现代计算机网络

李伟明 lwm@hust.edu.cn

1.4.4 传输层

- 问题:怎样实现远程进程间的数据传输
  - 主机-主机的包传输转化成进程-进程通信通道
  - 网络层结构，支持端应用程序--端到端协议
- 什么是连接？
  - 一条连接就是不同系统内的两个实体之间的一个临时性的逻辑关联通路(目IP,源IP,目端口,源端口，传输层协议（TCP/UDP）,五元组)
  - 在连接持续期间，每个实体都跟踪从对方到达和发送到对方的PDU，以便调节PDU的流量以及对丢失和损坏的PDU进行恢复。
- 互联网的全部功能，最基本、最小粒度的服务
  - 端到端数据传输

对传输层协议的希望与IP层现实

- 希望
  - 保障报文传输
  - 以发送相同的顺序传输报文
  - 每个报文最多传输一个拷贝
  - 支持任意长报文
  - 支持收、发之间的同步
  - 允许收方应用流控发方
  - 支持每个主机上的多个进程

- 现实（IP层提供的服务）
  - 丢包
  - 报文重排序
  - 对给定报文传输重复拷贝
  - 限制报文在某个有限大小
  - 在任意长延迟后传输报文
  - 以上是best-effort 层次上的服务，如IP

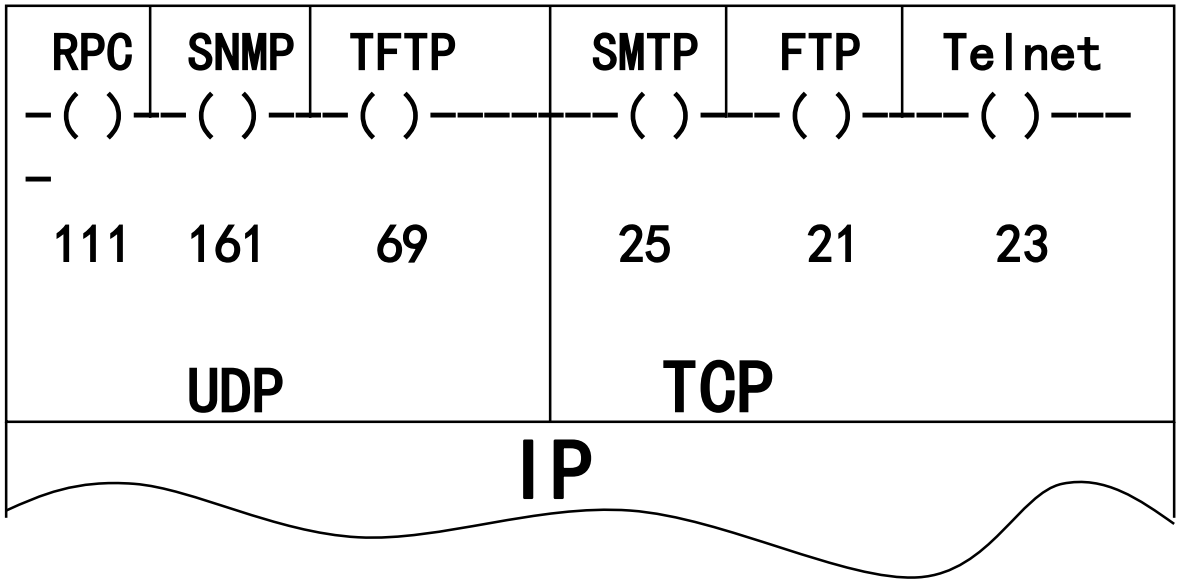
简单多路器-Multiplexer UDP 协议

- 最简单的传输层协议
  - 主机-主机的传输服务在IP协议上扩展成进程-进程的直接通信服务
  - 一台主机上可运行多进程, 需加一多路开关层,区别它们并共享网络

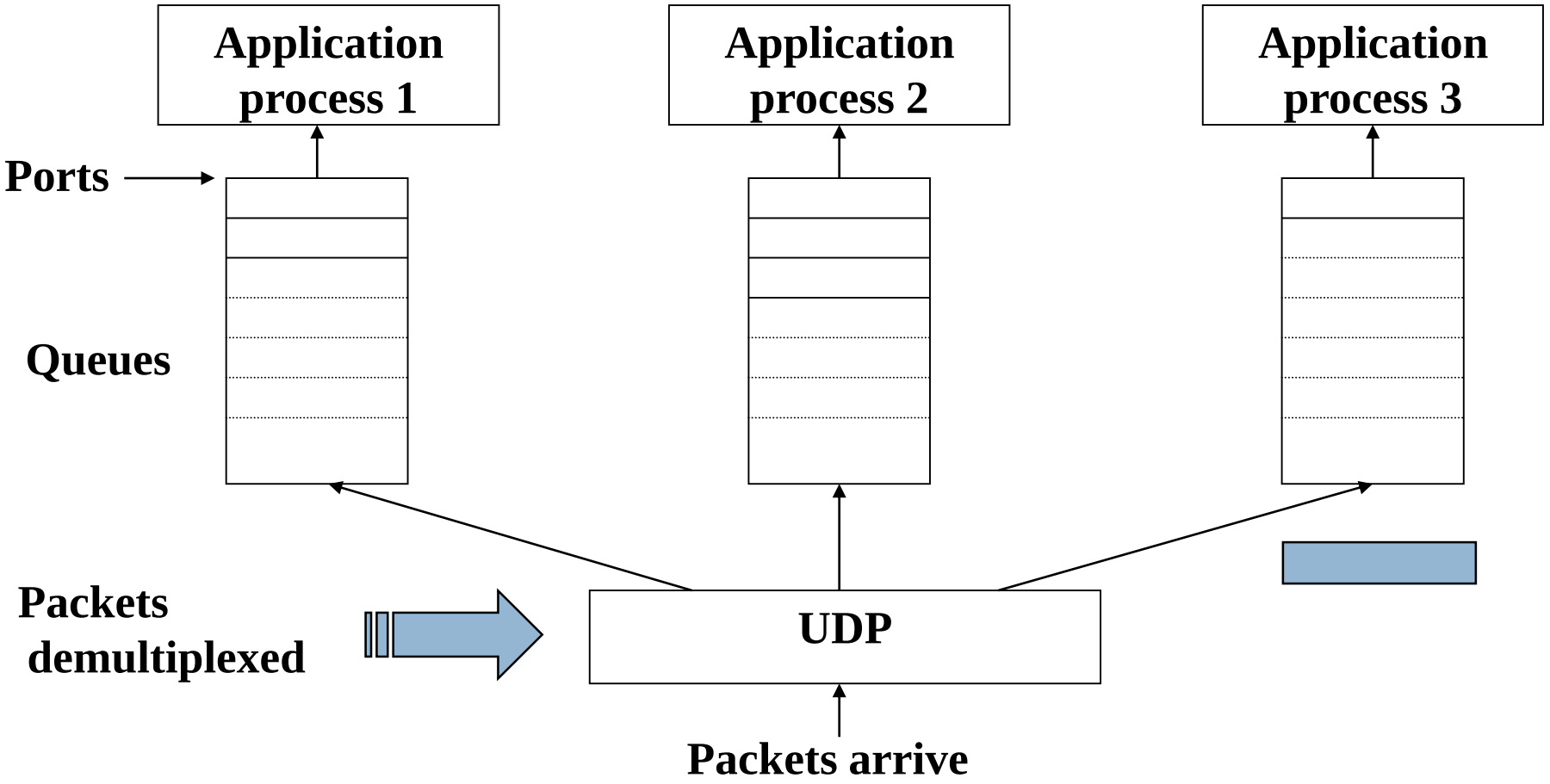
- UDP(User Datagram Protocol):
  - 最小/简单分路协议
  - just port numbers, and an optional checksum
  - no flow control, no congestion control, no reliability or ordering

端口的概念

- 区别源或目的主机上的通信进程
  - 端口:数字，传输层地址
  - 通信源/目的端标识 = 主机IP地址 + 端口号



UDP消息队列


Application process 1
Application process 2
Application process 3
Ports
Queues
Packets demultiplexed
UDP
Packets arrive

UDP协议

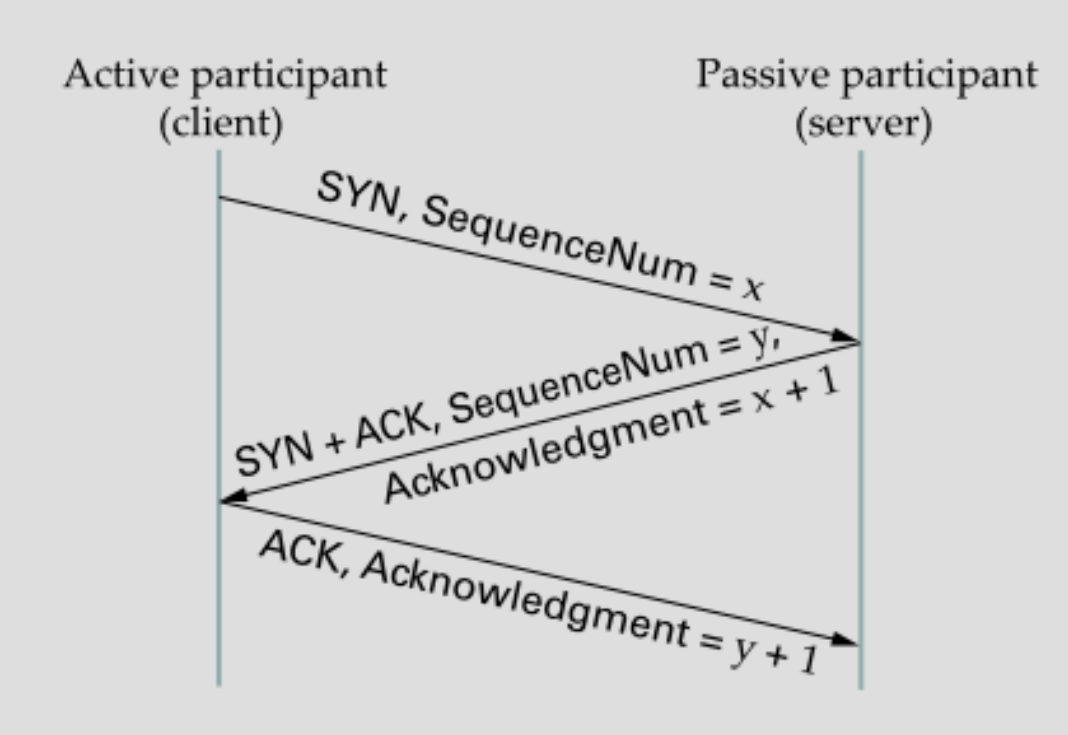
- 提供无连接服务，不保证数据完整到达目的地，减轻了网络的通信负担
- 适应C/S模式的简单请求/响应通信需要
- 应用程序要实施超时重传机制，并对数据包编号，但增加了应用程序的复杂性
- UDP可保留各报文间的边界，不把应用多次发送的数据合并成一个包发出去，且发包后不对该包缓存，这对简单请求/响应很方便
- 组播应用、多数音视频都建立在UDP之上。

