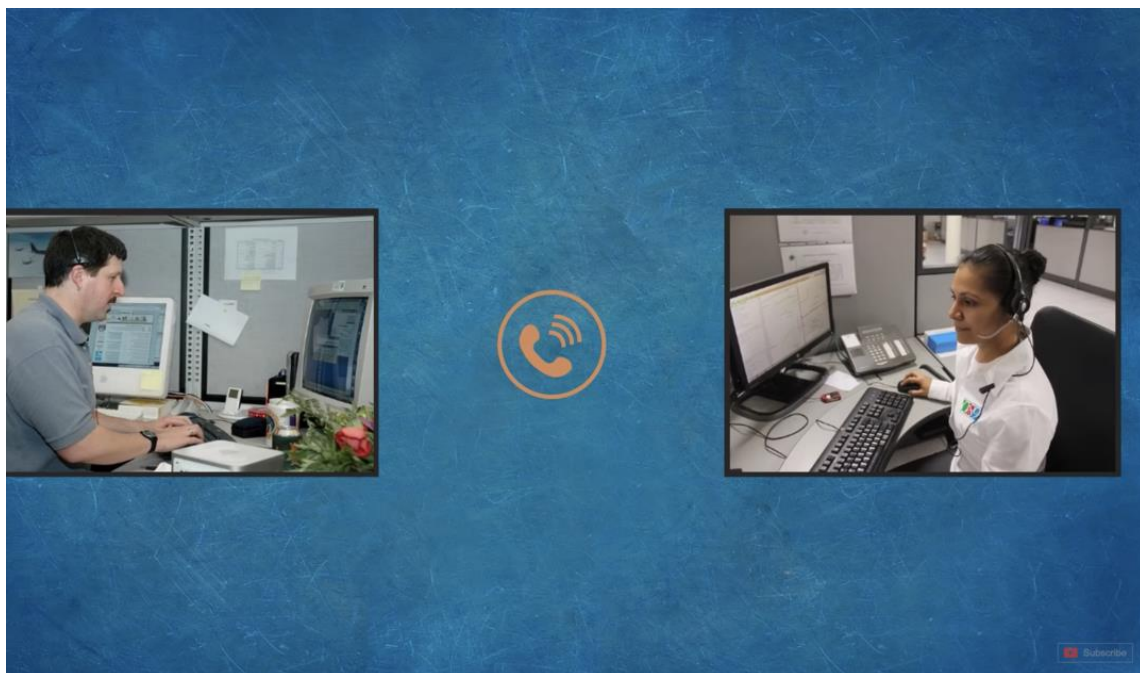


計算機網絡

OSI vs TCP/IP

一、OSI模型

前言



- 1. 历史上的对话
 - 如果你早期在一个公司的网络部门工作过，你可能会听到这样的电话对话：
 - 网管：「你好，这里是IT部门，你有没有尝试重启呢……」；
 - 员工：「没有，我这边是服务器的问题，我甚至无法固定它」；
 - 网管：「好的，听起来这应该是第3层的问题，我不负责这块，拜……」。
- 这就是很有历史的技术支持电话的对话内容，这个IT部门的网管提到了第几层的问题，其实他说的就是OSI模型的某一层，
 - 但这是什么意思呢？
 - 为什么有OSI模型这个东西，它的存在有什么意义？

一、OSI模型

OSI模型的诞生

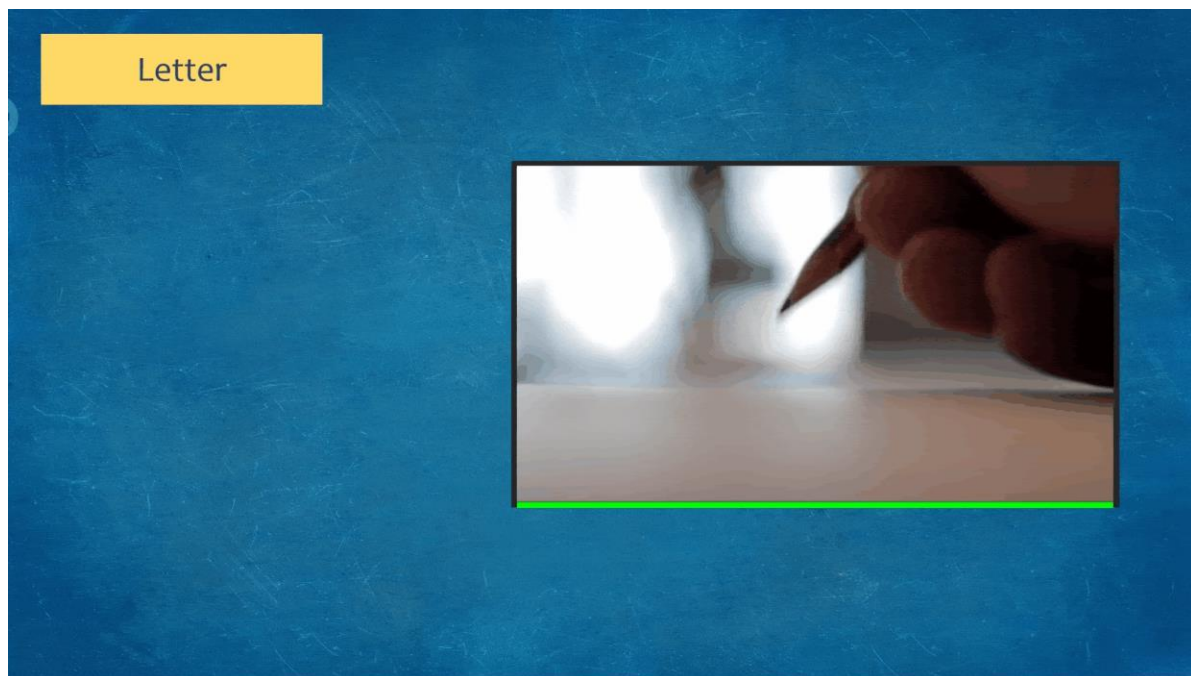


ISO International Organization for Standardization

- 网络上的所有设备（网络硬件和软件）需要一种通用的规则，才能实现彼此之间友好的交流。
- 「ISO」「国际标准化组织」将这些条例进行了整理规划，方便不同的人群了解不同的协议，这就是「OSI模型」
- 「网络协议」的分层也是为了完成这个目的，那就是将这个庞大的网络进行不同层级的划分，以便于不同的角色按照统一的「协议」生产出能够协同办公的软件或硬件。

