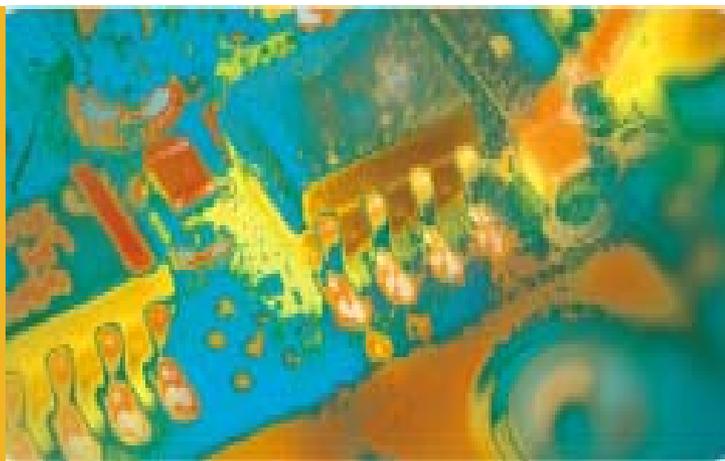




温州大学
WENZHOU UNIVERSITY

网络性能测试与分析



<http://network.wzu.edu.cn>

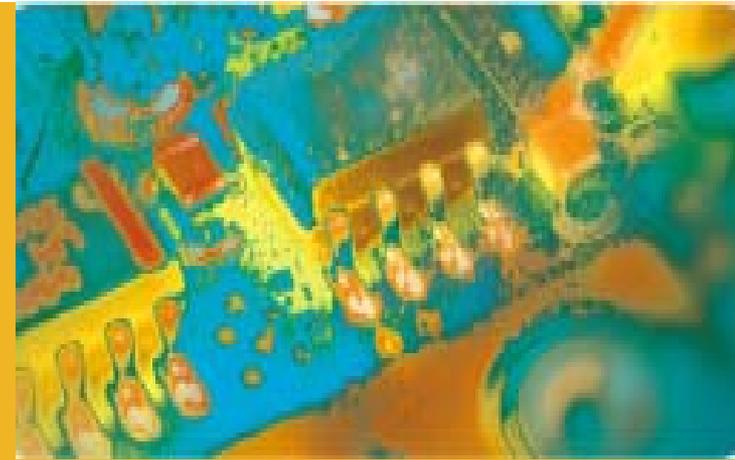
<http://www.spirent.com>

sxq@wzu.edu.cn



温州大学
WENZHOU UNIVERSITY

Chap2 二层以太网测试



<http://network.wzu.edu.cn>

[http:// www.spirent.com](http://www.spirent.com)

sxq@wzu.edu.cn

本章教学提要

●教学目标：

- ✓理解二层测试的必要性；
- ✓回顾以太网技术要点；
- ✓掌握以太网测试的主要技术指标；
- ✓了解以太网测试的**RFC**文档
- ✓掌握以太网测试方法学

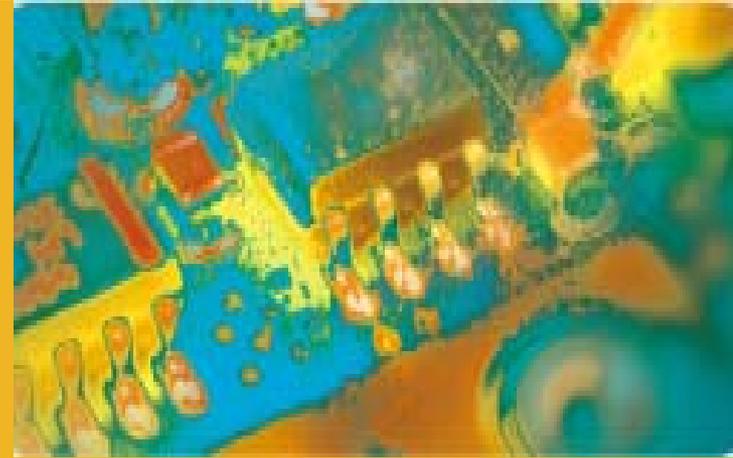
●教学难点/重点：以太网测试方法学

●教学时数：理论**6-8**学时



温州大学
WENZHOU UNIVERSITY

Chap2 二层以太网测试



Section 1 OSI二层测试的必要性

本节关注问题

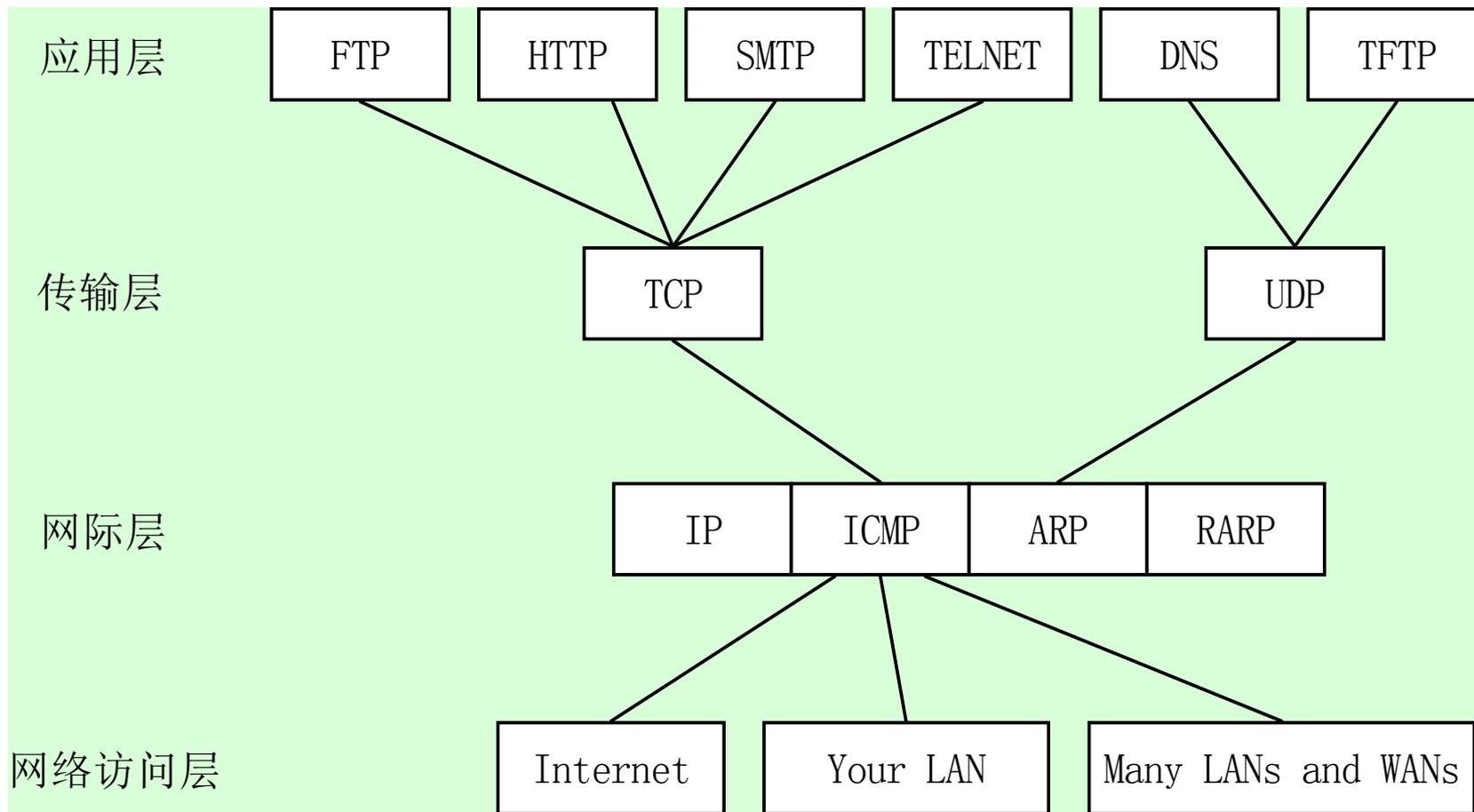
- 为什么要进行二层网络测试？
- 为什么需要关注二层的功能与性能？

OSI 二层功能概述



- 物理层涉及原始比特流的传输
- 数据链路层为相邻节点间提供可靠的帧传输服务。

TCP/IP模型中的网络访问层



- 该层包括了所有的局域网、城域网和广域网技术；
- 计算机网络数据传输的基础，构成了互连网通信的基础平台。

二层测试的必要性

- **根据网络分层模型，上层的功能实现是以下层所提供的服务为基础的；**
- **二层服务质量（如服务类型、数据传输质量等），最终影响网络高层的功能与性能。**
- **产生影响的因素包括：**
 - ✓ 技术选择
 - ✓ 设备选型
 - ✓ 拓扑设计

二层测试的运用

●网络设备的研发与生产过程

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

●网络规划与设计过程

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。

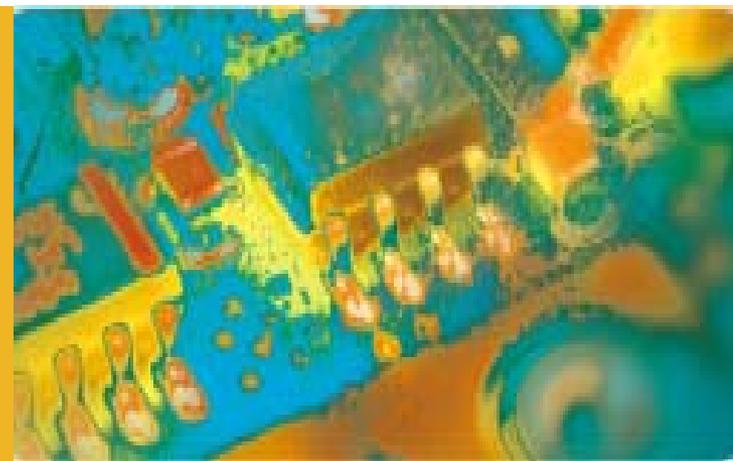
●网络运行过程

- ✓ 故障诊断
- ✓ 性能评价与分析
- ✓ 网络升级或优化



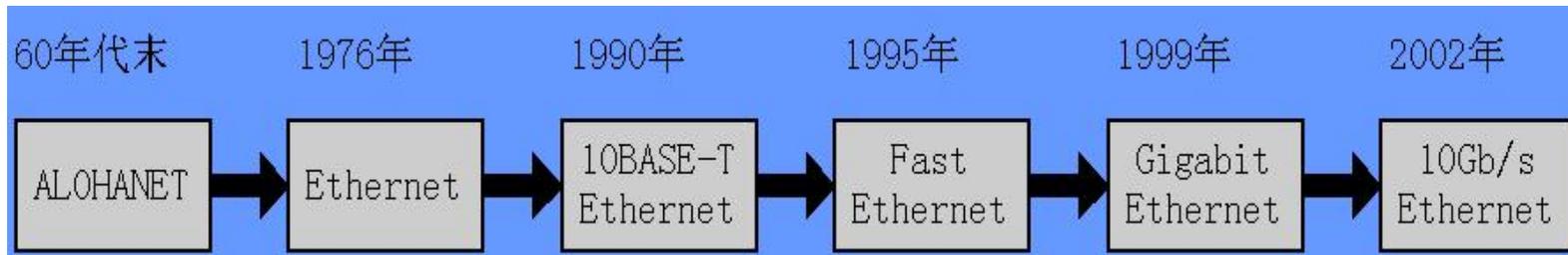
温州大学
WENZHOU UNIVERSITY

Chap2 二层以太网测试



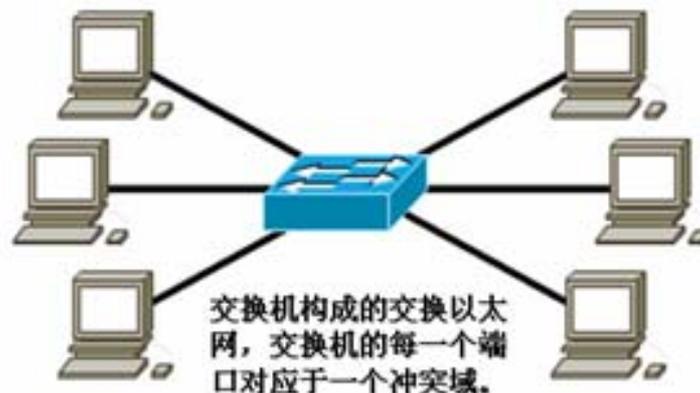
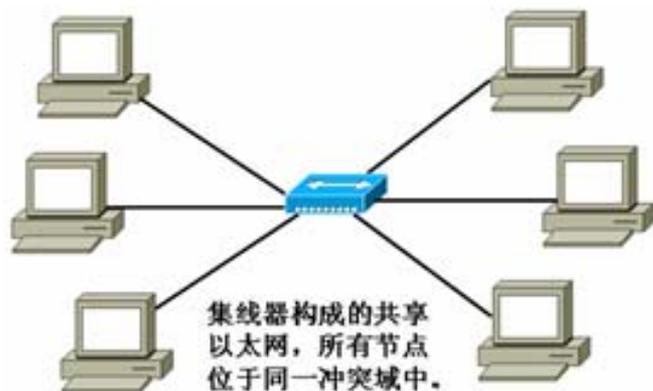
Section 2 以太网技术概述

以太网技术家族



名称	带宽	拓扑	组网方式	标准	适用范围
标准以太网	10M	总线/星型	共享/交换	IEEE802.3	局域网
快速以太网	10M	星型	共享/交换	IEEE802.3 u	局域网
千兆以太网	1G	星型	交换	IEEE802.3z/ IEEE802.3a b	局域网
万兆以太网	10G	星型 点对点	交换	IEEE802.3 ae	局域网城 域网