可靠字节流协议(TCP)

- TCP:更成熟的传输协议
  - 提供可靠,面向连接,按序字节流
  - 全双工,每个连接支持一对字节流,每个流一个方向
  - 流控机制:允许每个字节流的接收端在给定时间内限制其发送端的数据速率
  - 支持多路输出机制,允许一个主机上同时有多个会话对
  - 还提供拥塞控制机制
- 流控与拥控之差别:
  - 流控:防止发送超过接收者能力（速/量）, 是端到端的发送
  - 拥控:防止过多数据注入到网络中,从而引起交换机或链路超载,拥控是关于主机到网络的发送

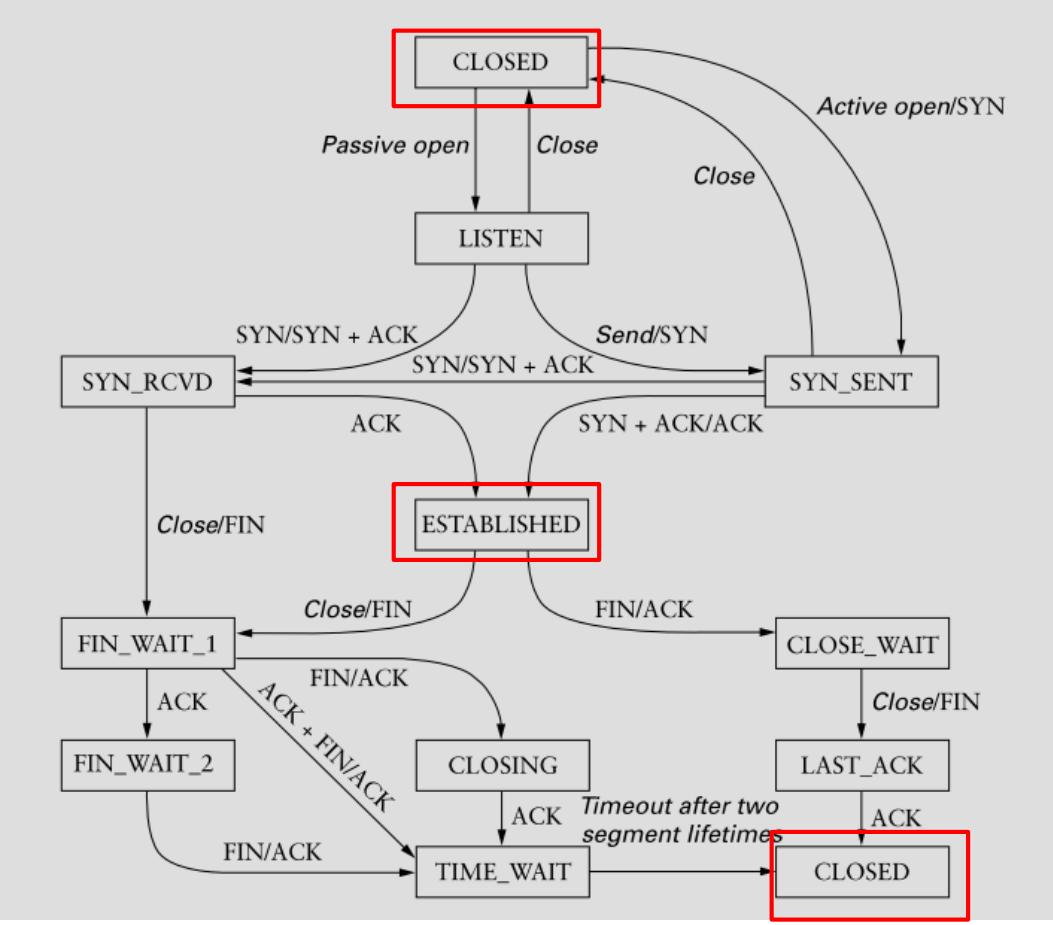
可靠字节流协议(TCP)

- TCP:连接需要建立和拆除



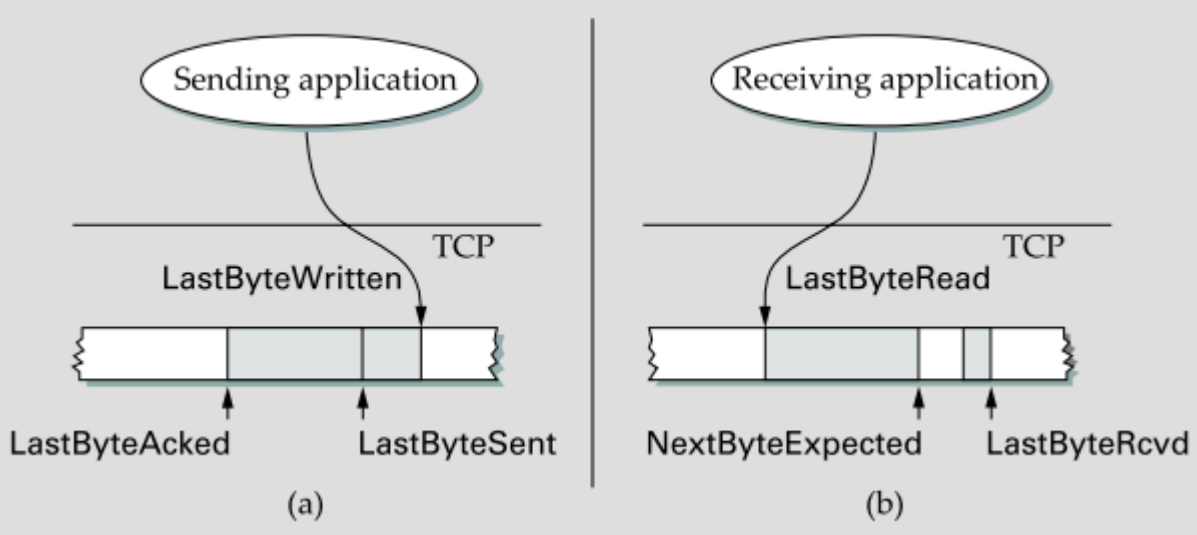
可靠字节流协议(TCP)

- TCP:是有状态的协议，状态机保证合理状态转换（event/action）



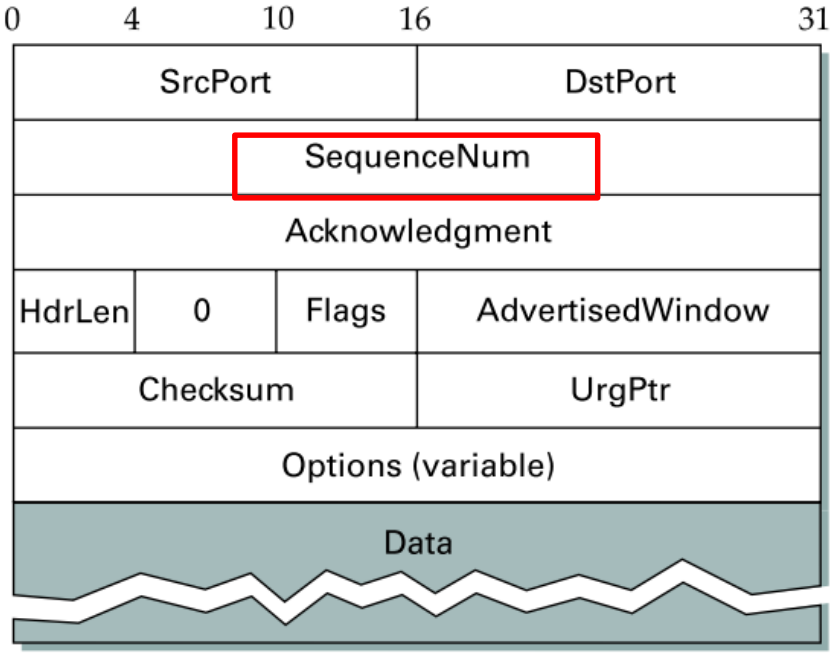
可靠字节流协议(TCP)

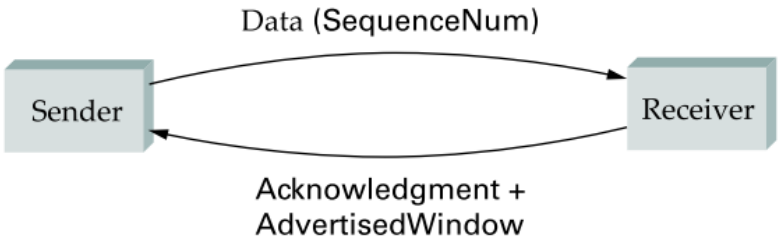
- TCP:滑动窗口（图中字节顺序从左到右，a为发送者，b为接收者）



可靠字节流协议(TCP)