一、OSI模型

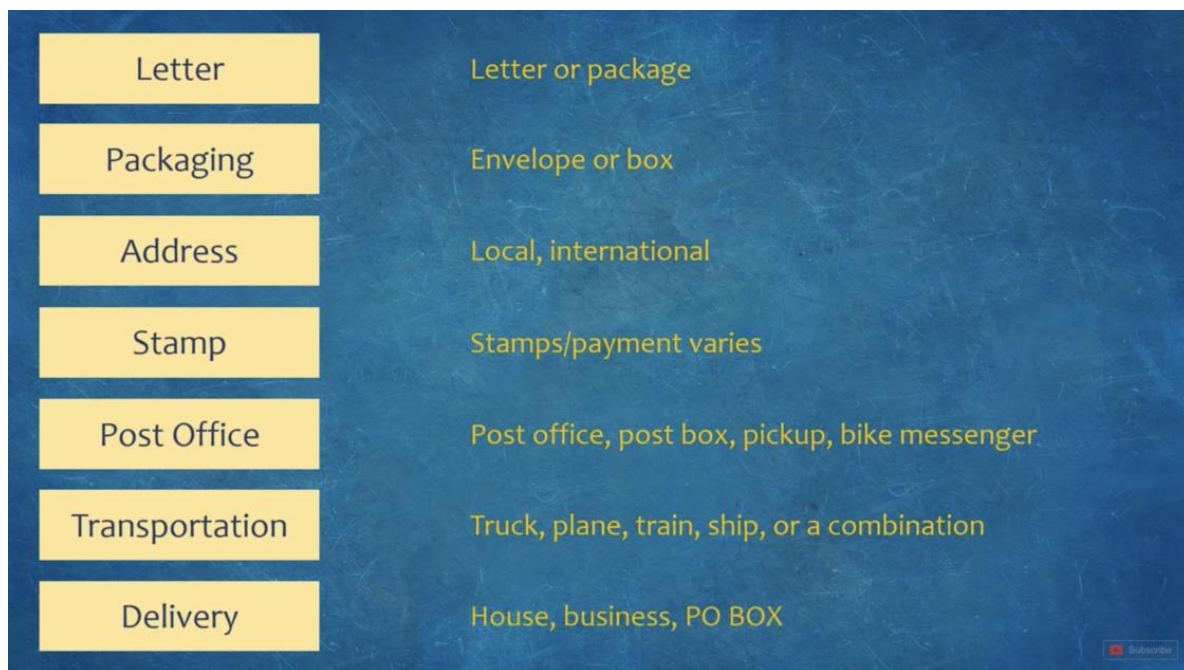
寄信的模型



- 写信
- 装进信封
- 填写地址
- 贴邮票
- 将邮件投入信箱
- 邮局工作人员取到信件，将其按地区划分，使用卡车运送到该地区邮局
- 当地邮差取到信件，并最终投递到对方手中