共享以太网与交换以太网



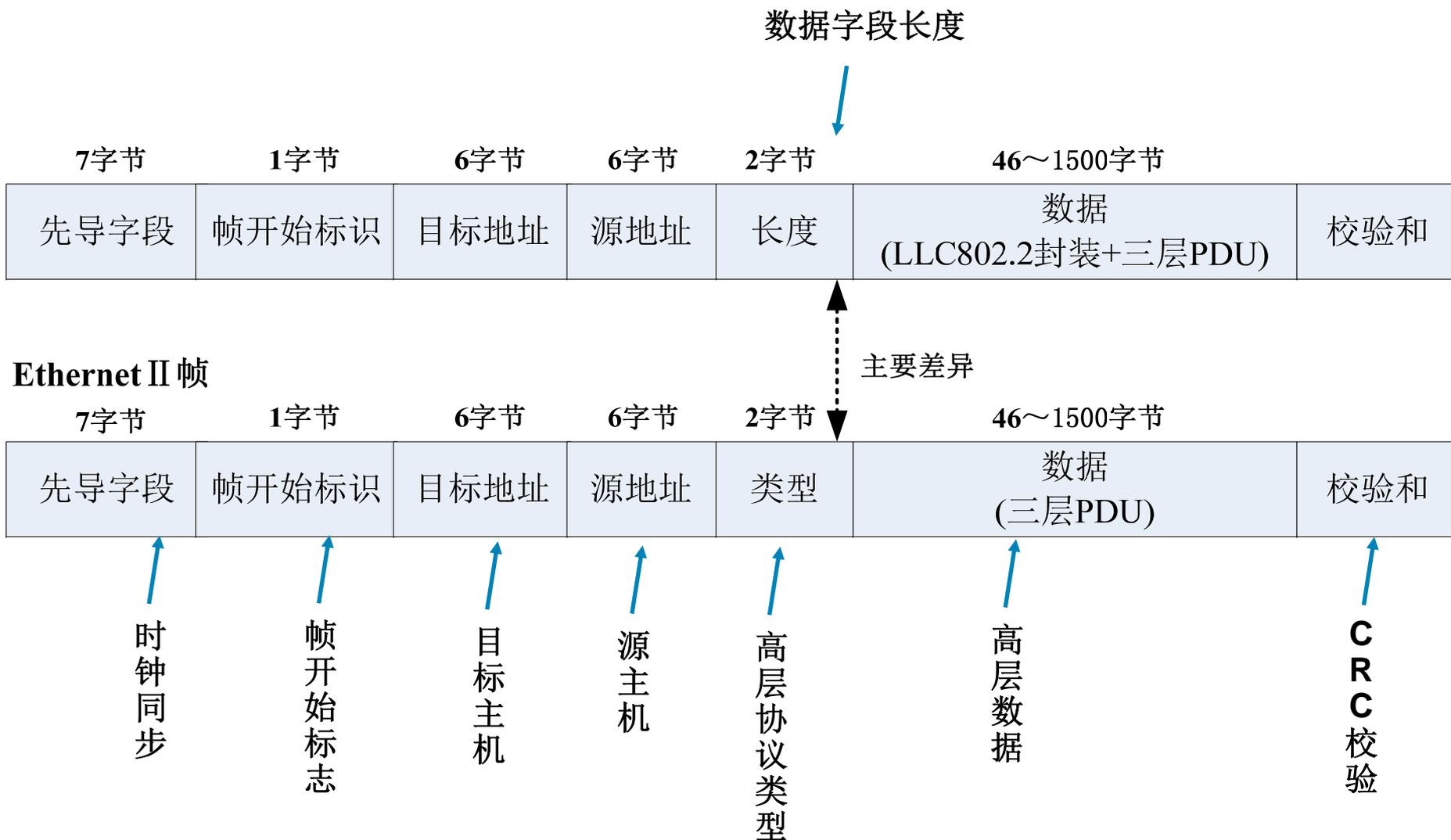
■ 共享以太网

- ✓ 总线拓扑**&**基于**HUB**的星型拓扑；
- ✓ 所有设备与节点位于同一个冲突域；
- ✓ 冲突影响了网络的运行性能。

■ 交换以太网

- ✓ 基于交换机的(扩展)星型拓扑；
- ✓ 交换机每一个端口对应于一个独立的冲突域；
- ✓ 逻辑划分提升了网络运行性能。

以太网帧结构



以太网帧的接收过程

❶ 判断帧的长度是否符合最短帧长度的要求以确定是否为有效帧；

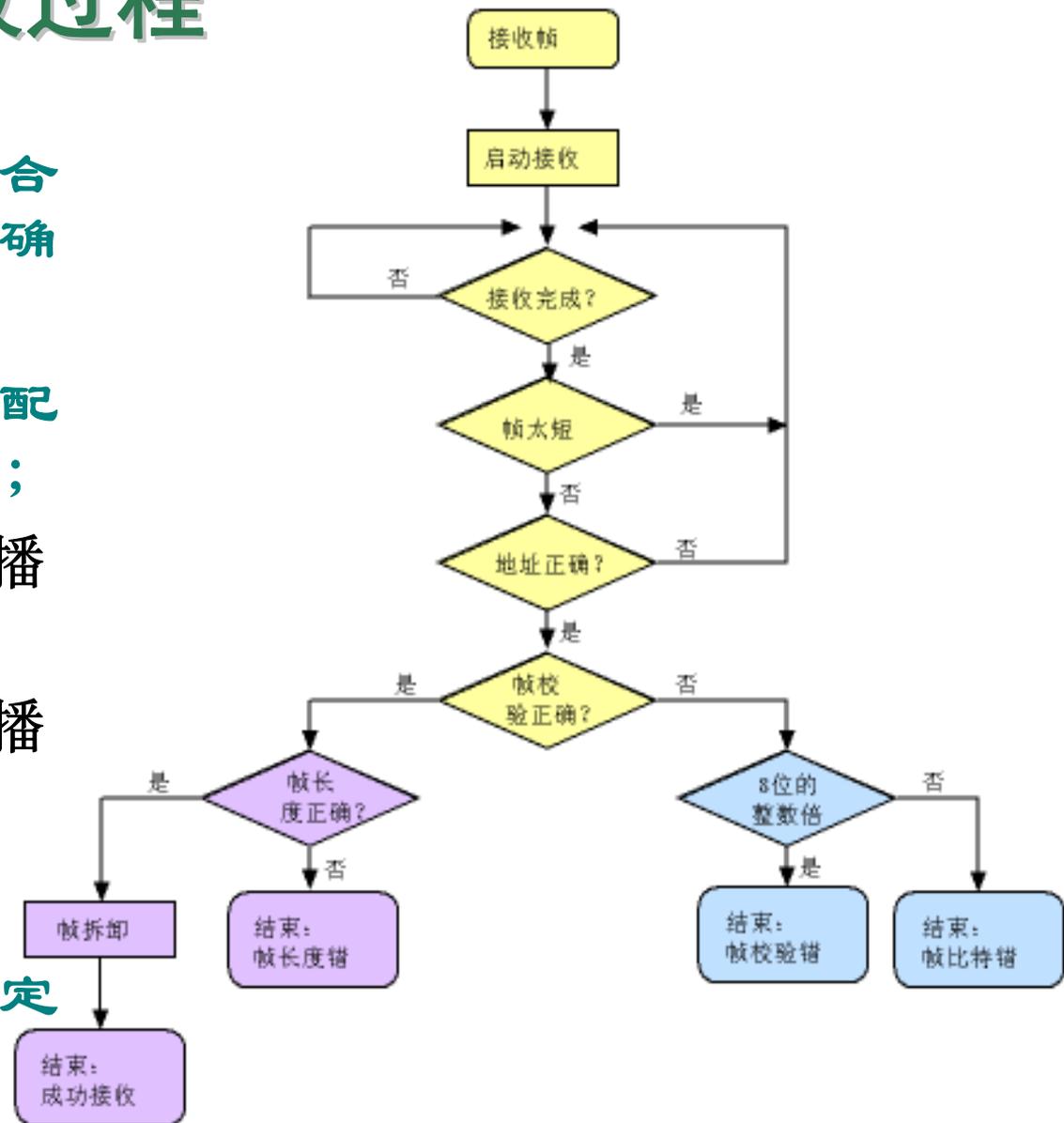
❷ 判断目标地址是否匹配以决定是否接收该帧；

1) 与本节点匹配的单播地址

2) 与本节点所在的组播组相同的组播地址

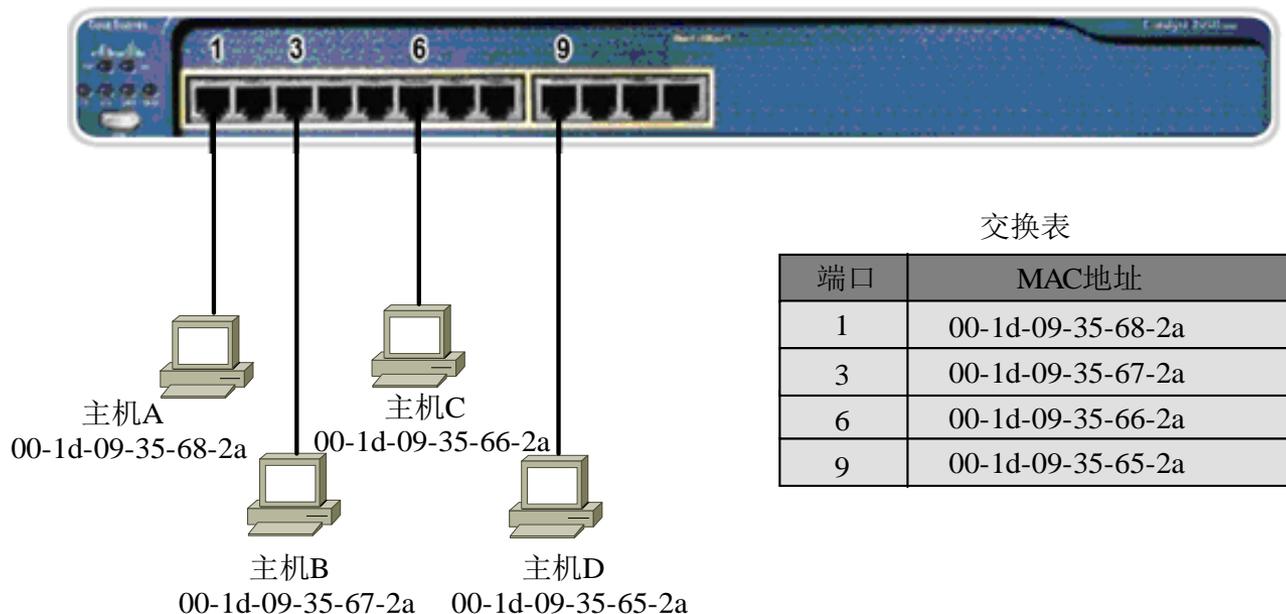
3) 广播地址。

❸ 判断帧的正确性以决定是否丢弃该帧。



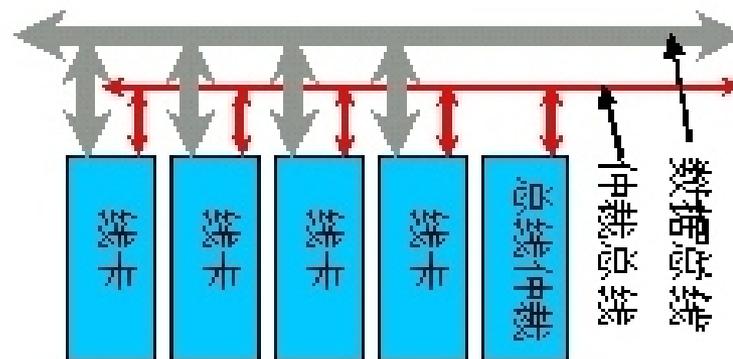
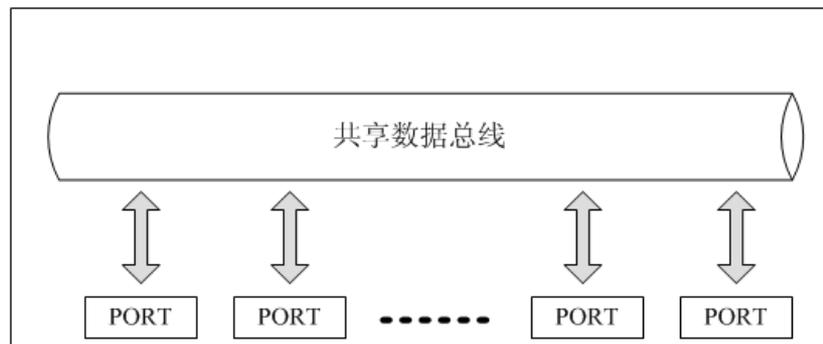
以太网交换机的工作原理

7



- 根据帧中的目的地址，通过查找地址表决定转发行为：
 - ✓ 若地址表显示目的节点与源节点位于交换机的同一端口 → 忽略帧；
 - ✓ 若地址表显示目的节点在交换机所连的某一端口，且与源节点不在同一端口 → 转发到目的端口；
 - ✓ 若目的**MAC**地址为广播地址 → 向除源端口外的所有端口转发帧；
 - ✓ 若地址表中找不到目的地址 → 向除源端口外的所有端口转发帧。

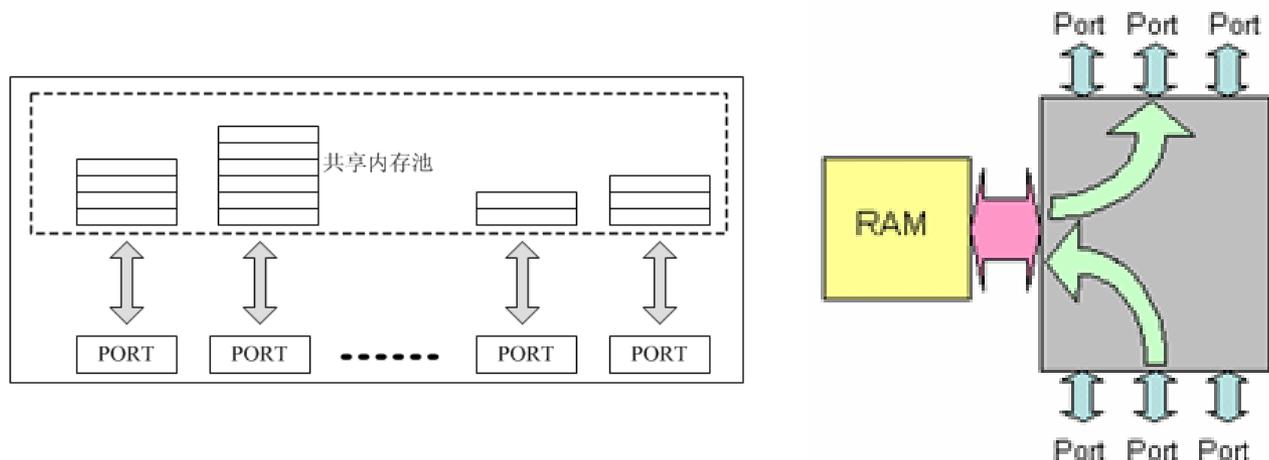
交换机的体系结构--共享总线型



● 两种数据交换实现方式

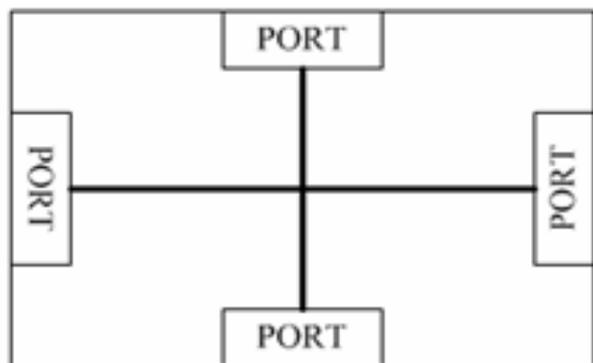
- ✓ 集中式交换方式中，由交换机中心处理器保存端口与目的**MAC**地址的映射表。
- ✓ 分布式交换方式中，每个端口在本地维持一个端口与目的**MAC**地址的映射表。
- 如果同时存在多个数据交换，采用时分技术，以分割时间片的方式进行数据交换。

交换机的体系结构--共享存储器



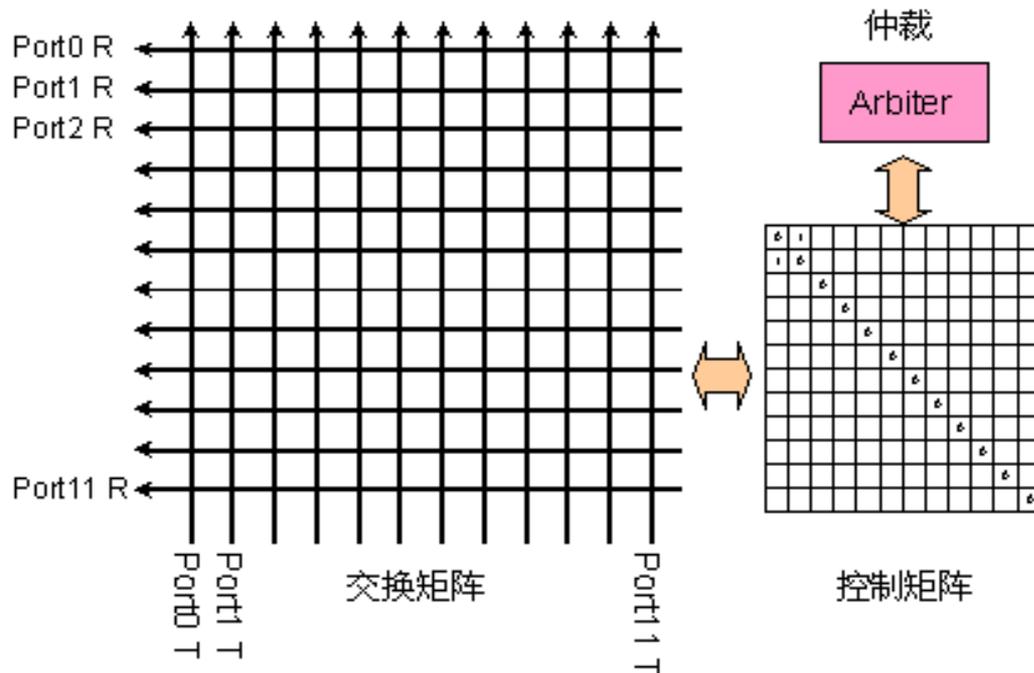
- 使用一个全局共享存储池进行数据传输交换
- 共享RAM一般由专用集成电路（ASIC）芯片管理。ASIC芯片通过查找地址表，找到与目的地址对应的目的端口，然后将数据发送至所对应的目的端口
- 简单，但受内存容量与速度限制，无法支持大容量交换（一般限制在20G-80G），且交换延时比较大。
- 一般适用于小容量交换机，或作为大容量交换机的线卡内部交换。

