, 朱洪波, 沈越泓, 蔡跃明, 程崇虎译, 《通用分组无线业务技术与应用》, 人民邮电出版社, 2004
- [7] Timothy Pratt, Charles Bostian, Jeremy Allnutt, 甘良才译, 《卫星通信》(第二版), 电子工业出版社, 2005
- [8] 李昌猷, “利用 GEO 卫星接入因特网”, 电子与信息学报, 1009-9336(2001)05-10
- [9] 韩松, 邓迎春, “卫星 TCP/IP 数据传输技术”, 现代电信科技, 2002Vol2: 11-15
- [10] Akyildiz I.F, Su W, Sankarasubramaniam Y, Cayirci E, “Wireless sensor network: A survey”, Computer Networks, 2002,38(4): 393-422
- [11] 802.15.4-2003 IEEE Standard for Information Technology-Part 15.4: Wireless Medium Access Control(MAC) and Physical Layer(PHY) specifications for Low Rate Wireless Personal Area Networks( LR-WPANS), 2003
- [12] J Postel, “Transmission control protocol”, RFC 793, 1981
- [13] M.Allman, V.Paxson, W.Stevens, “TCP congestion control”, RFC 2581, April, 1999
- [14] W.Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms”, RFC 2001, Jan 1997
- [15] 马刘非, 《卫星通信网络技术》, 国防工业出版社, 2003
- [16] 倪冰, 程玲, 杨龙祥, 《基于卫星的 Internet 关键技术》, 电力系统通信, 2003Vol.24: 13-16
- [17] <http://grouper.ieee.org/groups/802/11/>
- [18] The Institute of Electrical and Electronical Engineers, IEEE Std 802.11-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999 edition
- [19] The Institute of Electrical and Electronical Engineers, IEEE Std 802.11-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 1999 edition
- [20] <http://ieee802.org/15/pub/TG4.html>
- [21] 龙承志, 马玉秋, 沈树群, 《基于低速率的短距离无线通信网络新技术—ZigBee》, 数据通信, 2005No.1: 60-64
- [22] 钟晓峰, 王有政, 梅顺良, 王京, 《基于时分系统的无线自组织网络同步算法》, 清

华大学学报（自然科学版），2005Vol.45: 95-98

[23] 《TCP/IP 详解卷 1: 协议》，人民邮电出版社，2000

[24] V.Jacobson, "Congestion avoidance and control", Proc.SIGCOMM Symposium on Communication Architectures and Protocols, ACM SIGCOMM Stanford, CA, Aug 1988: 314-329

[25] V. Jacobson, "Modified TCP congestion avoidance algorithm", end2end-internet mailing list, April 30, 1990

[26] 《用 TCP/IP 进行网际互联第二卷: 设计、实现与内核》，人民邮电出版社，2000

[27] S.Floyd, T.Henderson, "The NewReno modification to TCP's fast recovery algorithm", RFC 2582, Apr 1999

[28] Vern Paxson, Mark Allman, "Computing TCP's Retransmission Timer", RFC 2988

[29] L.S.Brakmo, L.L.Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet", IEEE Journal on Selected Areas in Communications 13(8)1995: 1465-1480

[30] Lawrence S.Brakmo, Sean W.Omalley, Larry L.Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", Proceedings of ACM SIGCOMM'94, no.4: 24-35, October 1994

[31] Jong Suk Ahn, Peter B.Danzig, Zhen Liu, Limin Yan, "Evaluation with TCP Vegas: Emulation and Experiment", ACM SIGCOMM Computer Communications Review, vol.25: 185-195, August 1995

[32] Jeonghoon Mo, Richard J.La, Venkat Anantharam, Walrand, "Analysis and Comparison of TCP Reno and Vegas", INFOCOM'99 vol.3: 1556-1563

[33] M.Mathis, J.Mahdavi, S.Floyd, A.Romanow, "TCP selective acknowledgement options", RFC2081, October 1996

[34] S.Keshav, S.Morgan, "SMART retransmission: Performance with overload and random losses", Proc.IEEE INFOCOM'97, 1997

[35] A.Bakre, B.Badrinath, "I-TCP: Indirect TCP for mobile hosts", Proc.15th International Conf.on Distributed Computing Systems (ICDCS, 1995

[36] Prasun Sinha, Narayanan Venkitaraman, Raghupathy Sivakumar, Vaduvur Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks", Mobicom 99

[37] H.Balakrishnan, V.N.Padmanabhan, S.Seshan, R.Katz, "A Comparison of Mechanisms for Improving TCP performance over wireless links", Proc.ACM SIGCOMM'96, PaloAlto, CA, 256-269, Aug 1996

[38] R.Wang, M.Valla, M.Y.Sanadidi, M.Gerla, "Adaptive Bandwidth Share Estimation in TCP Westwood", In Proc.of Globecom 2002

[39] <http://www.isi.edu/nsnam/ns/>

[40] CMU Monarch Group, CMU monarch extensions to the NS-2 simulator URL: <http://monarch.cs.cmu.edu/cmu-ns.htm>

- [41] Kanth K, Ansari S, Melikri MH, "Performance enhancement of TCP on multihop ad hoc wireless networks", IEEE PWC, Pages: 90-94, Dec 2002
- [42] Eshak N, Baba MD, "Improving TCP performance in mobile ad hoc networks", IEEE APCC, Vol2:580-583, Sep 2003
- [43] Qianwen Lin, Kwang-Mien Chan, Kean-Soon Tan, Boon-Sain Yeo, "Performance analysis of ad-hoc networks partitioning on TCP", IEEE VTC, Vol4:2444-2448, May 2005
- [44] Chandran K, Ragbunathan S, Venkatesan S, Prakash R, "A feedback based scheme for improving TCP performance in ad-hoc wireless networks", IEEE DCS, Pages: 472-479, May 1998
- [45] Dongkyun Kim, Toh C K, Yanghee Choi, "TCP-BuS: improving TCP performance in wireless ad hoc networks", IEEE ICC, Vol3:1707-1713, June 2000
- [46] J. Liu, S. Singh, "ATCP: TCP for mobile Ad hoc networks", IEEE J-SAC, 19(7):1300-1315, July 2001
- [47] G.Anastasi, E.Ancillotti, M.Conti, A.Passarella, "TPA: a Transport Protocol for Ad hoc Networks", IEEE ISCC, pages:51-56, June 2005

## 致谢

在这篇论文即将结束之际，也是自己的研究生学习结束之时。搁笔之际，心情复杂。

在这短暂的两年半的时间里，自己学到了许多的东西，有做人的道理，有做学问的知识。可是，对科学的颠峰，感觉还是很遥远。

在这里，我要感谢我的恩师周金芳教授。是她，带领我走进一个向往已久的知识领域。也是从她身上，我学到了学习的方法和工作的经验。她那严谨治学的态度，渊博的专业知识和对事业的热爱，都深深地感染着我，并成为我今后学习、工作和生活地宝贵精神财富。

在这里，我还要感谢钱颖博士和贾晨军博士，曹猛虎硕士、孙妮硕士、曲亮硕士以及我的同学们，在研究生学习阶段，他们都给了我许多的关心和帮助。

最后，我想以一句话来结束我得论文——路漫漫其修远兮，吾将上下而求索。