一、OSI模型

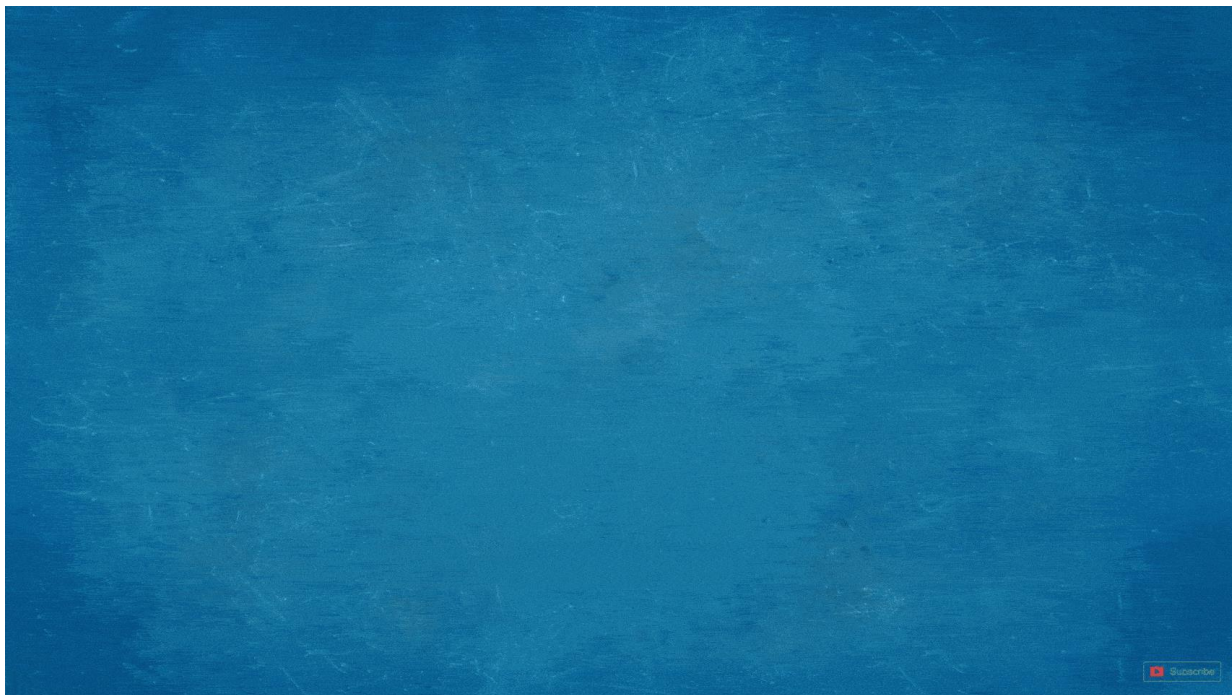
寄信的模型--模型中存在的变化



- 邮寄信件 / 包裹
- 使用信封 / 包装盒
- 邮寄本地 / 国际
- 张贴邮票 / 其他付款方式
- 邮局 / 邮政信箱 / 自行车信使
- 卡车 / 飞机 / 火车 / 轮船或其交通工具
- 房屋 / 公司 / 邮政信箱

一、OSI模型

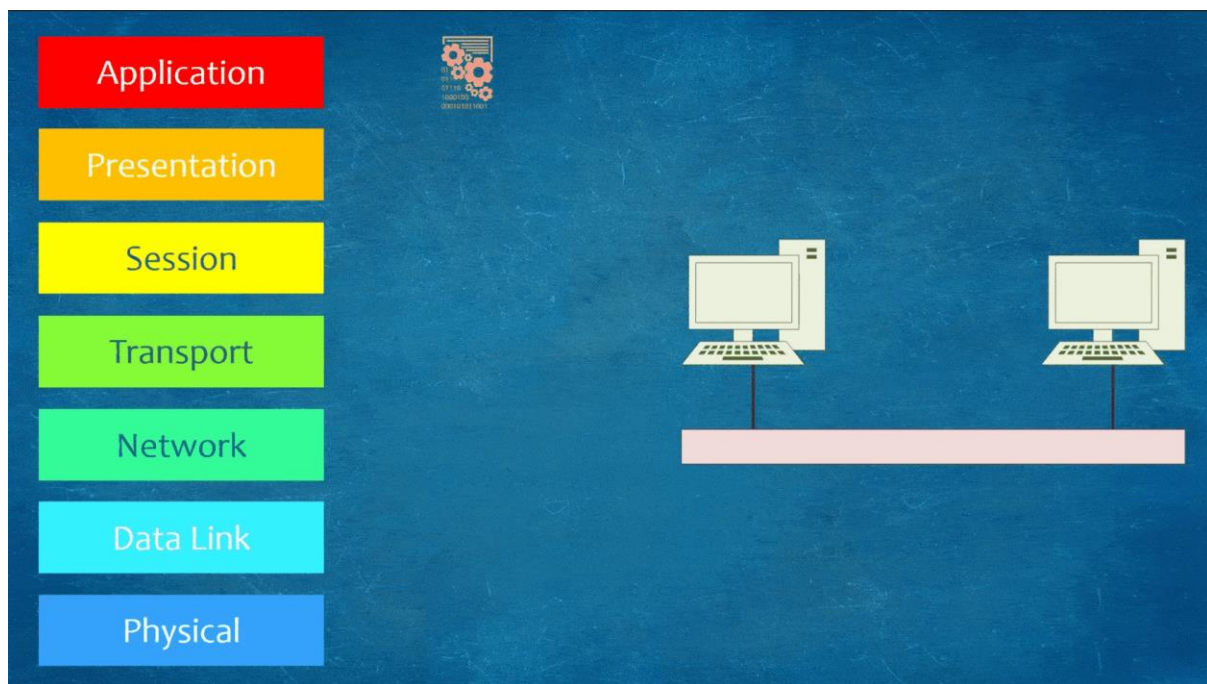
OSI模型层级划分



- 应用层【第七层】(Application)
- 表示层【第六层】(Presentation)
- 会话层【第五层】(Session)
- 传输层【第四层】(Transport)
- 网络层【第三层】(Network)
- 链路层【第二层】(Data Link)
- 物理层【第一层】(Physical)

