第二章 网络协议

协 议

0	1	4	H	E	L	L	O	S	T	U	D	E	N	T	S	X	X	X

什么是协议

- 协议是网络中计算机或设备之间进行通信的
- 一系列规则的集合。

协议示例

- 以发送消息“HELLO STUDENTS”为例。

常用协议

- IP、TCP、HTTP、POP3、SMTP

协议

计算机网络进行计算与计算之间的通信，在他们之间必须首先决定通信的“约束规则”，称为协议。即使不同的制造商生产的商品，只要使用相同的协议，它们之间就能够互相通信，计算机与计算机之间必须使用同一个协议，必须能够进行该协议所规定的处理。

协议栈

什么是协议栈

- 在网络中，为了完成通信，必须使用多层上的多种协议。这些协议按照层次顺序组合在一起，构成了协议栈(Protocol Stack)，也称为协议族(Protocol Suite)。

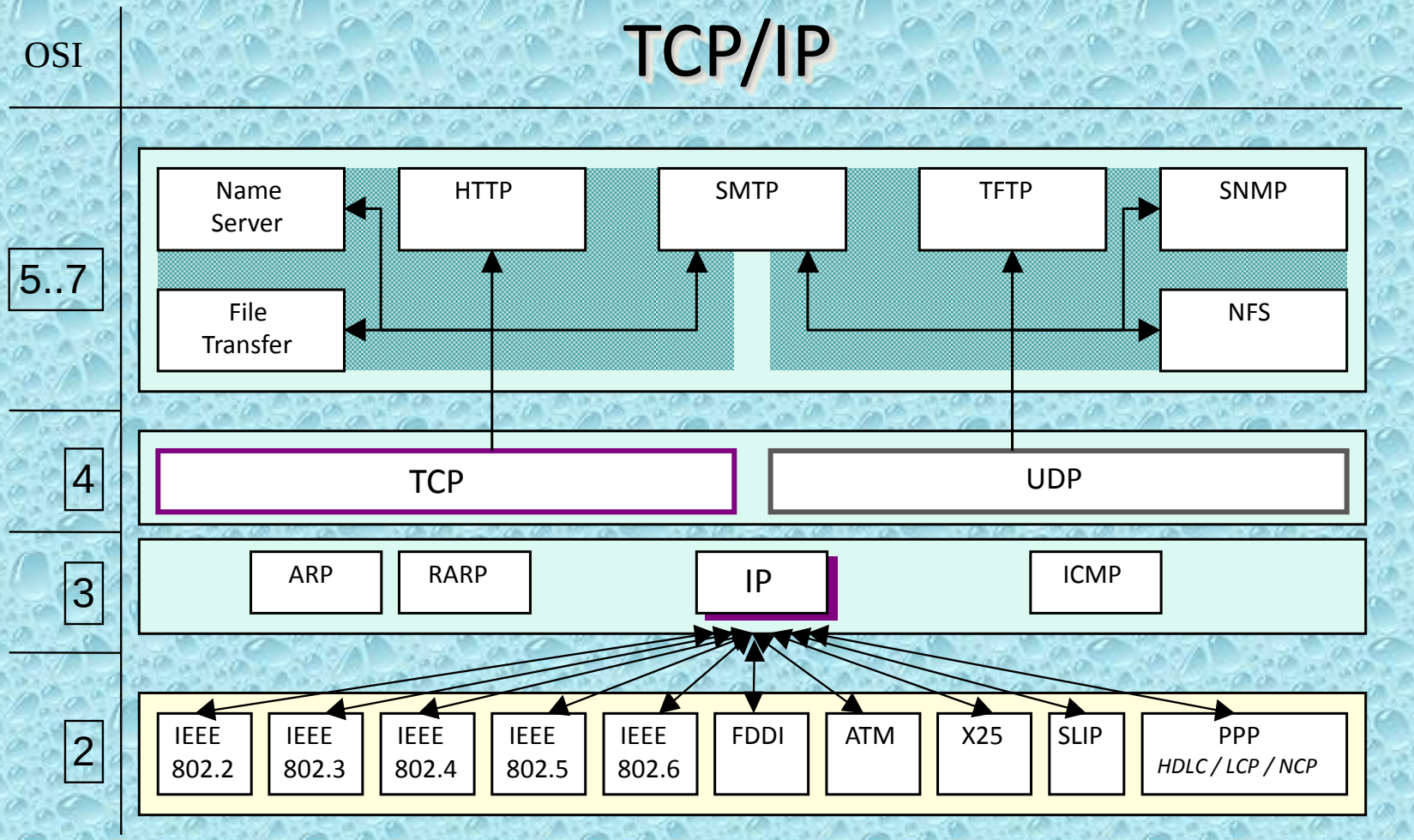
协议的作用及常见协议

- 协议的作用
- 一个网络协议的作用主要有两个：一是建立对等层之间的虚拟通信，二是实现层次之间的无关性。
- 层次间的无关性
- 所谓层次间无关性，就是指较高层次和相邻的相低层次进行通信时，只是利用较低层次提供的接口和服务，而不需了解低层实现该功能所采用的算法和协议的细节；较低层次也仅是使用从高层系统传送来的参数和控制信息，这就是层次间的无关性。

常见协议栈

- TCP/IP：工业标准、开放式协议，Internet网络的标准
- IPX/SPX：Novell开发的Netware操作系统使用的协议，IPX为网际数据包交换协议，工作在网络层，SPX为序列数据包交换协议，工作在传输层
- NetBIOS/NetBEUI：较小的协议栈，应用于IBM和早期的Windows系统，现在Windows仍然支持
- AppleTalk：Apple公司的Mac OS中所采用的网络协议

TCP/IP协议栈

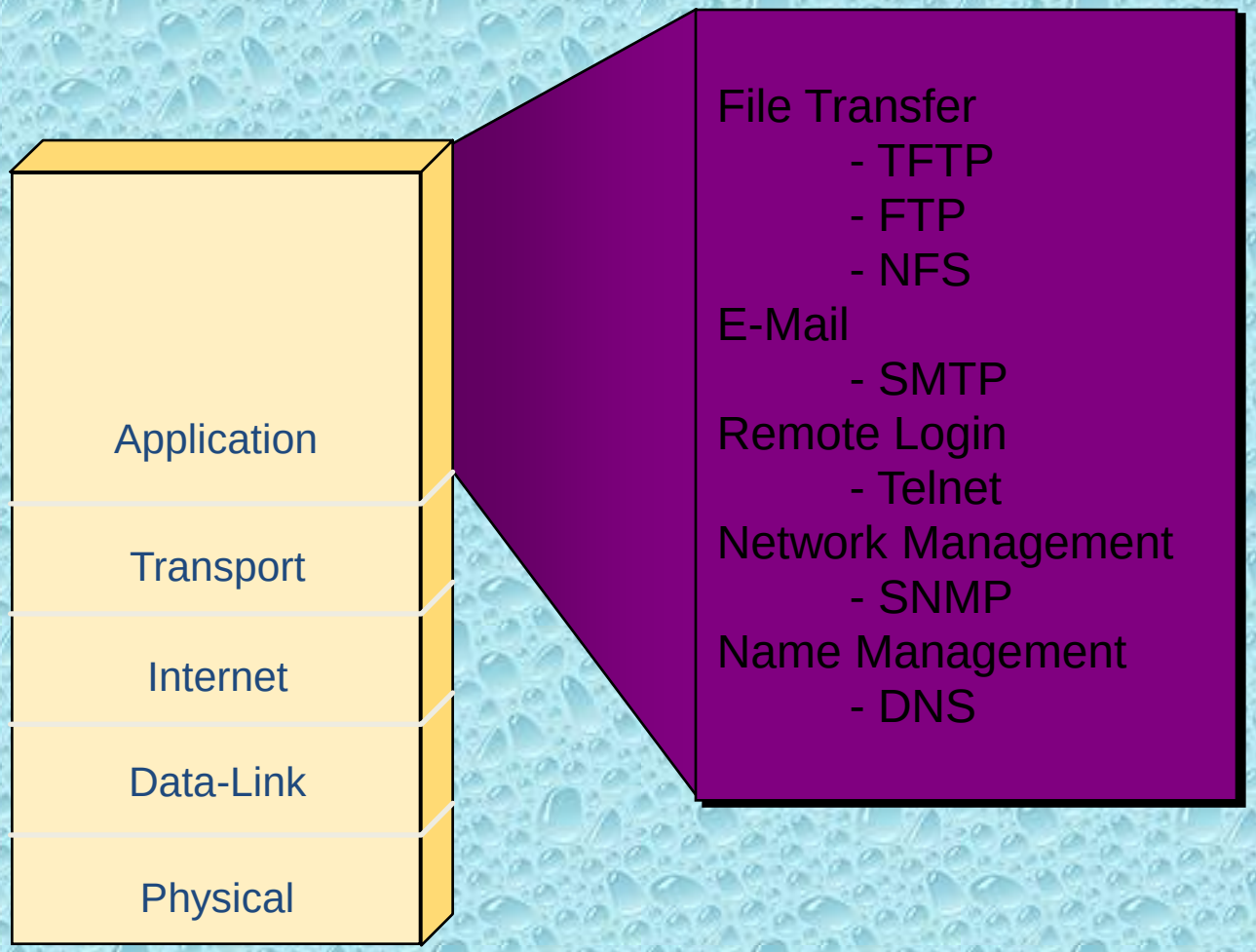

OSI
TCP/IP
5..7
4
3
2
Name Server
HTTP
SMTP
TFTP
SNMP
File Transfer
NFS
TCP
UDP
ARP
RARP
IP
ICMP
IEEE 802.2
IEEE 802.3
IEEE 802.4
IEEE 802.5
IEEE 802.6
FDDI
ATM
X25
SLIP
PPP
HDLC / LCP / NCP

TCP/IP各层协议简介

- 应用层

  应用层是TCP/IP协议组的顶层，所有的应用程序包含在这一层中；它们包括：HTTP，FTP，Telnet，SMTP，SNMP，DNS等。

应用层


Application
Transport
Internet
Data-Link
Physical
File Transfer
- TFTP
- FTP
- NFS
E-Mail
- SMTP
Remote Login
- Telnet
Network Management
- SNMP
Name Management
- DNS

Telnet

- Telnet是TCP/IP中的一种应用协议，可以为终端仿真提供支持。
- Telnet可使用户连接到主机上，使主机响应起来就像它直接连接在终端上一样。
- Telnet在发送端和接收端使用TCP的23号端口以进行专用的通信。

文件传输协议（File Transfer Protocol，FTP）

- FTP协议使用TCP20号和21号端口
  - 20号端口用于数据交换
  - 21号端口用于建立连接
  - 允许目录和文件访问，上传下载，不能远程执行文件

- 简单文件传输协议（Trivial File Transfer Protocol，TFTP）
  - TFTP是无连接的，使用UDP的69号端口
  - 用于当数据传输错误无关紧要而且无须安全性时的小型文件的传输

SMTP

- 简单邮件传输协议（Simple Mail Transfer Protocol，SMTP）是为网络系统间的电子邮件交换而设计的。使用 25 端口 。

- SMTP只需要在接收端的一个电子邮件地址即可发送邮件.
- POP3 协议用来接收邮件.使用110端口

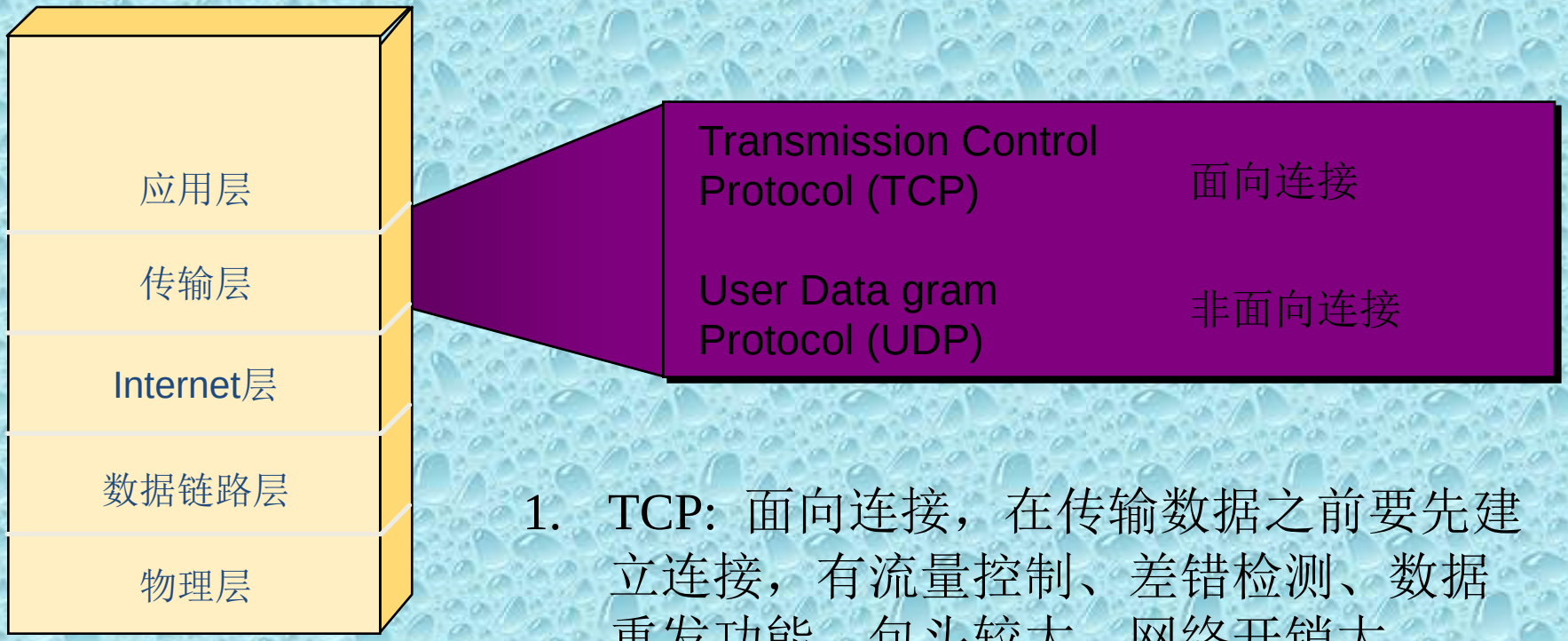
DNS

- 域名服务（Domain Name Service, DNS）
- 将域名转换为IP地址，或将IP地址转换为域名，用于解析完全合格域名（FQDN），例如 www.163.com。
- 使用53号端口

