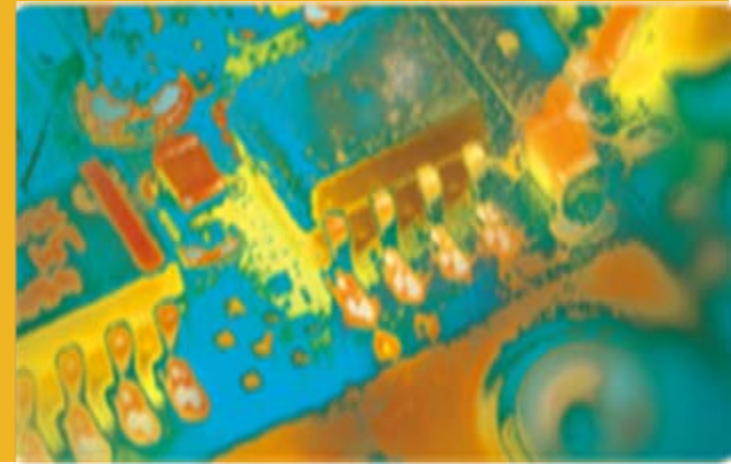




温州大学  
WENZHOU UNIVERSITY

# 第3章 三层网络测试



<http://network.wzu.edu.cn>  
<http://www.spirent.com>  
[linflag@163.com](mailto:linflag@163.com)

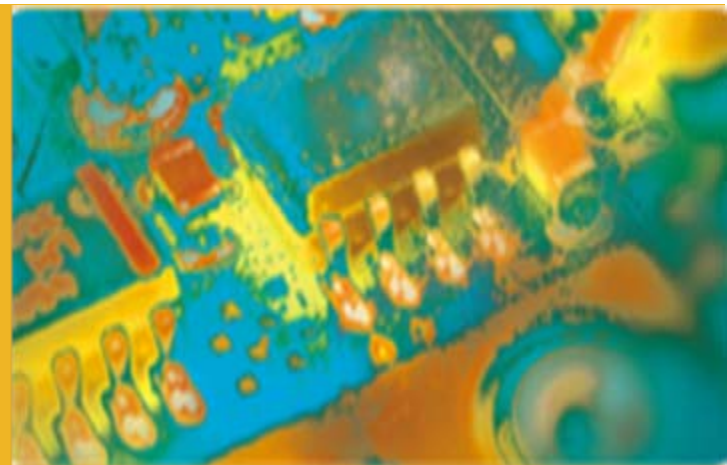
# 第3章学习导航

驱动问题	涉及的知识或技能点	学习要求
为什么要进行三层测试？	三层测试的必要性	理解
路由器是如何工作的？	<b>IP</b> 分组转发的工作原理	掌握
三层 <b>IP</b> 网络测试关注哪些问题？涉及哪些技术指标？	<b>IP</b> 网络测试的主要技术指标	掌握
如何进行测试？是否有相应的规则或方法可循？	<b>IP</b> 网络测试的 <b>RFC</b> 文档	理解
如何针对特定的测试进行方法设计？	<b>IP</b> 网络测试方法学	掌握