交换机的体系结构—纵横式矩阵



交叉点(Crosspoint):

- 交叉点结构中，数据传输通过交叉点进行
- 当端口数据量大，网络负载重的情况下会因交叉点的瓶颈而造成阻塞。

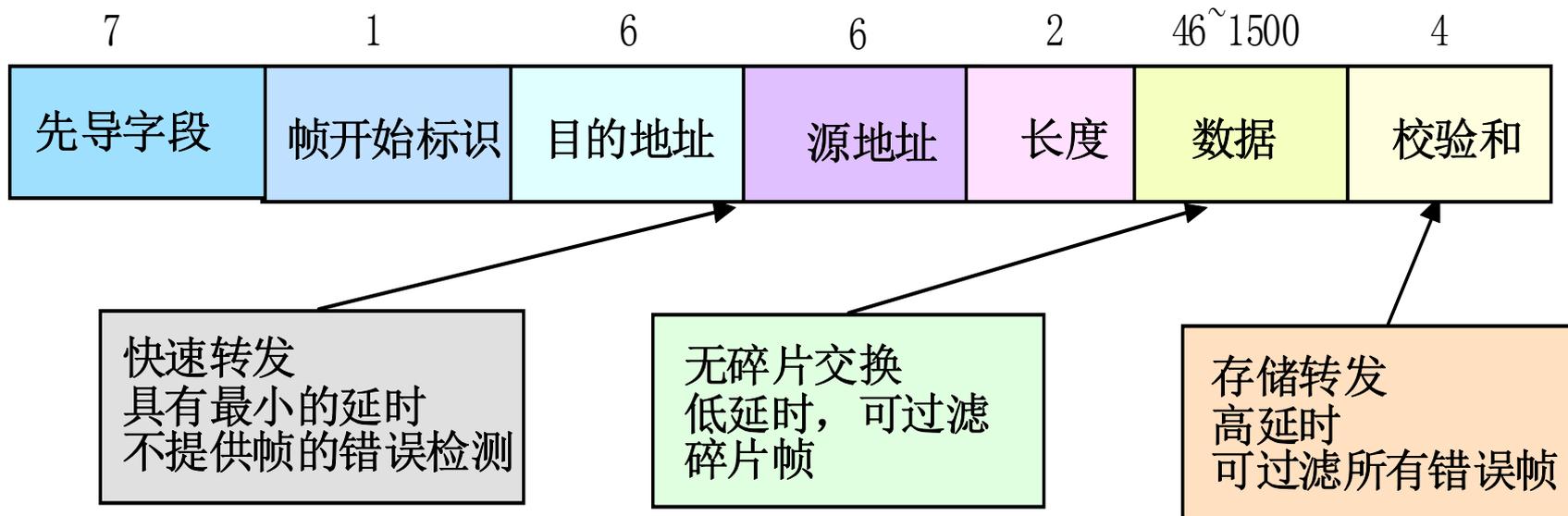


纵横式交换矩阵(Crossbar)结构

- 交叉开关矩阵或纵横式交换矩阵 (**Crossbar**) 是大容量高端交换机中普通采用的结构
- 内部的交换机矩阵和仲裁矩阵实现了无阻塞交换。

交换机的转发方式

字节

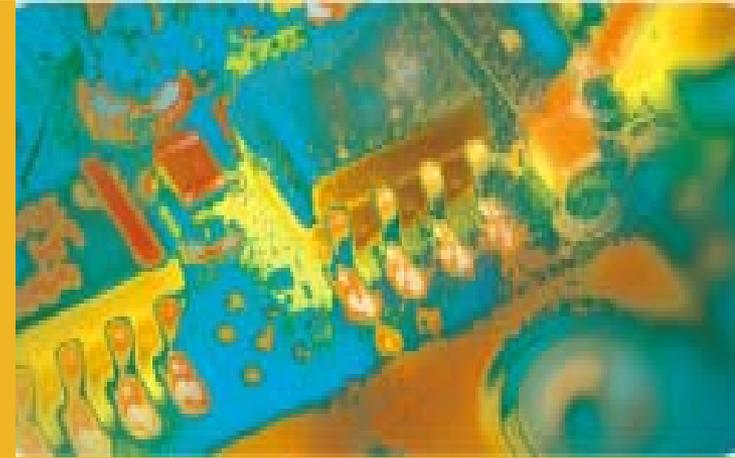


- 直接交换（**cut-through**）和存储转发（**store-and-forward**）
- 直接交换模式又分成快速转发（**fast-forward**）与无碎片（**Fragment-free**）交换。
- 实际交换机产品中，通常引入智能方式来进行选择→设置帧错误率的阈值，先采用直接交换，一旦帧错误率超过阈值，改用存储转发。



温州大学
WENZHOU UNIVERSITY

Chap2 二层以太网测试



Section 3 决定交换以太网性能的主要技术指标

如何确定指标？

● 二层功能包括：

- ✓ 帧的封装与拆封
- ✓ 基于**MAC**地址的帧接收与转发
- ✓ 流量控制
- ✓ 简单的差错控制

● 将这些功能可以分别归类两个不同层面上：

- ✓ 数据传输层面 → 数据传输指标
- ✓ 传输控制层面 → 传输控制指标

数据传输层面相关的指标

● 负载 (load)

信道或设备在单位时间内所承受的通信流量被称为负载。在网络二层，通常以单位时间内设备或网络所承载的帧数来衡量。

● 转发速率 (forwarding rate)

用以描述交换设备帧转发能力的指标。被定义成在某个特定负载下，交换机设备在单位时间内向目标端口成功转发的帧数。

不同设备的转发速率与交换机的体系结构、端口带宽、转发模式、设备的负载状况等因素有关。

对于给定的设备，在没有丢帧的理想状态下，转发率应该随着负载的增加而增加。当负载增加并接近线路传输容量时，会因丢帧而导致转发速率下降。

数据传输层面相关的指标(续1)

● 丢帧率(frame loss ratio)

丢失的数据帧占应转发帧的比例。应转发帧指那些应该被转发的合法帧，不包括那些过长、过短和错误的无效帧。

丢帧主要出现在负载过大时，因交换机存储容量、地址表查找、端口拥塞等方面的瓶颈而产生。

● 吞吐量(Throughput)

另一个描述交换设备数据包转发能力的指标，用以衡量交换机在不丢帧条件下每秒转发帧的极限能力。

定义成：在没有出现丢帧的条件下，能够传输给交换机让其转发到指定输出端口的每秒最大帧数。

数据传输层面相关的指标(续2)

● 突发 (burst)

在某个时间段内，一组以合法最小帧间隔传输的以太网帧被称为突发。突发通常由节点的突发数据流量引发。

一次突发传输中所包含的帧数被称为突发量 (**Burst size**)。突发量为**1**时，相当于无突发传输的恒定负载。

两次突发之间的时间间隔被称为突发间隔 (**Inter-burst gap**，简称**IBG**)。

交换机所能承载的突发量越大，可以处理的突发间隔越小，突发量分布的离散性越大，说明交换机处理突发数据流量的性能越好。

传输控制功能

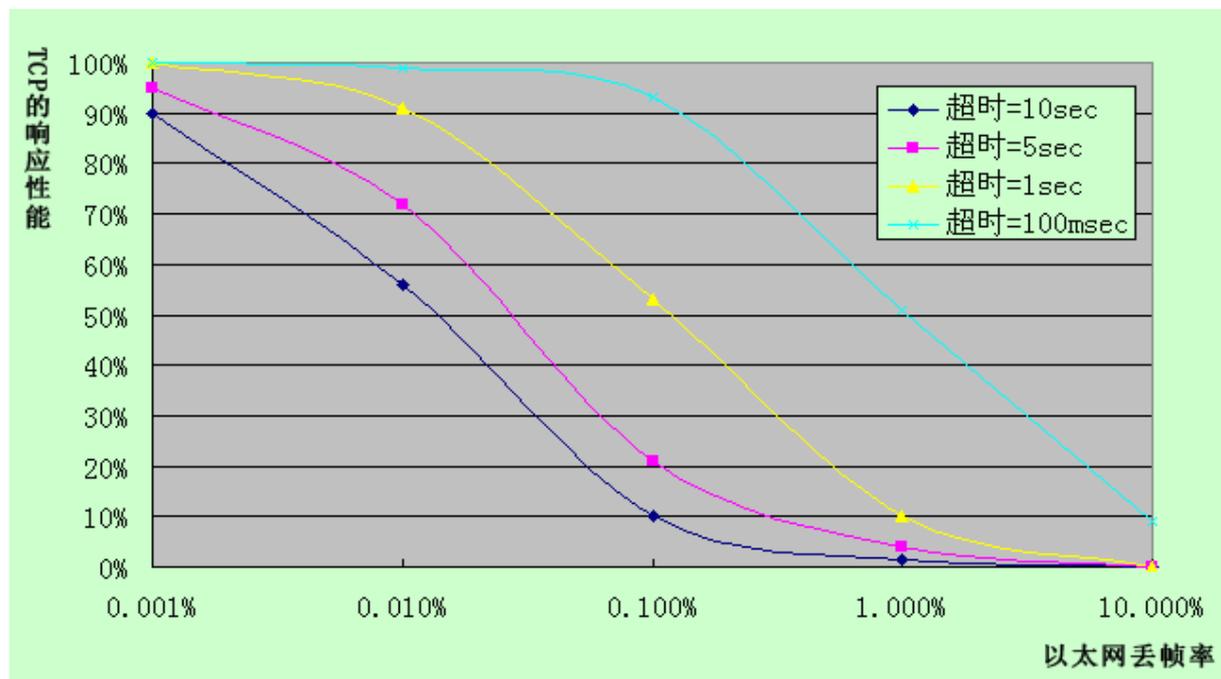
- 传输控制功能是为了正确实现基于二层地址的帧转发，避免或减少因拥塞而产生的丢帧，而提供的控制功能，如：
 - ✓ 地址处理
 - ✓ 拥塞控制
 - ✓ 广播处理
 - ✓ 流量隔离
- 控制功能实现或正确与否，直接影响数据层面上的转发性能。

传输控制层面指标-拥塞控制

● 拥塞控制 (frame loss ratio)

控制源端发送数据的数量及速度使其不超过接收端所能承受的能力，以避免产生帧的丢失，也称流量控制。

拥塞→帧丢失→网络运行性能下降(即使高层使用可靠传输协议如**TCP**)。



产生拥塞的主要原因控制

- 过载 (overload)

- 线端阻塞 (Head of line blocking)

因某个外出端口上的拥塞而阻碍了其他通往非拥塞端口流量的现象。

对于在输入队列中采用共享存储的交换机，当队列头有转发到阻塞端口的帧时，造成后继转发到非阻塞端口帧也必须等待。

拥塞控制-背压

●何为背压 (Backpressure)

当外出或输出端口出现拥塞现象时，被交换机用来通知发送端降低帧发送速度，以阻止外部数据源继续向拥塞端口传输帧的那些方法。

●常用方法

- ✓向流发送源回送拥塞(jam)信号
- ✓向流发送源回送先导位串
- ✓采用**IEEE 802.3x**流量控制协议
- 背压机制使发送到拥塞端口的流量得到减缓，可能会导致到非拥塞输出端口的发送流量也被减缓。

拥塞控制-前压

●前压 (Forward pressure)

当上游设备以小于最小帧间隙的间隔或以超过线速的速率向下游交换机发送流量时，往往会导致下游交换机出现接收缓冲(buffer)溢出甚至阻塞。

某些交换机可通过减缓输出队伍的饱和程度、禁止上游设备发送帧等强制性机制来消除或减少上述丢帧现象。

具有上述功能的交换机被认为具有前压机制。

不是以太网标准或协议所推荐的，因而不是所有的交换机都提供。

数据传输层面相关的指标(续1)

●地址处理 (address handling)

与交换机在二层地址学习与处理能力相关，包括：

✓地址缓存容量(Address Caching Capacity)

设备或端口模块所能拥有的最大MAC数，也就是地址交换表的最大容量。

✓地址学习速率(Address Learning Rate),

在没有广播和丢帧的情况下，交换机学习新地址的最大速度。

数据传输层面相关的指标(续2)

❶ 错误帧过滤 (Errored frame filtering)

错误帧(**error frame**)是指所有过长、过短、错位或含有错误校验序列的帧。

错误帧过滤可减少或避免因传输错误帧而产生的带宽浪费，以及后续转发设备因处理这些错误帧而付出不必要的资源消耗。

❷ 流量隔离 (traffic filtering)

衡量交换机对于**VLAN**流量隔离的功能。

数据传输层面相关的指标(续3)

● 广播 (broadcast)

与交换机广播性能相关，包括：

✓ 广播转发速率(**Broadcast forwarding rate**)

在某个指定的广播负载下，一秒内向它所在广播域的所有端口发送的广播帧的数量。

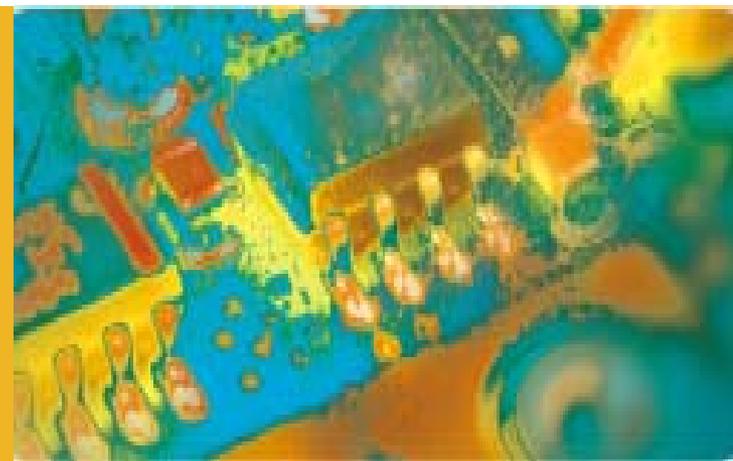
✓ 广播时延(**Broadcast latency**)

将广播帧转发到位于同一个广播域中的每个端口所需的时间。



温州大学
WENZHOU UNIVERSITY

Chap2 二层以太网测试



Section 4 二层以太网测试相关的RFC文档

二层测试相关的RFC文档

- 相关的RFC 文档包括RFC 1242、RFC 2544、RFC2285和RFC2889(? 图，提供发布时间)
- RFC 1242和RFC 2544具有一般性的指导意义，涵盖了OSI全部七层的测试。
- RFC 2285提供了基本术语
- RFC 2889针对LAN交换设备基准测试提供基本方法学

RFC2285中的基本术语

● DUT和SUT

DUT指被测试设备（**Device under test**），**SUT**指被测试系统（**System under test**）

● 单向流量和双向流量

单向流量（**Unidirectional traffic**）是指测试流在被测设备中以单向方式传输。当测试者为**DUT**加载单向流量时，是由**DUT**上的不同端口处理帧的接收与发送，输入端口与输出端口的角色是不重叠的。