- TCP报文格式：

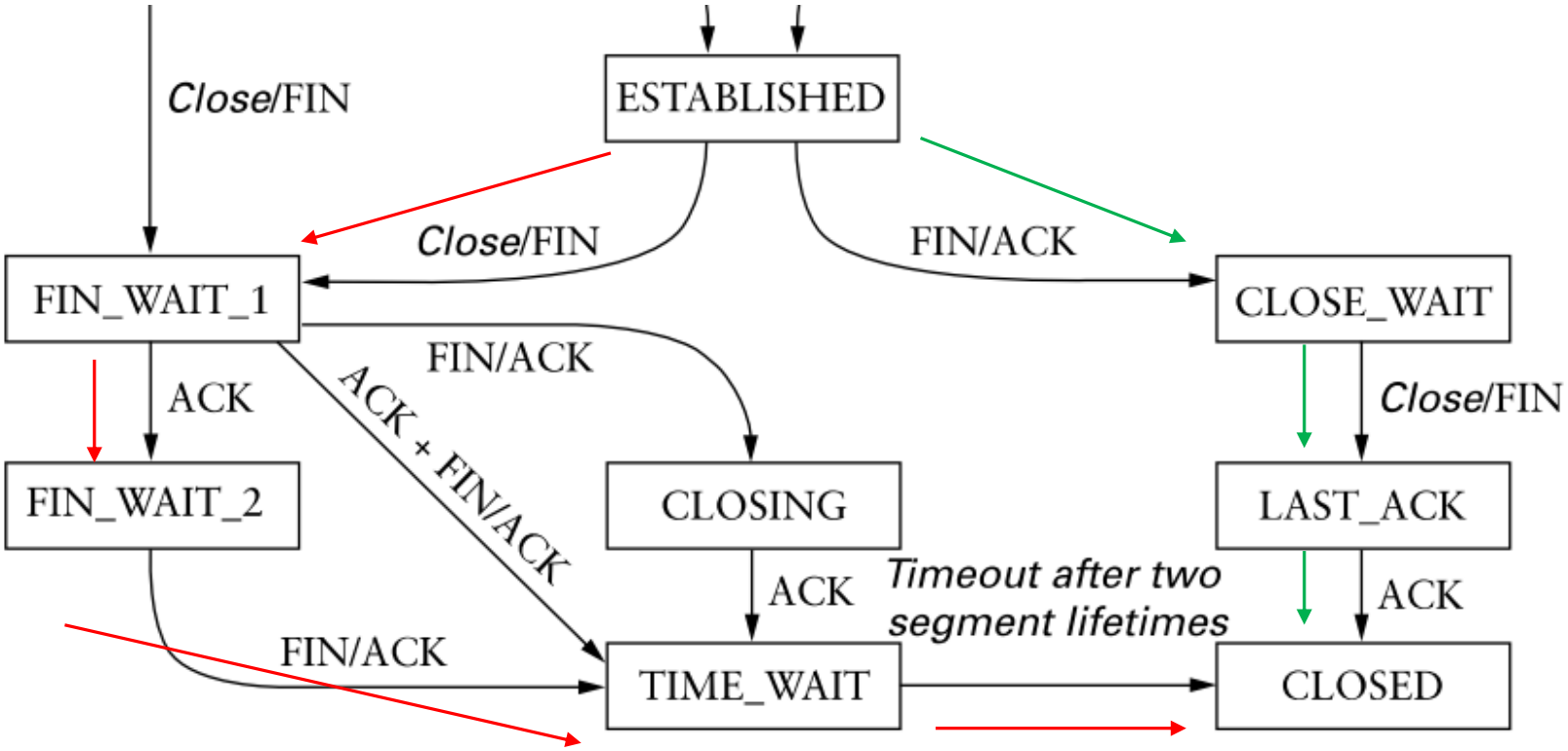




可靠字节流协议(TCP)

TCP结束（event/action）交换四个报文，四次握手

- This side closes first: ESTABLISHED → FIN_WAIT_1 →FIN_WAIT_2 →TIME_WAIT → CLOSED.
- The other side closes first: ESTABLISHED → CLOSE_WAIT → LAST_ACK→ CLOSED.



可靠字节流协议(TCP)

TCP在高速网络遇到问题:1）序列号回绕问题；2）发送窗口太小问题。

Bandwidth	Time until Wraparound
T1 (1.5 Mbps)	6.4 hours
Ethernet (10 Mbps)	57 minutes
T3 (45 Mbps)	13 minutes
Fast Ethernet (100 Mbps)	6 minutes
OC-3 (155 Mbps)	4 minutes
OC-12 (622 Mbps)	55 seconds
OC-48 (2.5 Gbps)	14 seconds

Bandwidth	Delay × Bandwidth Product
T1 (1.5 Mbps)	18 KB
Ethernet (10 Mbps)	122 KB
T3 (45 Mbps)	549 KB
Fast Ethernet (100 Mbps)	1.2 MB
OC-3 (155 Mbps)	1.8 MB
OC-12 (622 Mbps)	7.4 MB
OC-48 (2.5 Gbps)	29.6 MB

图中为假设RTT为100ms，要使得带宽满，需要的发送窗口（参见第一章题目）

可靠字节流协议(TCP)

RFC 1323（RFC7323）：TCP Extensions for High Performance

1. TCP Window Scale
2. Round-Trip Time Measurement
3. Protect Against Wrapped Sequence Numbers

Network Working Group
Request for Comments: 1323
Obsoletes: RFC 1072, RFC 1185

V. Jacobson
LBL
R. Braden
ISI
D. Borman
Cray Research
May 1992

TCP Extensions for High Performance

Status of This Memo

This RFC specifies an IAB standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This memo presents a set of TCP extensions to improve performance over large bandwidth*delay product paths and to provide reliable operation over very high-speed paths. It defines new TCP options for scaled windows and timestamps, which are designed to provide compatible interworking with TCP's that do not implement the extensions. The timestamps are used for two distinct mechanisms: RTTM (Round Trip Time Measurement) and PAWS (Protect Against Wrapped Sequences). Selective acknowledgments are not included in this memo.

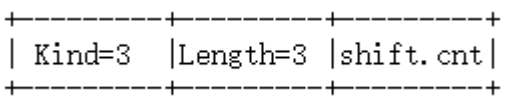
This memo combines and supersedes RFC-1072 and RFC-1185, adding additional clarification and more detailed specification. Appendix C summarizes the changes from the earlier RFCs.

TABLE OF CONTENTS

可靠字节流协议(TCP)

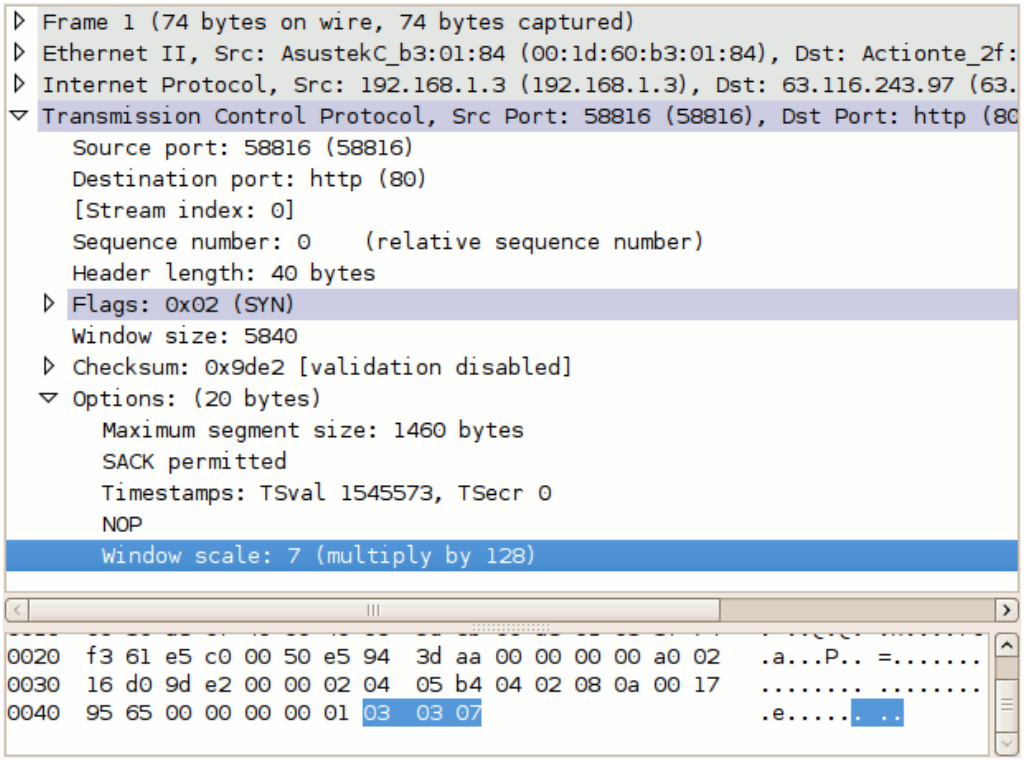
Window Scale Option

- TCP extension involves an option that defines a scaling factor for the advertised window.（one byte shift count，最大2**255倍扩展)
- 这个扩展选项仅仅在SYN发送一次，以后的Window Scale就固定下来
- 是一个3个字节的选项，Kind表示类型，长度3字节，最后一次字节表示Scale多少倍



可靠字节流协议(TCP)

TCP:发送窗口太小问题解决方法Window Scale

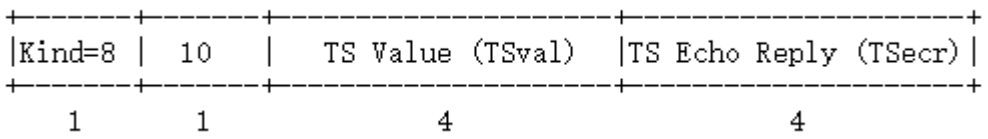


选项Timestamps表示发送这个报文的时间是4字节：1545573来自虚拟的时钟。

选项最后一字节表示Window Scale=07，表示Window Size要乘以2的7次方

可靠字节流协议(TCP)

- **TCP Round-Trip Measurement**
  - TCP增加了一个TCP Timestamps Option (TSopt) 的扩展选项，10个字节，包含了发送方提供的一个时间（TSval），以及接收方回应时的时间（Tsecr）。对于发送方第一个域有效，接收方回应两个域有效



  - TSopt是发送方在SYN报文中提出，以后每个报文都可以包含TSopt了
    1. 发送方在发送数据时，将一个timestamp(表示发送时间)放在包里面
    2. 接收方在收到数据包后，在对应的ACK包中将收到的timestamp返回给发送方(echo back)
    3. 发送发收到ACK包后，用当前时刻now - ACK包中的timestamp就能得到准确的RTT

可靠字节流协议(TCP)

- **PAWS — Protect Against Wrapped Sequence numbers针对的问题（序列号回绕问题）**
  - 假设发送方发了三个报文（Segment）A，B，C，A被阻塞，A2是其重传报文
  - 当接收端在接收到A2后，又接着确认到了数据包B，下一个想接收的数据是数据包C
  - 此时如果收到了数据包A(A从阻塞中恢复过来了，但并未真的丢失)，
  - 由于A与C的序列号是相同的。就会出现数据紊乱，没有做到可靠传输

- **解决办法：**
  1. TS.Recent存放着按序达到的所有TCP数据包的最晚的一个时间戳
  2. 如果收到的一个TCP数据包的timestamp值小于TS.Recnt就丢弃

TCP related papers

- An Analysis of TCP Reset Behavior on the Internet
- Strange Attractors and TCP/IP Sequence Number Analysis

主动响应Active Response：

《An Analysis of TCP Reset Behavior on the Internet》（SIGCOMM 2005）

- 结论：RSTs are surprisingly common on the Internet. They examined a year of SYN/FIN/RST packets from the University of Calgary's border and found that roughly 15% of all TCP flows were terminated by a RST packet *after* payload had already been sent in at least one direction. The reset rate was even higher for HTTP traffic, with 22% of the connections terminated by a client-side RST, and 3% by a server-side RST.