温州大學  
WENZHOU UNIVERSITY

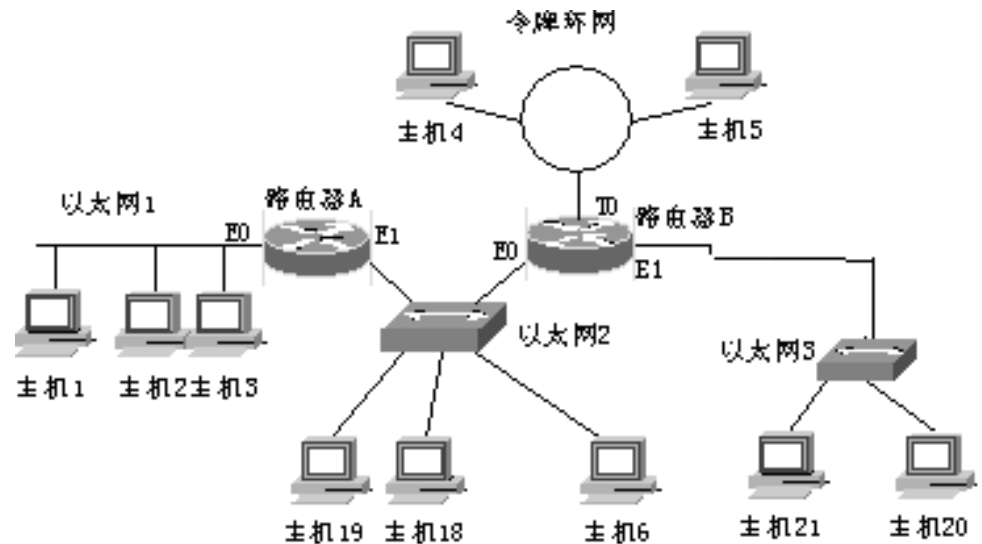
# 第7章 网络服务质量 测试



## 3.1 OSI三层测试的必要性

# OSI三层功能概述

- ✓ 任务：将分组从源主机传送到目的主机
- ✓ 解决方法：主机寻址方案、路由选择、异构网络互连
- ✓ 核心设备：路由器



# 三层测试的必要性

## ✓ 三层功能的重要性:

- 关系到整个通信子网的运行控制
- 直接影响到整个网络的通信质量

## ✓ 三层测试的运用:

### ➤ 研发与生产

- 阶段测试为产品研发过程提供有效的反馈信息;
- 产品研发完成时的合格测试;
- 为在运营商网络中使用的产品进行的入网认证测试。

### ➤ 网络规划部署

- 设备选型，特别是对指标有特殊要求，对功能与性能有严格规定，或对厂商承诺的指标有怀疑时。

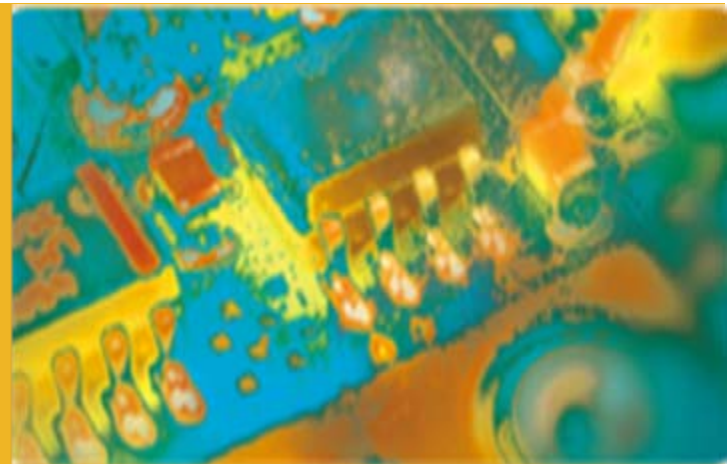
### ➤ 维护

- 故障诊断
- 性能评价与分析
- 网络升级或优化



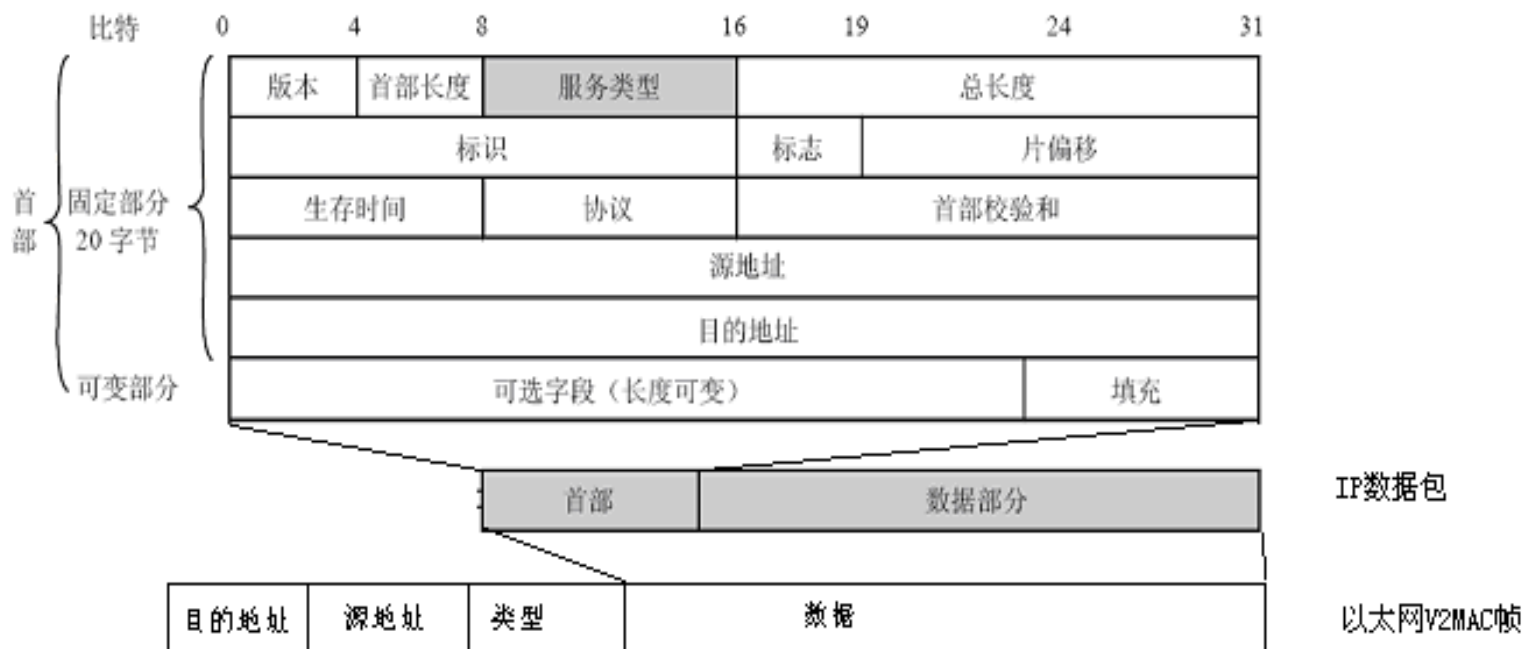
温州大学  
WENZHOU UNIVERSITY

# 第3章 三层网络测试



## 3.2 三层IP网络技术概述

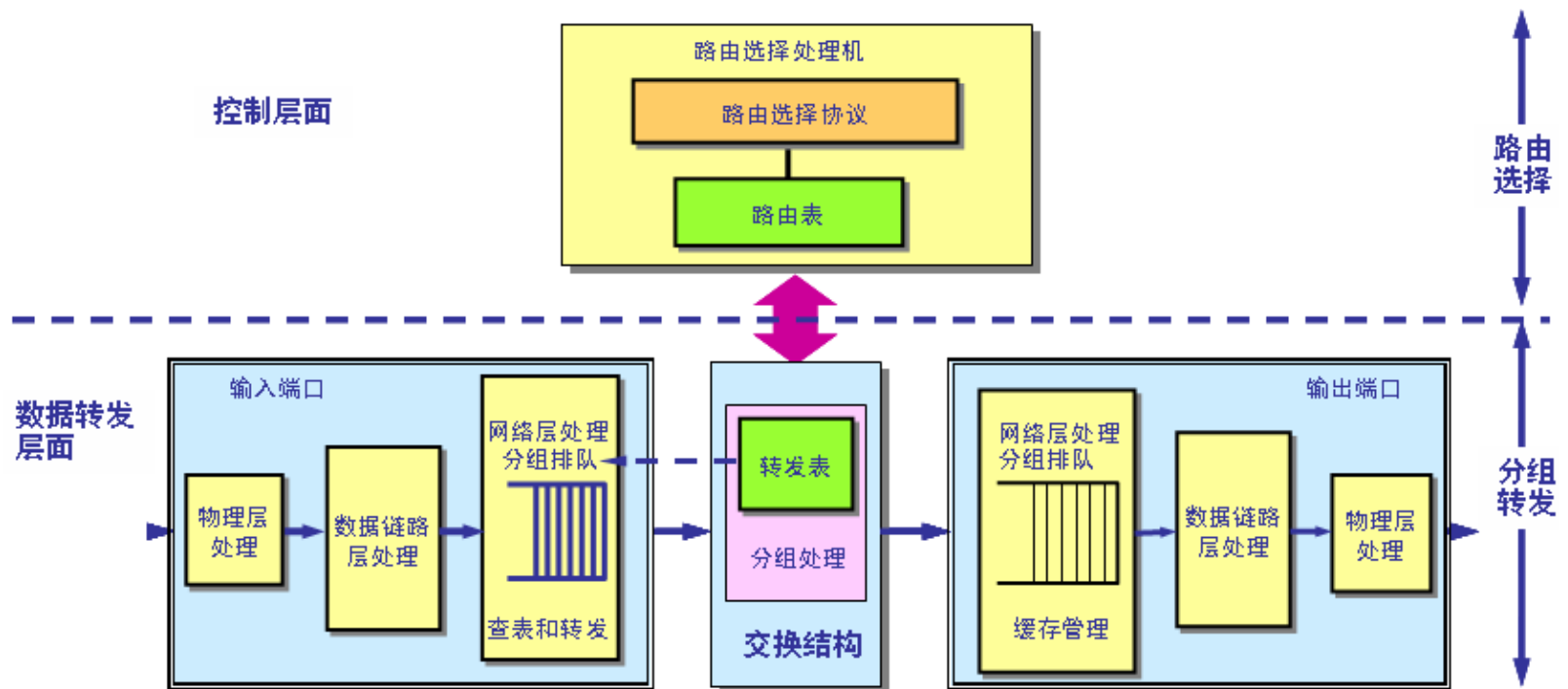
# IP协议



✓ 注意：封装和拆封过程中的二层和三层地址的对应关系

# 路由器的组成（1）

- 具有多个输入端口和多个输出端口的专用计算机
- 路由选择部分（控制层面）、分组转发部分（数据转发层面）





# 路由器的组成（2）

## 1.路由选择部分

- ✓ 核心构件:路由选择处理机
- ✓ 运行系统软件和路由协议，路由协议可以有不同的类型。
- ✓ 交换网络的拓扑结构信息
- ✓ 依照拓扑结构动态更新和维护路由表

# 路由器的组成（3）

## 2.数据包转发部分

- 功能：主要是根据控制层面生成的转发表（**FIB**）转发**IP数据包**。
- **注意**：转发表是根据路由表生成的，其表项和路由表项有直接对应关系，但转发表的格式和路由表的格式不同，它更适合实现快速查找

### （1）交换结构

- 作用：根据转发表对来自某个输入端口的数据包进行处理，并使之从一个合适的输出端口转发，以完成输入端口和输出端口之间的互连功能。
- 常用的**3种**交换结构：总线型交换、共享存储型交换和交叉开关型交换

# 路由器的组成（4）

## 2.数据包转发部分

### （2）输入端口

- 输入端口物理链路的连接点，也是数据包的接收点。
- 端口通常由线卡提供，一块线卡一般支持**4、8或16**个端口。每个输入端口都有三个模块，它们分别是对应于物理层、数据链路层和网络层的模块。
- 分组的处理：
  - 如果是路由交换信息——》路由选择处理机
  - 如果是数据——》查找转发表
  - 一个分组未处理完，又到了新的分组，则在**输入队列**中排队

# 路由器的组成（5）

## 2.数据包转发部分

### （3）输出端口

#### ✓ 主要功能：队列和缓冲管理

- 从交换结构接收数据包，然后将它们发送到路由器的线路接口卡上。
- 当交换结构传送的数据包的速率超过输出链路的发送速率时，来不及发送的数据包就暂时存放在队列中。
- 输出端口使用复杂的调度算法。
- 完成数据包的封装和支持物理链路协议等。

# 路由器的处理能力及影响

- ✓ 决定路由器处理数据包的能力的因素：
  - 输入端口的处理速率
  - 查找转发表的速率
  - 数据包交换的速率
  - 输出端口的处理速率
- ✓ 主要影响：排队？延时？队列的可用存储空间？  
丢弃数据包？
- ✓ 衡量路由器性能的重要参数是路由器每秒能够处理的数据包数。

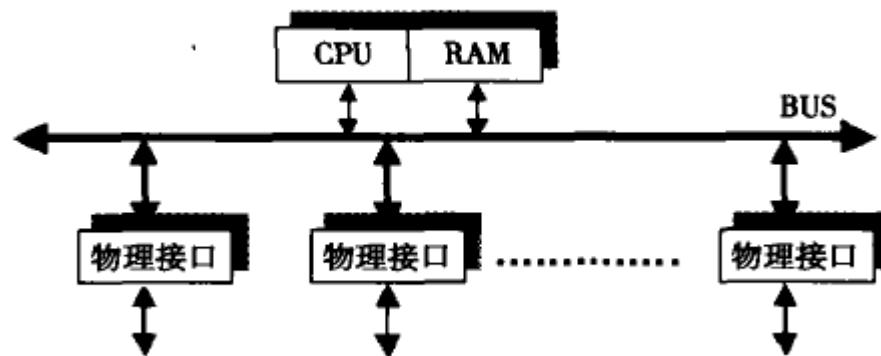
# 路由器的体系结构（1）

- ✓ 传输控制层面：运行在通用**CPU**系统中，多年来一直没有多少变化。在高可用性路由器设计中，大多采用双主控进行主从式备份，以保证传输控制平面的可靠性。
- ✓ 数据转发层面：为适应不同的线路速度，不同的系统容量，采用了不同的实现技术。
- ✓ 路由器的结构体系经历了从单总线单**CPU**结构到多总线多**CPU**、共享内存式结构，从支持简单的静态路由到自学习的动态路由的发展过程，技术演进速度非常快。