DHCP

- 动态主机配置协议(DHCP)服务器可以提供的信息有:
  1.IP地址
  2.子网掩码(subnet mask)
  3.域名(domain name)
  4.默认网关(default gateway)
  5.DNS

传输层概述



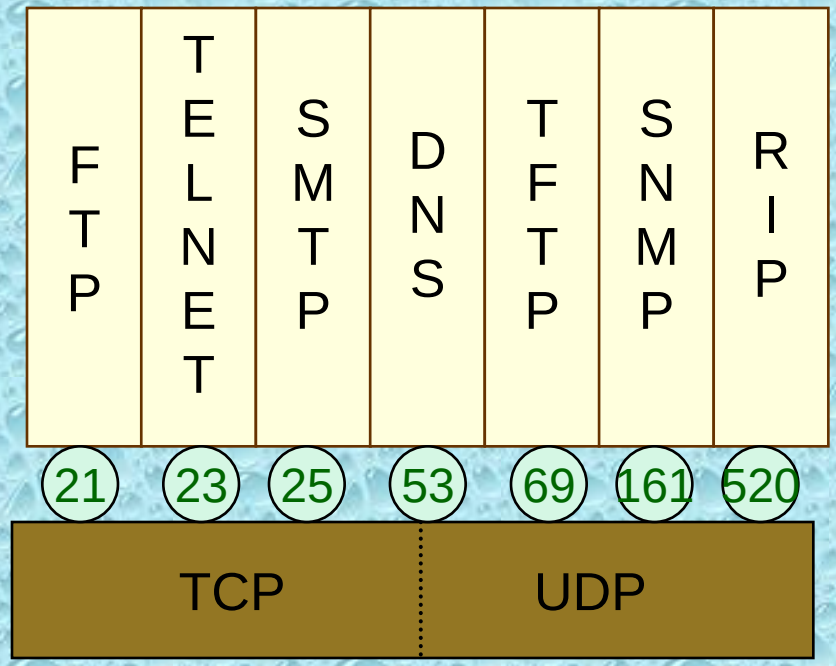
1. TCP：面向连接，在传输数据之前要先建立连接，有流量控制、差错检测、数据重发功能。包头较大，网络开销大。
2. UDP：无连接，直接发送数据，不进行流量控制，没有差错检测和数据重传功能。包头小，网络开销较小。

连接服务的类型

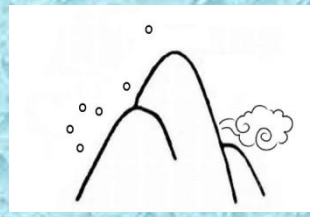
- 面向连接的服务(TCP)
  - 源端与目的端在通信前要先建立连接，然后在此连接上互相传输数据帧，每一个帧都被编号，数据链路层保证传送的帧被对方收到，且只收到一次，双方通信完毕后拆除连接。
- 无确认、无连接的服务(UDP)
  - 源端不需要建立连接就向目的端发送独立的数据帧，而目的端也不需要对收到的帧进行确认.

端口概述

- 端口号的范围：1-1023　　>1023
- 在TCP/IP协议的通信中，端口号是为了识别应用程序和各种服务而使用的号码，它包含在TCP协议和UDP协议的报头中.



端口



- 端口：由于网络通信所用的协议较多，且一台主机可能同时提供多种服务，为了标识和区分这些协议，引入了端口的概念，即每个协议都对应着一个端口，用端口号予以标识，因此每种服务也都有其自己的工作端口。若一台服务器提供多种服务，客户端可根据端口号访问到它所需要的服务。
- 端口号由16位二进制数组成，范围：1~65535

端口分类

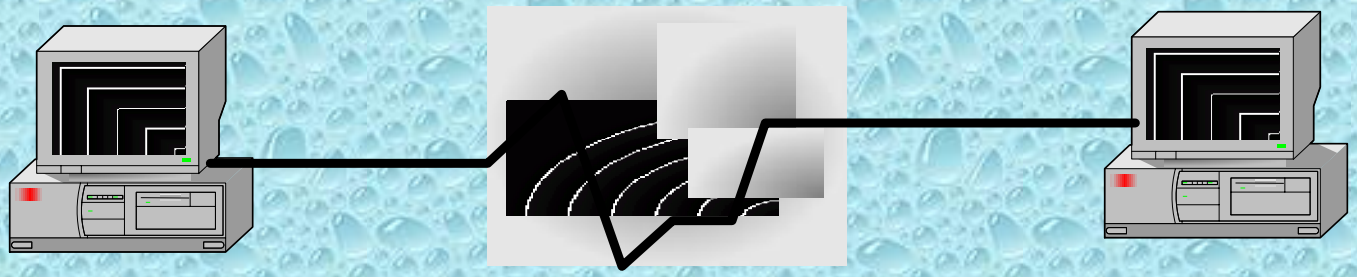
- 知名端口：网络协议默认使用的端口，用户不可随意使用。

  范围：1~1023

- 动态端口：计算机根据需要随机打开的端口，使用完毕即关闭。

  范围：1024~65535

TCP连接过程



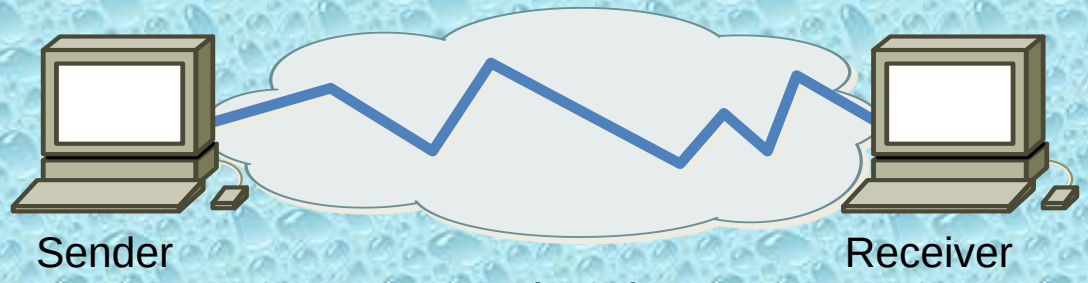
请求

同意&请求

确认

数据传送

TCP连接建立过程与“三次握手”



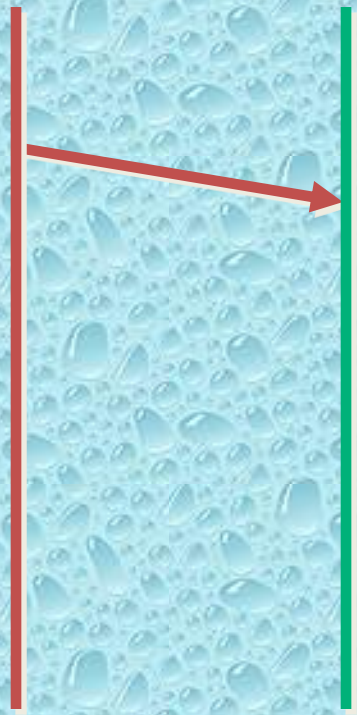
TCP 三次握手

Host A

Host B

1

发送 SYN
(seq=100 ctl=SYN)

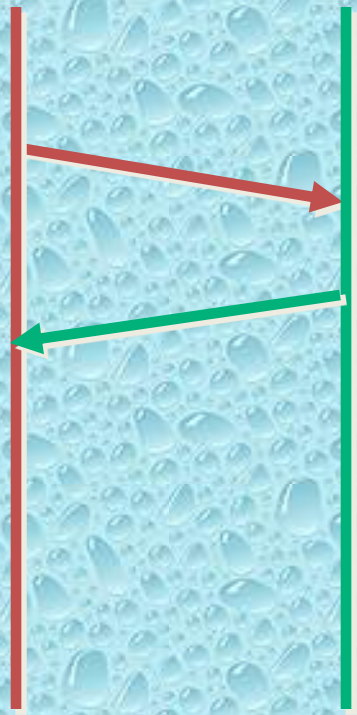


接收 SYN

TCP 三次握手

1

发送 SYN
(seq=100 ctl=SYN)



接收 SYN

发送 SYN, ACK
(seq=300 ack=101 ctl=syn,ack)

2

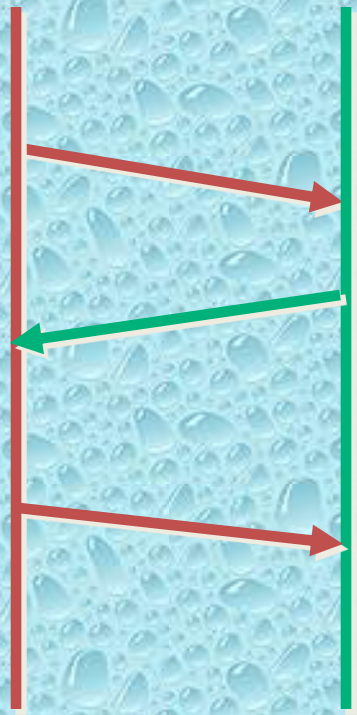
接收 SYN,ACK

TCP 三次握手

Host A

Host B

1 发送 SYN
(seq=100 ctl=SYN)



接收 SYN

2 发送 SYN, ACK
(seq=300 ack=101 ctl=syn,ack)

接收 SYN,ACK

3 建立会话发送 ACK
(seq=101 ack=301 ctl=ack)