- 方法：Tcpdump截获报文，用Bro分析

主动响应Active Response：

The measurement results in this paper focus on two traces.

The first covers the year-long period spanning October 1, 2003 through September 30, 2004. This trace contains 26,839,809,058 packets, comprising 7,893,035,860 TCP connections.

The second trace records all packets sent via the commercial Internet link between non-university clients and the campus Web server. There are 361,420 connections and 14,393,799 packets in this trace

注意TCP的REJECT情况：means that for every packet received an ICMP port unreachable packet is sent to the source address.

Example: Port 23 is set to REJECT:

```
08:29:33.908826 reddwarf.xix.com.2876 > megahard.xix.com.23: S
    611071769:611071769(0) win 32120
    <mss 1460,sackOK,timestamp 8136624[|tcp]> (DF) [tos 0x10]

08:29:33.908826 megahard.xix.com > reddwarf.xix.com:
    icmp: megahard.xix.com tcp port 23 unreachable [tos 0xd0]
```

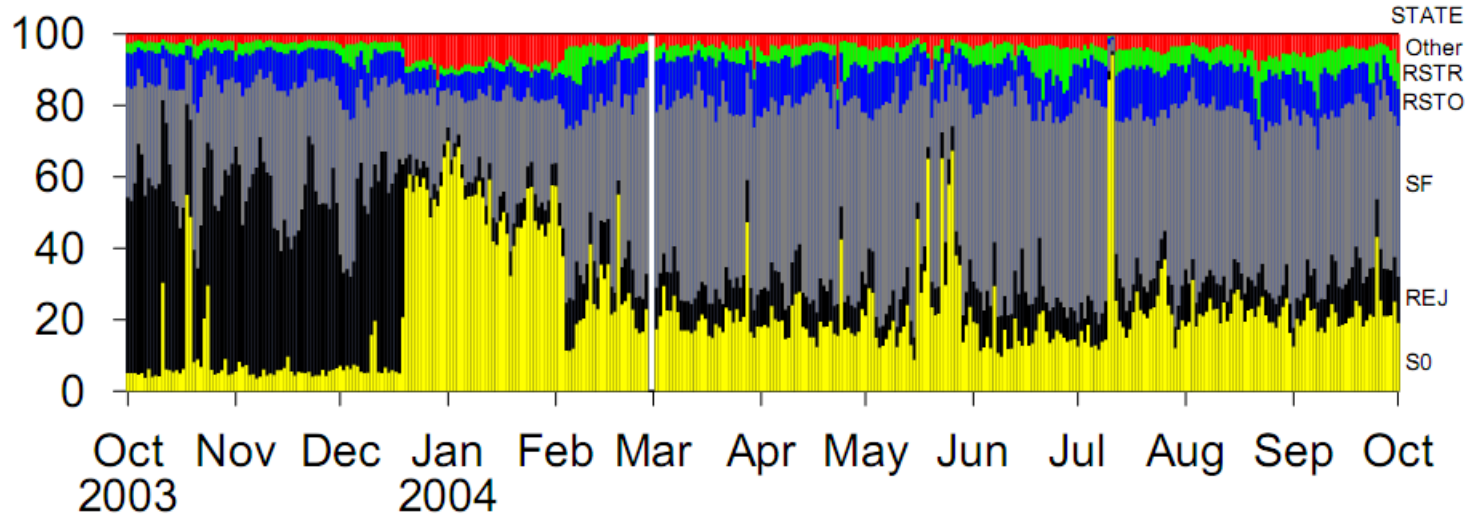
The response to the syn-packet (S) is an ICMP “port unreachable”.

主动响应Active Response：

RSTR: Reset from Server, RSTO: Reset from client

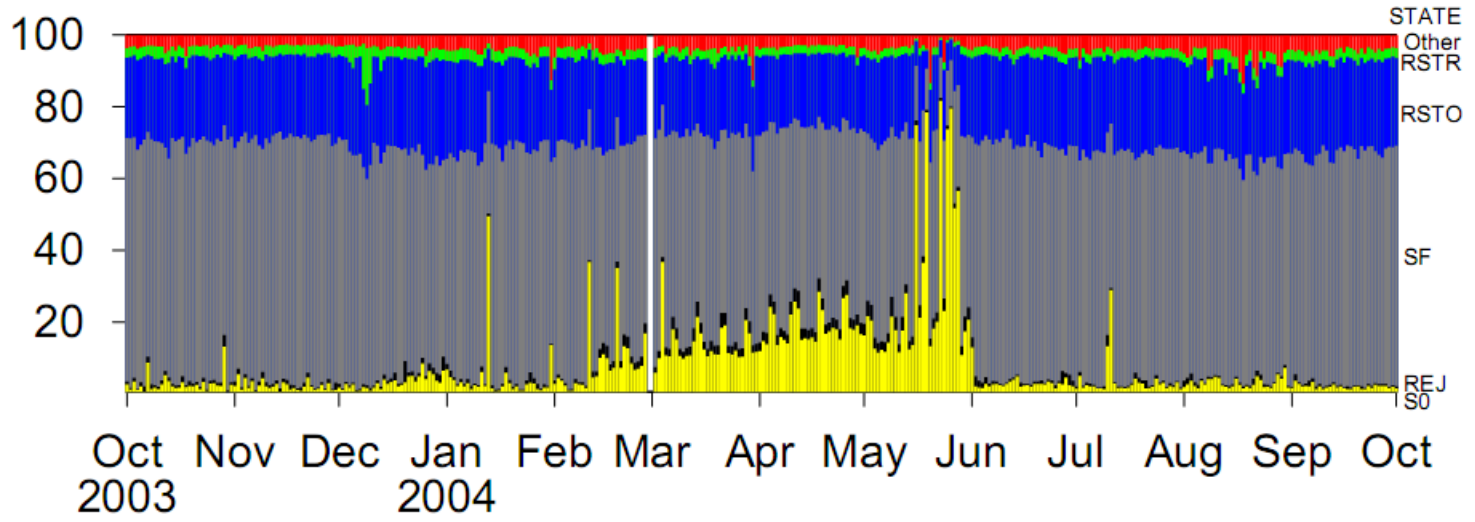
SF: normal, S0 and REJ: only syn or reject connection

All TCP Connections



主动响应Active Response：

RSTR: Reset from Server, RSTO: Reset from client
SF: normal, S0 and REJ: only syn or reject connection
Only HTTP Connections



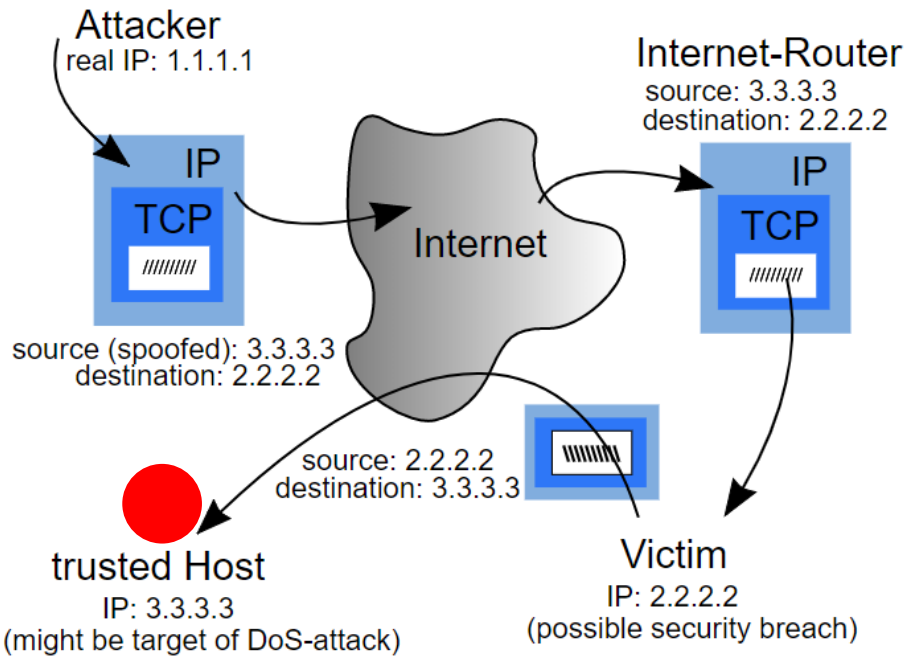
主动响应Active Response，结论：

The most prevalent anomaly is the absence of the normal FIN handshake for connection termination. Instead, connections are often reset by the client.

We believe that particular implementations of HTTP/TCP connection management cause this global trend.

IP Spoofing Attack

Kevin Mitnick’ s Christmas Day crack of Tsutomu Shimomura（下村努）'s machine, employed the IP spoofing and TCP sequence prediction techniques.



U.S. Department of Justice
United States Marshals Service

WANTED
BY U.S. MARSHALS

NOTICE TO ARRESTING AGENCY: Before arrest, validate warrant through National Crime Information Center (NCIC).
United States Marshals Service NCIC entry number: (NIC/ W721460021).

NAME: .....MITNICK, KEVIN DAVID
AKS (S): .....MITNIK, KEVIN DAVID
MERRILL, BRIAN ALLEN

DESCRIPTION:
Sex: .....MALE
Race: .....WHITE
Place of Birth: .....VAN NUYS, CALIFORNIA
Date(s) of Birth: 08/06/63; 10/18/70
Height: .....5'11"
Weight: 190
Eyes: .....BLUE
Hair: .....BROWN
Skin tone: .....LIGHT
Scars, Marks, Tattoos: .....NONE KNOWN
Social Security Number (s): 550-39-5695
NCIC Fingerprint Classification: ...DOPM20PM13DIPM19PM09

ADDRESS AND LOCALE: KNOWN TO RESIDE IN THE SAN FERNANDO VALLEY AREA OF CALIFORNIA AND LAS VEGAS, NEVADA

WANTED FOR: VIOLATION OF SUPERVISED RELEASE
ORIGINAL CHARGES: POSSESSION UNAUTHORIZED ACCESS DEVICE; COMPUTER FRAUD
Warrant Issued: CENTRAL DISTRICT OF CALIFORNIA
Warrant Number: 9312-1112-0154-C

DATE WARRANT ISSUED: NOVEMBER 10, 1992

MISCELLANEOUS INFORMATION: SUBJECT SUFFERS FROM A WEIGHT PROBLEM AND MAY HAVE EXPERIENCED WEIGHT GAIN OR WEIGHT LOSS
VEHICLE/TAG INFORMATION: NONE KNOWN OFTEN USES PUBLIC TRANSPORTATION

If arrested or whereabouts known, notify the local United States Marshals Office, (Telephone: 213-894-2485).
If no answer, call United States Marshals Service Communications Center in McLean Virginia.
Telephone (800)336-0102; (24 hour telephone contact) NLETS access code is VAUSM0000.