# 路由器的体系结构（2）

## 1. 第一代单总线单CPU结构路由器

- ✓ 实例：接入式路由器，如Cisco 2501
- ✓ 特点：处理速度慢、系统容错性差、价格较低

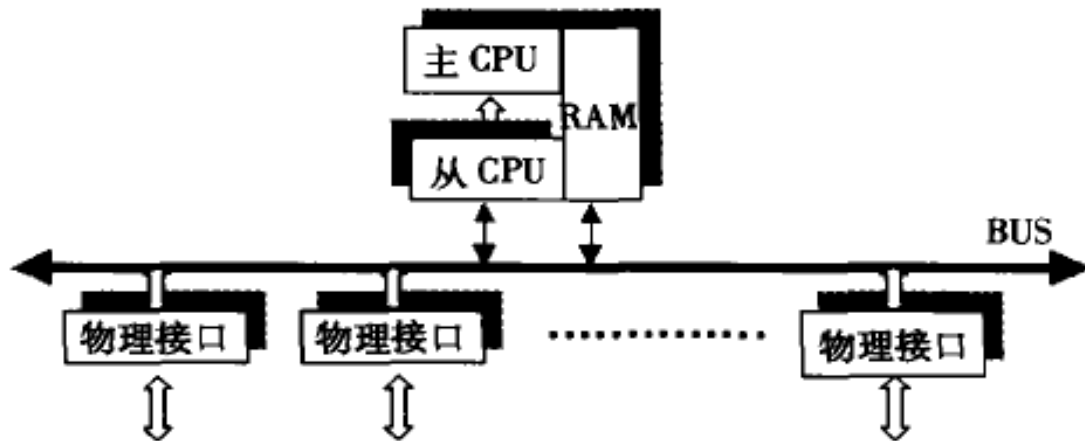


# 路由器的体系结构（3）

## 2. 第二代单总线主从CPU结构路由器

✓ 特点：第1代的简单扩展，对系统的容错性能和速度的提高贡献非常有限

➤ 实例：3Com公司的NetBuilder 2



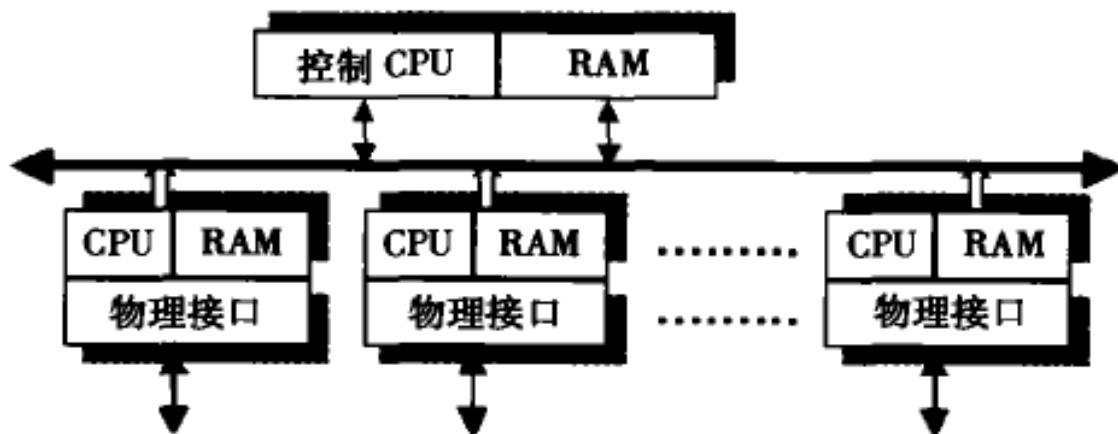


# 路由器的体系结构（4）

## 3. 第三代单总线对称多CPU结构路由器

✓ 特点：提高了网络接口处理能力、转发决策由软件来实现、采用共享总线

➤ 实例：北电的Bay BCN

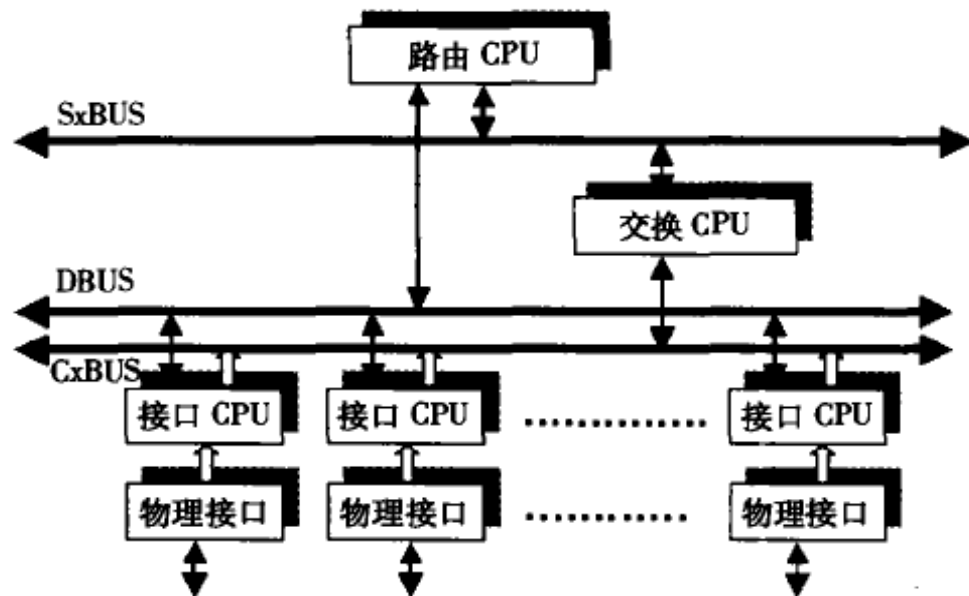


# 路由器的体系结构（5）

## 4. 第四代多总线多CPU结构路由器

✓ 特点：结构非常复杂，性能和功能也非常强大，采用硬件快速查找路由表

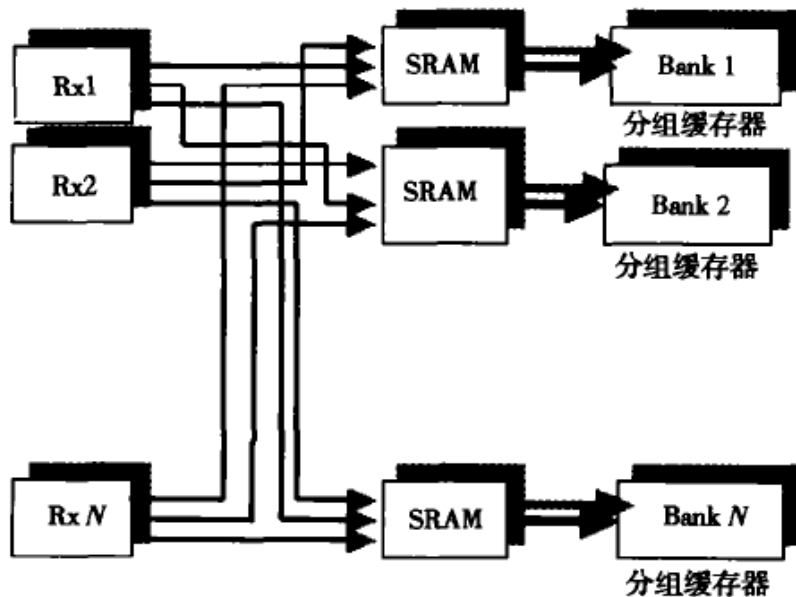
➤ 实例：Cisco7000系列



# 路由器的体系结构 (6)

## 5. 第五代共享存储型结构路由器

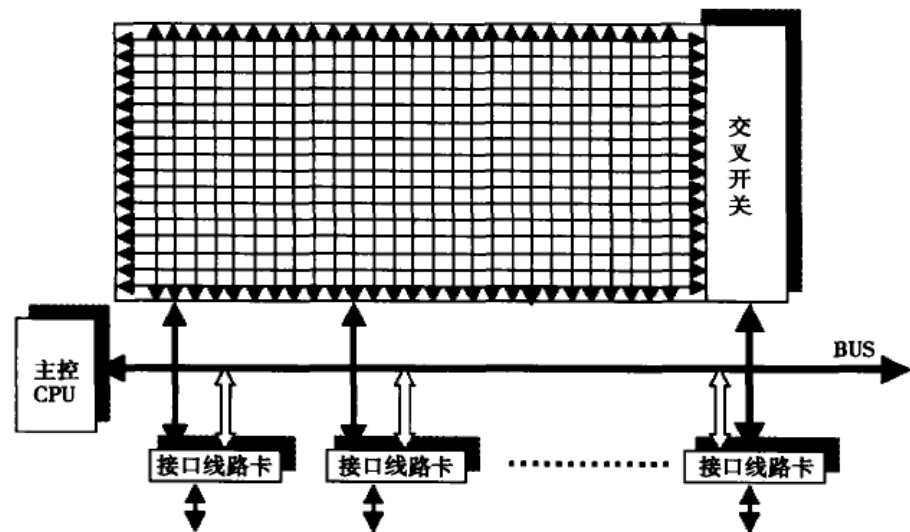
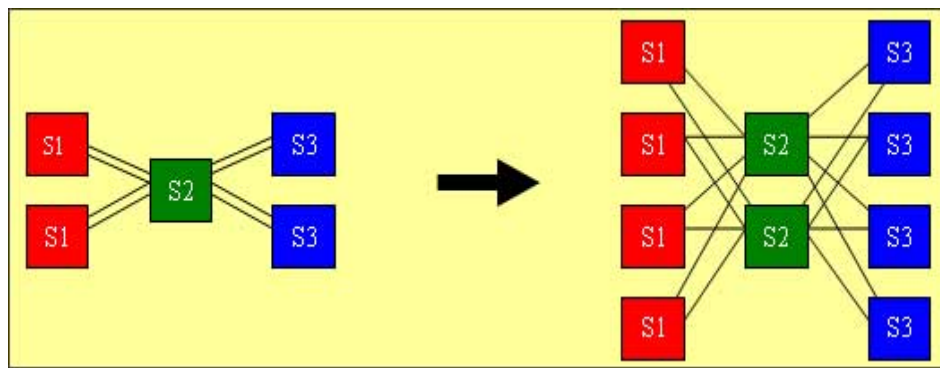
- ✓ 特点：交换带宽主要由存储器的带宽决定、不利于进一步发展