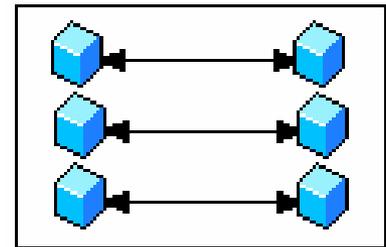
双向流量（**Bidirectional traffic**）是指每个端口在进行接收流量的同时也在进行发送流量。当测试者为**DUT**加载双向流量时，所有从测试仪表接收测试流量的端口同时也在向测试仪表回送测试流量，每个端口同时承担输入端口与输出端口的角色。

RFC2285中的基本术语(续1)

●非网状流量、部分网状流量和全网状流量

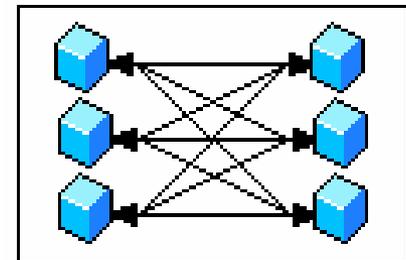
- ✓非网状流量(**Non-meshed traffic**)，也称端口对 (**Port Pair**)

如用于对端口转发性能进行测试。



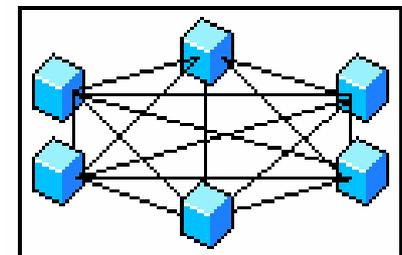
- ✓部分网状流量(**Partially meshed traffic**)，也称也被称作骨干 (**Back Bone**)

如用于非对称交换机之上行/下行端口的转发性能测试。



- ✓完全网状流量(**Fully meshed traffic**)，也称也被称作骨干 (**Back Bone**)

如用于对交换机进行整机性能测试。



RFC2285中的基本术语(续2)

期望负载、提交负载、最大提交负载、过载

✓期望负载(Intended load, 简称Iload)

指外部信息源企图传输给**DUT/SUT**让其转发到指定输出端口的每秒帧数。

✓提交负载 (Offered load, 简称Oload)

指外部信息源能够被观察或测量到的传输给**DUT/SUT**让其转发到指定输出端口的每秒帧数。

✓最大提交负载 (Maximum offered load, MOL)

指外部信息源每秒能够传送给**DUT/SUT**并让其向指定输出端口转发的最大帧数。

✓过载(Overloading)

超过媒介允许的最大传输速率向**DUT/SUT**施加负载。

RFC2285中的基本术语 (续2)

● 转发速率相关的术语

✓ 转发速率(Forwarding rate, 简称FR)

指定提交负载下，一台设备能够被观测到的每秒钟内成功向正确目的端口传送的帧数。

✓ 最大提交负载下的转发速率(Forwarding rate at maximum offered load, 简称FRMOL)

一台设备在最大提交负载的情况下能够被观测到的每秒钟内成功向正确目的端口转发的帧数。

✓ 最大转发速率(Maximum forwarding rate, 简称MFR)

一系列经过重复或迭代测试所获得的转发速率测量值中的最大值。注意，该值往往发生在最大提交负载之前。

RFC2889简介

- 为LAN交换设备的基准测试提供方法学，由RFC 2544的方法学扩展而来

- 测试对象：

交换机转发性能（**Forwarding performance**）、拥塞控制（**Congestion control**）、延迟（**Latency**）、地址处理（**Address handling**）和错误过滤（**Error filtering**）

- 文档结构：

备忘录、介绍、要求以及安全机制、参考文献等辅助性说明外，核心内容分为测试设置、帧格式与长度和基准测试三大部分。

- 基准测试：

涉及测试目标、参数设置、测试过程、测量方法和测试报告格式等方面

RFC2889所提供的基准测试

- 全网状互联条件下的吞吐量、丢帧率和转发速率（**Fully meshed throughput, frame loss and forwarding rates**）
- 部分网状互连条件下的一对多/多对一（**Partially meshed one-to-many/many-to-one**）
- 部分互连的多个设备（**Partially meshed multiple devices**）
- 部分网状互连条件下的单向通信流量（**Partially meshed unidirectional traffic**）
- 拥塞控制（**Congestion Control**）

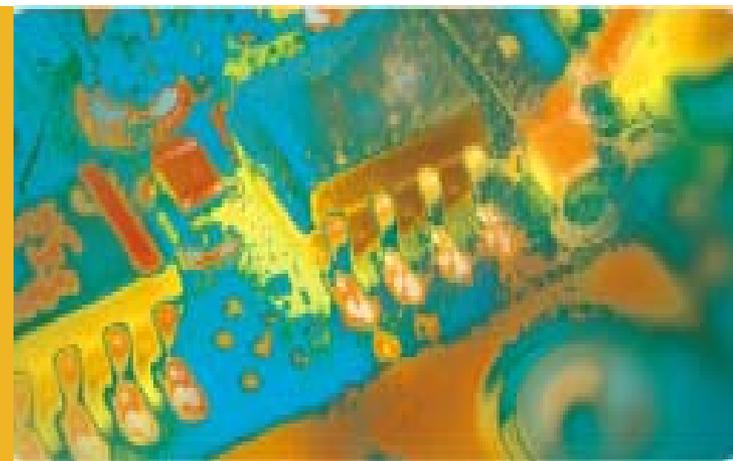
RFC2889所提供的基准测试(续)

- 转发压力和最大转发速率 (**Forward Pressure and Maximum Forwarding Rate**)
- 地址缓冲容量 (**Address Caching Capacity**)
- 地址学习速率 (**Address Learning Rate**)
- 错误帧过滤 (**Erred frames filtering**)
- 广播帧转发和延迟 (**Broadcast frame Forwarding and Latency**)



温州大学
WENZHOU UNIVERSITY

Chap2 二层以太网测试



Section 5 二层以太网测试的基本方法

测试方法及基本内容

- 测试方法是指针对一项具体的测试目标，围绕如何实施测试所提出的方法与方案。
- 主要内容包括：
 - ✓ 测试目标
 - ✓ 测试环境（包括测试拓扑、测试流、测试参数与变量）
 - ✓ 测试相关的算法设计
 - ✓ 测试结果的统计与报告
 - ✓ 测试前的准备工作

测试设置中的二层地址学习

●目的：

在测试开始之前让被测试交换机学习测试中将要用到的**MAC**地址。

●要求：

先学习→再验证→后测试

●注意1：

发送地址学习帧的速率不可太大(建议：**≤50**帧每秒)，以免因**DUT**地址学习速率上的限制而导致地址学习失败。

●注意2：

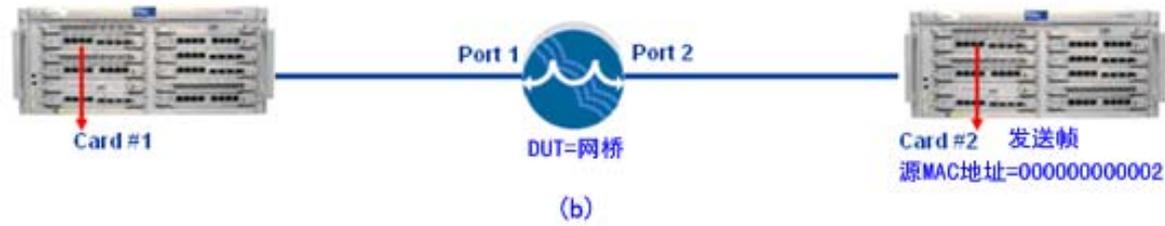
调整被测试交换机的地址老化时间，使得其足够长(建议：大于测试学习时间、测试持续时间、配置测试设备时间的和)，确保在测试结束前所学到的地址都不被老化。

二层地址学习举例

- 测试拓扑：测试仪表上的**Card#1**作为发送方要向作为接收方**Card#2**发送数据流，两者构成了一对一的非网状流



- **Card#2**向二层DUT设备发送地址学习帧



- DUT设备完成关于**Card#2**的MAC地址的学习



- 学习完成之后，**Card#1**才向**Card#2**开始发送测试数据流



测试帧的长度选择

- 理论上，选择任何一种长度在**64**字节到**1518**字节之间的测试帧都是被允许的。
- 帧长选择影响交换机的转发率、丢帧率和吞吐量等性能指标
- 为了全面地反映被测试设备的性能，有必要在不同的帧长度下运行有关的以太网测试。
- 三种常见的测试帧长方案
 - ✓完备性的测试方案：对所有长度的帧，进行测试。
 - ✓快速的测试方案：选择某些有代表性的帧长进行测试。
 - ✓基于统计的测试方案：以对设备所在网络环境的帧长监测与帧长分布统计信息为基础，确定相应的测试帧长。

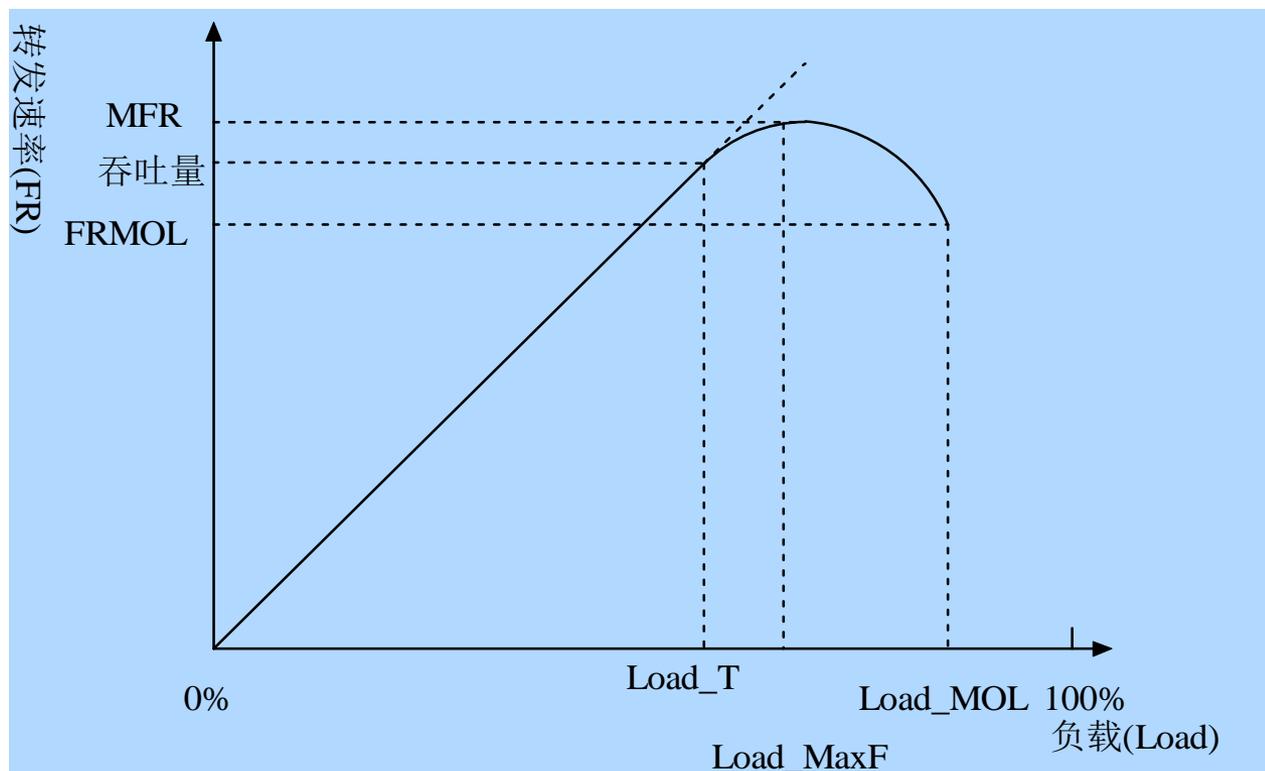
测试帧的格式

- 测试仪表仅对测试帧进行统计，为此需要对普通帧与测试帧加以必要的区分。
- 在基本格式上，测试帧与普通以太网帧相同，但需要加入一个独特的签名字段(**signature field**)。
- 签名字段(**signature field**)要有足够的可区分性，以帮助测试仪表从所接收的流量中提取出测试帧流量，并过滤掉那些不属于提交负载的普通帧流量。

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)	Parallel bit
Crc16 Byte1	Crc16 Byte0	TCP UDP Checksum Cheater (Byte1)	TCP UDP Checksum Cheater (Byte0)	

关于负载、吞吐量、丢帧率和转发率的深入理解

- 转发率、丢帧率和吞吐量是描述交换机转发性能的主要技术指标。
- **RFC2889**中，半数的基准测试直接针对上述指标
- 负载、吞吐量、丢帧率和转发速率的关系可以**FR-Load**曲线描述。



吞吐量测量的步进查找法

- 定义变量当前负载**Load**与丢帧率，定义初始负载**Load₀**，步进长度 **Δ Load**
- 将初始负载**Load₀**赋值给当前负载，并判断当前负载下的丢帧率是否为零
- ✓ **If 为零**， **Load** ← **Load** + Δ **Load**，再度观测丢帧率
若丢帧率不为零，停止对**Load**的步进，并将该负载下的转发率作为所要查找的吞吐量；否则，继续步进。
- ✓ **If 不为零**， **Load** ← **Load** - Δ **Load**，再度观测丢帧率
若丢帧率为零，停止对**Load**的继续步进，并将该负载下的转发率作为所要查找的吞吐量；否则，继续步进。
- 测试准确度取决于步长的大小，步长越小，所报告的吞吐量越接近实际结果，但步进次数越多，查找速率越慢。

吞吐量测量的二分迭代查找法

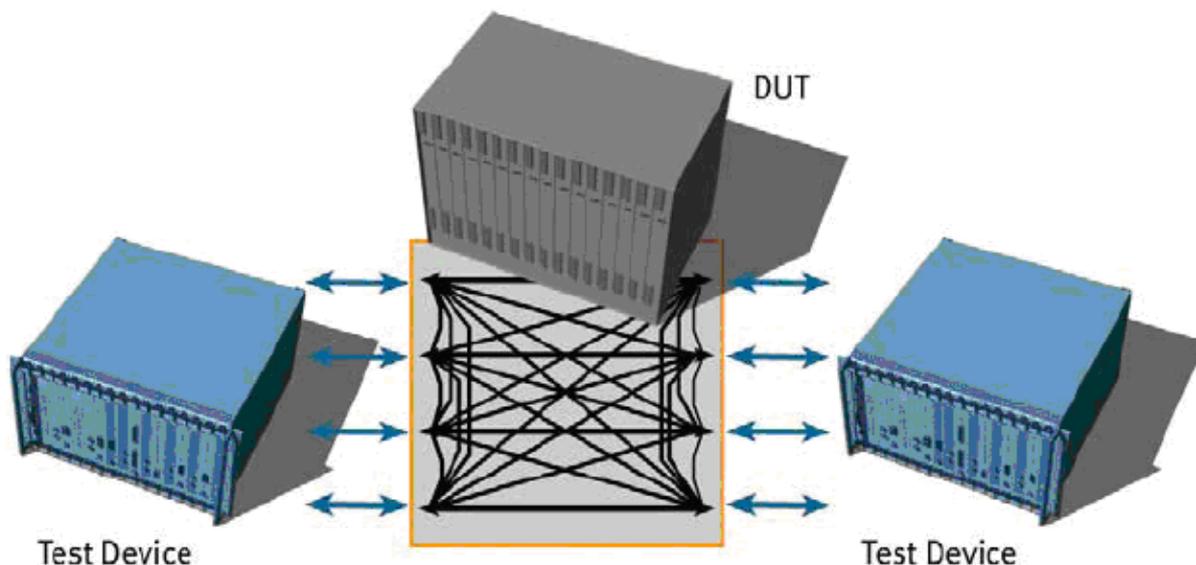
- 定义变量当前负载 **Load** 与丢帧率，定义常量最小负载 **Load_Min**、最大负载 **Load_Max**、初始负载 **Load₀** 和迭代分辨率 (**Resolution**)。
- 将初始负载 **Load₀** 赋值给当前负载，并判断当前负载下的丢帧率是否为零
- ✓ **If 为零**， $\text{Load} \leftarrow \text{Load} + (\text{Load_Max} - \text{Load}) / 2$ ，再度观测丢帧率，反复迭代，一旦丢帧率不为零或间隔小于迭代分辨率时停止迭代，并将当前负载下的转发率作为所要查找的吞吐量；
- ✓ **If 不为零**， $\text{Load} \leftarrow \text{Load} - (\text{Load} - \text{Load_Min}) / 2$ ，再度观测丢帧率，反复迭代，一旦丢帧率为零或间隔小于迭代分辨率时停止迭代，并将当前负载下的转发率作为所要查找的吞吐量。
- 测试准确度由迭代分辨率决定，迭代分辨率越小，吞吐量越接近实际结果，迭代次数越多