一、OSI模型

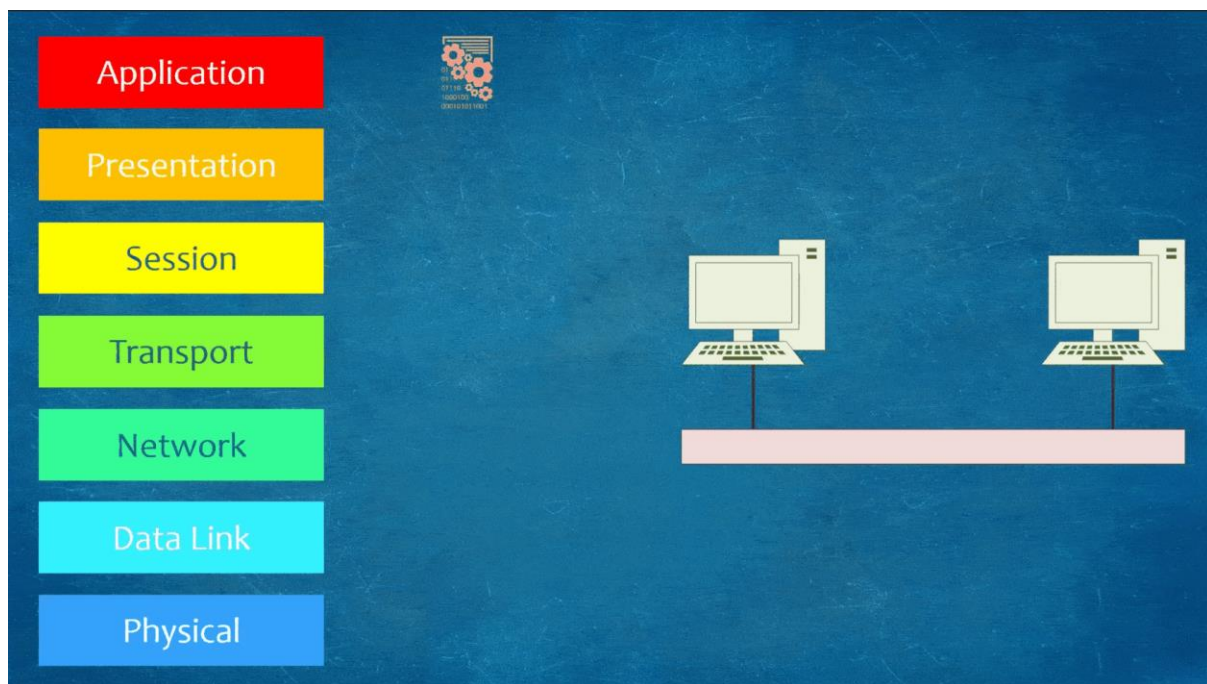
数据传输



- OSI模型」各层是如何协同工作
- 在这个案例中，我们有两台用于测试的主机，分别是「主机A」和「主机B」，我们用「主机A」准备一个信息，将其传递到「主机B」，在这个传递过程中，OSI的每一层都参与了怎样的工作？

一、OSI模型

数据传输 應用-表示-會話

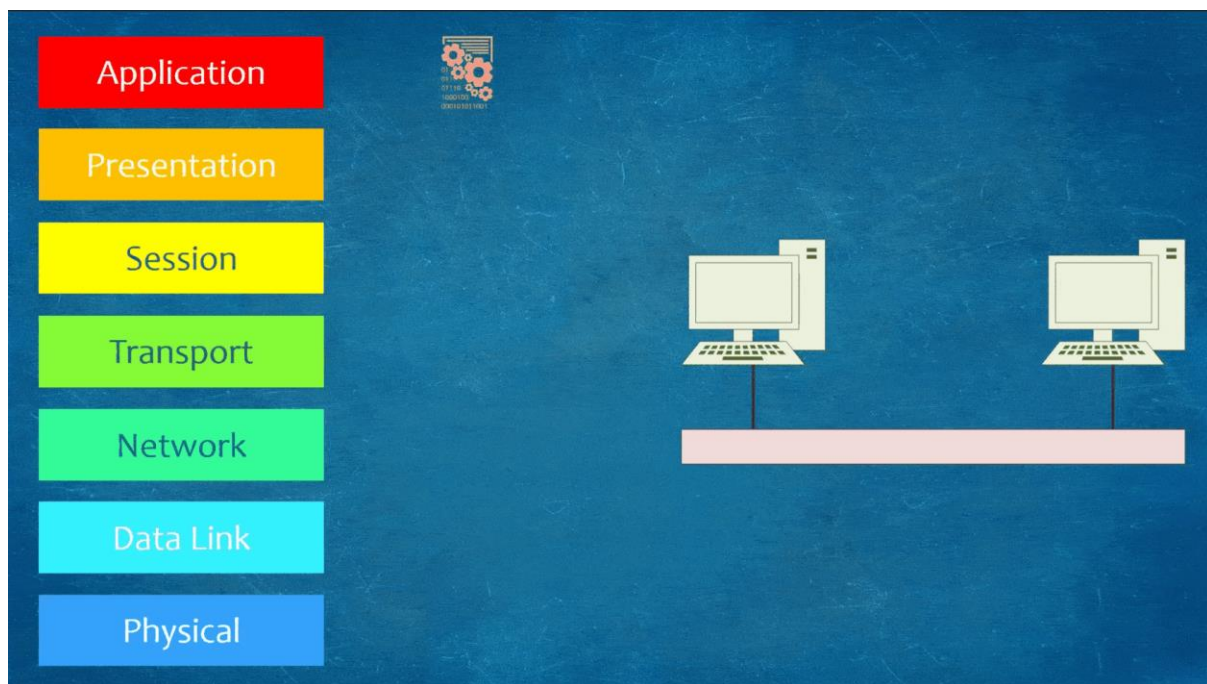


ISO International Organization for Standardization

- 4.1 应用层 （數據內容）
 - 数据的出现是始于应用层的，（例如使用「主机A」通过FTP向「主机B」传输一个文件信息）应用程序都属于应用层的内容（例如：QQ、微信、Web浏览器等都属于应用层）；应用程序准备好信息后，就可以向表示层传递了。
- 4.2 表示层 （數據內容）
 - 表示层收到应用层传来的文件信息后，会把数据转换为能与接收者「主机B」的系统格式兼容并适合传输的格式（表示层可能对数据进行加密、压缩、编码等操作，无论如何，这都是为了迎合「主机B」），操作完毕，数据会传递到会话层。
- 4.3 会话层 （數據內容）
 - 会话层接收到表示层传来的数据信息后，负责在数据传输中设置和维护两台计算机之间的通信连接，以保证这段连接不会因中断而造成不必要的麻烦。建立好会话连接后，数据继续向下传递。