# 路由器的体系结构 (7)

## 6. 第六代交叉开关 / 交换式体系结构路由器

- ✓ 特点：采用基于硬件专用芯片**ASIC**的交叉开关、系统的吞吐量大
- ✓ 实例：高性能路由器



# 路由器的体系结构（8）

## 7. 第七代多级交换路由器

- ✓ 特点：采用网络处理器，对新的应用需求与协议变化的适应能力强
- ✓ 实际上，IP路由器速度的快速提高，还归功于采用了以下3种新技术：
  - 硬件体系结构
  - ASIC技术
  - 三层交换

# 路由器工作原理（1）

路由器必须完成两个基本功能：路由处理和  
数据包转发处理

- ✓ 路由处理：收集网络拓扑信息并动态形成路由表（**RIB**）
- ✓ 数据转发层面：根据转发表（**FIB**）转发IP数据包
- ✓ **RIB** 与**FIB**关系：**RIB** ——》**FIB**，供转发数据包时的快速查找

# 路由器工作原理（2）

## ✓ 主要流程：

- 线路输入：信号恢复、译码和**CRC**校验，放进输入**FIFO**队列
- 数据存储
- 数据包头部分析：主要是三层目的**IP**地址，甚至包括二层和四层数据头部
- 数据包头部修改：包括**TTL** 值减**1**、更新**IP**头部校验和、替换二层**MAC**地址等，——》输出**FIFO**队列
- 线路输出

# 路由器类型 (1)

## 1. 按性能档次分高、中低端路由器

- ✓ 通常按照背板交换能力划分，如**>40Gbps** 高端路由器，**25Gbps~40Gbps**之间 中端路由器
- ✓ 综合指标

## 2. 从结构上分为模块化路由器和非模块化路由器

- ✓ 模块化结构：中、高端路由器
- ✓ 非模块化结构：低端路由器



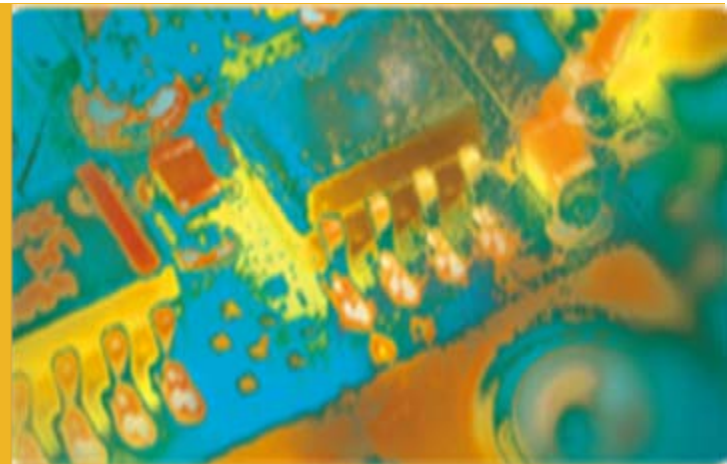
# 路由器类型（2）

- 3.按所处网络位置划分通常把路由器划分为边界路由器和中间节点路由器
- 4.从性能上分为线速路由器和非线速路由器
- 5.从应用划分为通用路由器与专用路由器。
- 6.从功能上划分，可将路由器分为骨干级路由器、企业级路由器和接入级路由器



温州大學  
WENZHOU UNIVERSITY

## 第3章 三层网络测试



### 3.3 决定三层IP网络性能的主要指标

# 数据传输层面相关的指标（1）

- ✓ 关注的是路由器基于数据包的数据转发能力
- ✓ 主要考察数据包的转发是否准确、有无丢包、转发延迟多大等

# 数据传输层面相关的指标（2）

## 1. 吞吐量

- ✓ 定义：指在没有丢包的情况下，路由设备能够转发的最大速率。
- ✓ 两层含义：
  - 保证待转发的每一个包都能够从输入端口准确地转发到适当的输出端口
  - 被测设备在不丢包的前提下，单位时间内能够转发的最大包数量。
- ✓ 吞吐量的衡量单位：每秒转发的数据包的数量。
- ✓ 网络设备性能的首要指标

# 数据传输层面相关的指标（3）

## 1. 吞吐量

### ✓（1）整机吞吐量：

✓指设备整机的包转发能力，是设备性能的重要指标。整机吞吐量通常小于路由器所有端口吞吐量之和。

### ✓（2）端口吞吐量：

✓指路由器在某端口上的包转发能力。

# 数据传输层面相关的指标（4）

## 2. 延迟

指分组的最后一位比特进入路由器到第一位比特离开路由器的时间间隔

产生原因：

- ✓ 转发数据包时完成的一系列操作，如计算路由、地址解析等需要一定时间
- ✓ 二是数据包等候处理时需要排队

语音、图像和视频：对时间极为敏感

# 数据传输层面相关的指标（5）

## 3. 丢包率

- ✓ 指路由器在稳定状态负载下，由于缺乏资源而不能被网络设备转发的包占有所有应该被转发的包的百分比。
- ✓ 丢包率 =  $(X - Y) \times 100 / X$
- ✓ 主要反映网络设备在超负荷运转下的性能
- ✓ 丢包率与吞吐量：在吞吐量的范围内丢包率为**0**，超过吞吐量就会引起丢包

# 真实的故事 - IPTV业务（无丢包）



采用思博伦通信  
**Anue**模拟丢包



# 真实的故事 - IPTV业务 (0.5%丢包)



采用思博伦通信  
**Anue**模拟丢包

# 真实的故事 - IPTV业务 (5%丢包)



采用思博伦通信  
**Anue**模拟丢包

# 数据传输层面相关的指标（6）

## 4. 背对背

- ✓ 含义：指的是在一段较短的时间内，以合法的最小帧间隙在传输媒介上连续发送固定长度的包不引起丢包时的包数量。
- ✓ （被测试设备再某个速率下能转发的最长的连续帧数目）
- ✓ **IEEE**规定的以太网帧间的最小帧间隙为**96**比特。
- ✓ 该指标用于测试路由器缓存能力。
- ✓ 对于吞吐量为**100%**线速的路由器来说，该指标值为无限大。

# 数据传输层面相关的指标（7）

## ✓ 背对背帧思考

- 突发帧（组帧）是互联网的一种现实
- 许多应用数据会被分段为一组帧，任何帧的丢弃将会影响数据重组
- 网络中的一些设备能够根据请求产生高达**64K**字节的数据块，网络将会把这些数据块分段成一组连续的帧进行发送。（例如使用**NFS**或**rdump**的设备）
- 路由更新信息也可能被分段到几个报文中，丢帧将可能导致路由不可达

# 数据传输层面相关的指标（8）

## 5. 时延抖动

- ✓ 含义：指时延的变化。
- ✓ 对时延抖动敏感的业务，如语音、视频，该指标才有测试的必要性。

# 数据传输层面相关的指标（9）

## 6. 背板能力

- ✓ 背板指路由设备输入与输出端口间的物理通路，如共享背板、可交换式背板
- ✓ 背板能力只能在设计中体现，一般无法直接测试。
- ✓ 背板能力能够体现在路由器吞吐量上，背板能力通常大于依据吞吐量和测试包长所计算的值。

# 数据传输层面相关的指标（10）

## 7.系统恢复

- ✓ 系统恢复特指路由设备从超载行为状况下恢复到正常转发状态的行为特征，主要表现为系统恢复速度的快慢。

## 8.系统重启

- ✓ 系统重启时间越小，即因电源开关或系统重置而产生的等待时间就越小，网络的丢包也就越少，由此产生的网络不可用时间也就越短。

# 传输控制层面相关的指标

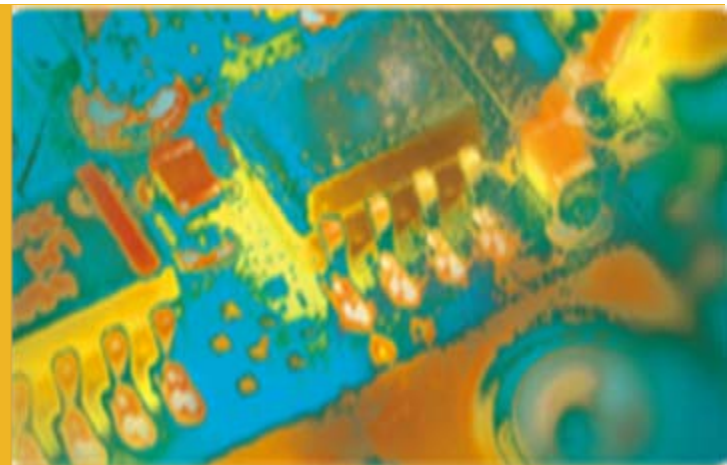
- ✓ 因网络类型的差异，考虑的问题也会有所不同
- ✓ 主要指标：
  - 路由震荡
  - 路由表容量
  - 路由收敛时间等