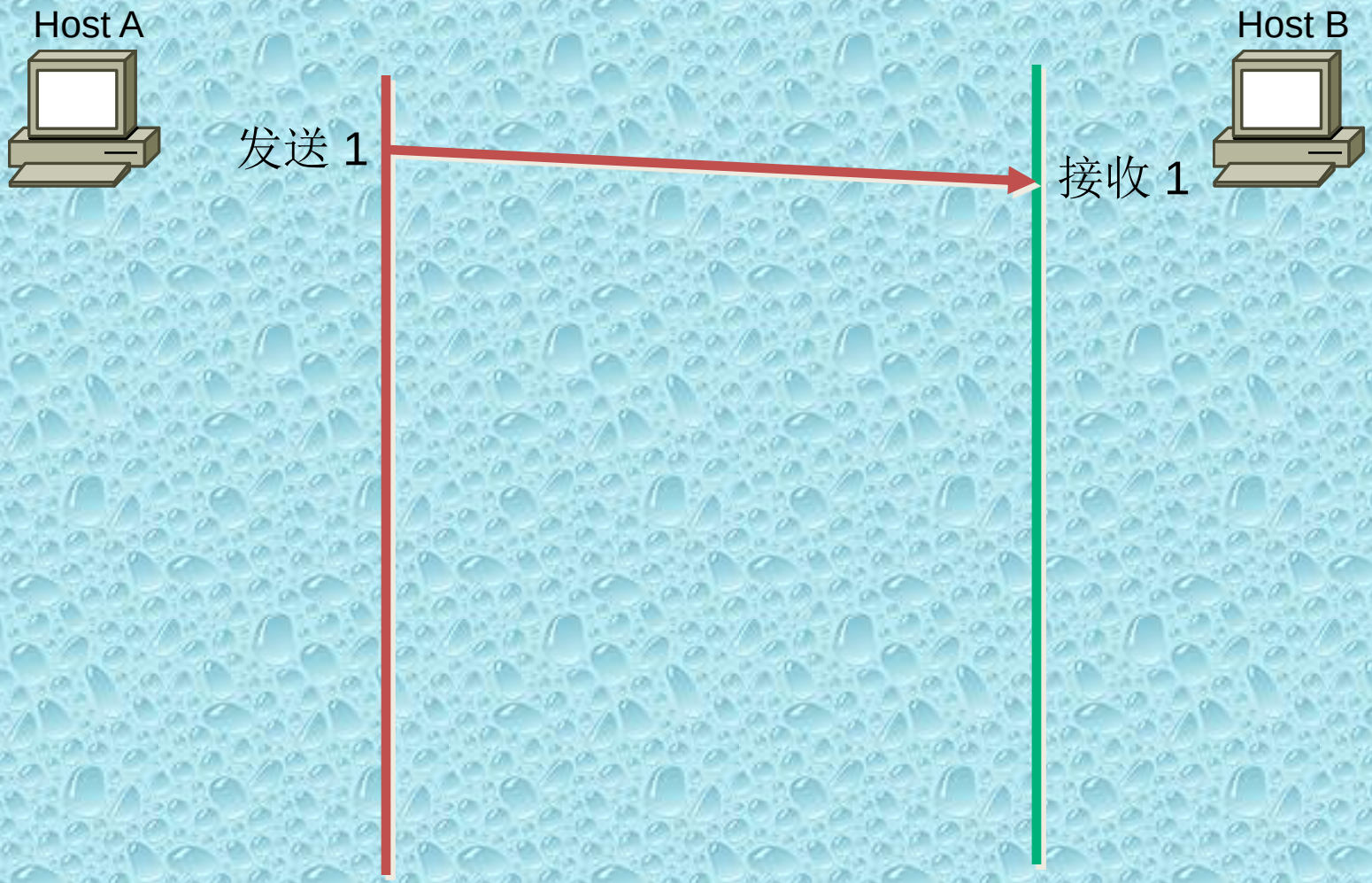
TCP 简单确认

Host A

Host B

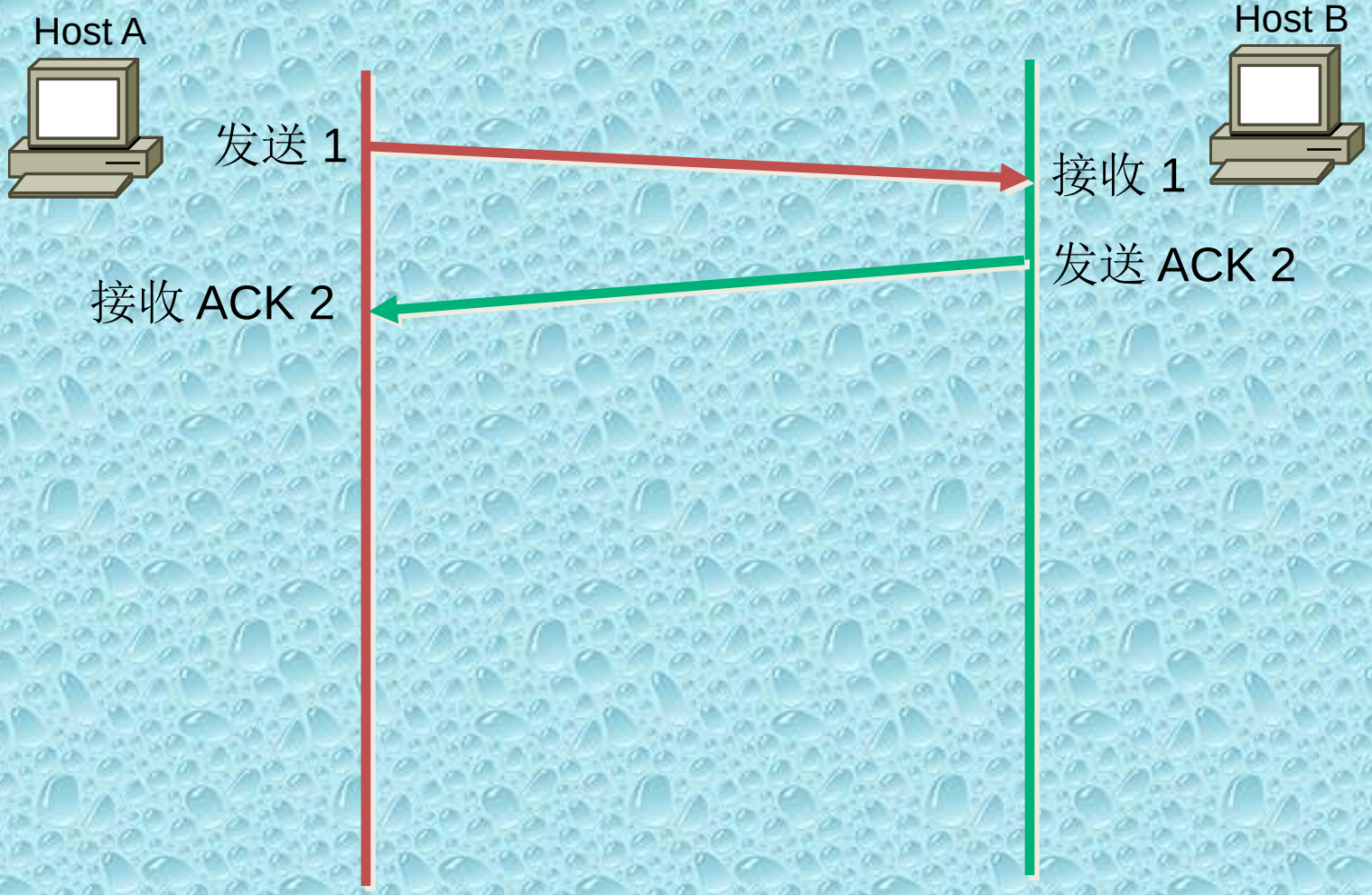
滑动窗口 = 1

TCP 简单确认



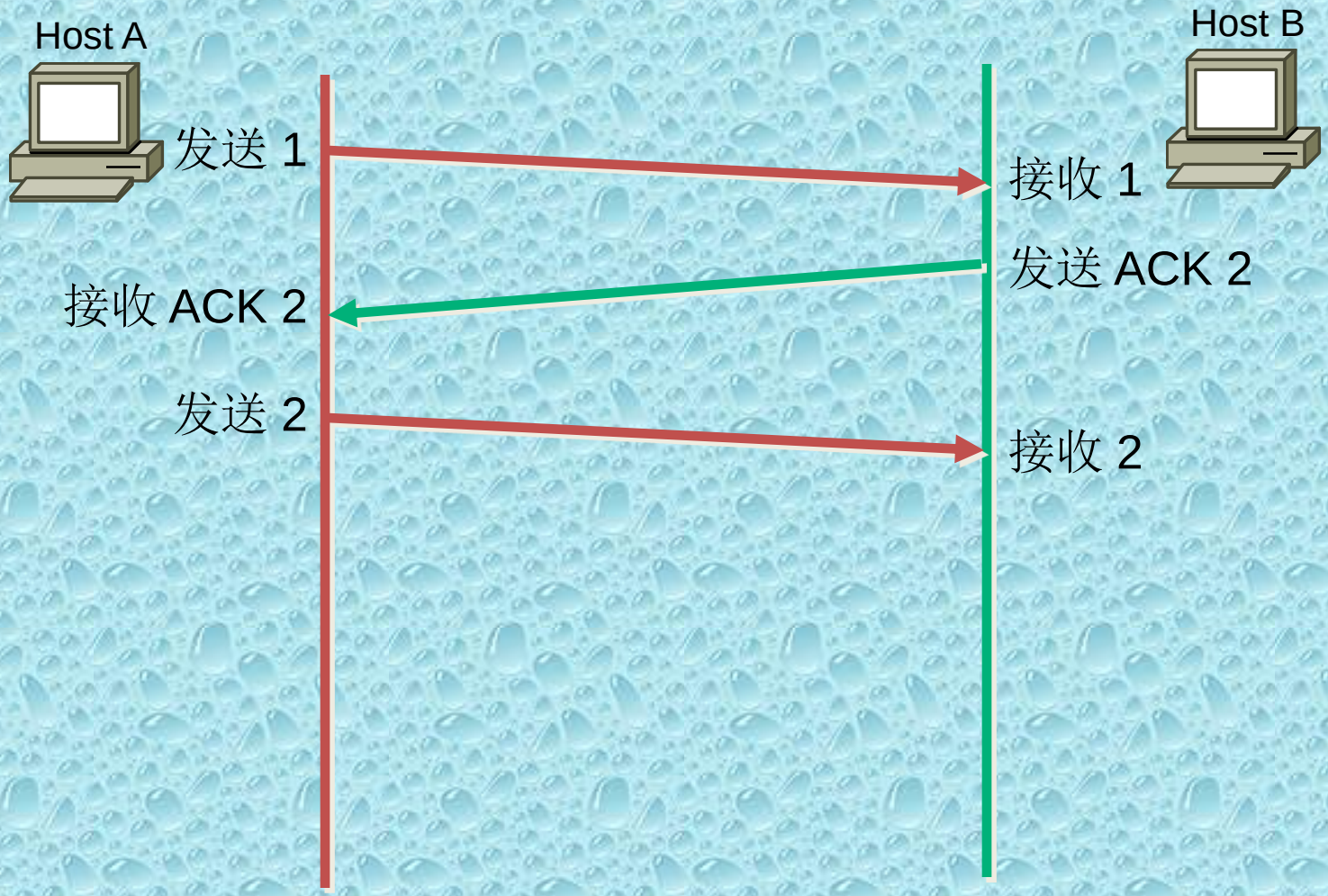
滑动窗口 = 1

TCP 简单确认



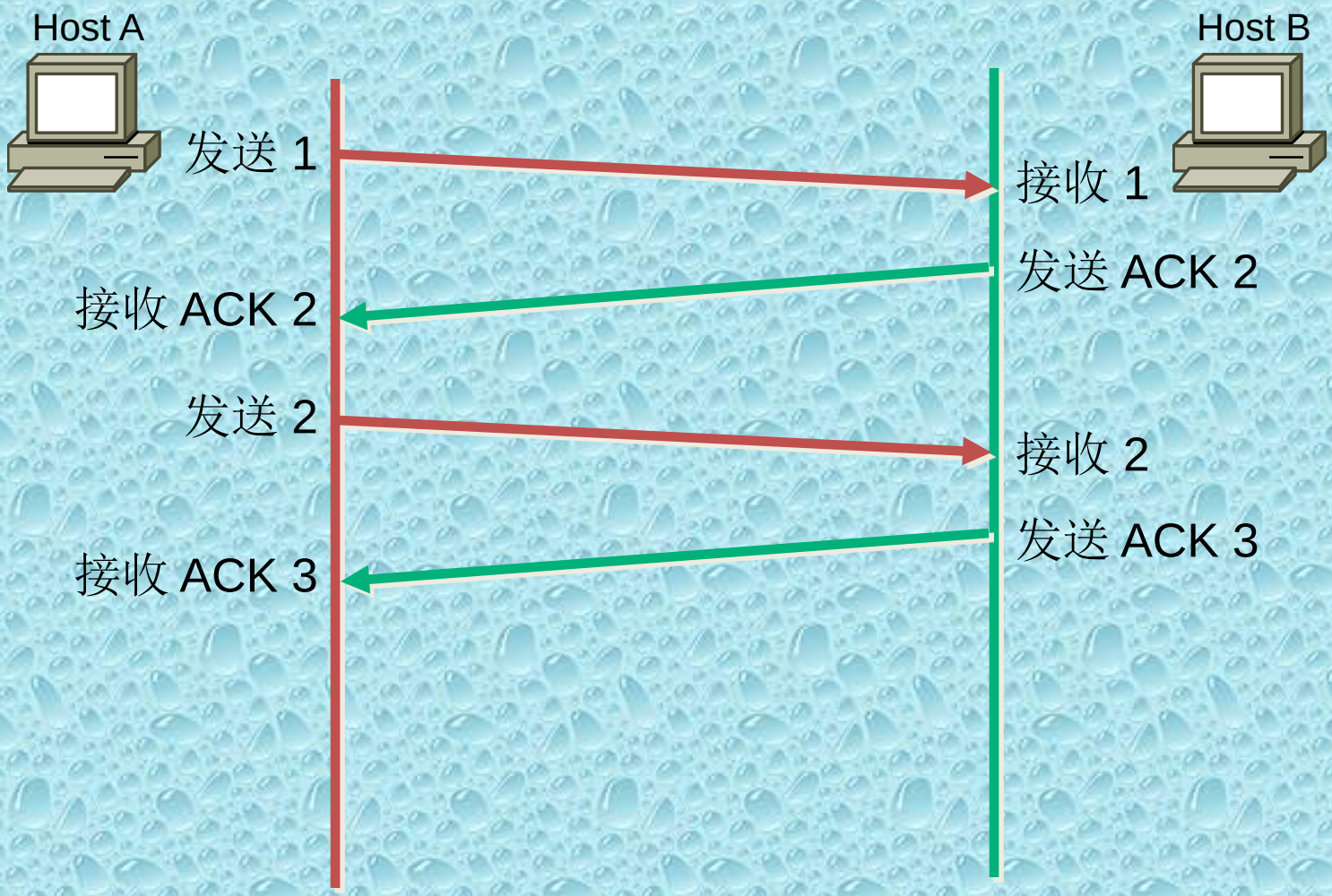
滑动窗口 = 1

TCP 简单确认



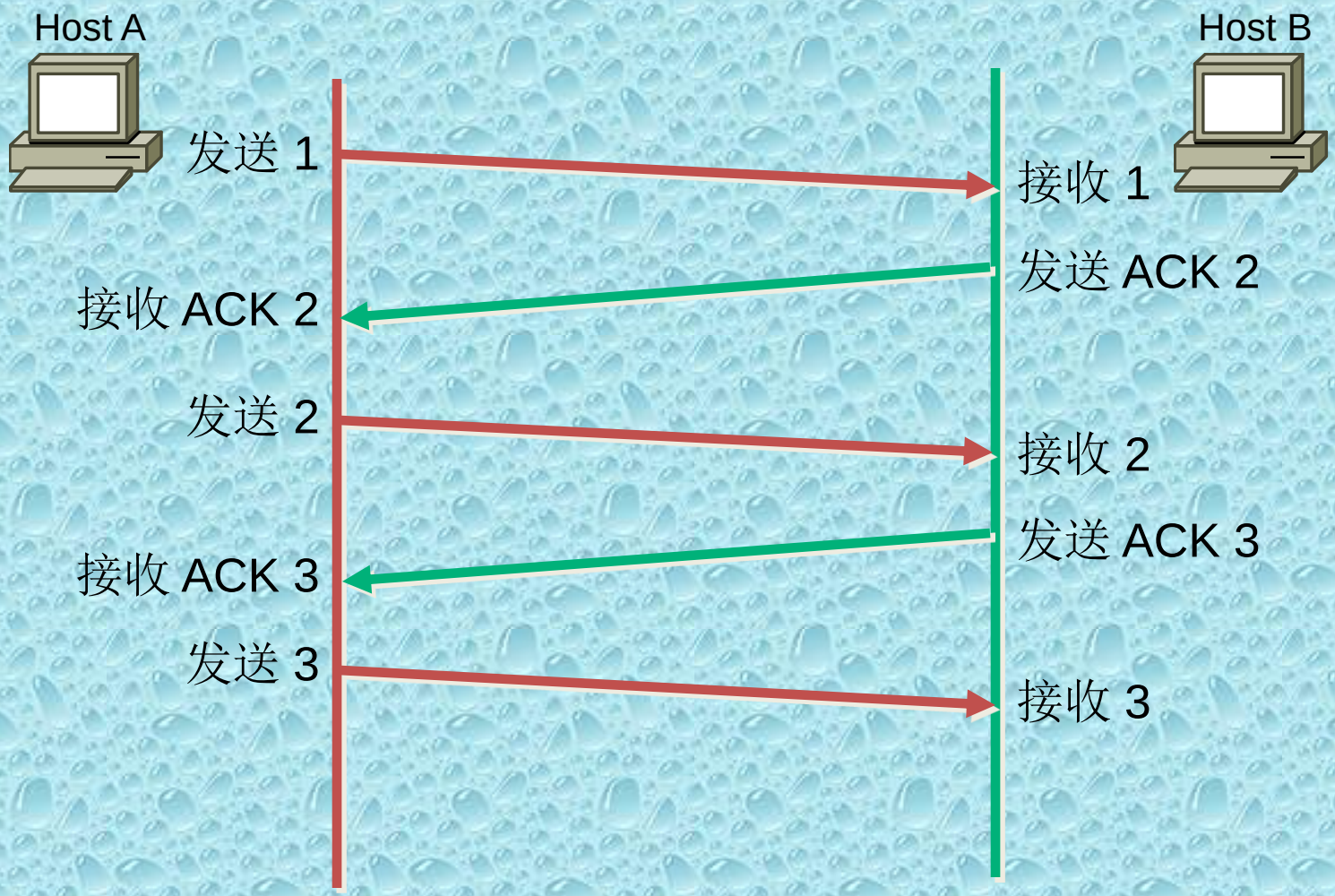