一、OSI模型

数据传输 傳輸層



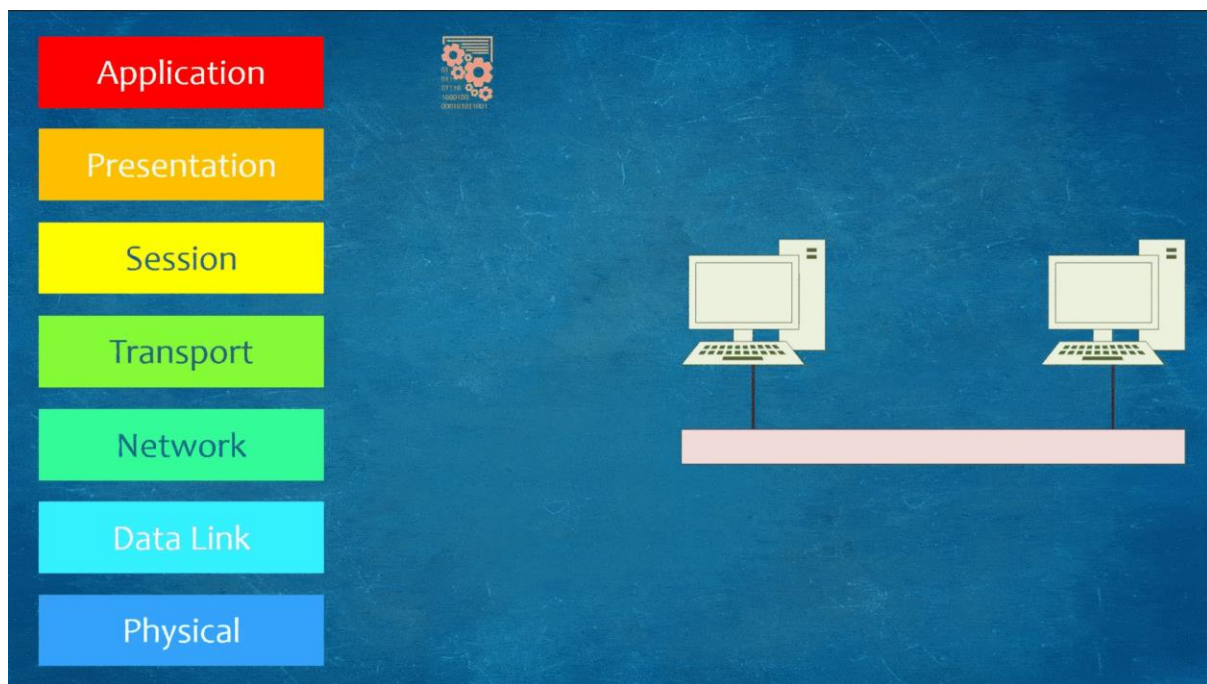
ISO International Organization for Standardization

• 传输层（[源目端口] | 数据内容）

- 虽然看起来上三层已经把大量信息处理完毕了，但这里可能遇到一些问题，例如应用层传递的文件是一个非常大的文件，在传输过程中一旦会话层的连接中断了，这意味着我们需要重新传递这个文件，这就变得非常麻烦。
- 如果我们可以把这个大文件拆分成小块传输，这个问题就迎刃而解了。这就是传输层的功劳，它可以对数据传输进行流量控制，也就是大文件分成多块进行传输。就好比 we 搬家，一辆车装不下，那就多装几辆车，然后让这些车同时出发，同时呢，为了避免道路阻塞，它还可以选择多条路来进行传输，这就是传输层的流量控制、避免拥塞和多路复用的体现。
- 同时，传输层还会给这些数据追加一个「源目端口」，一方面告知对方信息打哪来，一方面要让信息知道它们要到哪去（例如：我使用 Google 浏览器访问百度网站，这里就会添加了 Google 的端口和百度服务器的端口）。

一、OSI模型

数据传输 網絡層



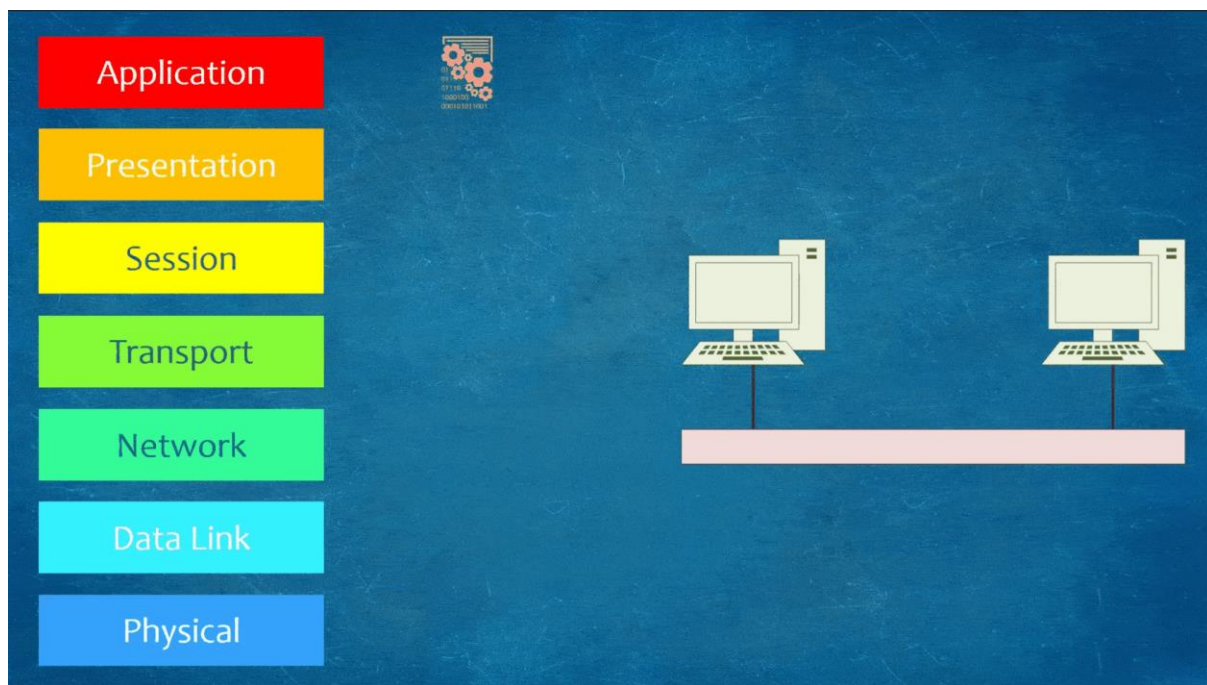
ISO International Organization for Standardization