Form USM-132 (Rev. 3/2/82)

PRIOR EDITIONS ARE OBSOLETE AND NOT TO BE USED

November 1992

Strange Attractors and TCP/IP Sequence Number Analysis

A paper by Michal Zalewski in 2001。

Michal Zalewski is one of the 15 most influential people in security and among the 100 most influential people in IT。

Strange Attractors and TCP/IP Sequence Number Analysis

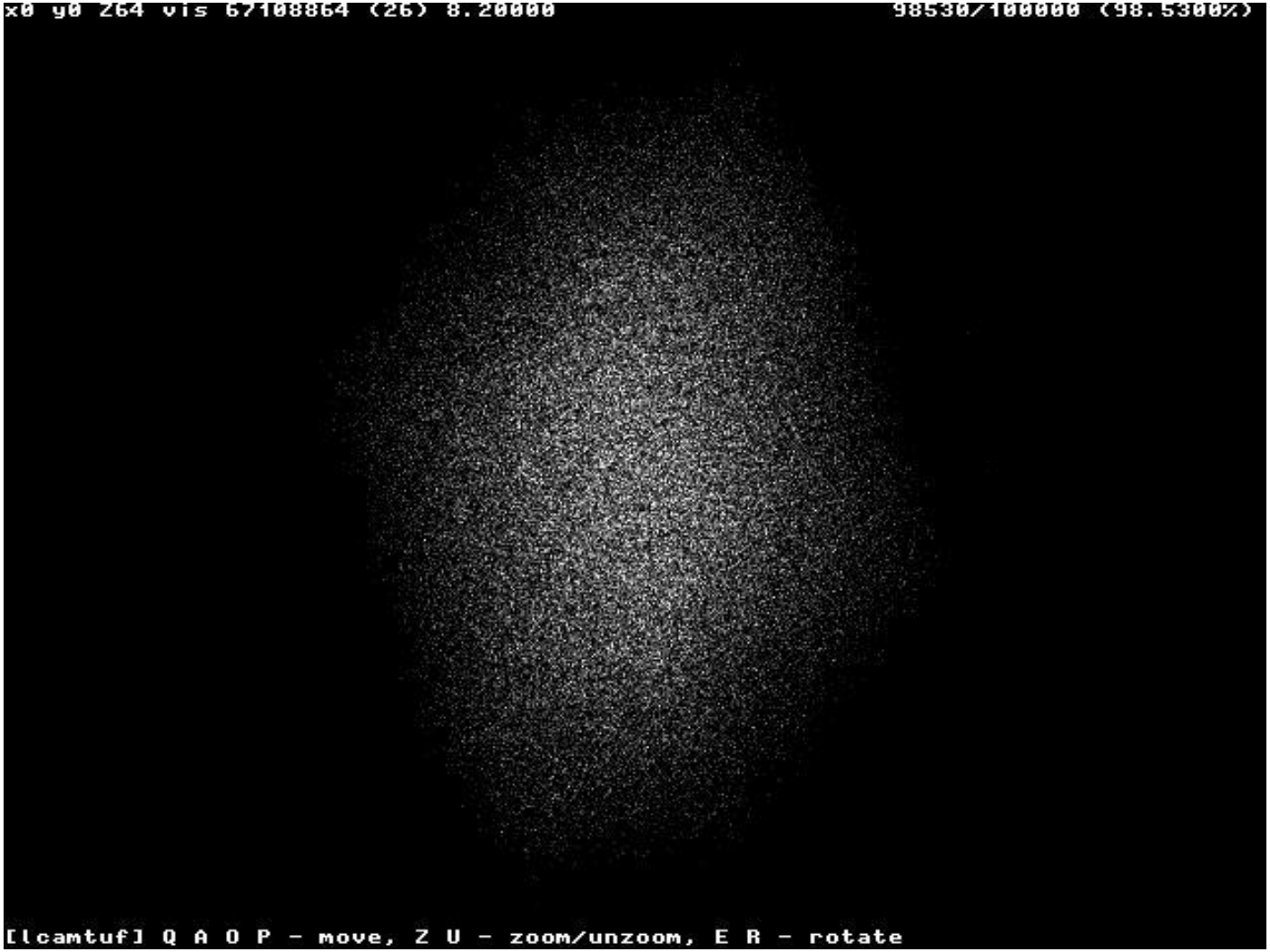
A paper by Michal Zalewski in 2001

- Studied, and graphed the randomness of Initial Sequence Numbers of various operating systems.
- Graphs the output of 100,000 **ISNs** for each OS.
- Attempts an ISN attack on each OS, and lists the difficulty for each.

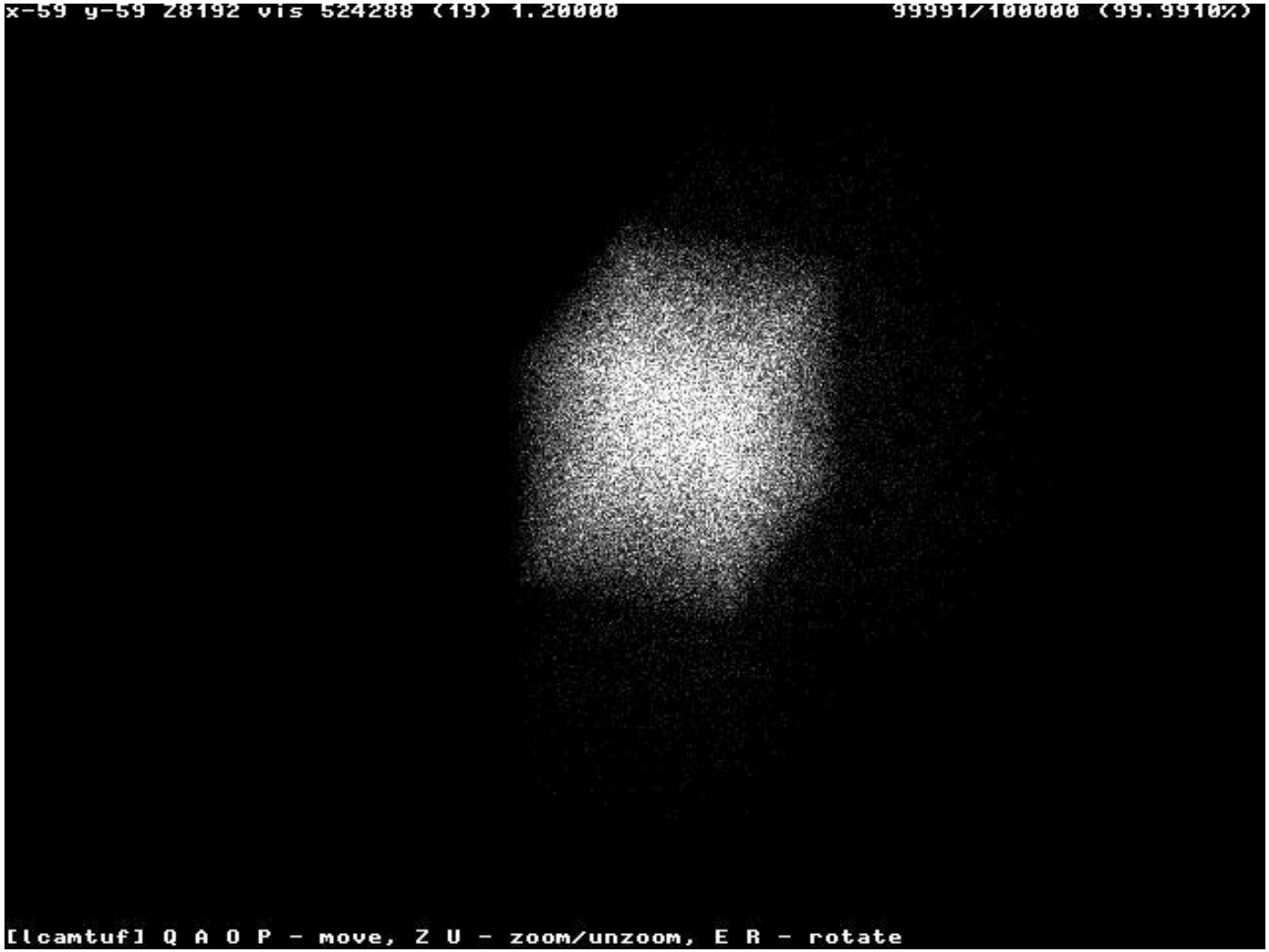
Final Verdict:

OpenBSD is great, Linux is pretty good. Others have big problems.