温州大学  
WENZHOU UNIVERSITY

## 第3章 三层网络测试



### 3.4 三层IP网络性能测试相关的 RFC文档

# 相关RFC文档（1）

- ✓ RFC1242、RFC2544、RFC2285、RFC2889、RFC2328、RFC3222 等

## 3.4.1 RFC1242概述

- ✓ 为网络互连设备的基准测试定义了基本术语和概念
  - 背靠背、恒定负载、数据链路帧大小、丢帧（包）率、帧间空隙、延迟、重启行为、吞吐量

# 相关RFC文档（2）

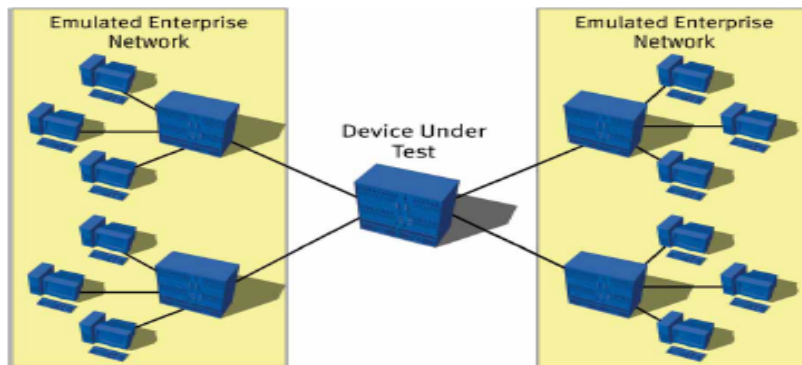
## 3.4.2 RFC2544概述

- ✓ 通常被称为网络测试的“圣经”
- ✓ 对网络二层到七层性能的衡量都有着重要指导作用
- ✓ 最主要内容是基准测试
  - 吞吐量（Throughput）
  - 丢包率（Packet Loss Rate）
  - 延迟（Latency）
  - 背靠背（Back-to-back）
  - 系统恢复（System recovery）
  - 重启（Reset）

# RFC 2544/1242—网络通信测试的基础

思博伦通信

提供业界标准的  
RFC2544测试



测试产品:

**Spirent TestCenter**

**SmartBits**

**AX/4000**

## 主要测试指标

- ✓吞吐量—Throughput
- ✓延迟—Latency

- ✓丢包率—Frame Loss Rate
- ✓背对背缓冲—Back to Back Frames

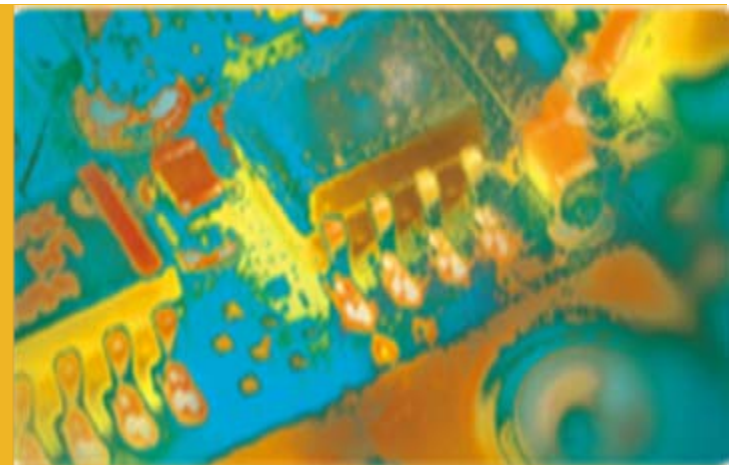
## 适用的场景

- ✓几乎适用于我们目前所有的2~3层设备和网络的测试
- ✓交换机、路由器、防火墙、VPN... ..
- ✓各种类型和拓扑的网络



温州大学  
WENZHOU UNIVERSITY

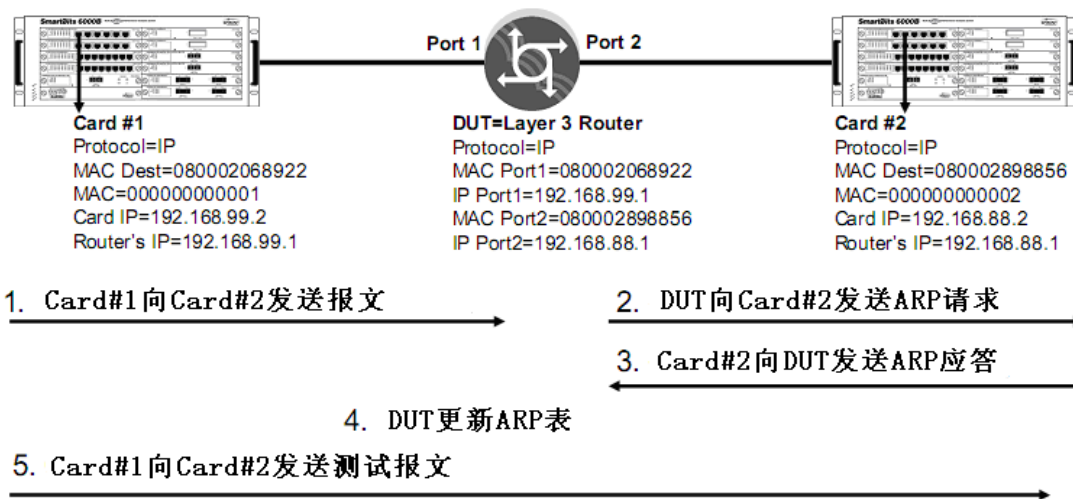
## 第3章 三层网络测试



### 3.5 三层IP网络性能测试的基本方法

# 测试设置中的地址学习

- 技术指标关注数据包、以帧为对象定义每条测试流量
- 目的：确保被测设备能够在稳定的工作状态下直接转发每个**IP**测试数据包
- 要求：先学习，再验证，后测试



# 吞吐量的测试方法（1）

- ✓ 测试目的：找到在没有丢包的情况下路由器能够转发的最大速率
- ✓ 判定唯一标准：零丢包
- ✓ 两个方面的性能：整机吞吐量、端口吞吐量

	参与端口	测试对象
端口吞吐量	一对同种类型端口	端口和线卡的能力
整机吞吐量	全部或者多种类型的端口	背板或者交换网络的能力

# 吞吐量的测试方法（2）

- ✓ 假设每次测试时速率的变化量为 $dv$ ，如果在某个速率 $v$ 下的包能够正确转发，但在速率 $v+dv$ 时包出现丢失，则 $v$ 就是吞吐量。
- ✓ 具体测试过程：
  - 步进
  - 二分搜索法
  - 混合



# 吞吐量的测试方法（3）

## ✓ 几点注意：

- 初始帧速率可以取理论最大传输速率的某个百分比，也可以通过前一个测试案例的结果加以适当调整获得。
- 由于包丢失是一个随机过程，为使测量结果尽量接近路由器的实际性能，测试时要持续较长的时间，选取较小的变化量，重复多次测量求平均值。
- 吞吐量随帧大小的不同而改变。

# 吞吐量的测试方法（4）

## ✓ 测试条件：

- 时长：每次持续**60**秒以上
- 帧尺寸：**RFC2544**对以太网测试帧大小的建议为**64, 128, 256, 512, 1024, 1280, 1518**字节

# 吞吐量的测试方法（5）

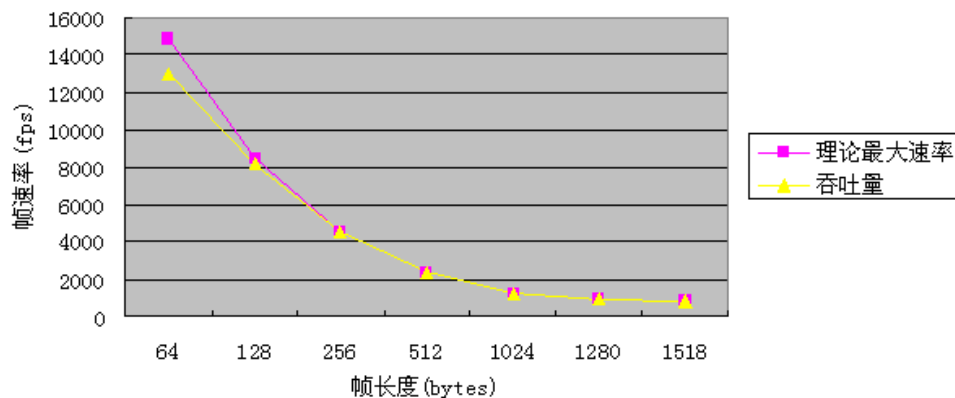
各种网络下，帧长与最大理论速率的关系

帧长(bytes)	以太网(fps)	16Mb令牌环(fps)	FDDI(fps)
64	14880	24691	152439
128	8445	13793	85616
256	4528	7326	45620
512	2349	3780	23585
768	1586	2547	15903
1024	1197	1921	11996
1280	961	1542	9630
1518	812	1302	8138

# 吞吐量的测试方法 (6)

测试结果:

吞吐量



帧长度 (bytes)	理论最大速率 (fps)	吞吐量 (fps)
<b>64</b>	<b>14880</b>	<b>13000</b>
<b>128</b>	<b>8445</b>	<b>8200</b>
<b>256</b>	<b>4528</b>	<b>4500</b>
<b>512</b>	<b>2349</b>	<b>2349</b>
<b>1024</b>	<b>1197</b>	<b>1197</b>
<b>1280</b>	<b>958</b>	<b>958</b>
<b>1518</b>	<b>812</b>	<b>812</b>