• 4.5 网络层

- 数据传输的问题解决了，如果我们要传递的信息是被异地主机接收，就需要靠ip地址来确定异地的网段。所以网络层会给数据追加一个「源ip地址」，再添加一个「目的ip地址」，这就解决了数据从哪里来，要到哪里去的问题了~
- [源ip地址] | [源端口] | 数据内容

一、OSI模型

数据传输 數據鏈路層



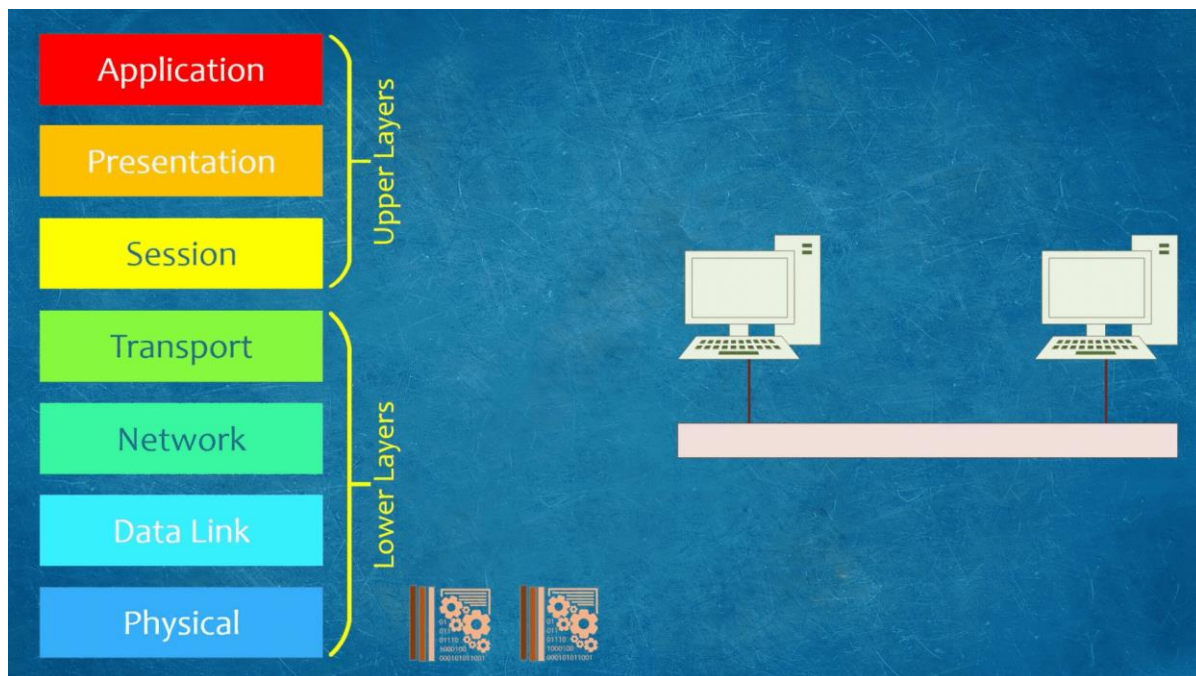
ISO International Organization for Standardization

• 4.6 数据链路层

- 网络层解决了异地网段的问题，但数据走到这里还无法确定要传送到该网段下的那一台机器，因此，数据链路层会把「主机A」和「主机B」的「MAC地址」再进行追加，这就是「源MAC地址」与「目的MAC地址」，拥有了这个唯一的烙印，「主机B」就一定能被「主机A」找到。
- 同时，数据链路层在数据的尾部追加了一个「FSC帧校验」，因为我们的数据再传输层有可能已经被切片，因此，为了检测数据的完整性，就有了这个「FSC帧校验」。
- [源目Mac地址] | [源目ip地址] | [源目端口] | 数据内容 | [FSC帧校验]