吞吐量测量的混合查找法

- 步进算法具有较二分迭代更快的查找速率，而二分迭代法具有较步进算法更准确的查找结果。
- 混合算法的思想 → 兼顾查找速率与查找准确度。
- ✓ 首先，采用步进查找算法来快速找到或逼近**DUT**的吞吐量所对应的负载大概范围或区间；
- ✓ 然后，在上述区间内进行二分迭代查找。
- 举例：

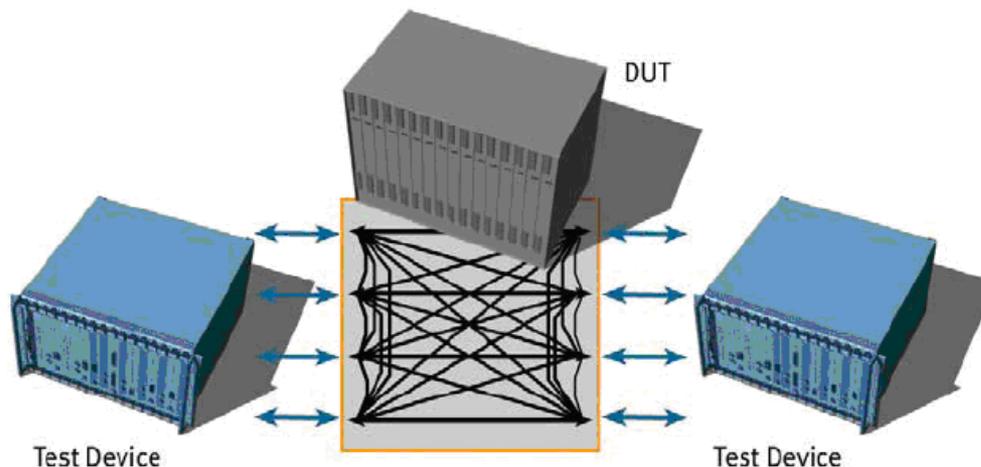
初始负载**Load₀** 时，未有丢帧现象。在每**n**次步进时首次观测到丢帧，那么就中止后续的步进过程，转而在区间**[Load 0+(n-1)ΔLoad, Load 0+nΔLoad]**内进行二分迭代查找。
- 若事先对**DUT**吞吐量所对应的区间有所了解，可直接在该区间内运行二分迭代查找算法，以加快整个测试进程。

全网状吞吐量、丢帧率和转发速率测试



- 全网状吞吐量、丢帧率和转发率测试是为了确定 **DUT** 在全网状流量下的吞吐量、丢帧率和转发率，可反映出交换机的整机交换转发性能。

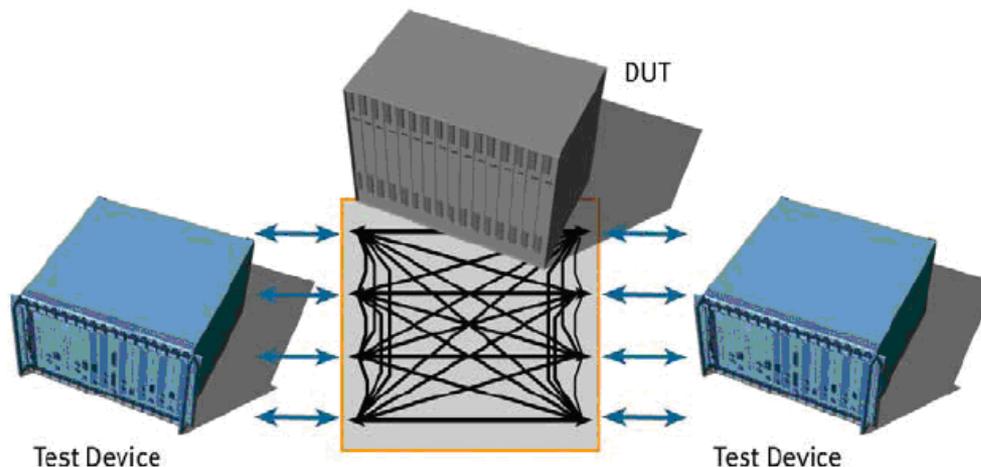
全网状吞吐量、丢帧率和转发速率测试方法



● 测试参数设置

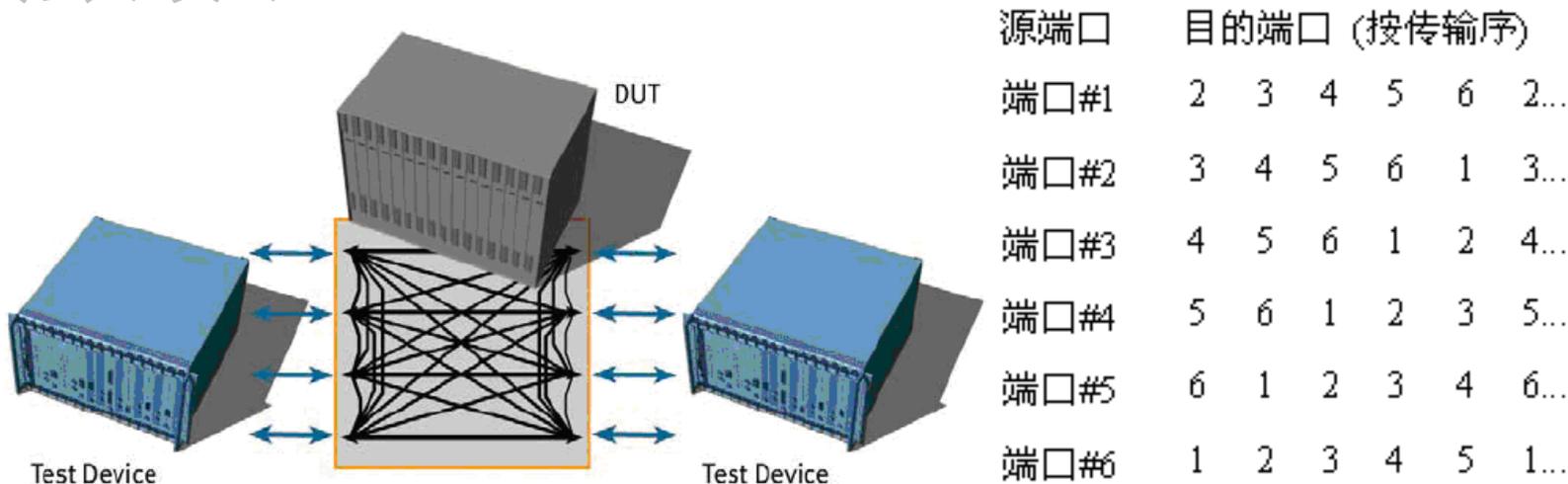
- ✓ 测试帧长设定为**64,128,256,512,1024,1280**和**1518**字节，至少要选择其中的五个不同帧长实施测试；
- ✓ 突发帧（**burst**）中的帧间隙（**IFG**）被指定成**96**比特长度的最小合法帧间隔，以能够反映极限**IFG**下的交换机性能。
- ✓ 突发帧的长度被设定成在**1**到**930**帧之间变化，以更接近实际网络环境中的突发帧分布状况；
- ✓ 在端口工作模式上，可选择半双工或者全双工。。

全网状吞吐量、丢帧率和转发速率测试方法(续1)



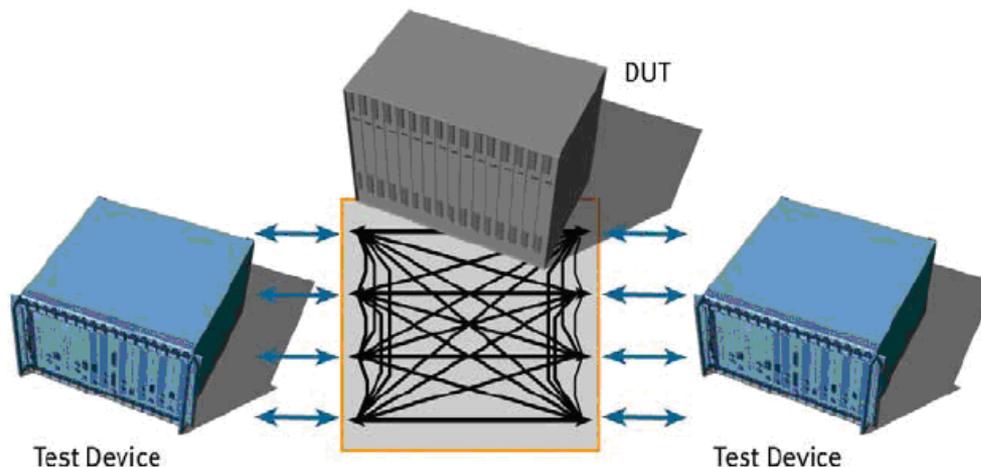
- 为了反映**DUT**在执行不同数量地址查找时的交换能力，需要让测试仪所生成的测试流中包含若干不同的帧目的地址：
 - ✓ 按照 2^n 的方式来进行，即按照 **1, 2, 4, 8, 16, 32, 64, 128, 256,**这样的数量去改变设置，参考值为**1**。
 - ✓ 对于在每端口采用多个地址的测试来讲，测试帧中的源地址和目标地址对还应是随机分布的，以真实反映**DUT**的地址查找性能。

全网状吞吐量、丢帧率和转发速率测试方法(续2)



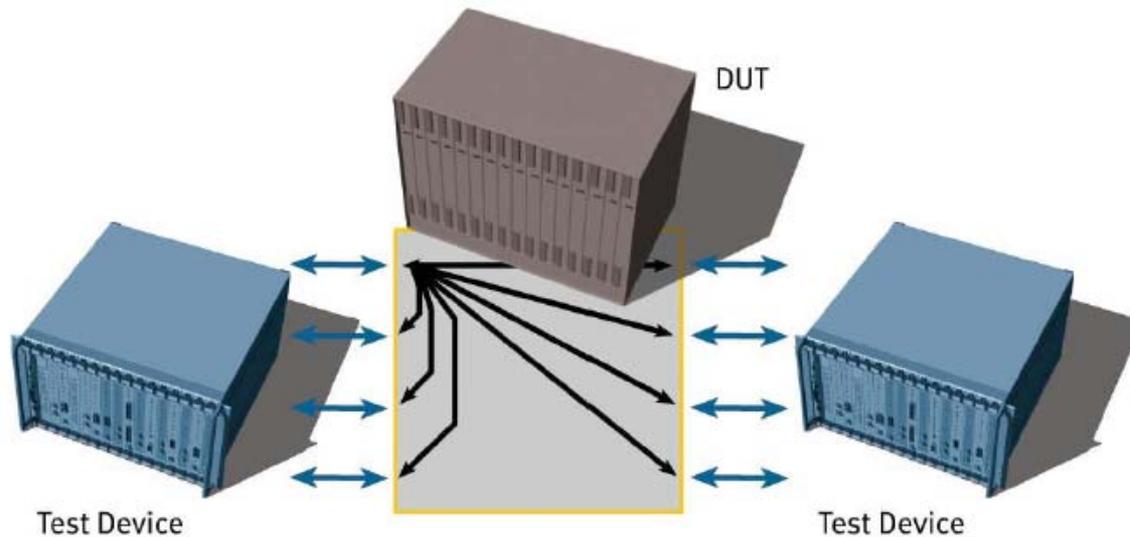
- 整个测试期间所有的端口平等的而且满负载工作，以确保整个测试的平衡：
- ✓ 在不同的目的端口上，目的地址在数量上有相等的分布，每一个目的端口都不会超负荷。
- ✓ 每个端口以循环轮转的方式发送测试帧给其它端口。
- 测试时长，**RFC2889**建议在**1至300秒**之间调整，推荐的测试时间为**30秒**。

全网状吞吐量、丢帧率和转发速率测试方法(续3)



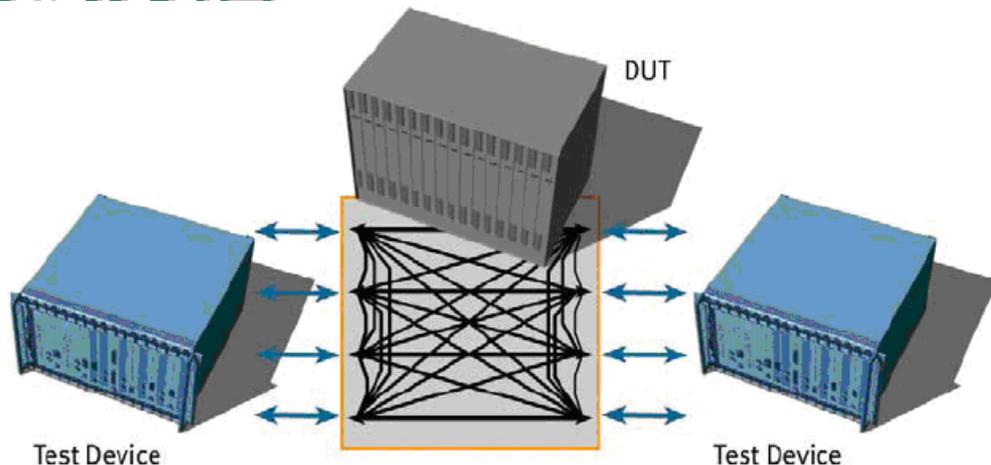
- 对测试结果进行统计时，测试仪表的接收端口必须只将那些源自测试仪表的测试帧才能被统计成接收帧(**Received Frames**)，而忽略任何源于**DUT**的非测试帧。
- 测试结果的报告上，对于负载，规定以媒质的最大理论负载的百分比表示；
- 转发率应当报告为每秒内设备能够被观察到的成功转发到正确目的接口的测试帧的数量，同时作为对一个特定提交负载的响应，要指出所对应的提交负载值。

部分网状一对多/多对一吞吐量、丢帧率和转发速率测试



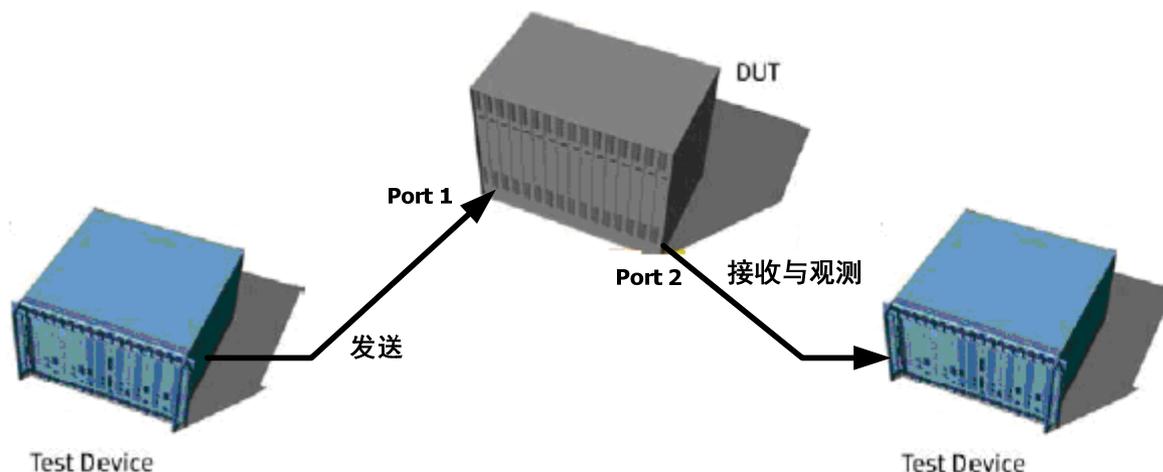
- 为了确定当从**DUT**的一个端口传输到多个端口或从多个端口传输到一个端口时的吞吐量
- 被用来确定**DUT**利用一个以太网端口转发来自多个以太网端口的交换流量的能力，如用于确定一款非对称交换机上行端口为所有其他非上行端口进行数据帧转发的能力。