ISN Graph of: Linux 2.2



ISN Graph of: OpenBSD 2.8



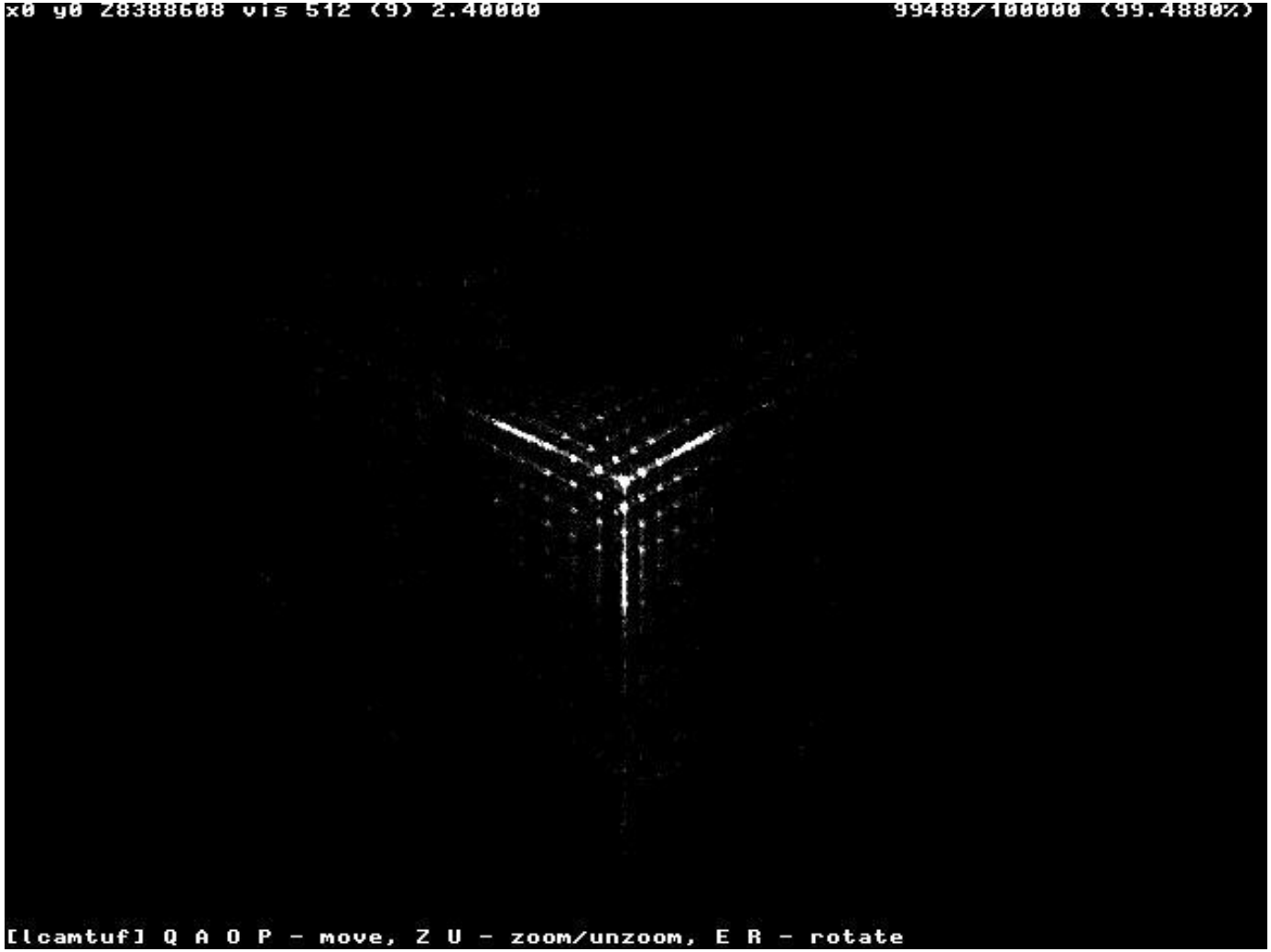
ISN Graph of: OpenBSD 2.9 rewritten by Niels Provos

ISN Graph of: Solaris, weak mode

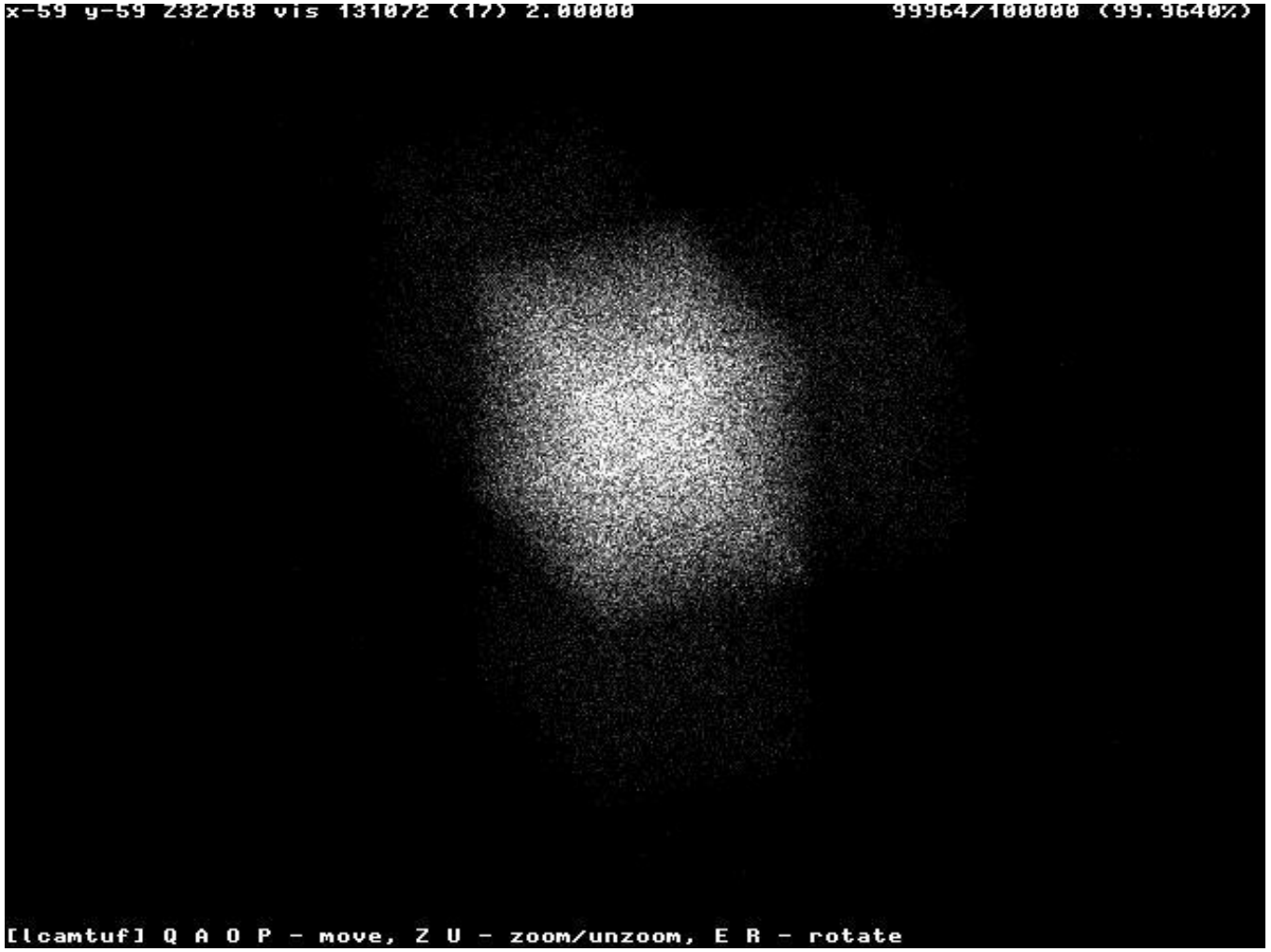
ISN Graph of: Solaris, strong mode

ISN Graph of: Windows 95

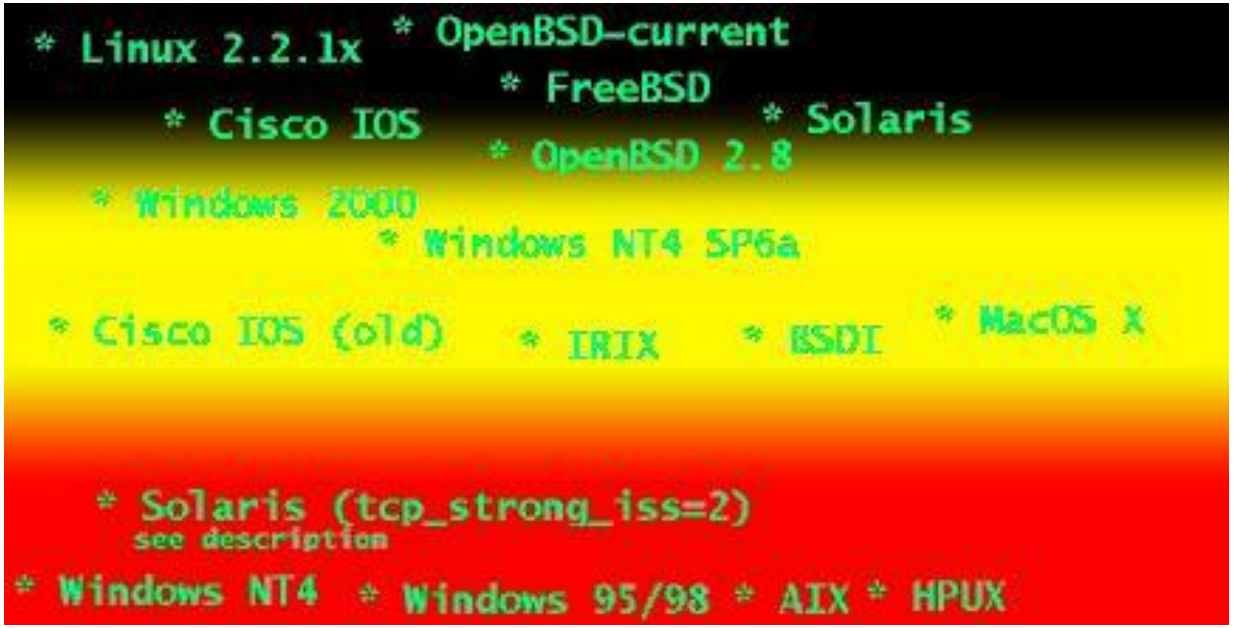
ISN Graph of: Windows 98SE



ISN Graph of: Windows 2000

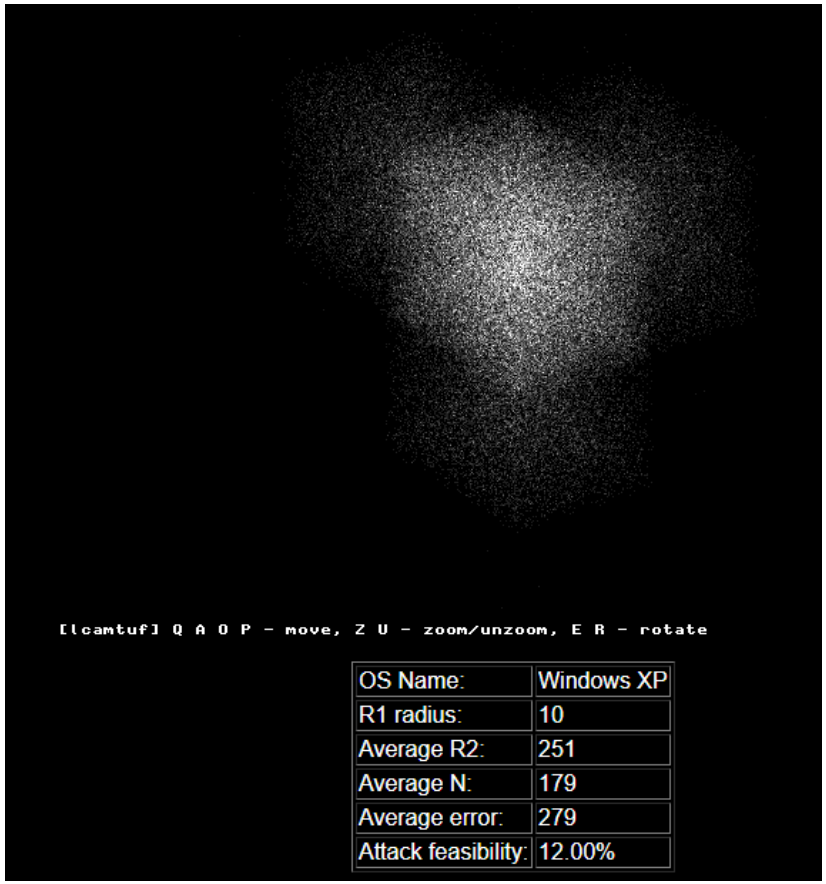


The Result


* Linux 2.2.1x
* OpenBSD-current
* FreeBSD
* Cisco IOS
* Solaris
* OpenBSD 2.8
* Windows 2000
* Windows NT4 SP6a
* Cisco IOS (old)
* IRIX
* BSDI
* MacOS X
* Solaris (tcp_strong_iss=2)
see description
* Windows NT4
* Windows 95/98
* AIX
* HPUX