滑动窗口 = 1

TCP 简单确认



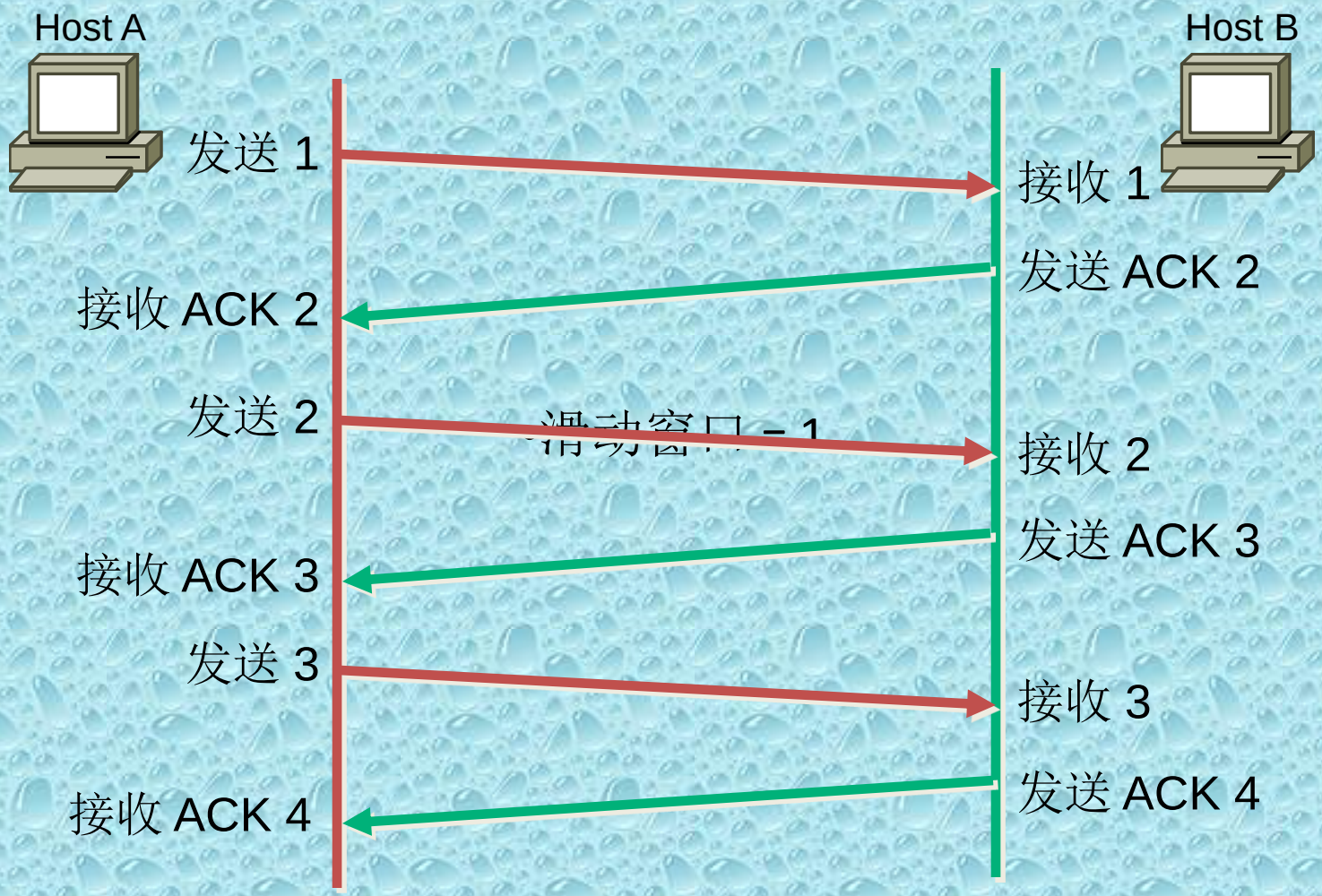
滑动窗口 = 1

TCP 简单确认

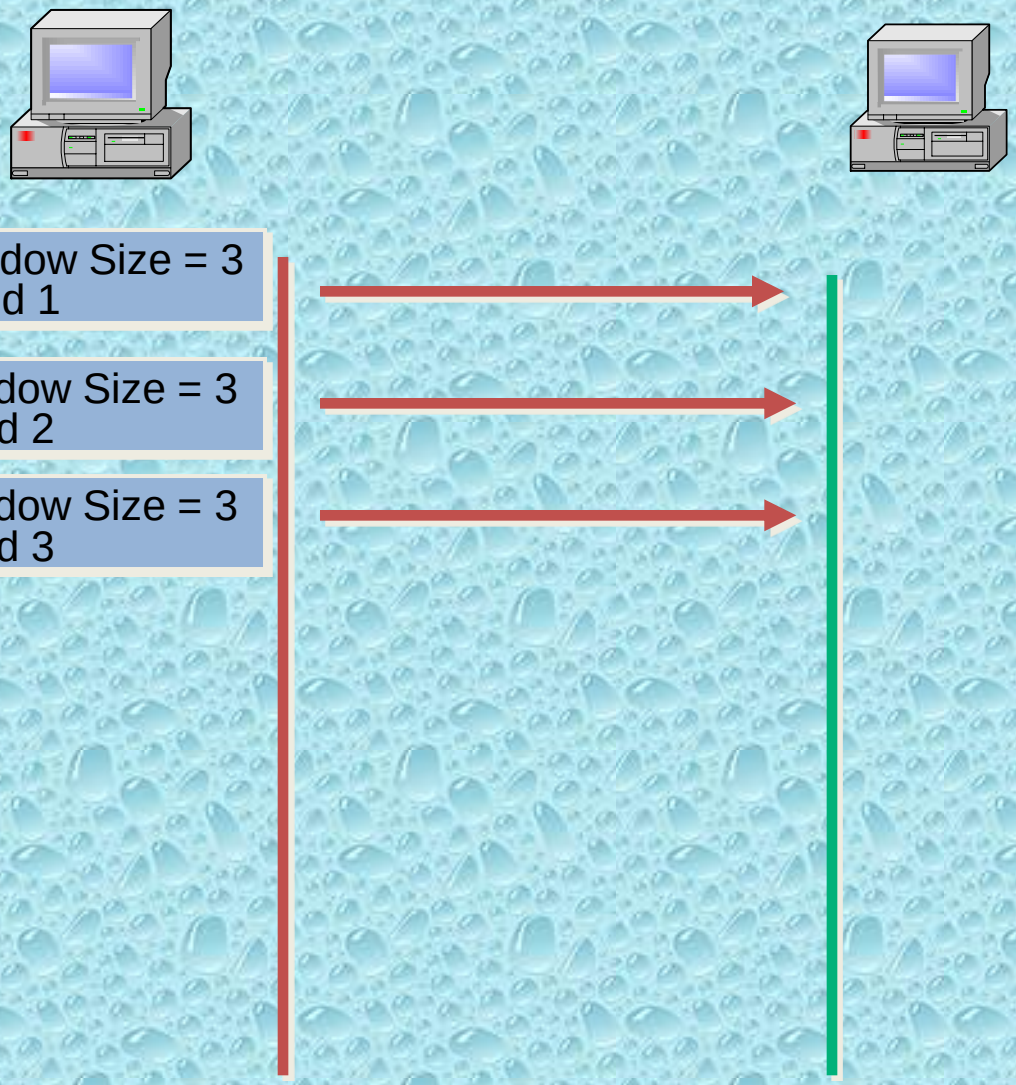


滑动窗口 = 1

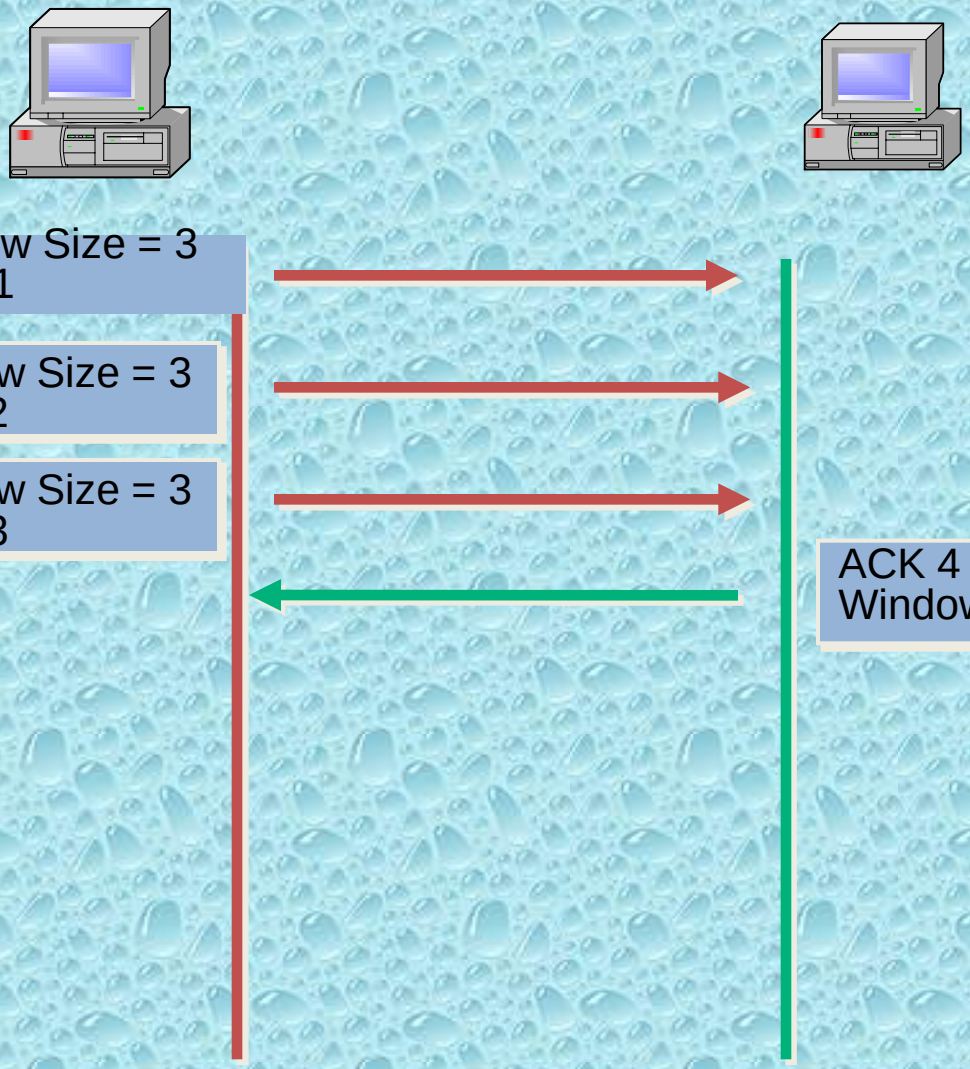
TCP 简单确认


Host A
Host B
发送 1
接收 1
发送 ACK 2
接收 ACK 2
发送 2
接收 2
发送 ACK 3
接收 ACK 3
发送 3
接收 3
发送 ACK 4
接收 ACK 4