部分网状一对多/多对一吞吐量、丢帧率和转发速率测试方法



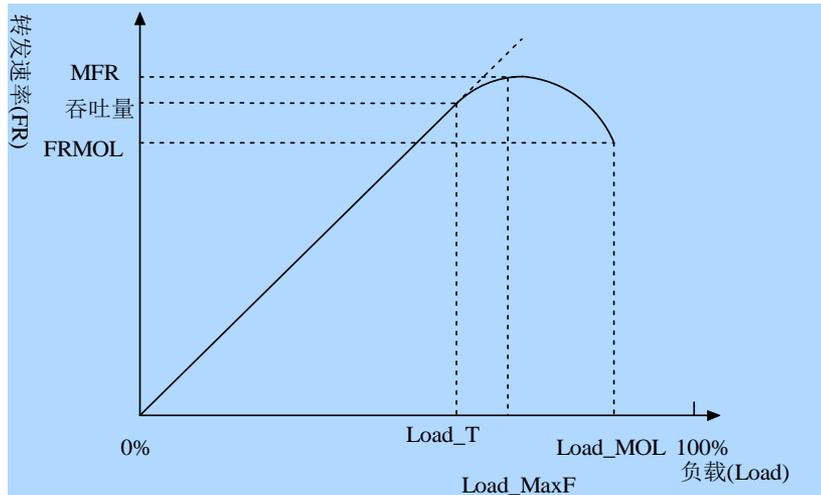
- 主要的测试参数设置、测试过程、测试结果的统计以及测试结果的报告方式均可参照全网状吞吐量测试。
- 与全网状吞吐量测试的主要区别在于流量方式：
 - ✓ 当测试为多对一时，来自多个端口的测试帧必须发往一个端口；
 - ✓ 当测试为一对多时，来自一个端口的测试帧必须以循环轮转方式发往多个端口。

前压和最大转发速率的测试-最大转发速率测试



- 最大转发速率测试的目的是为了度量**DUT**设备在负载发生变化时的转发率峰值。
- 测试仪表生成的测试帧作为负载传输到**DUT**的第一个端口(**port 1**)，观测与度量在**DUT**的第二个端口(**port 2**)进行。
- 帧大小、双工模式、测试时间等主要测试参数设置和测试结果统计方法上都可以参考全网状测试的做法。

前压和最大转发速率的测试-最大转发速率测试(续)



- 为了能够在负载 **Load_T** 和 **Load_MAX**之间找到最大转发速率，须引入步进查找算法
- 设置一个合适的步进长度(**Step**，简称步长)，对负载进行步进，从一组重复的**DUT**转发率测试结果中找到其中的峰值作为最大转发速率(**MFR**)最终报告值。
- 步长值越小，测量结果越精确，但测试所需要的次数就越多。

```
CONSTANT    #定义有关的常量
    MOL = ... frames/sec;

VARIABLE    #定义有关的变量
    MFR    := 0 frames/sec;
    ILOAD := starting throughput in frames/sec; {Offered load}
    STEP   := ... frames/sec; {Step Size}

BEGIN      #开始测试流程
    ILOAD := ILOAD - STEP;
    DO
    BEGIN
        ILOAD := ILOAD + STEP
        IF (ILOAD > MOL) THEN
        BEGIN
            ILOAD := MOL
        END
        AddressLearning; {Port 2 broadcasts with its source address}
        Transmit(ILOAD); {Port 1 sends frames to Port 2 at Offered load}
        IF (Port 2 Forwarding Rate > MFR) THEN
        BEGIN
            MFR := Port 2 Forwarding Rate; {A higher value than before}
        END
    END
    WHILE (ILOAD < MOL); {ILOAD has reached the MOL value}
    DONE
```

传输控制层面功能或性能测试的设计思路

●思路一：

直接给出一些传输控制层面的功能或性能指标，然后对这些指标进行直接的测试；

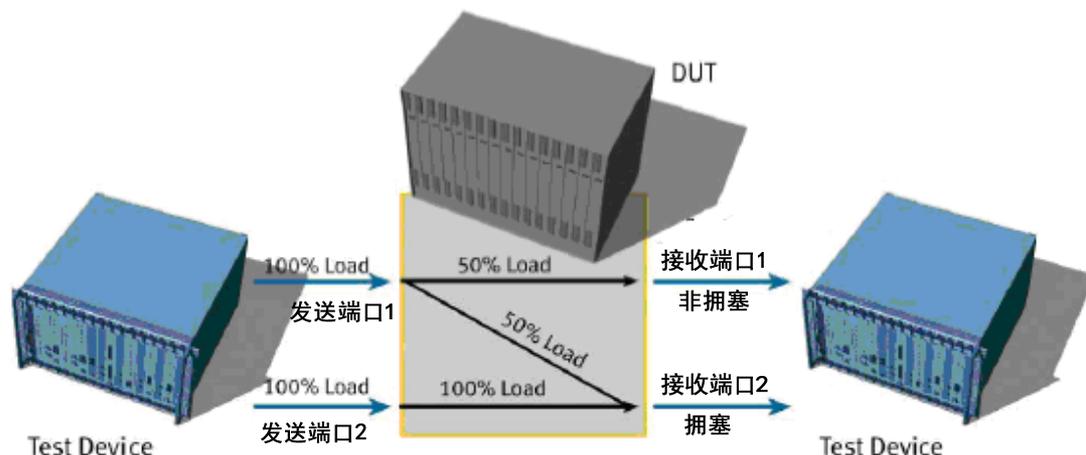
●思路二

控制层面的功能实现与否及其效果最终必然会反映到数据转发层面上来→提出一些测试数据转发层面的性能指标，通过对些指标的测试来间接反映传输控制层面的功能与性能。

●比较两种思路：

思路二采用数据转发层面的指标来测试控制层面功能，更具有说服力。

拥塞控制功能的测试

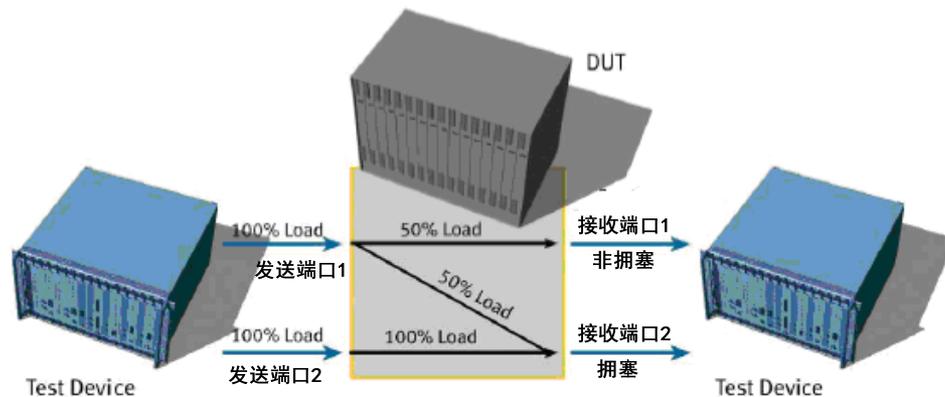


● 采用设计思路二

由DUT上的4个具有相同MOL的端口组成一个测试块。两个端口作为源发送端口，另两个为目标接收端口且分别被配置成拥塞(**congested**)端口和非拥塞(**uncongested**)端口

- 通过观测到拥塞控制机制对拥塞端口和非拥塞端口所产生的不同效果或影响，并可通过对不同结果的比对来确定DUT上是否执行了拥塞控制功能，是否采用了背压机制，是否存在线头阻塞现象。
- 测试帧长、帧间隔、双工模式、每端口地址数和测试时长等，可参照前面吞吐量、丢帧率和转发率测试中的做法

拥塞控制功能的测试



● 判断DUT是否执行了拥塞控制功能

若DUT没有执行拥塞控制，那么对拥塞端口而言，在150%的超载下，测试帧的丢失比例将要达到33%。

若拥塞端口的丢帧率为零，即没有检测到帧丢失，则表明在DUT内有背压机制存在

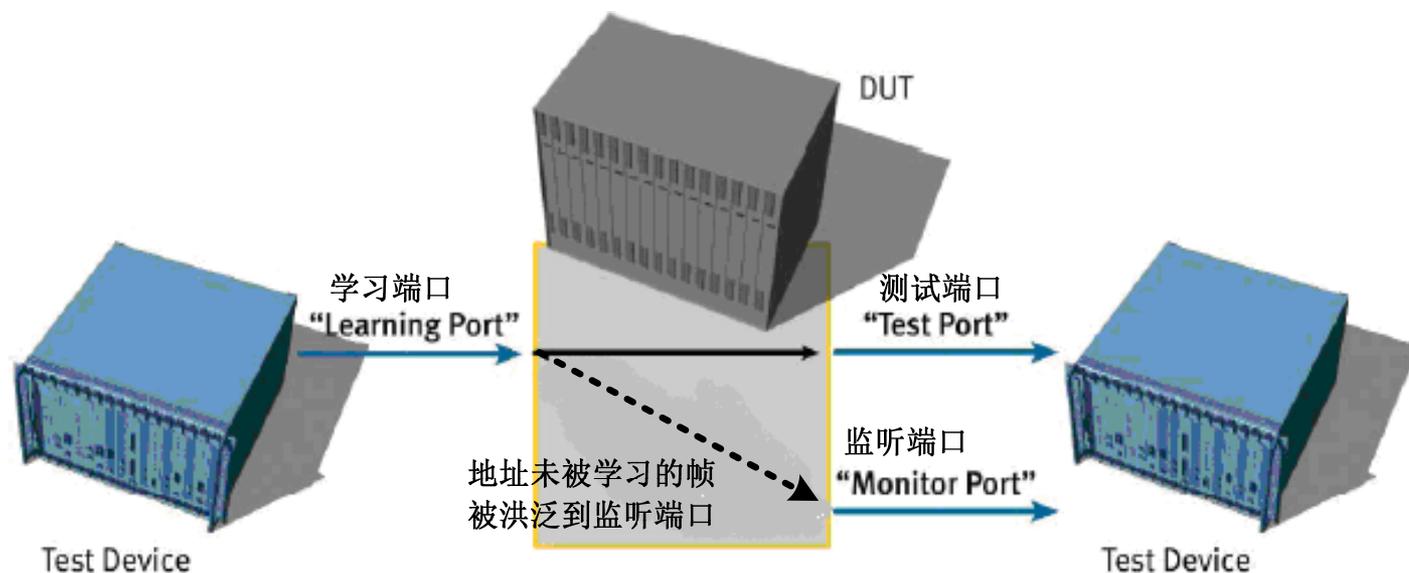
对于非拥塞端口，若丢帧率报告为0，但是检测到的最大转发率又低于50%的MOL，则表明在DUT内有背压机制存在

● 判断DUT中是否存在线端阻塞

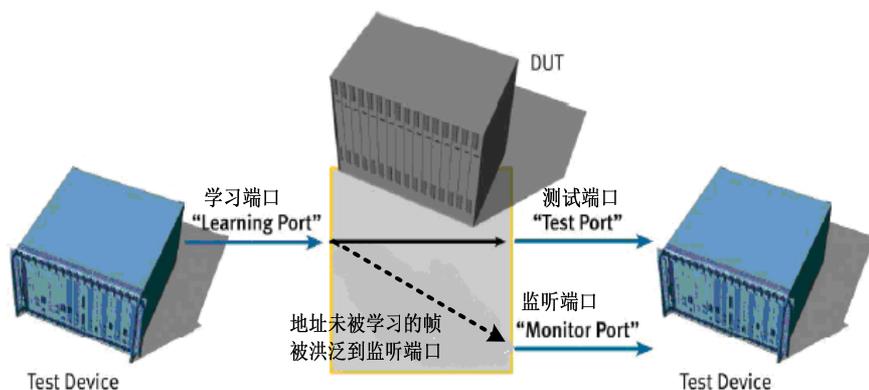
若非拥塞端口检测到有帧丢失，即非拥塞端口的丢帧率不等于零时，则表明DUT中出现了列头阻塞现象。

地址处理功能的测试-地址容量测试

- 地址缓冲容量的测试是为了确定在**RFC 2285**中定义的**LAN**交换设备地址缓冲的能力。
- 设计思想：地址处理属于传输控制层面的功能→利用数据转发层面指标测试来验证传输控制层面功能



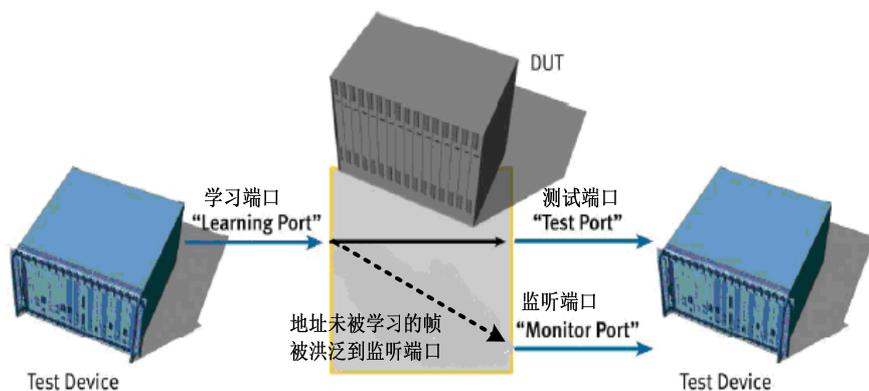
地址处理功能的测试-地址容量测试



- 使用**DUT**上的至少三个端口。分别作为学习端口(**Learning Port**)、测试端口(**Test Port**)和监测监听端口(**Monitor Port**)。

- 为了达到通过数据层面的测试来度量**DUT**地址表容量的目的，采用了将测试端口(**Tport**)所接收的测试帧回传到学习端口，并由**DUT**上的监听端口担当检查端口，以监听是否有洪泛帧或错误转发的帧：
 - ✓ 若监听测端口所统计到的洪泛帧计数不等于零，即它收到了洪泛帧，说明在此前的地址学习过程中，地址表已经发生溢出；
 - ✓ 若监听测端口的洪泛帧计数为零，即它没有收到洪泛帧，说明在此前的地址学习中，**DUT**还拥有足够的地址缓存空间，来确保所有的源地址被**DUT**所学习并写入地址表。

地址容量测试中的查找算法



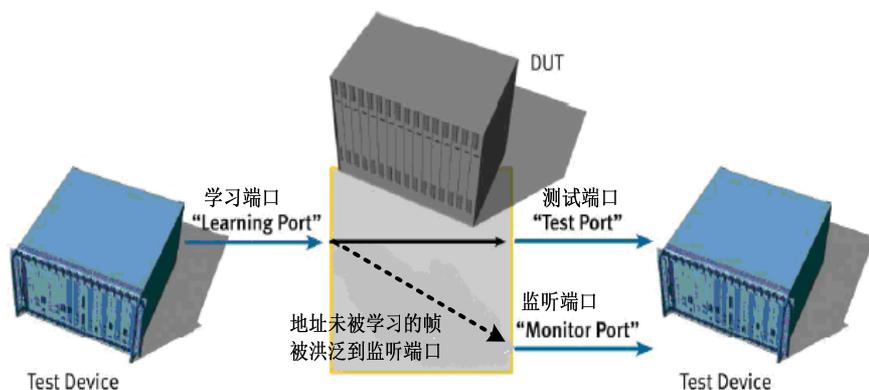
- 为了确定地址表容量，也需要采用某种查找算法来进行反复的测试，以找到DUT所能支持的最大地址数目。
- 二分迭代是目前普遍采用的算法。