Strange Attractors and TCP/IP Sequence Number Analysis - One Year Later

一年以后，再次对各种OS的TCP ISN产生进行评价



流控制传输协议:ＳＣＴＰ

- RFC 4960,Oct.2000y
  - SCTP：Stream Control Transmission Protocol
  - 最初设计用于在IP上传输电话信令SS7（可靠/边界），把SS7信令网络的一些可靠特性引入IP，以后扩大了一些其它应用
- 信令类需求
  - Multi-homing，Multi-streaming（different IPs，same port）
  - Message boundaries (with reliability*)
  - Improved SYN-flood protection
  - Tunable parameters (Timeout, Retrans, etc.)
  - A range of reliability and order (full to partial to none)along with congestion control
- UDP/TCP很难满足
  - UDP不可靠、无连接、无顺序、有边界；信令需要面向连接/可靠性！
  - TCP有可靠、有连接、有顺序、无边界；信令需要边界性/部分有序！

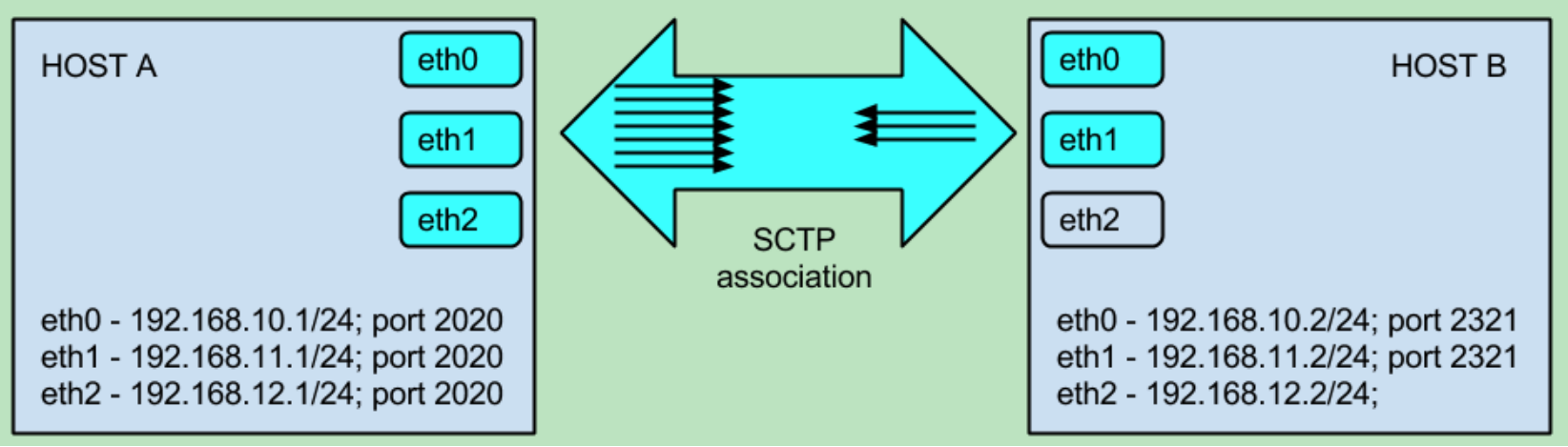
SCTP 关键特点

- Multi-homing improved robustness to failures
  - In TCP,连接仅在 <IP addr,port> 与 <IP addr, port>之间进行；如果接口down，整个连接down
  - In SCTP， For multi-homed, 每端可列出许多 IP addresses ;如果接口down ，仍可通过任何其它地址保持连接
- Multi-streaming reduced delay
  - 部分保序. 消减 Head of Line (HOL) 阻塞
  - In TCP, 所有数据保序; 队列头的丢失导致整个数据段延迟交付
  - In SCTP, 你可发送多达 64K 的独立流, 每个保序流独立,某个流上的丢失并不导致其它流延迟交付
- Message boundaries preserved easier coding
  - In TCP 打包并不保留报文的边界
  - In SCTP 保护报文边界,应用层协议容易写入，编码简单！

SCTP 关键特点

Multi-homing Multi-streaming reduced delay

- 一个SCTP的连接在建立的时候通过协商，两端可以包含多个IP地址和端口
- SCTP有心跳监控报文
- 一旦某个接口出问题，还可以继续传输



SCTP 关键特点

- Improved SYN-flood protection *more secure*
  - TCP 易受 SYN flooding攻击;
  - SCTP 采用四次握手，保护免受SYN flooding攻击
- Tunable parameters (Timeout, Retrans, etc.) *more flexibility*
  - TCP 参数调整只有系统管理员才能进行，实施内核的改变和锁定等
  - SCTP 参数可由socket basis调整
- Congestion controlled unreliable/unordered data *more flexibility*
  - TCP 虽有拥控，但不能做不可靠/失序的交付
  - UDP 虽能做不可靠/失序的交付, 但没有拥控
  - SCTP 总有拥控，且能在部分/全范围提供可靠性、保序的服务
  - SCTP, 可靠/不可靠数据都能在相同连接上多路

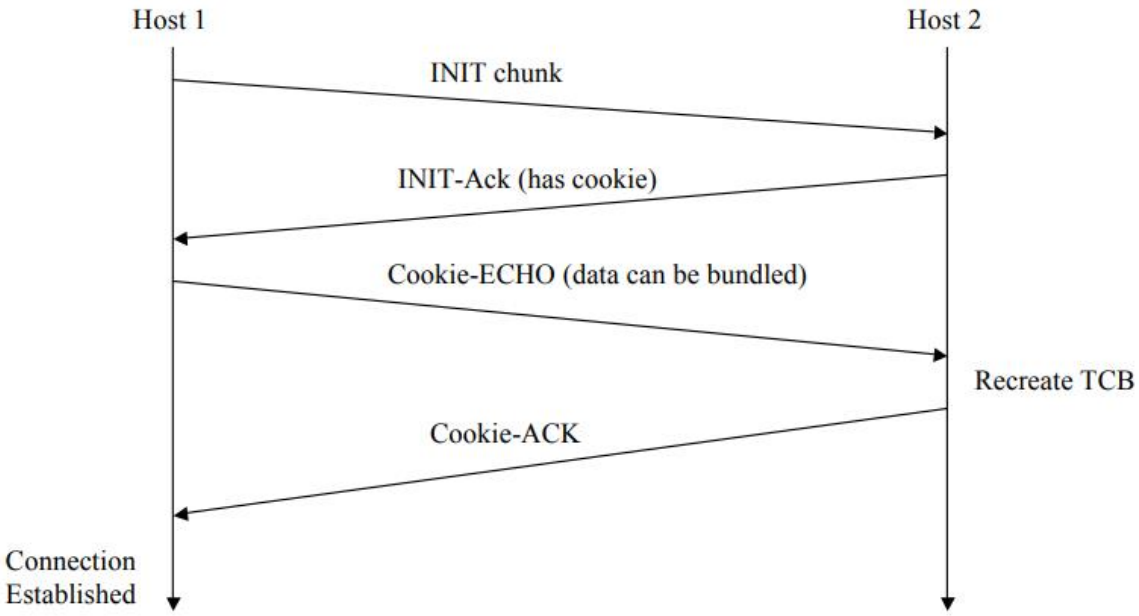
4次握手建立连接

“A” 发送 INIT 块到 “Z”，

“Z” 响应 INIT ACK 块. 其中一个Cookie (该Cookie是对时间等信息的带密钥Hash)

“A” 发送 COOKIE ECHO 块到 “Z”. 可与DATA 块绑定

“Z” 回答 COOKIE ACK 到 “A”，可与DATA 块绑定



实验1（Linux内核如何在三次握手中实现cookie机制）

4次握手建立连接

SCTP报文由两个大的部分组成：

1. The common header, which occupies the first 12 bytes and is highlighted in blue
2. The data chunks, which occupy the remaining portion of the packet.

Bits	0–7	8–15	16–23	24–31
+0	Source port		Destination port	
32	Verification tag			
64	Checksum			
96	Chunk 1 type	Chunk 1 flags	Chunk 1 length	
128	Chunk 1 data			
...	...			
...	Chunk N type	Chunk N flags	Chunk N length	
...	Chunk N data			