TCP 窗口



TCP 流量控制


Window Size = 3
Send 1
Window Size = 3
Send 2
Window Size = 3
Send 3
ACK 4
Window Size = 3

TCP 窗口

Window Size = 3
Send 1

Window Size = 3

Window Size = 3
Send 2

Window Size = 3
Send 3

ACK 4
Window Size = 3

Window Size = 3
Send 4

Window Size = 3
Send 5

Window Size = 3
Send 6

ACK 7
Window Size = 3

TCP 流量控制

发送方

Window size = 3
Send 1

Window size = 3
Send 2

Window size = 3
Send 3

接收方

TCP 流量控制

发送方

接收方

Window size = 3
Send 1

Window size = 3
Send 2

Window size = 3
Send 3

ACK 3
Window size = 2

数据 3 被丢弃

TCP 流量控制

发送方

接收方

Window size = 3
Send 1

Window size = 3
Send 2

Window size = 3
Send 3

ACK 3
Window size = 2

数据 3 被丢弃

Window size = 2
Send 3

Window size = 2
Send 4

TCP 流量控制

发送方

接收方

Window size = 3
Send 1

Window size = 3
Send 2

Window size = 3
Send 3

ACK 3
Window size = 2

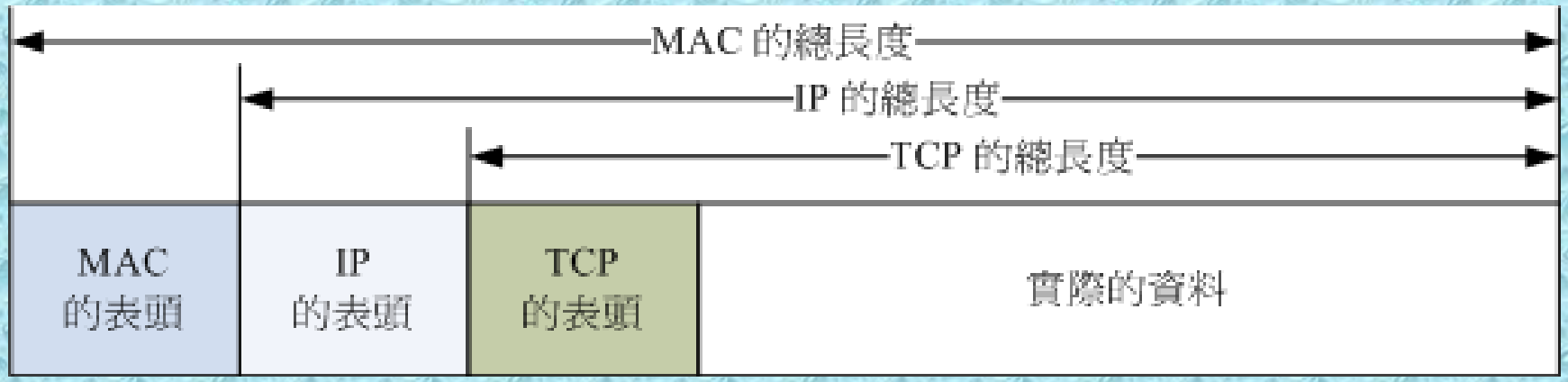
数据 3 被丢弃

Window size = 3
Send 3

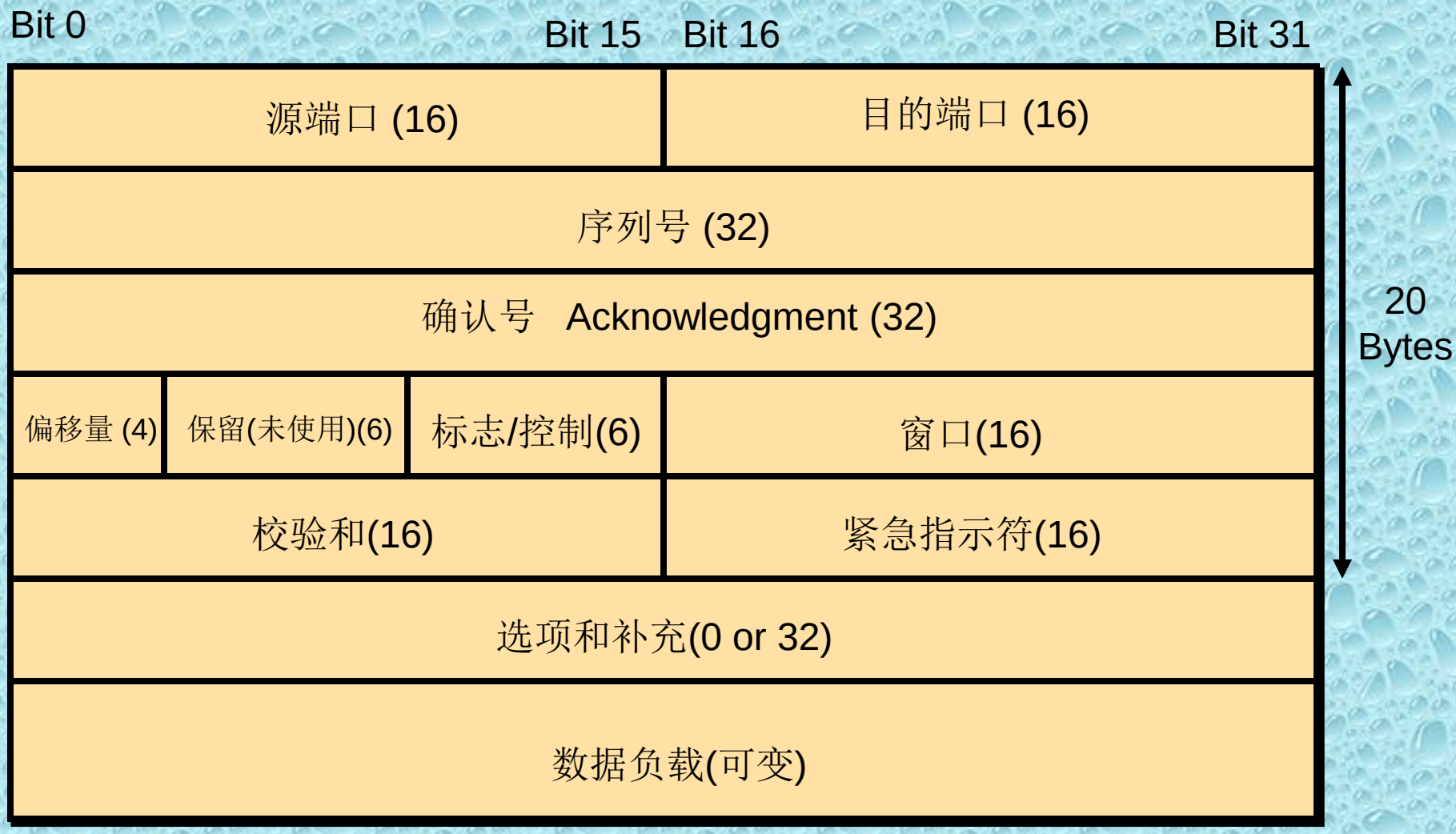
Window size = 3
Send 4

ACK 5
Window size = 2

数据结构



TCP段格式


Bit 0
Bit 15
Bit 16
Bit 31
源端口 (16)
目的端口 (16)
序列号 (32)
确认号 Acknowledgment (32)
偏移量 (4)
保留(未使用)(6)
标志/控制(6)
窗口(16)
校验和(16)
紧急指示符(16)
选项和补充(0 or 32)
数据负载(可变)
20 Bytes

TCP 段格式

- 源端口(Source Port)：呼叫端口号
- 目的端口(Destination Port)：被叫端口号
- 序号(Sequence Number)：标记数据段的顺序
- 确认号(Acknowledgment Number)：下一个段的序号
- 报头长度(HLEN)：报头的字节数,又称偏移量
- 保留域(Reserved)：为0
- 编码位(Code Bits)：控制功能(会话的建立和终止)
- 窗口(Window)：发送的字节数
- 校验和(Checksum)：报头和数据字段的校验和
- 紧急指针(Urgent Pointer)：紧急数据的末尾
- 选项(Option)：当前定义项，TCP段的最大值
- 数据(Data)：上层协议的数据

User Datagram Protocol (UDP）

- UDP协议的是无连接(connectionless),即不可靠,因为它不与对方进行协商并连接,它也不会给数据段标号,也不关心数据段是否到达接受方。

UDP 段结构

Source port (16)	Destination port (16)
Length (16)	Checksum (16)
Data (if any)	

UDP协议的用途和特征

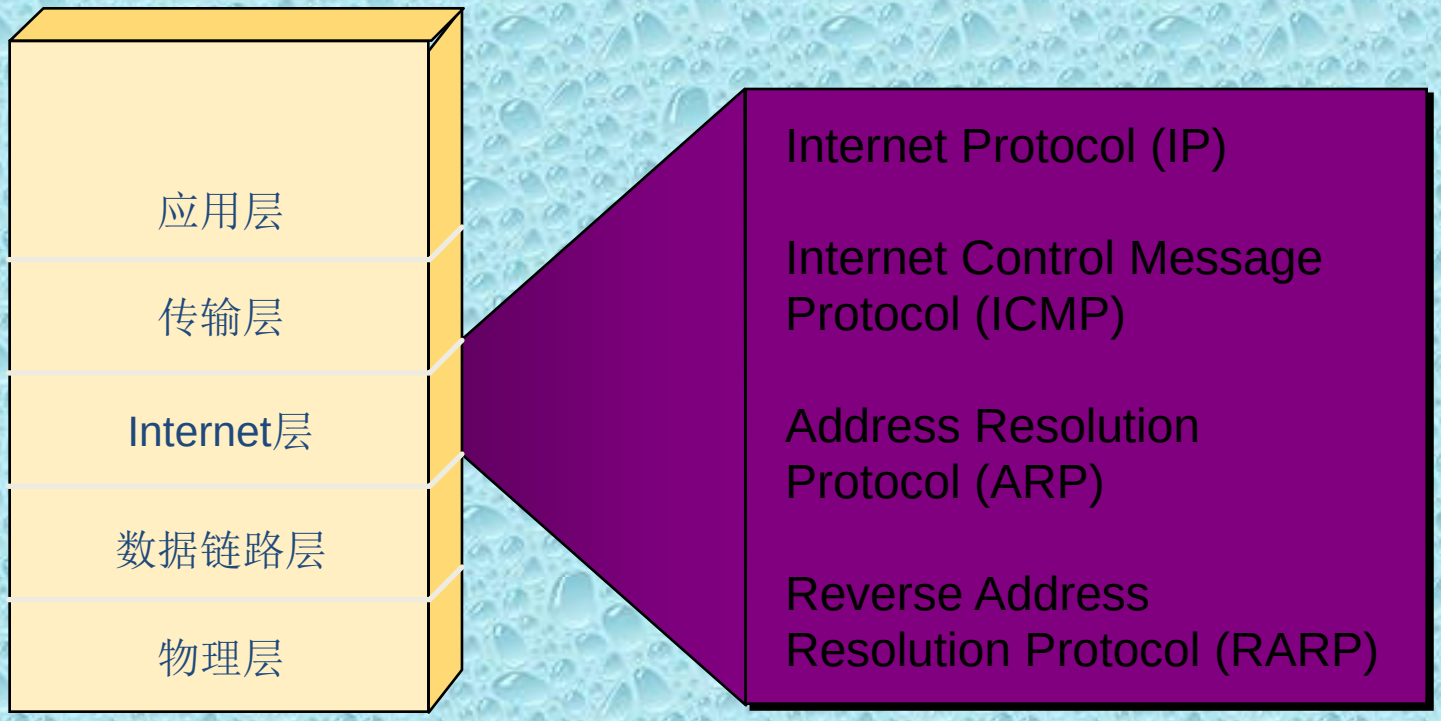
- UDP协议只是使用IP协议提供了无连接的通信服务，所以
- 无论何时都能够发送数据。而且，由于它处理比较简单，
- 所以能够进行高速的处理。UDP协议适合以下几方面的应
- 用：
- 总包数比较小的通信；
- 动画和声音的多媒体通信；