- 采用二分迭代算法时，需要设定的测试参数包括最小地址数**Low**、最大地址数**High**和初始地址数**N**。
- 测试端口以初始地址数发送测试帧，并判断监听端口的洪泛帧计数是否为零
- **If 为零**， $N \leftarrow N + (High - N) / 2$ ，再度观测监听端口的洪泛帧计数，反复迭代，一旦监听端口的洪泛帧计数不为零时停止迭代，且将当前地址学习数作为地址表容量；
- **If 不为零**， $N \leftarrow N - (N - Low) / 2$ ，再度观测监听端口的洪泛帧计数，反复迭代，一旦监听端口的洪泛帧计数为零时停止迭代，且将当前地址学习数作为地址表容量。

地址容量二分迭代查找算法伪代码示例

- **CONSTANT** #定义测试过程所用到的常量
 - **AGE = ...;** {value greater than DUT aging time}
 - **MAX = ...;** {maximum address support by implementation}
 - **VARIABLE** #定义测试过程所用到的变量
 - **LOW := 0;** {Highest passed value}
 - **HIGH := MAX;** {Lowest failed value}
 - **N := ...;** {user specified initial starting point}
 - **BEGIN** #开始测试
 - **DO**
 - **BEGIN**
 - **PAUSE(AGE);** {Age out any learned addresses} #清除当前地址表中的内容
 - **AddressLearning(TPort);** {broadcast a frame with its source Address and broadcast destination} #测试前进行TPort目的地址的学习
- **AddressLearning(LPort);** {N frames with varying source addresses to Test Port} #学习端口TPort发送具有不同源地址的学习帧
- **Transmit(TPort);** {N frames with varying destination addresses corresponding to Learning Port} # TPort向LPort回送帧
- **IF (MPort receive frame != 0) OR (LPort receive frames < TPort transmit) THEN**
 - **BEGIN** {Address Table of DUT/SUT was full}
 - **HIGH := N;**
 - **END**
 - **ELSE**
 - **BEGIN** {Address Table of DUT/SUT was NOT full}
 - **LOW := N;**
 - **END**
 - **N := LOW + (HIGH - LOW)/2;**
 - **END WHILE (HIGH - LOW >= 2);**
 - **END** {Value of N equals number of addresses supported by DUT/SUT}

地址容量测试中的若干注意点



合适的地址发送速率

地址发送速率是指测试仪表单位时间内提供给**DUT**学习的新地址数量。不可过大，以免因超过了**DUT**的地址学习速率，而产生地址无法被学习的现象。建议学习帧的提供速率在每秒**50**帧或**50**帧以下。

设置合适的地址老化时间

- ✓ 不可太小，须长于在指定速率下生成测试帧所需的时间，以免过多的老化帧
- ✓ 不可太长，以免每次迭代测试时需要较长的地址表清空时间

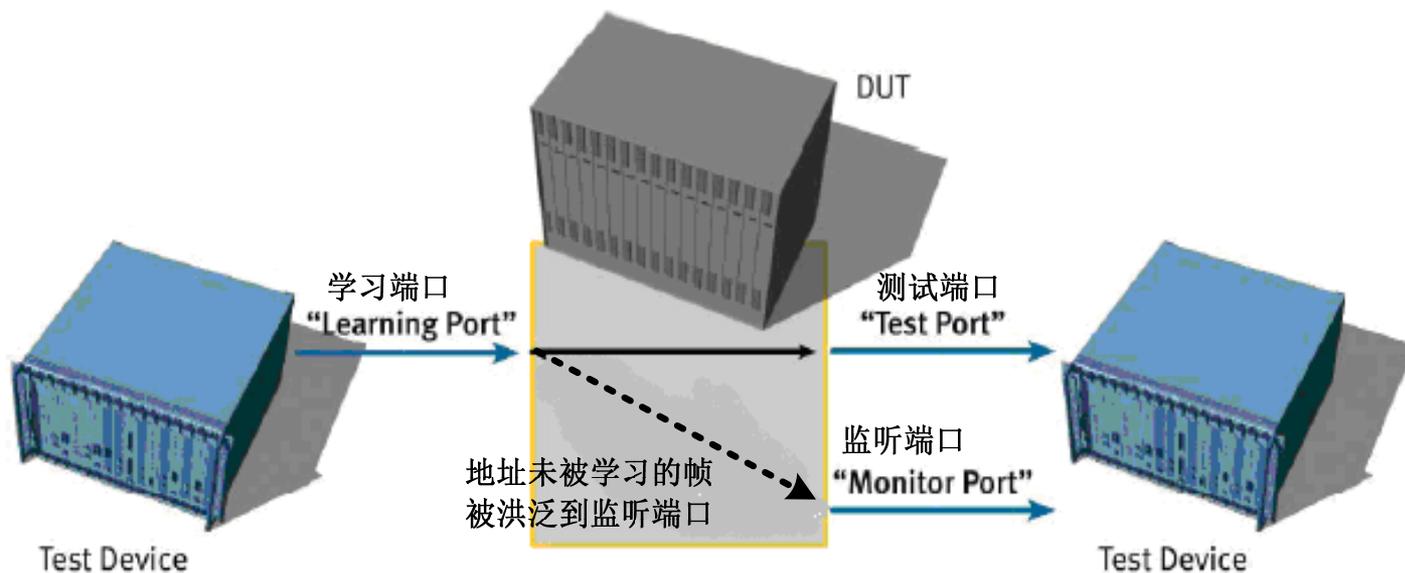
合适的初始化地址数

设置成**1**到**DUT**所支持的最大地址数之间的任何值

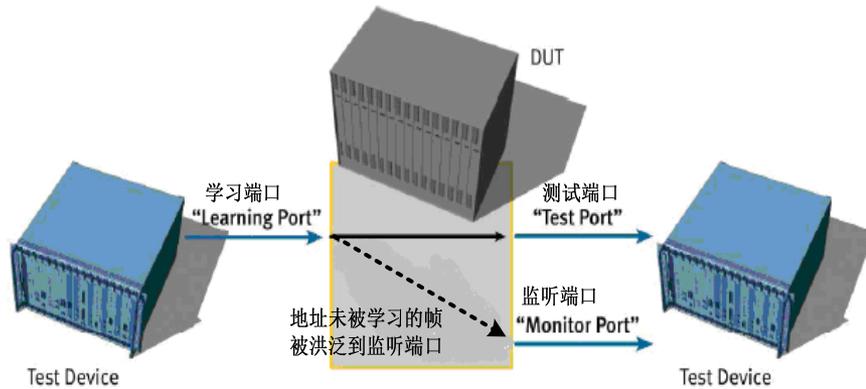
过小或过大(与预期的地址容量相比)都可能导致迭代次数增加，测试时间增长。

地址处理功能的测试-地址学习速率测试

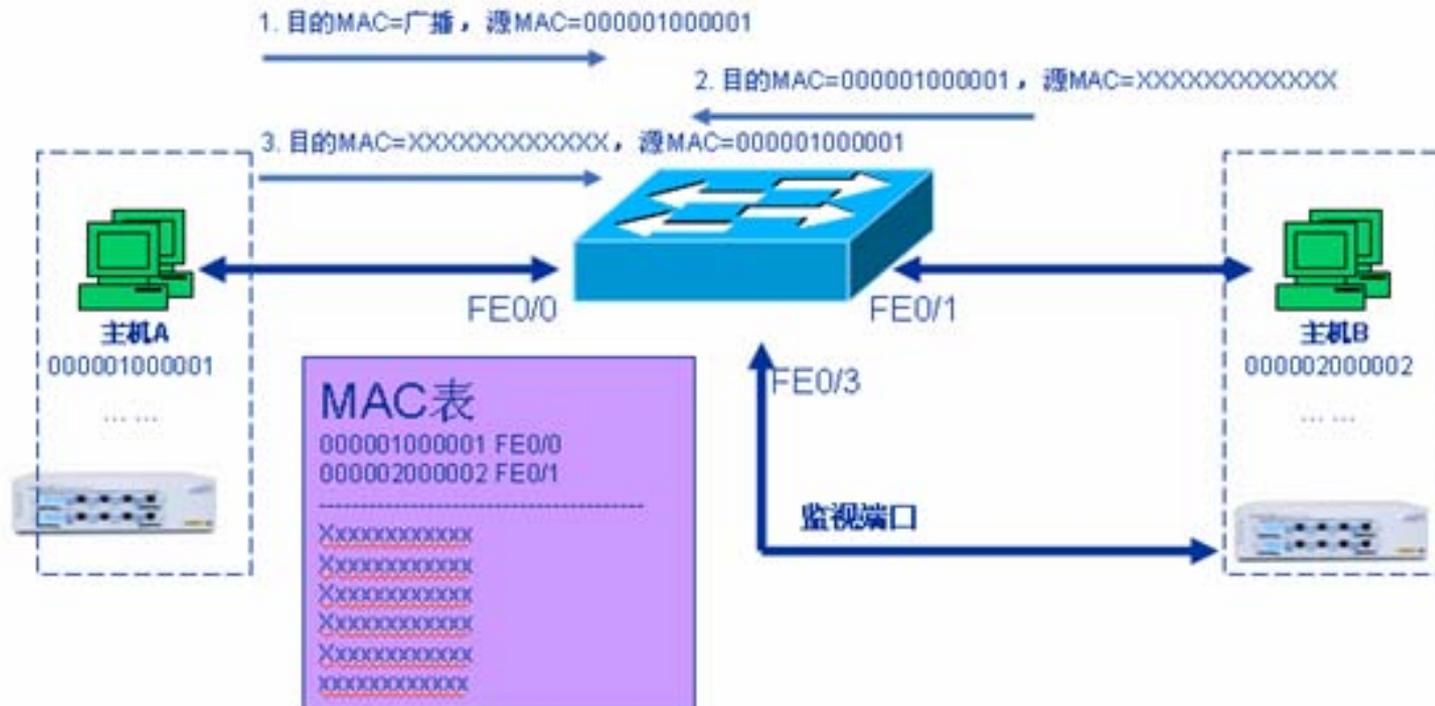
- 目的在于确定**LAN**交换设备的极限地址学习速率。
- 设计思想：地址处理属于传输控制层面的功能→利用数据转发层面指标测试来验证传输控制层面功能
- 与地址容量测试类似的测试方法：
 - ✓ 类似的测试拓扑
 - ✓ 类似的二分迭代查找算法



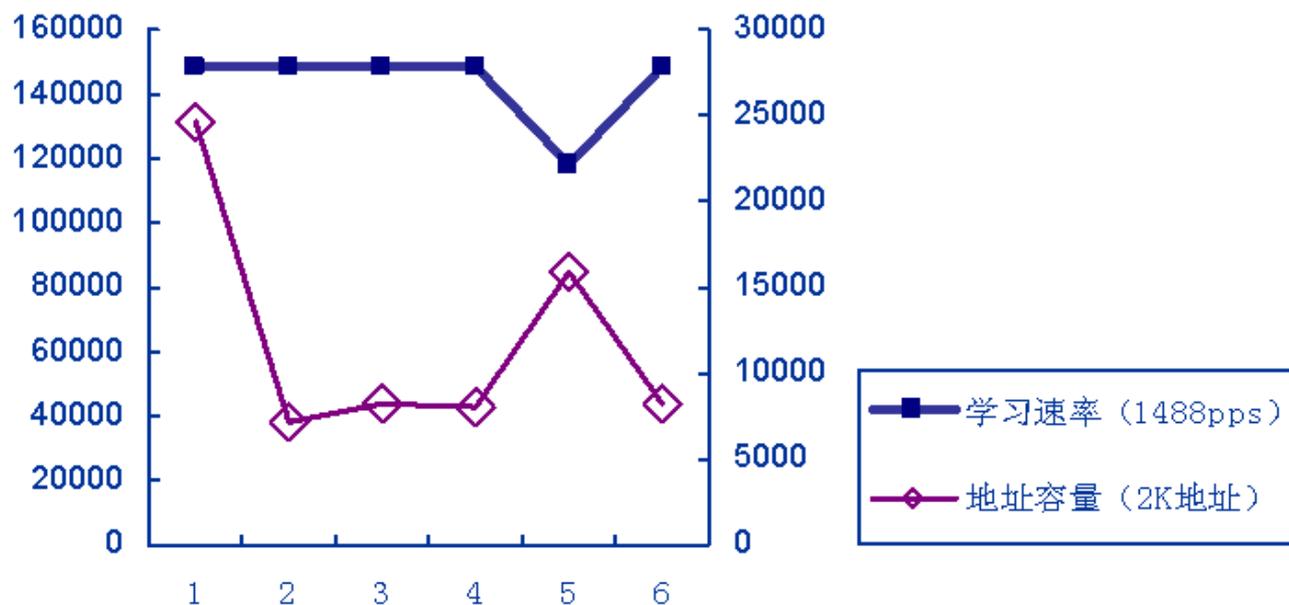
地址处理功能的测试-地址学习速率测试



- 区别在于：基于学习帧发送速率的二分迭代法
- 测试参数设定涉及最小地址发送速率、最大发送速率和初始地址发送速率



交换机地址处理能力测试结果举例

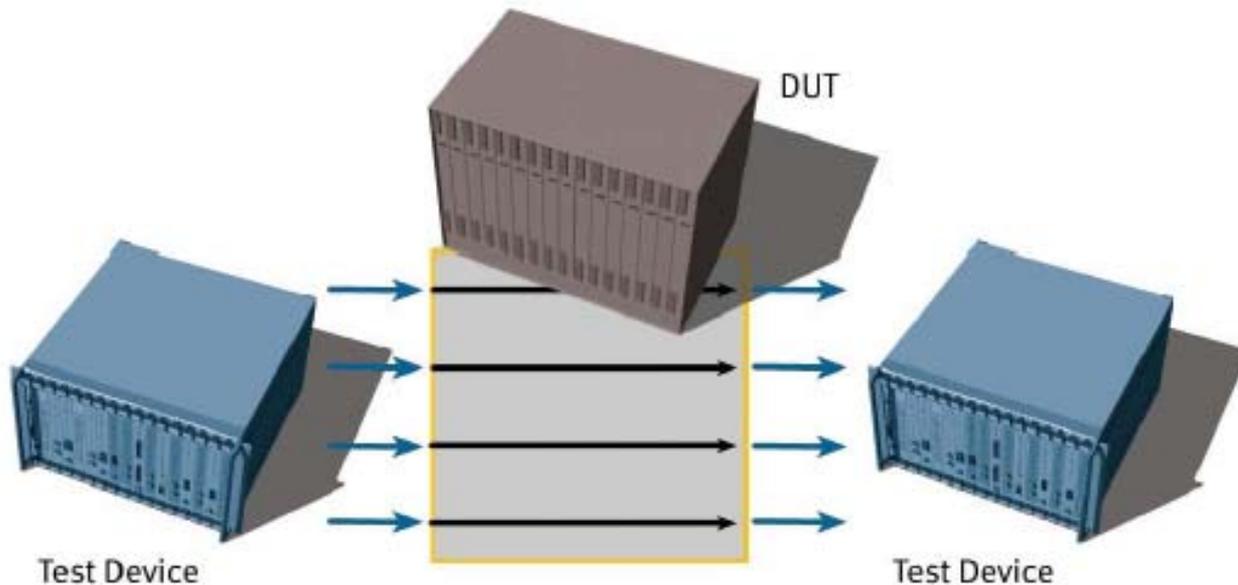


- 横轴的数字表示不同的交换机，左纵轴表示地址表容量，右纵轴表示地址学习速率
- 图给出了六款不同的交换机的地址学习速率与地址容量
- 不同款的交换机在地址学习能力上可能存在较大的差异