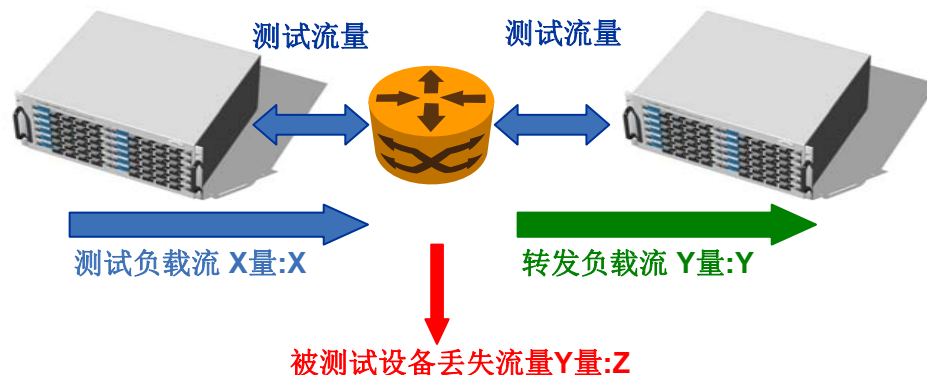
# 丢包率的测试方法（1）

- ✓ 测试目的：确定**DUT**在不同的负载和帧长度条件下的丢包率。
- ✓ 考虑**DUT**在**超负荷**情况下的异常行为，包括包的丢失、重发、乱序和错发等
- ✓  $(( \text{发送包数量} X - \text{接收包数量} Y ) * 100 ) / \text{发送包} X \text{数量}$

# 丢包率的测试方法（2）

- ✓ 影响丢包率的因素：
  - 包长度
  - 包发送速率
- ✓ 丢包率和吞吐量密切相关
- ✓ 主要反映网络设备在超负载运转情况下的处理策略
  - 对各种不同类型帧的丢弃策略

# 丢包率的测试方法 (3)



- ✓ 实际测试时负载的范围应介于吞吐量和最大传输速率之间
- ✓ 测试的帧需要编号，接收时要通过查对每个帧的编号判断可能出现的各种不同的异常情况。
- ✓ 在统计接收到的帧数时，帧的重发和乱序不能算作丢失，但不能重复统计重发帧。

# 丢包率的测试方法（4）

- ✓ 关心设备的整体性能和在某一具体应用的行为
- ✓ 3种测试流量模式：
  - 1) 一对端口发送和接收：**DUT**一个端口接收到的数据经另一个端口转发。
  - 2) 部分网状多对一端口测试：**DUT**多个端口接收到的数据都由一个端口转发。
  - 3) 多对多全网状测试：**DUT**每一个端口接收到的数据都要送给其余任意一个端口。



# 丢包率的测试方法（5）

两种不同的测试过程：

## 1) 吞吐量未知的情况下

- 在第一次的测试中，应该以测试帧长相应的**100%**的最大传输速率（**maximum frame rate**）发送一定数量的帧，并计算丢包率。
- 接下来使用最大传输速率的**90%**，之后用**80%**，依此类推，直到在整个测试序列当中，出现两个连续的测试没有出现丢包为止。
- 传输速率最大的测试步长是最大传输速率的**10%**，建议小于**10%**，鼓励使用更合适的步长来进行测试。

## 2) 如果事先已经测试得到吞吐量，测试时将吞吐量设为最低负载，然后逐渐增加负载，观察丢包的情况。

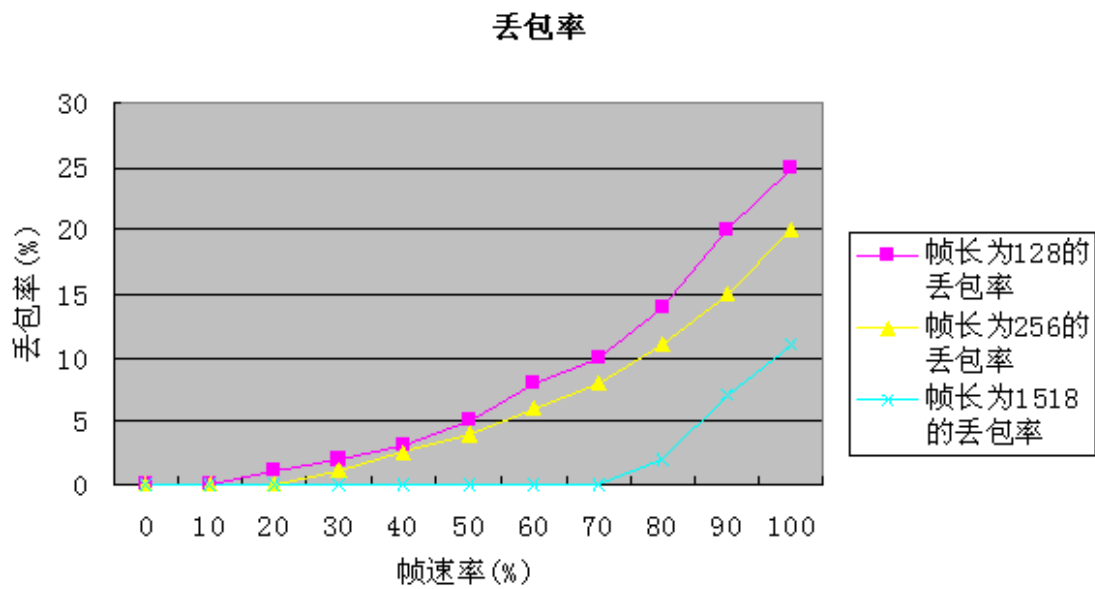
# 丢包率的测试方法（6）

## ✓ 测试条件：

- 发送的持续时间最小为**60秒**
- 最大传输速率
- 吞吐量
- 不同格式的帧要分别测试

# 丢包率的测试方法 (7)

## ✓ 测试结果



# 延迟的测试方法（1）

- ✓ 是基准测试的重点
- ✓ 测试目的：确定数据包经过**DUT**传输的所需要的时间
- ✓ 反映**DUT**执行转发操作的速度以及转发队列的处理情况，延迟越小，说明路由器处理数据包的速度越快

# 延迟的测试方法（2）

✓ 延迟测试必须要测得以下两个参数：

实施困难

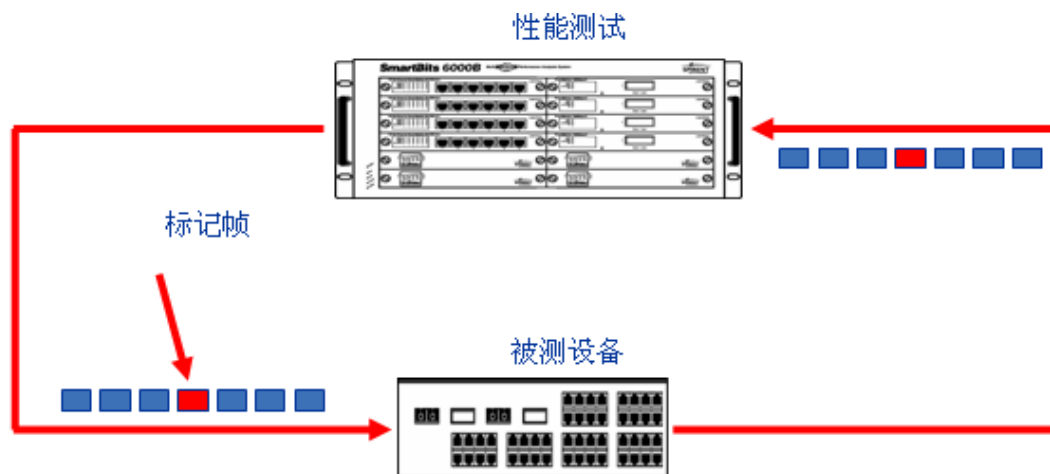
- (1) 输入帧的最后一位到达输入端口的时刻；
- (2) 输出帧的第一位出现在输出端口的时刻。

✓ 可用的方法：

- 整个报文的延迟是和报文中任意位的延迟是相等的，引入了标记帧方法来测试延迟。

# 延迟的测试方法（3）

- ✓ 对带有标记的帧的要求：
  - 不能在传输过程中丢失
  - 被转发的时候网络设备应该已经工作在稳定状态，即标记帧设置在测试流的中间



# 延迟的测试方法（4）

## ✓ 延迟与数据包长度

- 包长越长则延迟越大，但延迟并不是随着包长的变化而线性变化

## ✓ 延迟与链路速率（发包速率）

- 链路速率需限定在路由器端口吞吐量范围内，超过吞吐量测试延迟指标没有意义

# 延迟的测试方法（5）

## ✓ 具体的测试过程：

- 首先，为了确保在不丢包的情况下进行测试，要确定**DUT**在各个帧长下的吞吐量。
- 然后，针对每一个特定的帧长，以已经确定好的不超过吞吐量的发送速率发送该帧的数据流。
- 该发送过程应该维持至少**120**秒钟。在数据流发送**60**秒之后，在其中的某一个帧中打上与实现无关的标记，并记录下该帧发送的时间戳**A(timestamp A)**。
- 测试设备的接受端必须能够识别该标记信息，并记录下接收到携带该标记的帧的时间戳**B(timestamp B)**。
- 根据**RFC1242**所定义，延迟由时间戳**B**减去时间戳**A**来获得。