- 没有顺序号和确认号
- 由上层应用（应用层程序）来保证传输的可靠性

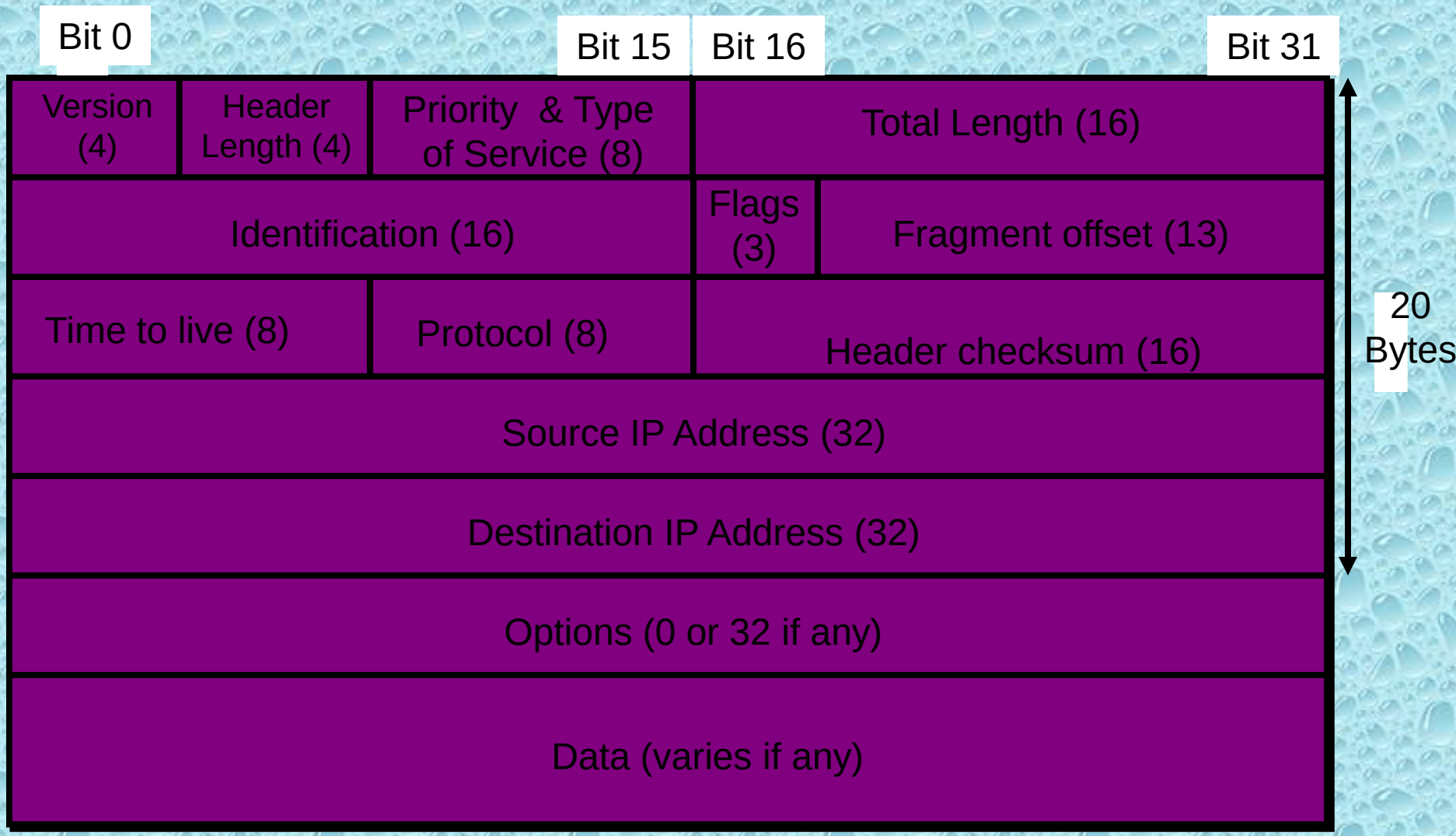
TCP 与 UDP 比较

- 1.TCP协议在传送数据段的时候要给段标号;UDP协议没有
  2.TCP协议可靠；UDP协议不可靠
  3.TCP协议是面向连接； UDP协议采用无连接
  4.TCP协议负载较高； UDP协议低负载
  5.TCP协议的发送方要确认接受方是否收到数据段;UDP反之
  6.TCP协议采用窗口技术和流量控制;UDP协议没有

Internet 层概述



IP 数据


Bit 0
Bit 15
Bit 16
Bit 31
Version (4)
Header Length (4)
Priority & Type of Service (8)
Total Length (16)
Identification (16)
Flags (3)
Fragment offset (13)
Time to live (8)
Protocol (8)
Header checksum (16)
Source IP Address (32)
Destination IP Address (32)
Options (0 or 32 if any)
Data (varies if any)
20 Bytes

IP包说明

- Version
  - 版本 (VER)。表示的是 IP 版本，目前的 IP 规格多为版本 4 (version 4)，所以这里的数值通常为 0x4 (注意：封包使用的数字通常都是十六进制的)。
- Internet Header Length
  - 表头长度 (IHL)。我们从IP包规格中看到前面的 6 行为header ，如果Options沒有设定的話，也就只有5行的长度; 我们知道每行有32bit 也就是4byte;那么，5列就是20byte了。

Type of Service

服务类型(TOS)。这里指的是 IP 封包在传送过程中要求的服务类型，其中一共由 8 个 bit 組成，每组 bit 组合分别代表不同的意思:

000.....	Routine	設定 IP 順序，預設為 0 ，否則，數值越高越優先
...0....	Delay	延遲要求， 0 是正常值， 1 為低要求
....0...	Throughput	通訊量要求， 0 為正常值， 1 為高要求
.....0..	Reliability	可靠性要求，0 為正常值， 1 為高要求
......00	Not Used	未使用

IP包说明

- Total Length
  - 封包总长 (TL),包括表头和数据的总和.
- Identification
  - 识别码 (ID)。每一个IP封包都有一個 16bit 的唯一识别码。当程序产生的数据通过网络传送时，都会在网络层拆散成封包形式发送，当封包进行重组时，这个ID就是依据。
- Flag
  - 标志 (FL)。这是当封包在传送过程中进行最佳组合时使用的三个bit的识别记号。

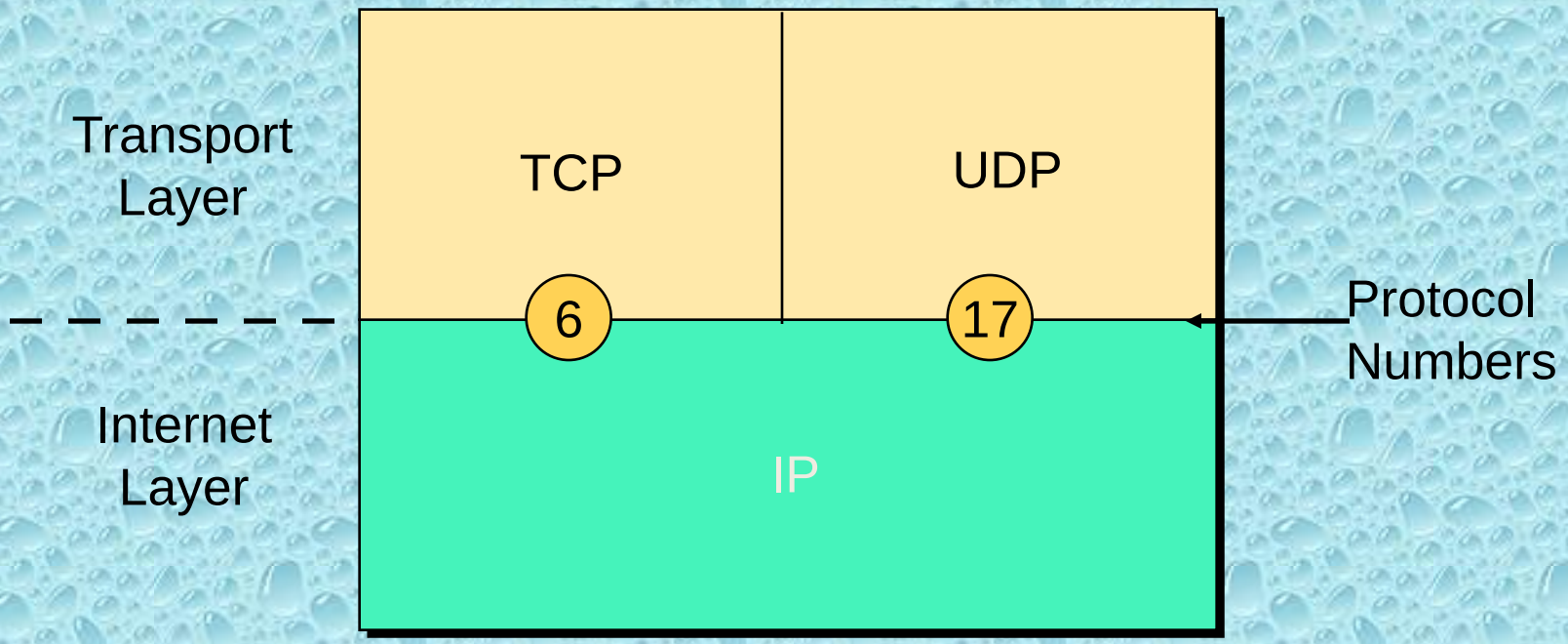
000.	當此值為 0 的時候，表示目前未被使用。
.0..	當此值為 0 的時候，表示封包可以被分割，若為 1 則不能被分割。
..0.	當上一個值為 0 時，此值為 0 就示該封包是最後一個封包，如果為 1 則表示其後還有被分割的封包。

IP包说明

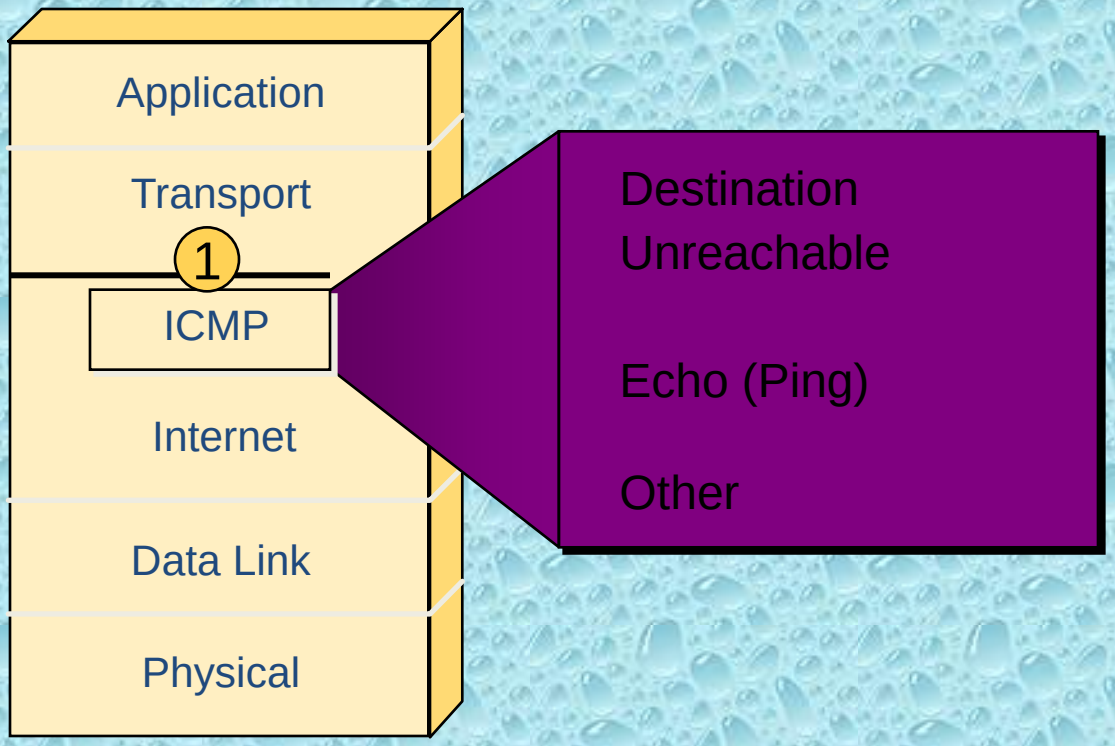
- Fragment Offset
  - 分割定位 (FO)。当一个大的封包在经过一些传输单位较小的路径时。会被切割为碎片再进行传送，由于网络情况或其他因素的影响，其抵达顺序并不一定会和发送时相同，所以当封包进行切割时，会对各片段做好定位记录，这样在重组的时候就可以对号入座了。 如果封包沒有被切割，那么 FO 的值为“0”。
- Time To Live
  - 存活時間 (TTL)。