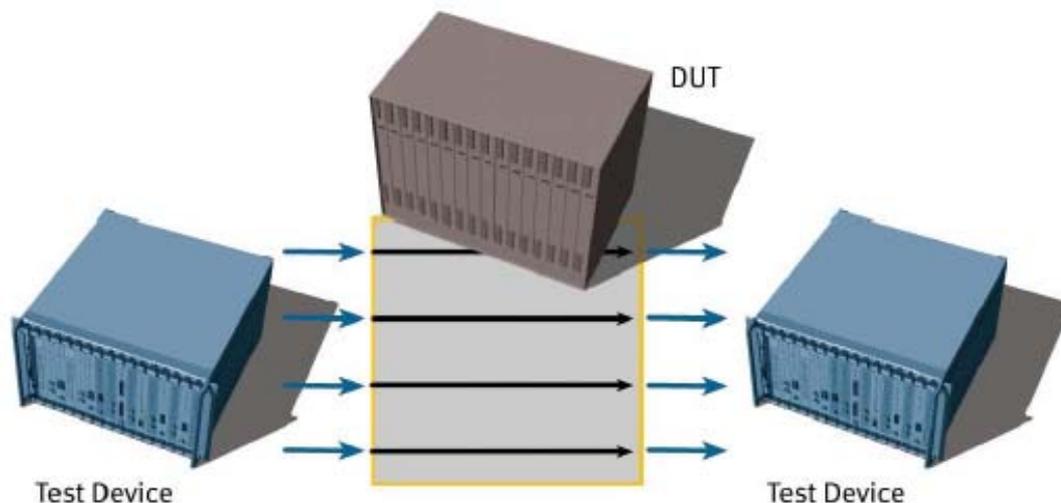
错误帧过滤功能的测试

- 目的在于为了确定交换机在错误或反常帧情况下的行为
- 错误帧过滤功能属于传输控制层面功能→

采用以数据层面的观测来检验传输控制层面功能的设计思想。

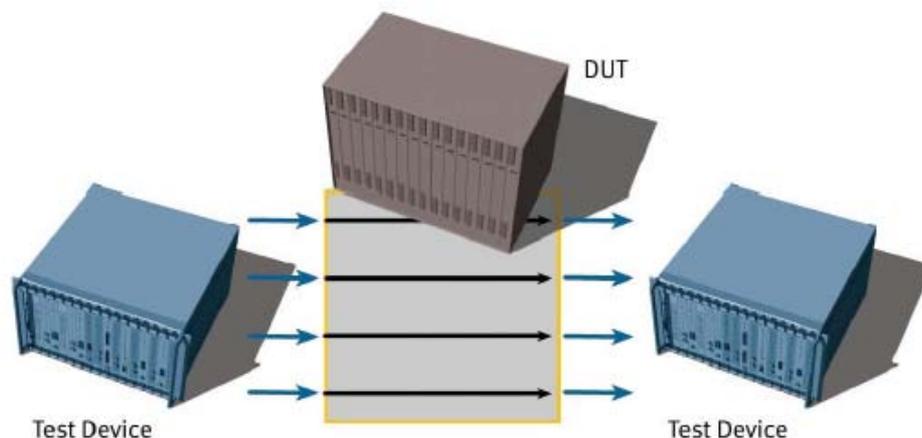


错误帧过滤功能的测试方法



- 采用一组共**8**个端口，其中**4**个作为测试帧的源端口，另**4**个作为目的端口暨监测端口。由测试仪表所生成的错误帧经过**DUT**的**4**个源端口后向**4**个目的暨监测端口发送
- 在目的/监测端口，通过观测是否有错误帧的流量泄漏，来判断**DUT**是否提供了错误帧过滤功能以及所能过滤的错误帧类型
 - ✓ 若某种类型的错误帧能够被观测到，则标记为“失效 (**fail**)”
 - ✓ 若某种类型的错误帧未被观测到，则标记为“通过 (**pass**)”

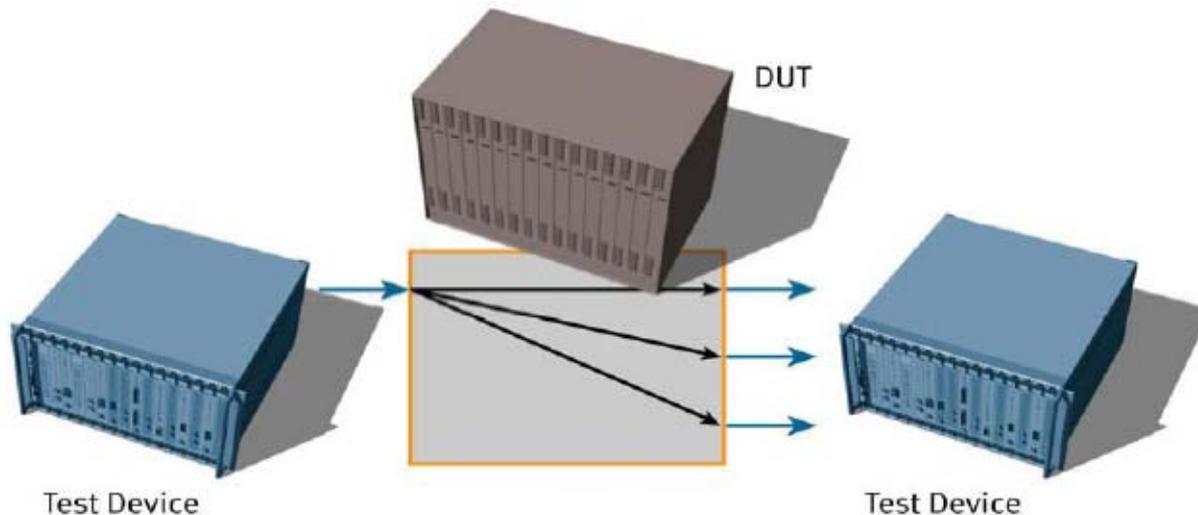
错误帧过滤功能的测试分析



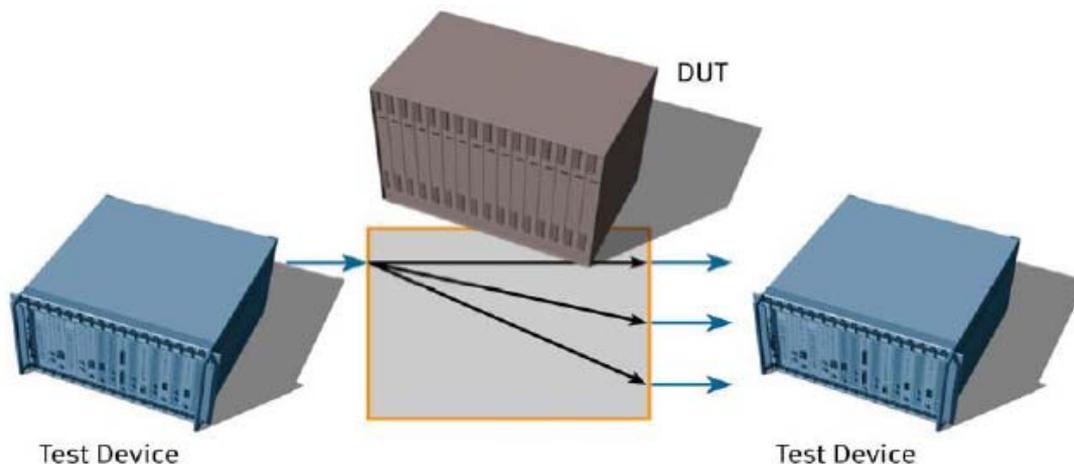
- 根据对各种类型错误帧观测结果的统计与分析：
 - ✓ 若对所有指定类型的错误帧，标记状态为“通过”→**DUT**启用了错误帧过滤功能，且能过滤所有给定类型的错误帧；
 - ✓ 若对某些指定类型的错误帧标记状为“通过”，而另一些类型错误帧的标记状态为“失效”，→**DUT**提供了错误帧过滤功能，但只能过滤某些特定类型的错误帧
 - ✓ 若所有指定类型的错误帧，标记状态为“失效”→**DUT**未提供错误帧过滤功能，或错误帧过滤功能失效。

广播帧转发性能的测试

- 目的在于确定交换机对于广播帧的处理能力，包括：
 - ✓ 广播转发率(**Broadcast forwarding rate**): 一台交换设备在单位时间内向所在广播域的所有端口发送的广播帧数量
 - ✓ 广播时延(**Broadcast latency**): 交换机将广播帧转发到位于同一广播域中的所有端口所需的时间。
- 分别对应于**RFC2889**所提供的广播帧转发和延迟基准测试

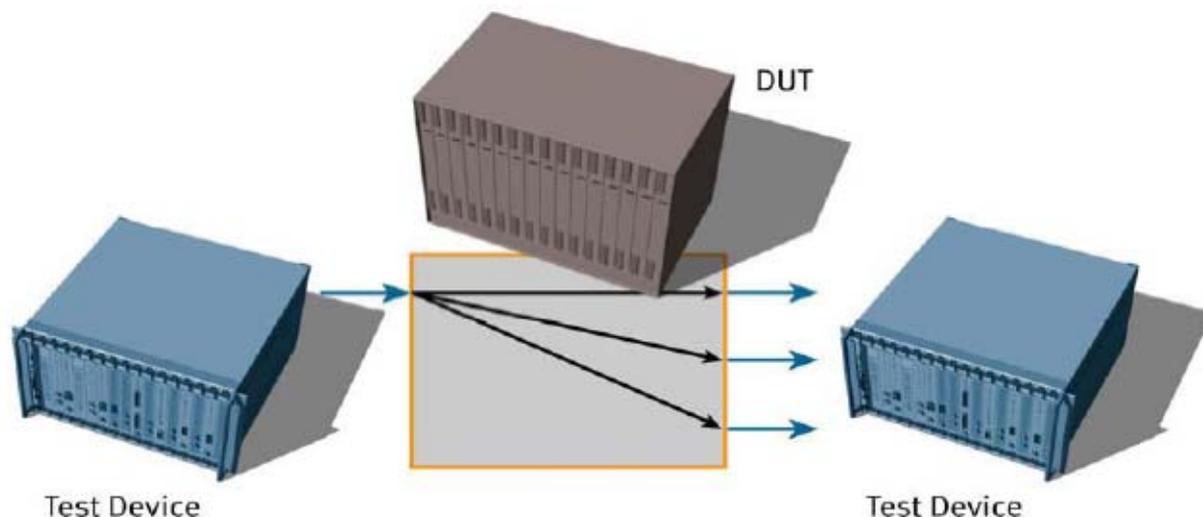


广播帧转发性能的测试拓扑



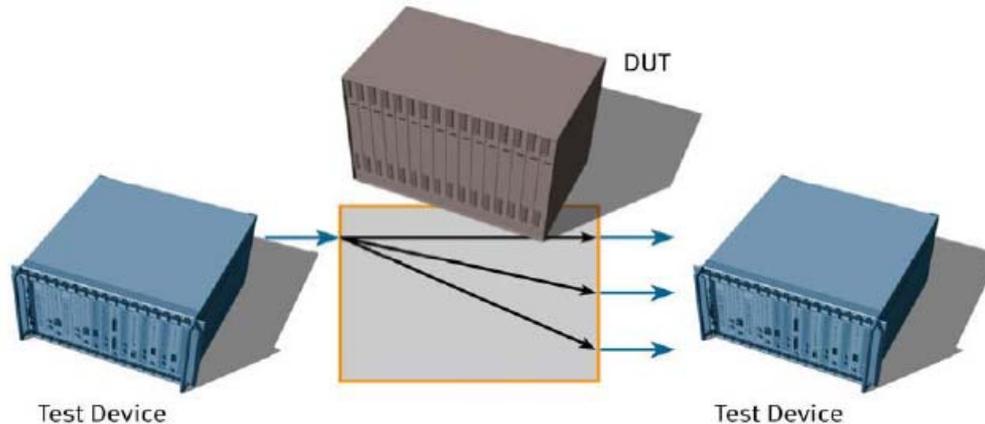
- 由1个广播发送端口和若干个广播接收端口组成一个测试块
- 且确保测试块中的源端口与接收端口位于同一广播域中→若DUT上进行了VLAN划分，那么要确保测试块中的源端口与接收端口位于同一个VLAN中。

广播帧转发性能测试方法-广播吞吐量



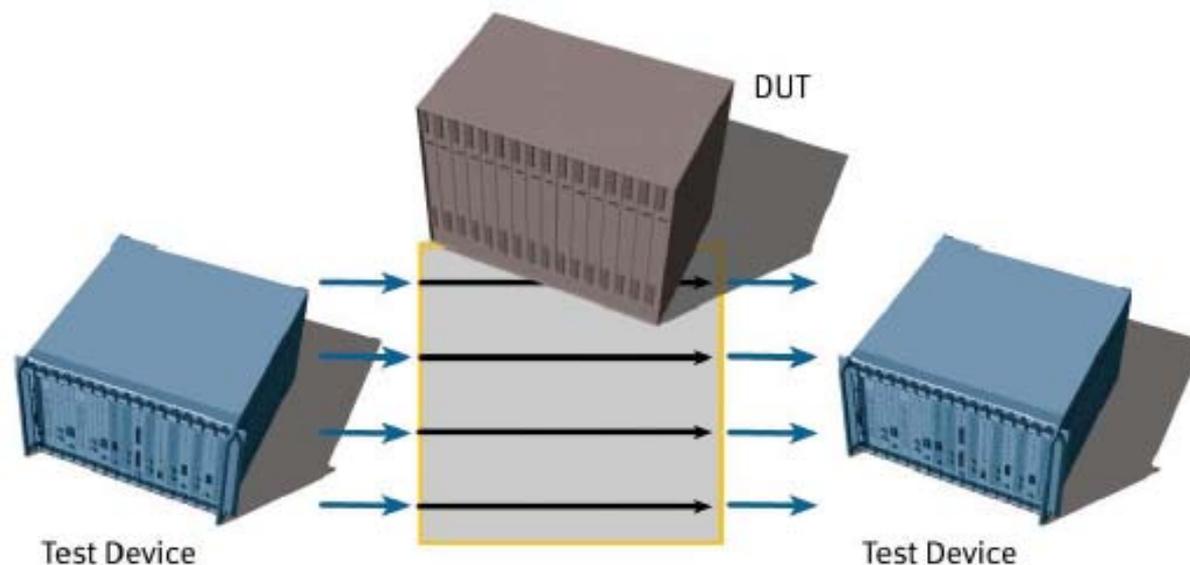
- 测试仪表所生成的目的地址为“**FFFFFFFFFFFFFF**”的广播测试流量通过**DUT**上的唯一源端口发送到各个广播接收端口。
- 在接收端口，测试仪表只选择对广播测试流量进行转发率和帧丢失率的度量。
- 通过测试不同负载下的广播帧转发率和帧丢失率，找到在零丢帧率下的最大广播帧转发率，即广播吞吐量。

广播帧转发性能测试方法-广播延迟



- 测试仪表在每次测试时仅发送一个测试帧，测试帧中包含了一个时戳(**Time Stamp**)
- 在接收端口，通过该时戳来计算帧的广播转发延迟。
- 对于给定的**DUT**，同一广播域中涉及的“**UP**”或接收端口越多，需要复制并转发的广播帧数就越多，因此可能带来转发延时的增大。
- 同一广播域中的不同“**UP**”或接收端口，可能会存在延迟值的漂移，因此若有多个“**UP**”端口在接收广播流量，则需要对每个接收端口的延迟进行度量并取平均值，以作为广播帧延迟的最终值。

广播帧转发性能测试的优化



●因帧的大小会影响**DUT**的广播转发性能→

需要采用多个不同的帧长来实施广播吞吐量和帧延迟测试，以更加全面地反映**DUT**的广播转发性能。