# 延迟的测试方法（6）

## ✓ 测试条件：

- 由于每次延迟测试存在抖动，该测试必须反复进行**20**次以上
- 覆盖不同帧尺寸进行测试
- 取记录的平均值

# 延迟的测试方法（7）

## ✓ 存在如下局限：

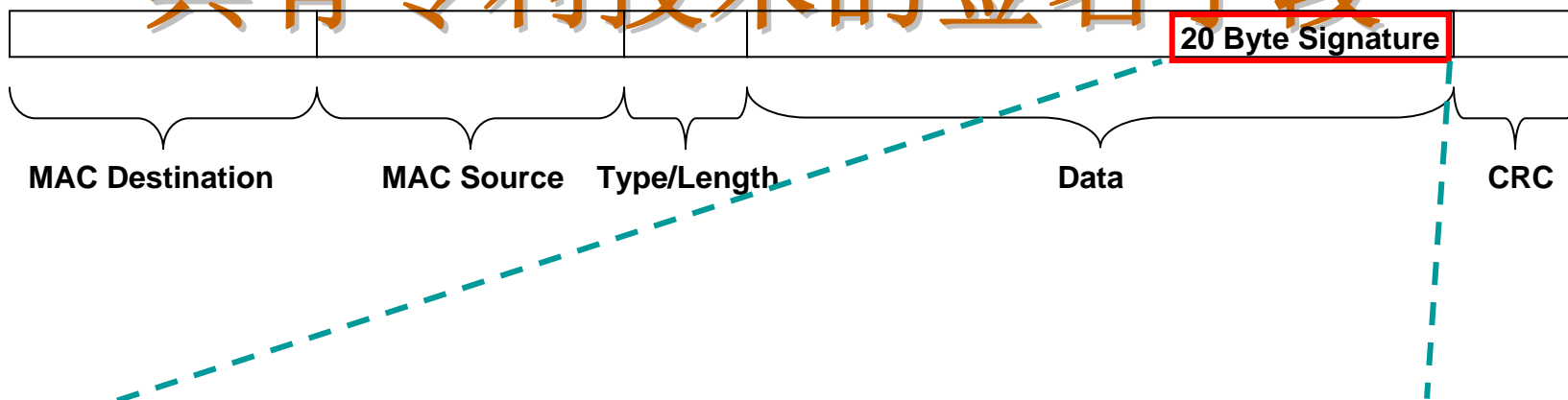
- 在测试流量中，将中间一个标记帧的延迟测试结果作为整个测试的结果；
- 没有考虑双向链路延迟可能不同；
- 必须要在没有丢包的条件下进行测试，因此必须先测试吞吐量；
- 单次测量结果的偏差可能较大，需要对20次以上的结果进行平均。

# 延迟的测试方法（8）

- ✓ 改进：给每个测试帧都打上特有的签名字段（**Signature**）
- ✓ 好处有：
  - 即使在有丢包的条件下延迟测量也能进行；
  - 不必按照**RFC2544**进行**20**次的测试，仅需进行较少次测试，通过求平均就可以得到测试结果；
  - 在一轮测试中可以同时完成延迟和吞吐量的测试；
  - 通过统计计算可获得除延迟之外新的延迟测量指标，如平均延迟、最大延迟、最小延迟、延迟分布等扩展延迟参数

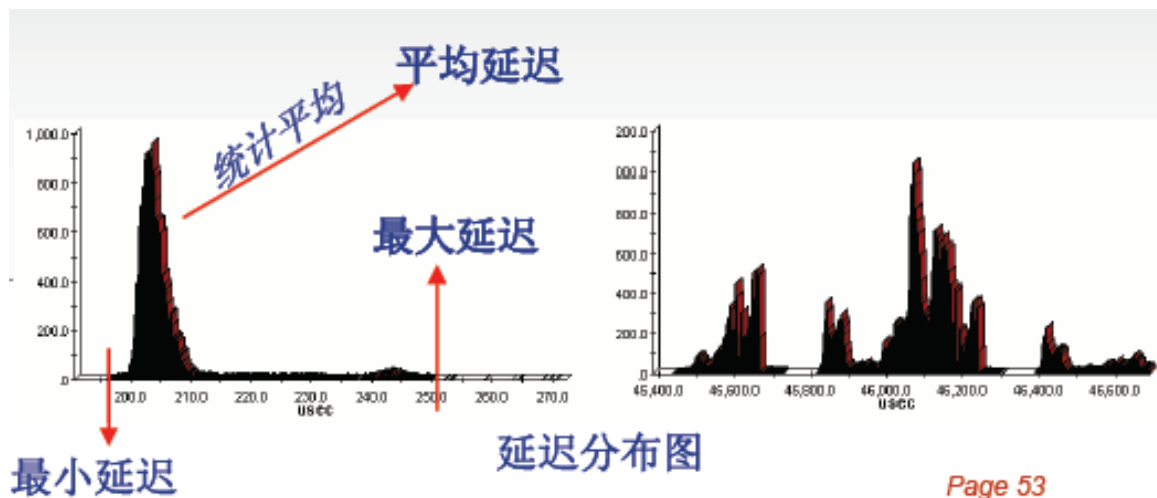
# Spirent TestCenter

## 具有专利技术的签名字段



Inverse of Sequence Byte0	Stream ID (MSB) Byte3	Stream ID Byte2	Stream ID Byte1
Stream ID Byte0	Sequence (MSB) Byte5	Sequence Byte4	Sequence Byte3
Sequence Byte2	Sequence Byte1	Sequence Byte0	Timestamp (2.5ns) Byte3
Timestamp (2.5ns) Byte2	Timestamp (2.5ns) Byte1	Timestamp (2.5ns) Byte0	TS (2.5ns) (bits 37:32) <small>PRBS t</small>
Crc16 Byte1	Crc16 Byte0	TCP UDP Checksum Cheater (Byte1)	TCP UDP Checksum Cheater (Byte0)

# 延迟的测试方法 (9)



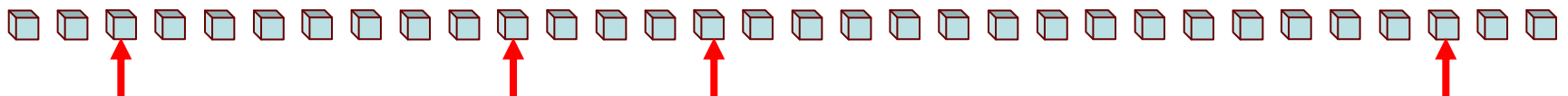
Page 53

延迟结果样表

帧长度 (bytes)	帧速率 (fps)	延迟 (ms)
64	13000	450
128	8200	480
256	4500	502
512	2349	562
1024	1197	658
1280	958	704
1518	812	775

# 背对背帧的测试方法（1）

- ✓ 测试目的：通过向被测设备发送具有合法最小帧间隙的突发数据包，确定被测设备在不丢包的情况下能够处理的最大数据包数目，从而考察路由器接口对于**突发数据的缓存能力**。
- ✓ 不同缓存区及分配策略：
  - 共享缓存区结构
  - 输入缓存区结构
  - 输出缓存区结构
  - 其他缓存区结构



# 背对背帧的测试方法（2）

- ✓ 背对背测试与吞吐量测试 的区别：
  - 吞吐量测试重在转发引擎的转发能力
  - 背对背测试重在接口缓存能力
- ✓ 联系：
  - 吐量不能达到最大理论值时，有必要进行背对背测试

# 背对背帧的测试方法（3）

- ✓ 测试流量模型：
  - 多介质混合
  - 一对端口
  - 部分网状
  - 全网状等情况
- ✓ 测试帧的长度也要覆盖各种情况



# 背对背帧的测试方法（4）

## ✓ 测试过程：

- 在每轮背对背测试中，以最大帧速率向**DUT**传送特定数量的测试帧，并统计**DUT**转发的帧数。
- 如果出现丢帧，则减少帧数（或测试持续时间）；否则增加帧数（或测试持续时间）。重复上述测试过程，直至找到没有丢帧时的最大帧数。
- **RFC2544**没有规定测试采用的算法，但最常用的算法是二分搜索算法。

# 背对背帧的测试方法（5）

## ✓ 测试条件：

- 发送具有合法的最小帧间隙的突发数据包的持续时间必须大于等于**2秒钟**
- 并且测试应该至少进行**50次**
- 最终取平均值。

# 背对背帧的测试方法（6）

✓ 测试结果表示：

帧长度 (bytes)	帧数 (平均数)
<b>64</b>	<b>37200</b>
<b>128</b>	<b>21112</b>
<b>256</b>	<b>11320</b>
<b>512</b>	<b>5872</b>
<b>1024</b>	<b>2992</b>
<b>1280</b>	<b>2395</b>
<b>1518</b>	<b>2030</b>

# 系统恢复的测试方法（1）

- ✓ 进入超载行为的状况时，重新分配资源。当负载恢复到正常水平时，资源应当重新分配并最终回到正常值。
- ✓ 目的是确定**DUT**从超载行为状况下恢复的速度的快慢。
- ✓ 如果**DUT**吞吐量测试结果接近媒介的**100%**最大速率，则不需要进行系统恢复测试。