  Protocol
  - 协议类型(PROT)。这里指的是该封包所封装的协议类型。

协议域

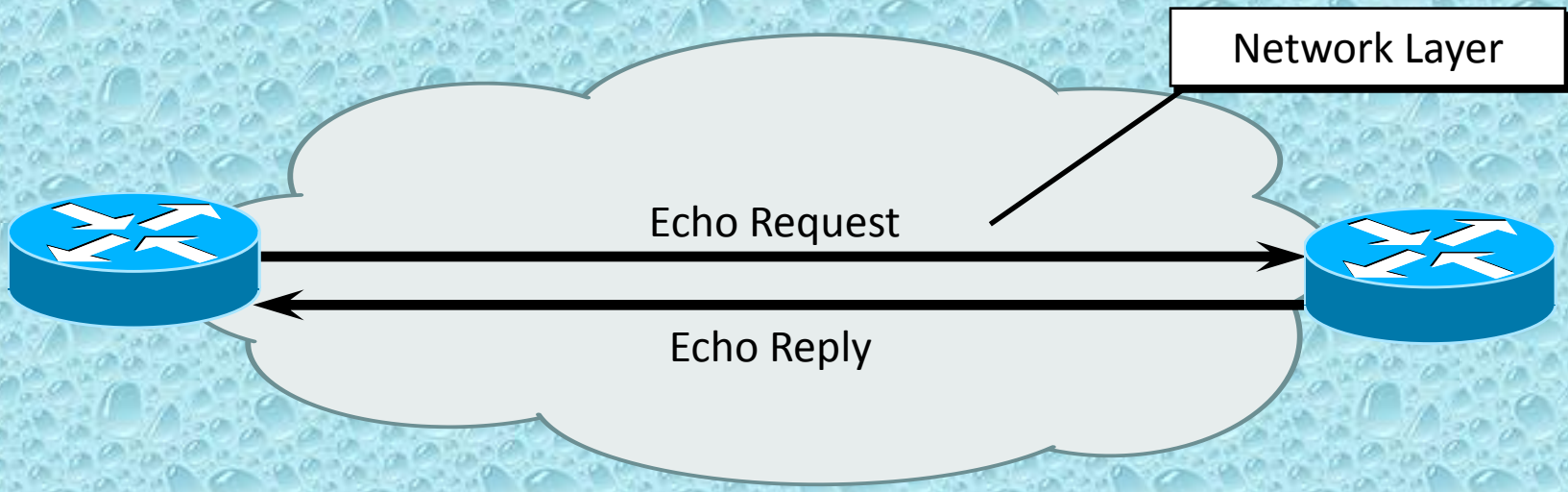

Transport Layer
TCP
UDP
6
17
Protocol Numbers
Internet Layer
IP

ICMP协议



ICMP 的全称是 Internet Control Message Protocol (网络控制信息协议)。从技术角度来说，ICMP 就是一个“错误侦测与回报机制”，其目的就是让我们能够监测网络的连线状况，也不能确保连线的准确性。

ICMP协议



Ping命令用于检查网络的可达性，Icmp Echo Reply消息表示目的节点可达。

Ping命令的结论

Ping的主要功能是确定一个给定的IP地址是否可以到达。如果ping执行成功，则暗示：

1）从源到目的节点存在一条可以工作的路径；

2）目标IP地址对应的机器在正常工作；

3）从目标节点到源节点存在一条可以工作的路径。

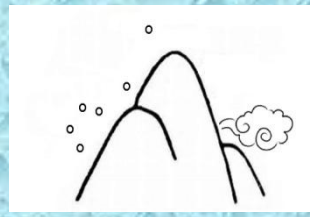
但是，从源到目标的路径与从目标回源节点的路径可能不一致。即不对称路由。

ARP协议——提出问题

- IP地址将不同的物理地址统一起来，将物理地址隐藏．上层软件使用IP地址标识节点。
- 只有两台机器知道物理地址时才能进行实际的通信。
- 分组到达目的物理网络后，发送分组的计算机需把目的主机的IP地址映射到它的物理地址上。

需经中介路由器的发送，发送方必须将中介路由器的IP地址映射到它的物理地址上。

Ip地址和mac地址的区别



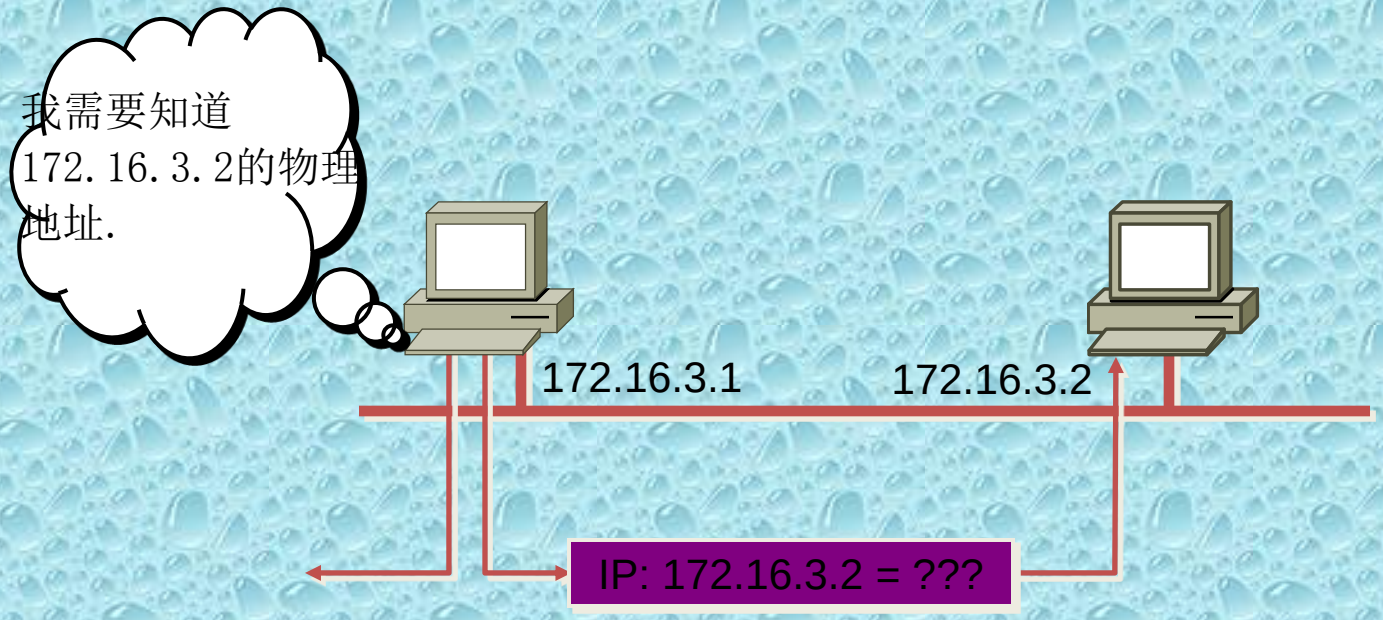
物理地址是在单个网络内部对一个计算机进行寻址时所使用的地址。在局域网中物理地址被固化在网卡的ROM中，物理地址也称为硬件地址或MAC地址。IP地址有32 bit，物理地址有48 bit。

在IP层的互连网上，我们看到的是IP数据报，在数据报的首部中写明源地址和目的地址。

在具体的物理网络的链路层，我们看到的是MAC帧，IP数据报被封装在MAC帧里面。

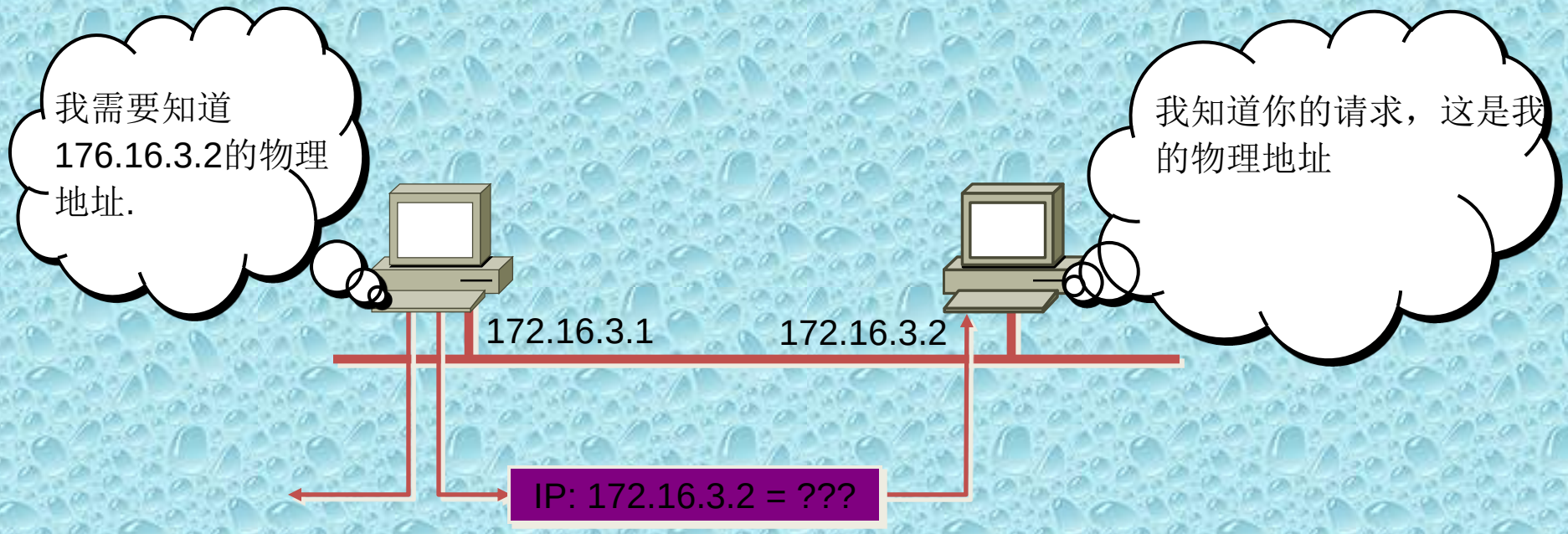
互连在一起的网络的硬件地址体系可能各不相同，但IP层抽象的互连网却屏蔽了下层的这些很复杂的细节。