4次握手建立连接

SCTP中的Chunk分为几十种类型，INIT Chunk用来协商可靠连接，Payload data用来传输数据。

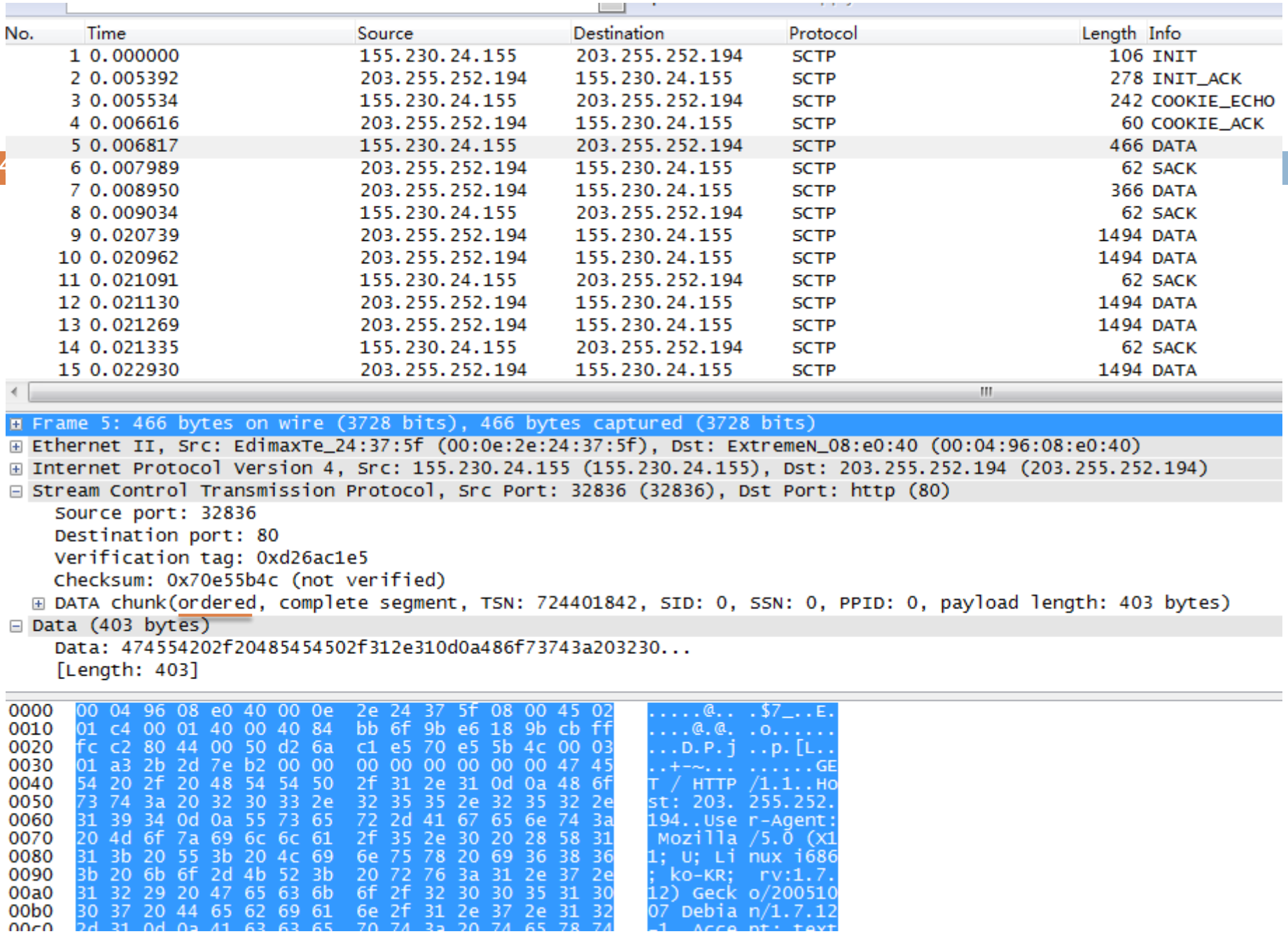
0 = Payload data
1=INIT
2=INIT-ACK
3=SACK
4=Heart Beat Request
5=Heart Beat ACK
6=Abort
7=Shut Down
8= Shut Down Ack …

+	Bits 0 - 7	8 - 11	12	13	14	15	16 - 31
0	Chunk type=0	Reserved	I	U	B	E	Chunk length
32	TSN						
64	Stream identifier						Stream sequence number
96	Payload protocol identifier						
128	Data						

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.170.8	192.168.170.56	SCTP	78	INIT
2	0.000296	192.168.170.56	192.168.170.8	SCTP	174	INIT_ACK
3	0.000783	192.168.170.8	192.168.170.56	SCTP	150	COOKIE_ECHO
4	0.001001	192.168.170.56	192.168.170.8	SCTP	50	COOKIE_ACK
5	0.002212	192.168.170.8	192.168.170.56	SCTP	1102	DATA DATA
6	0.002459	192.168.170.56	192.168.170.8	SCTP	1118	SACK DATA DATA
7	0.003116	192.168.170.8	192.168.170.56	SCTP	1102	DATA DATA
8	0.003323	192.168.170.56	192.168.170.8	SCTP	1118	SACK DATA DATA
9	0.004016	192.168.170.8	192.168.170.56	SCTP	1102	DATA DATA
10	0.007184	192.168.170.8	192.168.170.56	SCTP	1102	DATA DATA
11	0.007257	192.168.170.56	192.168.170.8	SCTP	1118	SACK DATA DATA
12	0.007656	192.168.170.8	192.168.170.56	SCTP	590	SACK DATA
13	0.007872	192.168.170.56	192.168.170.8	SCTP	1118	SACK DATA DATA
14	0.007928	192.168.170.56	192.168.170.8	SCTP	574	DATA
15	0.008871	192.168.170.8	192.168.170.56	SCTP	1102	DATA DATA

```
⊞ Frame 5: 1102 bytes on wire (8816 bits), 1102 bytes captured (8816 bits)
⊞ Ethernet II, Src: AsustekC_b1:0c:ad (00:e0:18:b1:0c:ad), Dst: 3com_45:e4:55 (00:60:08:45:e4:55)
⊞ Internet Protocol Version 4, Src: 192.168.170.8 (192.168.170.8), Dst: 192.168.170.56 (192.168.170.56)
⊟ Stream Control Transmission Protocol, Src Port: 7 (7), Dst Port: 7 (7)
    Source port: 7
    Destination port: 7
    Verification tag: 0x00000eb0
    Checksum: 0xcfbb0406 (not verified)
  ⊟ DATA chunk(unordered, complete segment, TSN: 1560164255, SID: 0, SSN: 0, PPID: 0, payload length: 512 bytes)
    ⊞ Chunk type: DATA (0)
    ⊞ Chunk flags: 0x07
      Chunk length: 528
      TSN: 1560164255
      Stream Identifier: 0x0000
      Stream sequence number: 0
      Payload protocol identifier: not specified (0)
⊞ Data (512 bytes)
⊟ Stream Control Transmission Protocol
  ⊟ DATA chunk(unordered, complete segment, TSN: 1560164256, SID: 1, SSN: 0, PPID: 0, payload length: 512 bytes)
    ⊞ Chunk type: DATA (0)
    ⊞ Chunk flags: 0x07
      Chunk length: 528
      TSN: 1560164256
```

```
0000  00 60 08 45 e4 55 00 e0  18 b1 0c ad 08 00 45 10   .`.E.U.. ......E.
0010  04 40 00 00 40 00 40 84  60 98 c0 a8 aa 08 c0 a8   .@..@.@. `.......
0020  aa 38 00 07 00 07 00 00  0e b0 cf bb 04 06 00 07   .8...... ........
0030  02 10 5c fe 37 9f 00 00  00 00 00 00 00 00 00 00   ..\.7... ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	155.230.24.155	203.255.252.194	SCTP	106	INIT
2	0.005392	203.255.252.194	155.230.24.155	SCTP	278	INIT_ACK
3	0.005534	155.230.24.155	203.255.252.194	SCTP	242	COOKIE_ECHO
4	0.006616	203.255.252.194	155.230.24.155	SCTP	60	COOKIE_ACK
5	0.006817	155.230.24.155	203.255.252.194	SCTP	466	DATA
6	0.007989	203.255.252.194	155.230.24.155	SCTP	62	SACK
7	0.008950	203.255.252.194	155.230.24.155	SCTP	366	DATA
8	0.009034	155.230.24.155	203.255.252.194	SCTP	62	SACK
9	0.020739	203.255.252.194	155.230.24.155	SCTP	1494	DATA
10	0.020962	203.255.252.194	155.230.24.155	SCTP	1494	DATA
11	0.021091	155.230.24.155	203.255.252.194	SCTP	62	SACK
12	0.021130	203.255.252.194	155.230.24.155	SCTP	1494	DATA
13	0.021269	203.255.252.194	155.230.24.155	SCTP	1494	DATA
14	0.021335	155.230.24.155	203.255.252.194	SCTP	62	SACK
15	0.022930	203.255.252.194	155.230.24.155	SCTP	1494	DATA

```
⊞ Frame 5: 466 bytes on wire (3728 bits), 466 bytes captured (3728 bits)
⊞ Ethernet II, Src: EdimaxTe_24:37:5f (00:0e:2e:24:37:5f), Dst: ExtremeN_08:e0:40 (00:04:96:08:e0:40)
⊞ Internet Protocol Version 4, Src: 155.230.24.155 (155.230.24.155), Dst: 203.255.252.194 (203.255.252.194)
⊟ Stream Control Transmission Protocol, Src Port: 32836 (32836), Dst Port: http (80)
    Source port: 32836
    Destination port: 80
    Verification tag: 0xd26ac1e5
    Checksum: 0x70e55b4c (not verified)
  ⊞ DATA chunk(ordered, complete segment, TSN: 724401842, SID: 0, SSN: 0, PPID: 0, payload length: 403 bytes)
⊟ Data (403 bytes)
    Data: 474554202f20485454502f312e310d0a486f73743a203230...
    [Length: 403]
```

```
0000  00 04 96 08 e0 40 00 0e  2e 24 37 5f 08 00 45 02   .....@.. .$7_..E.
0010  01 c4 00 01 40 00 40 84  bb 6f 9b e6 18 9b cb ff   ....@.@. .o......
0020  fc c2 80 44 00 50 d2 6a  c1 e5 70 e5 5b 4c 00 03   ...D.P.j ..p.[L..
0030  01 a3 2b 2d 7e b2 00 00  00 00 00 00 00 00 47 45   ..+-~... ......GE
0040  54 20 2f 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   T / HTTP /1.1..Ho
0050  73 74 3a 20 32 30 33 2e  32 35 35 2e 32 35 32 2e   st: 203. 255.252.
0060  31 39 34 0d 0a 55 73 65  72 2d 41 67 65 6e 74 3a   194..Use r-Agent:
0070  20 4d 6f 7a 69 6c 6c 61  2f 35 2e 30 20 28 58 31    Mozilla /5.0 (X1
0080  31 3b 20 55 3b 20 4c 69  6e 75 78 20 69 36 38 36   1; U; Li nux i686
0090  3b 20 6b 6f 2d 4b 52 3b  20 72 76 3a 31 2e 37 2e   ; ko-KR;  rv:1.7.
00a0  31 32 29 20 47 65 63 6b  6f 2f 32 30 30 35 31 30   12) Geck o/200510
00b0  30 37 20 44 65 62 69 61  6e 2f 31 2e 37 2e 31 32   07 Debia n/1.7.12
00c0  2d 31 0d 0a 41 63 63 65  70 74 3a 20 74 65 78 74   -1..Acce pt: text
```

SCTP课后练习

1. 安装WireShark
2. 打开sctp.cap观察sctp心跳报文
3. 打开sctp-www.cap观察sctp传输http报文