# 系统恢复的测试方法（2）

- ✓ 以吞吐量的速率的**110%**或根据介质允许的理论速率发送一个测试流，流的发送时间持续至少**60**秒钟。
- ✓ 之后，为了使系统恢复到正常状况，在时间戳**A**，将发送速率减少至原来速率的**50%**。通过比较发送数据包与接收数据包的序号，找到丢失的最后一个数据包，并记录发送这个数据包的时刻（时间戳**B**）。
- ✓ 系统恢复时间由时间戳**B**减去时间戳**A**确定。
- ✓ 该测试应该多次进行，求平均值便可以得到测试结果。

# 系统恢复的测试方法（3）

帧长度（字节）	帧速率（fps）	恢复时间（毫秒）
<b>64</b>	<b>14300</b>	<b>2800</b>
<b>128</b>	<b>8445*</b>	<b>2750</b>
<b>256</b>	<b>4528*</b>	<b>2730</b>
<b>512</b>	<b>2349*</b>	无
<b>1024</b>	<b>1197*</b>	无
<b>1280</b>	<b>958*</b>	无
<b>1518</b>	<b>812*</b>	无

# 系统重启的测试方法（1）

- ✓ 主要集中在系统重启时间的测试上
- ✓ 引发系统重启的**3**种类型：
  - 1) **reset**引起重启：使用**DUT**设备软件的**reset**方式进行软件系统重启。
  - 2) 硬件重启：使用**DUT**设备硬件的**reset**按钮进行硬件系统重启。
  - 3) 电源中断重启：与上述相似，不同之处在于重启**DUT**时需要关闭电源**10**秒钟。

# 系统重启的测试方法（2）

- ✓ 以所得到的最小帧长吞吐量的速率持续地发送数据流。
- ✓ 重启**DUT**。
- ✓ 从数据包开始转发起一直监测输出端口，分别记录复位前测试端口接收到的最后数据包的时间（时间戳**A**）和复位后测试端口接收到的第一个数据包的时间（时间戳**B**）。
- ✓ 重启时间由时间戳**B**减去时间戳**A**所得。
- ✓ **注意**：为避免引入新的延迟，采用直连网络或静态路由



# Edit Command Sequence

Command Category Filter:

All Commands

Command Name P/F

- Link OAM: Res...
- Link OAM: Sta...
- Link OAM: Sta...
- Link OAM: Sta...
- Link OAM: Sta...
- Link OAM: Sta...
- Link OAM: Sta...
- Link OAM: Sto...
- Link OAM: Sto...
- Link OAM: Sto...
- Link OAM: Sto...
- Link OAM: Sto...
- Link OAM: Sto...
- Link OAM: Sto...
- PPP: Abort Em...
- PPP: Connect ...
- PPP: Disconne...
- RFC 2544: Fr...
- RFC 2544: Lat...
- RFC 2544: Th...
- Routing: Adve...
- Routing: Clear...



New Move Up Move Down Add Commands To

Command Name	P/F	Description
[-] RFC 2544: Throughput Test 1		Encapsulates an RFC 2544 Throughput test group
BenchmarkTestStartCommand 2		Starts a benchmark test.
BenchmarkTestLearningStartCommand 4		Starts benchmark test learning.
[-] Trial Loop (0)		Executes commands in the loop once for each iteration.
[-] BenchmarkIterateTrialCommand 2		Handle trial related test logic.
[-] Frame Size Loop (0)		Executes commands in the loop once for each iteration.
[-] Benchmark Iterate Frame Size 2		Modify streamblock frame size.
[-] Load Loop (0)		Executes commands in the loop once for each iteration.
Benchmarking: Benchmark Throughput 1		Adjusts port load values to search for throughput when used in a loop.
Set Traffic Duration 2		Sets the duration of traffic generation.
Clear All Results 2		Clears all results.
Start Analyzer 33		Turn on analyzer.
Wait 4 (2 seconds)		Blocks sequencer execution for specified amount of time.
Start Traffic 5		Begin transmitting traffic.
Wait For Traffic Stop 2		Wait for traffic to stop.
Wait 5 (15 seconds)		Blocks sequencer execution for specified amount of time.
Stop Analyzer 33		Turn off analyzer.
BenchmarkTestStopCommand 2		Stops a benchmark test.

Command Description  
Reboot specified equipment.

Drag-n-drop, double-click or use add/delete buttons to insert new commands from the command list (left side) into the sequencer (right side). Double-click commands in sequencer to edit command properties. Sequencer command order can be modified using drag-n-drop or up/down buttons. Holding the Shift key for drag-n-drop operations will insert the new command before the selected index. By default, commands are inserted in sequencer after the selected index. If there is no index selected in the sequencer, new commands are appended to the sequencer.

OK Cancel

# Edit Command Sequence

Command Category Filter:

All Commands

Command Name

P/F

Admin: Reboot Equipment

Application Layer: Wait on Load Pro...

Basic: Activate/Deactivate StreamBl...

Basic: Clear All Protocol Results

Basic: Clear All Results

Basic: Clear All Traffic Results

Basic: Clear Chart Results

Basic: Group

Basic: Iterate Frame Size

Basic: Iterate Load Size

Basic: Iterate Multicast Group Count

Basic: Iterate Throughput

Basic: Iterate Trial Command

Basic: Loop

Basic: Restart Auto Negotiation

Basic: Run External Process

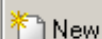
Basic: Save Capture Data

Basic: Save Iteration Results

Basic: Save Results

Command Description

Reboot specified equipment.



Command Name	P/F	Description
[-] RFC 2544: Frame Loss Test 1		Encapsulates an RFC 2544 Frame Loss test group
BenchmarkTestStartCommand 1		Starts a benchmark test.
BenchmarkTestLearningStartCom...		Starts benchmark test learning.
[-] Trial Loop (6)		Executes commands in the loop once for each iteration.
BenchmarkIterateTrialComma...		Handle trial related test logic.
[-] Frame Size Loop (1)		Executes commands in the loop once for each iteration.
Benchmark Iterate Frame ...		Modify streamblock frame size.
[-] Load Loop (1)		Executes commands in the loop once for each iteration.
Benchmark Iterate Loa...		Modify port/streamblock load.
Set Traffic Duration 1		Sets the duration of traffic generation.
Clear All Results 1		Clears all results.
Start Analyzer 1		Turn on analyzer.
Wait 1 (2 seconds)		Blocks sequencer execution for specified amount of time.
Start Traffic 4		Begin transmitting traffic.
Wait For Traffic Stop 1		Wait for traffic to stop.
Wait 2 (15 seconds)		Blocks sequencer execution for specified amount of time.
Stop Analyzer 1		Turn off analyzer.
BenchmarkTestStopCommand 1		Stops a benchmark test.



Drag-n-drop, double-click or use add/delete buttons to insert new commands from the command list (left side) into the sequencer (right side). Double-click commands in sequencer to edit command properties. Sequencer command order can be modified using drag-n-drop or up/down buttons. Holding the Shift key for drag-n-drop operations will insert the new command before the selected index. By default, commands are inserted in sequencer after the selected index. If there is no index selected in the sequencer, new commands are appended to the sequencer.

OK

Cancel

# Edit Command Sequence

Command Category Filter:

All Commands

Command Name

P/F

Admin: Reboot Equipment

Application Layer: Wait on Load Pro...

Basic: Activate/Deactivate StreamBl...

Basic: Clear All Protocol Results

Basic: Clear All Results

Basic: Clear All Traffic Results

Basic: Clear Chart Results

Basic: Group

Basic: Iterate Frame Size

Basic: Iterate Load Size

Basic: Iterate Multicast Group Count

Basic: Iterate Throughput

Basic: Iterate Trial Command

Basic: Loop

Basic: Restart Auto Negotiation

Basic: Run External Process

Basic: Save Capture Data

Basic: Save Iteration Results

Basic: Save Results

Command Description

Reboot specified equipment.



Command Name	P/F	Description
[-] RFC 2544: Latency Test 1		Encapsulates an RFC 2544 Latency test group
BenchmarkTestStartCommand 4		Starts a benchmark test.
BenchmarkTestLearningStartCommand 8		Starts benchmark test learning.
[-] Trial Loop (2)		Executes commands in the loop once for each iteration.
[-] BenchmarkIterateTrialCommand 4		Handle trial related test logic.
[-] Frame Size Loop (0)		Executes commands in the loop once for each iteration.
[-] Benchmark Iterate Frame Size 4		Modify streamblock frame size.
[-] Load Loop (0)		Executes commands in the loop once for each iteration.
Benchmark Iterate Load Size 2		Modify port/streamblock load.
Set Traffic Duration 4		Sets the duration of traffic generation.
Clear All Results 4		Clears all results.
Start Analyzer 35		Turn on analyzer.
Wait 8 (2 seconds)		Blocks sequencer execution for specified amount of time.
Start Traffic 7		Begin transmitting traffic.
Wait For Traffic Stop 4		Wait for traffic to stop.
Wait 9 (10 seconds)		Blocks sequencer execution for specified amount of time.
Stop Analyzer 35		Turn off analyzer.
BenchmarkTestStopCommand 4		Stops a benchmark test.



Drag-n-drop, double-click or use add/delete buttons to insert new commands from the command list (left side) into the sequencer (right side). Double-click commands in sequencer to edit command properties. Sequencer command order can be modified using drag-n-drop or up/down buttons. Holding the Shift key for drag-n-drop operations will insert the new command before the selected index. By default, commands are inserted in sequencer after the selected index. If there is no index selected in the sequencer, new commands are appended to the sequencer.

OK

Cancel