ARP协议

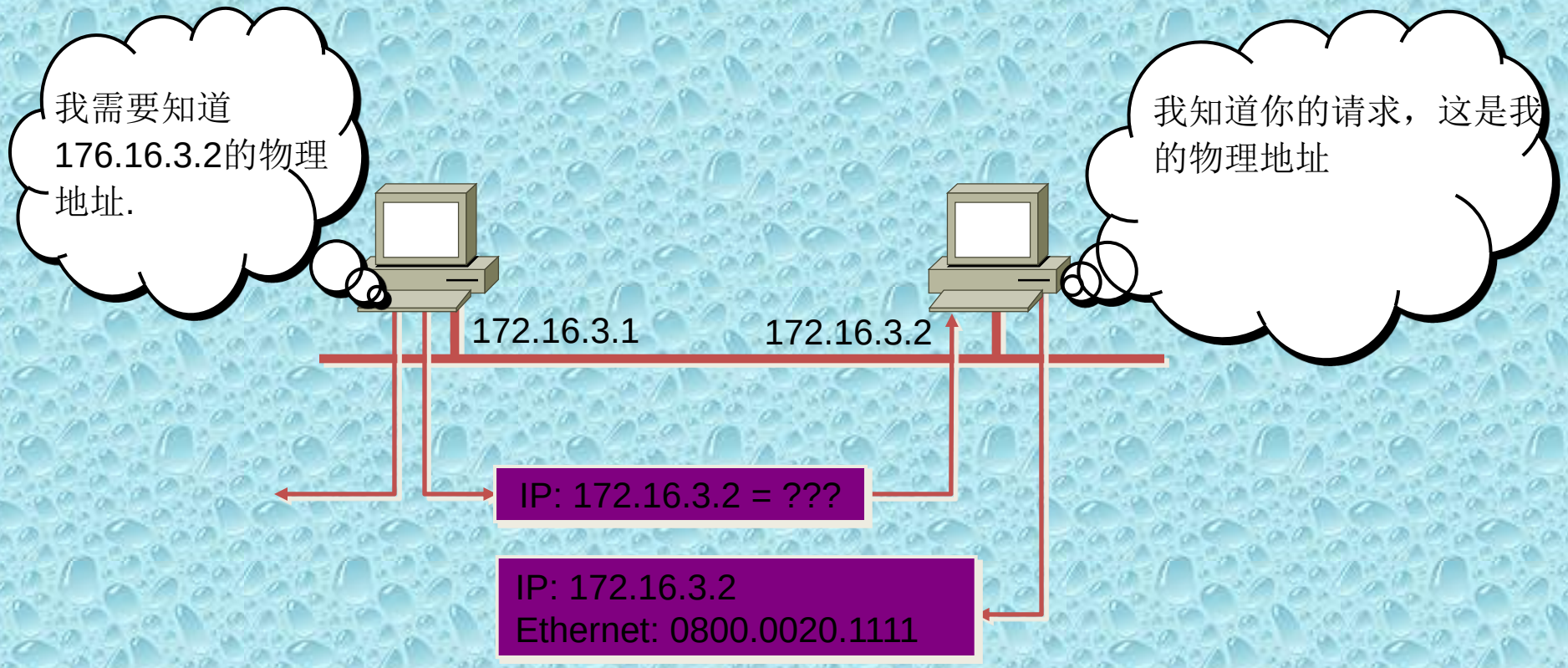


源计算机利用ARP协议向网络发广播，寻找目标主机的MAC地址。

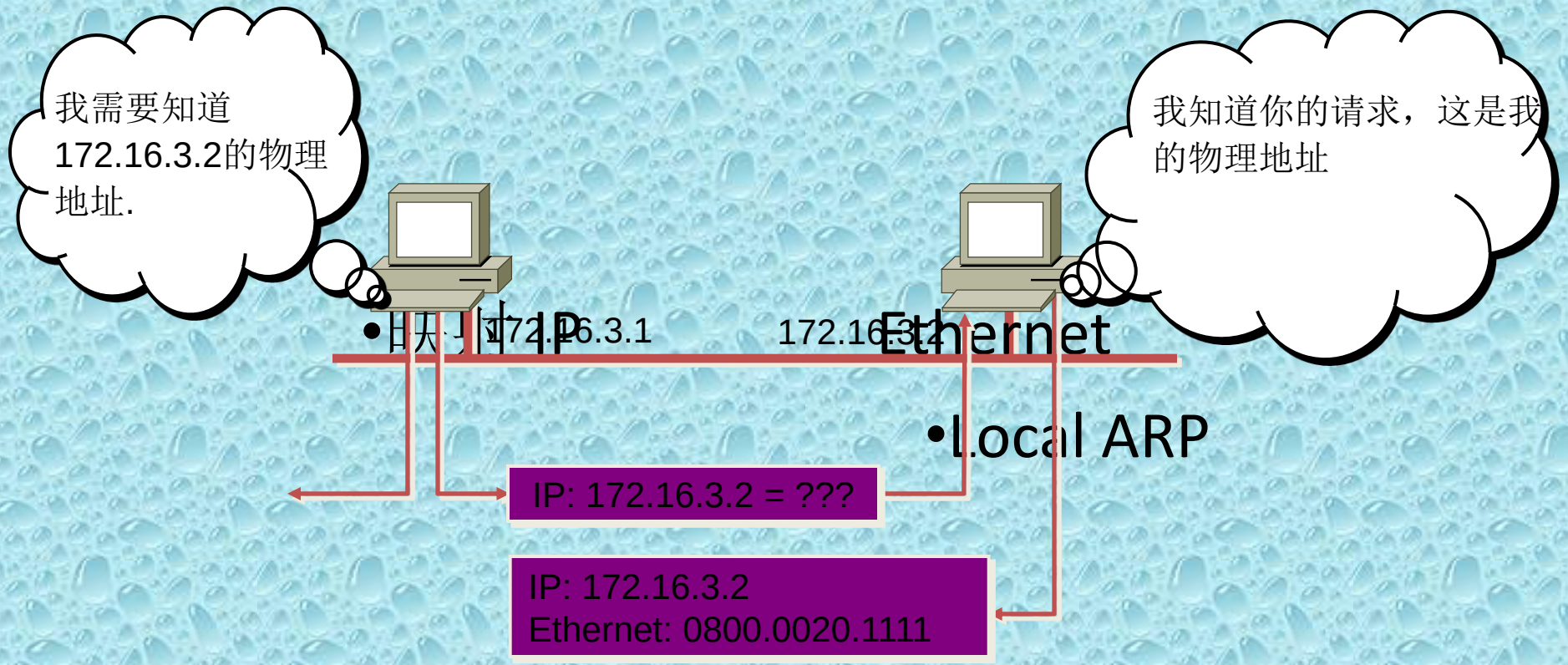
ARP协议


我需要知道176.16.3.2的物理地址.
我知道你的请求，这是我的物理地址
172.16.3.1
172.16.3.2
IP: 172.16.3.2 = ???

ARP协议


我需要知道176.16.3.2的物理地址.
我知道你的请求，这是我的物理地址
172.16.3.1
172.16.3.2
IP: 172.16.3.2 = ???
IP: 172.16.3.2
Ethernet: 0800.0020.1111

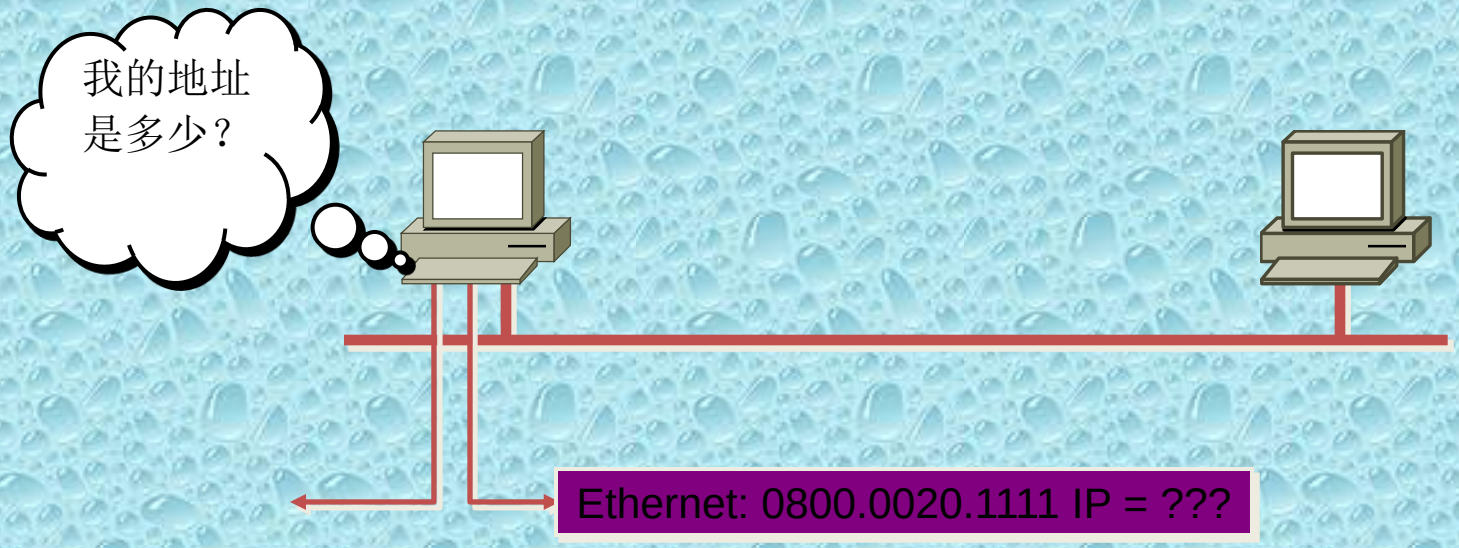
ARP协议


我需要知道172.16.3.2的物理地址.
我知道你的请求，这是我的物理地址
172.16.3.1
172.16.3.2
IP
Ethernet
Local ARP
IP: 172.16.3.2 = ???
IP: 172.16.3.2
Ethernet: 0800.0020.1111

ARP协议

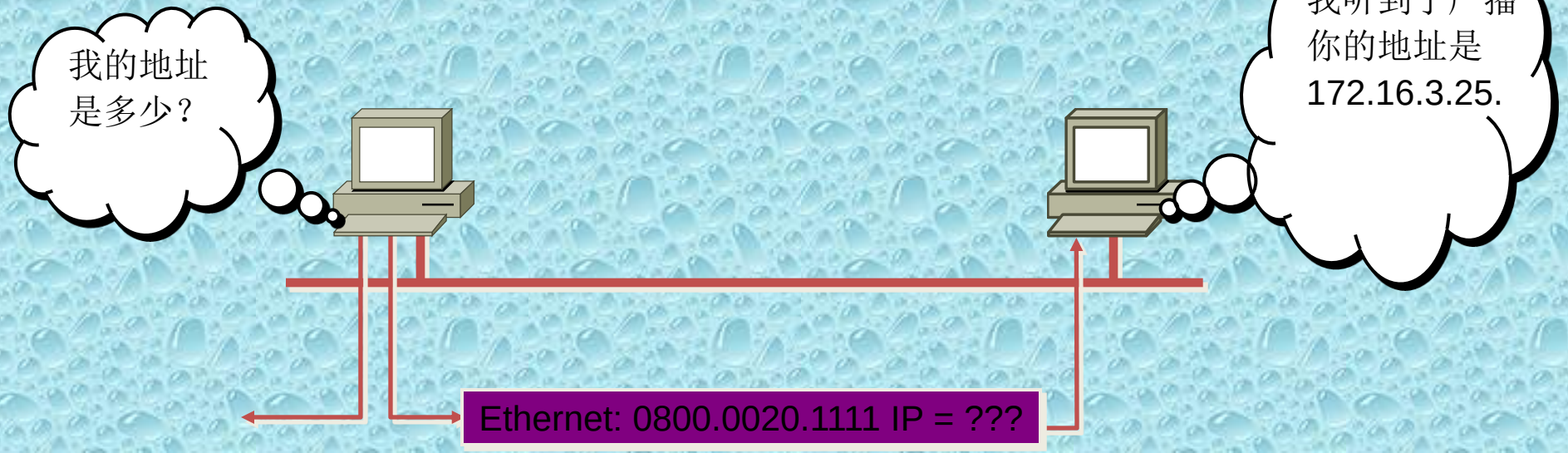
- 正常情况下，PC机只对本地网络主机进行ARP广播来查找目标主机的MAC地址，对非本地目标主机，直接把IP数据包发给默认网关，由该路由器来转发IP包。

RARP协议

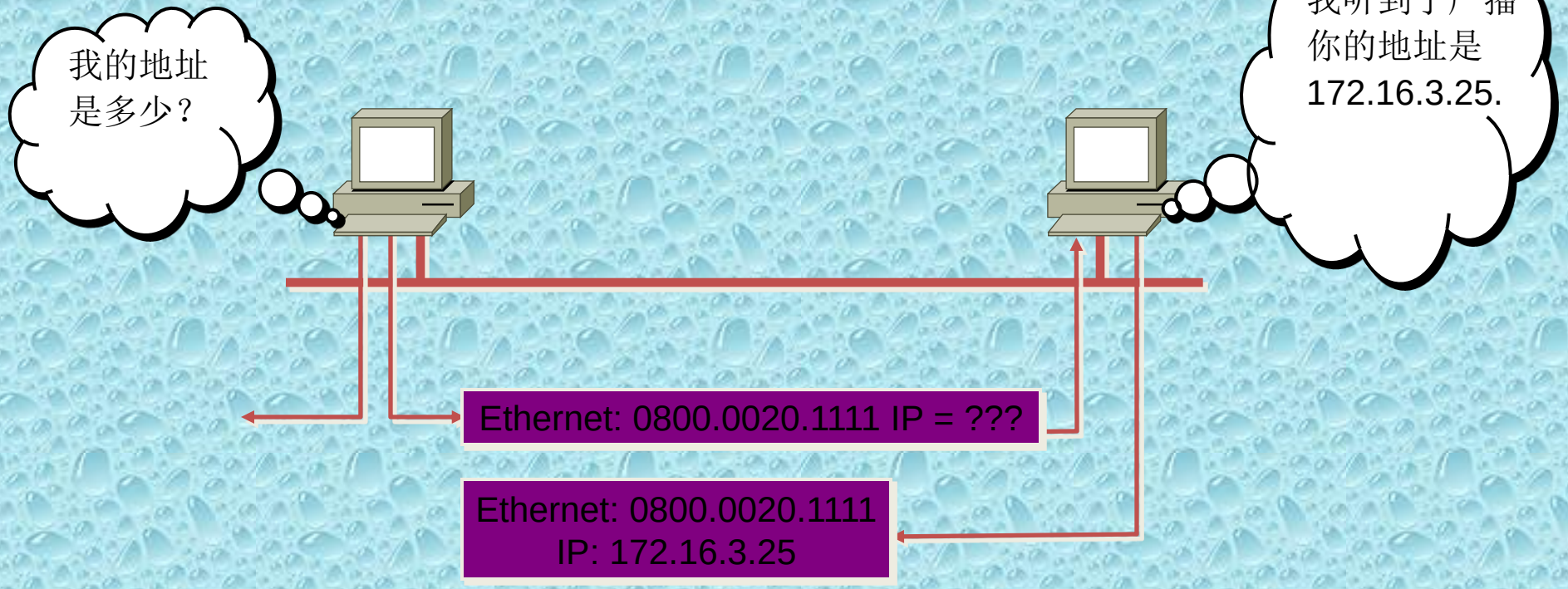


在DHCP环境中，PC机只知道自己的MAC地址，需要通过RARP协议发送广播来获取自己的IP地址。

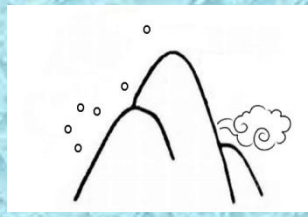
RARP协议



RARP协议


我的地址是多少？
我听到了广播你的地址是172.16.3.25.
Ethernet: 0800.0020.1111 IP = ???
Ethernet: 0800.0020.1111
IP: 172.16.3.25

RARP协议



我的地址是多少？

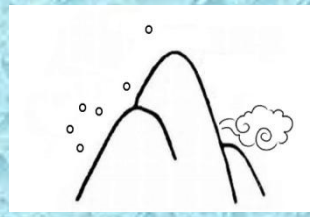
我听到了广播你的地址是172.16.3.25.

- 映射 Ethernet

Ethernet: 0800.0020.1111 IP = ???

Ethernet: 0800.0020.1111
IP: 172.16.3.25

TCP/IP各层协议简介



- 网络接口层
  - 网络接口层负责将数据放置在网络媒介上，从网络媒介上接收数据。它包括以太网（IEEE802.3）、异步传输模式（ATM）、帧中继(FR)和令牌环(IEEEE802.5)这样的协议。

TCP/IP数据帧的封装过程