一、OSI模型

数据传输 物理層



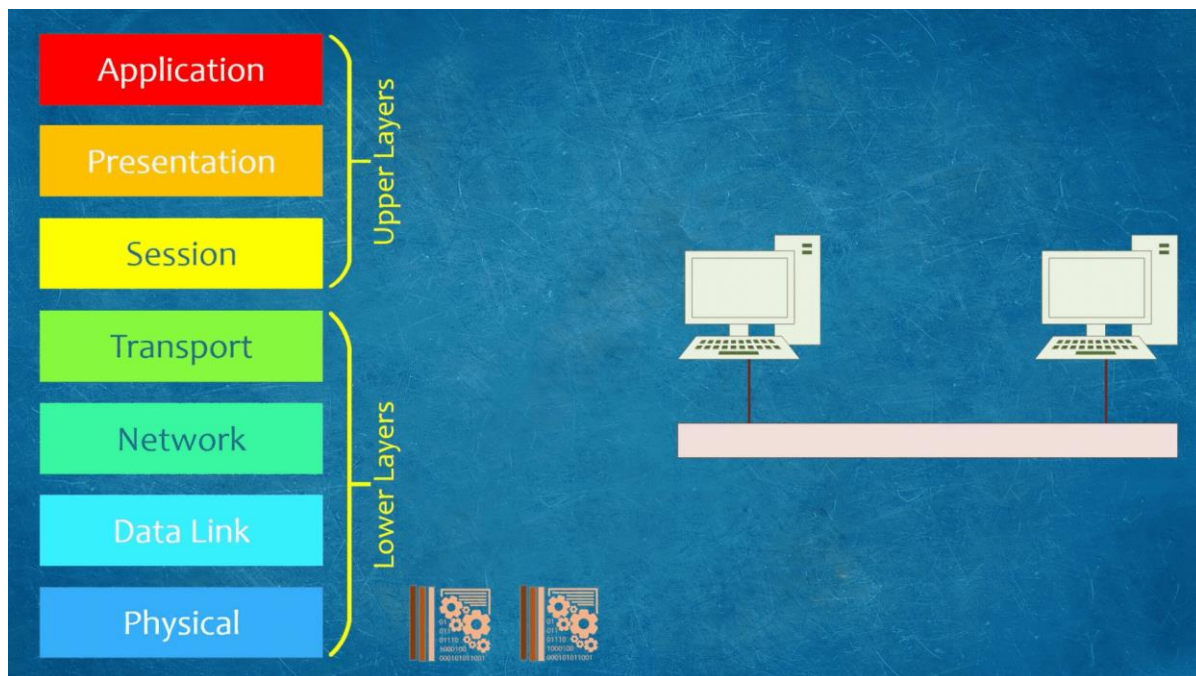
ISO International Organization for Standardization

• 4.7 物理层

- 数据通过OSI的六层磨练，还差最后一层才能将信息传递出去，只要能满足这最后一层对数据格式的需求，物理层就会给我们准备一条路，并把数据发送出去。那物理层识别什么格式的数据呢？01代码，无论是电信号还是光信号，都只能传递这种类型的数据，因此，物理层会将数据转换成 01代码并实现数据传递。
- 10011101011101011011010000
1101101011

一、OSI模型

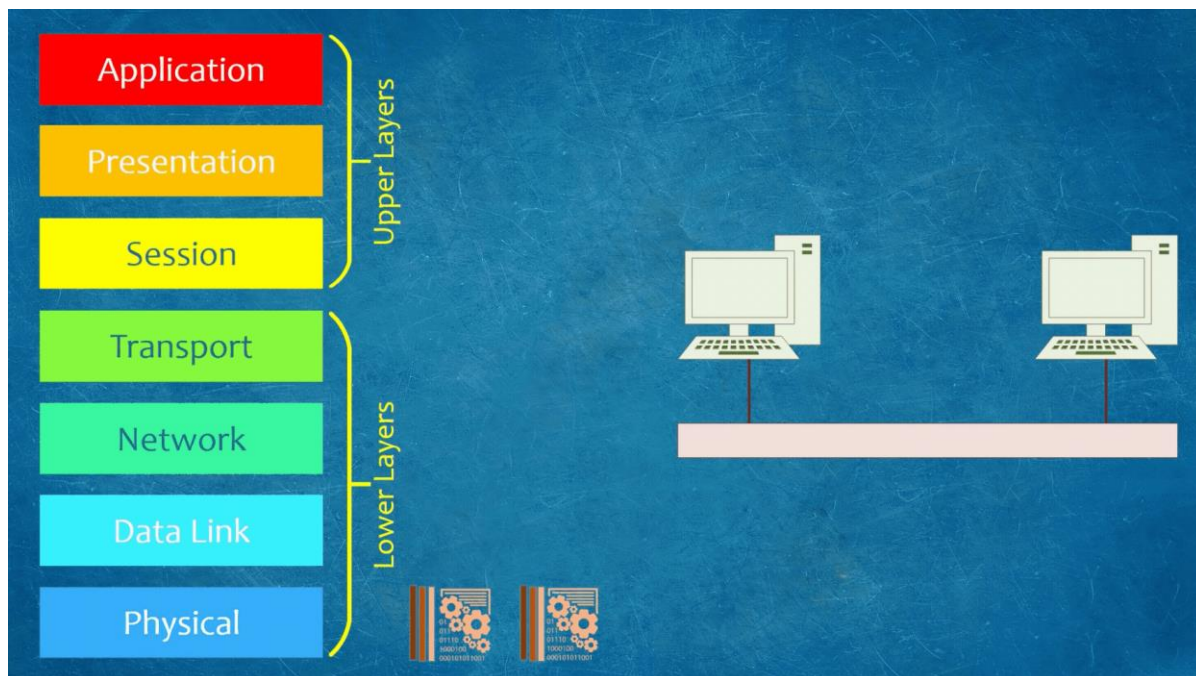
数据接收



- 数据自应用层诞生，进行了层层封装，并最终通过物理层的线路将数据发送到了另一台设备上，那接收到数据的这台设备，也就是「主机B」该如何处理这些数据呢？

一、OSI模型

数据接收 物理層

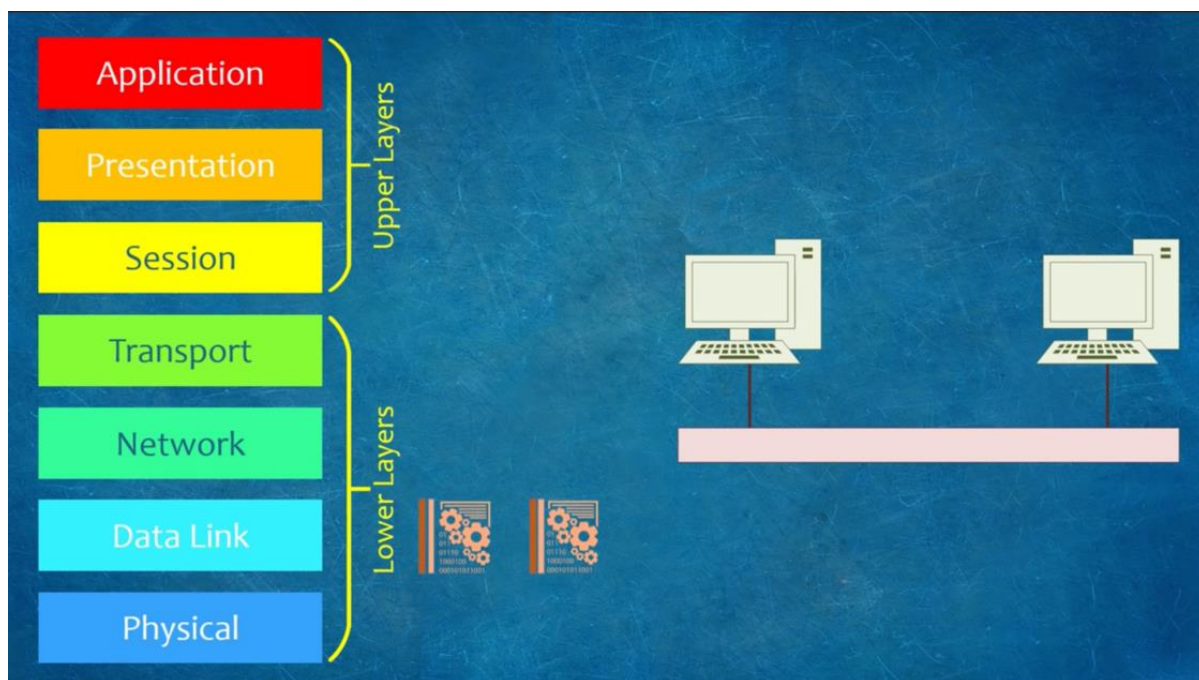


• 5.1 物理层

- 收到「主机A」发来的01代码，并对代码进行解析，将其转换成数据链路层需要的数据格式。【此时，数据已经转换为数据链路层可识别的数据格式】
- 100111010111010110110100001101101011

一、OSI模型

数据接收 數據鏈路層



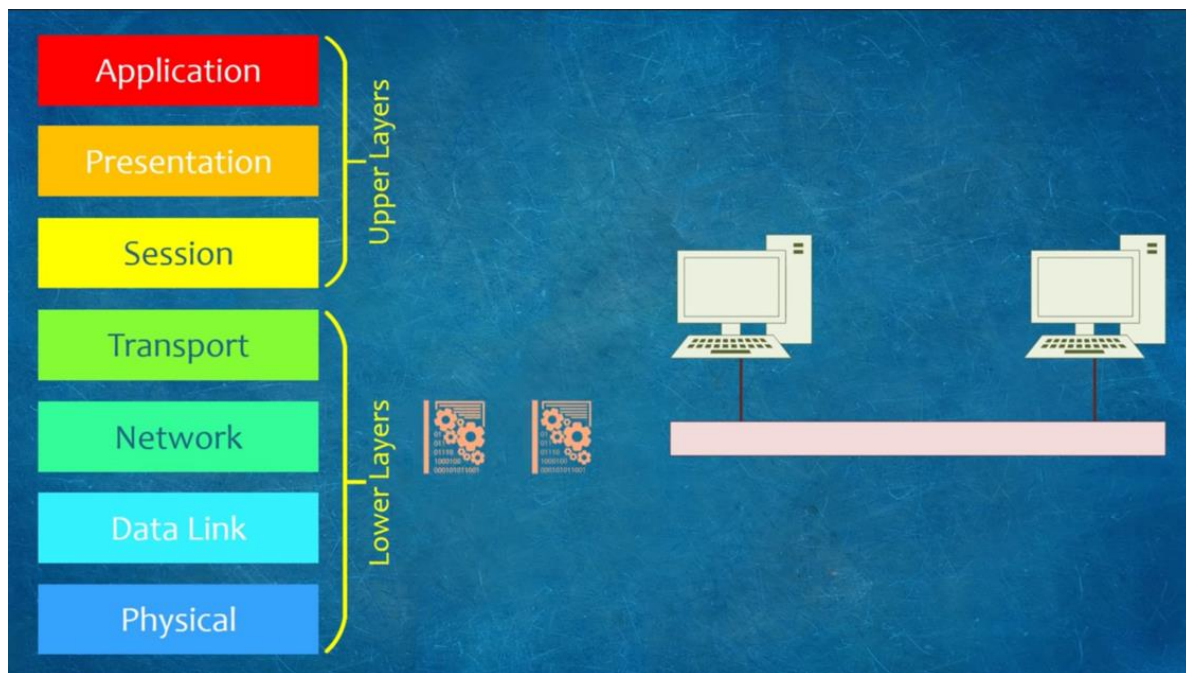
ISO International Organization for Standardization

• 5.2 数据链路层

- 数据链路层收到由物理层解析后的数据，并先进行「FSC帧校验」，检测数据的完整性，然后确定「目标MAC地址」是否正确，最后剥离「源目MAC地址」及「FSC帧校验」内容，随后将处理后的数据帧交给网络层。【此时，已经通过MAC地址找到目标设备】
- [源目Mac地址] | [源目ip地址] | [源目端口] | 数据内容 | [FSC帧校验]

一、OSI模型

数据接收 網絡層



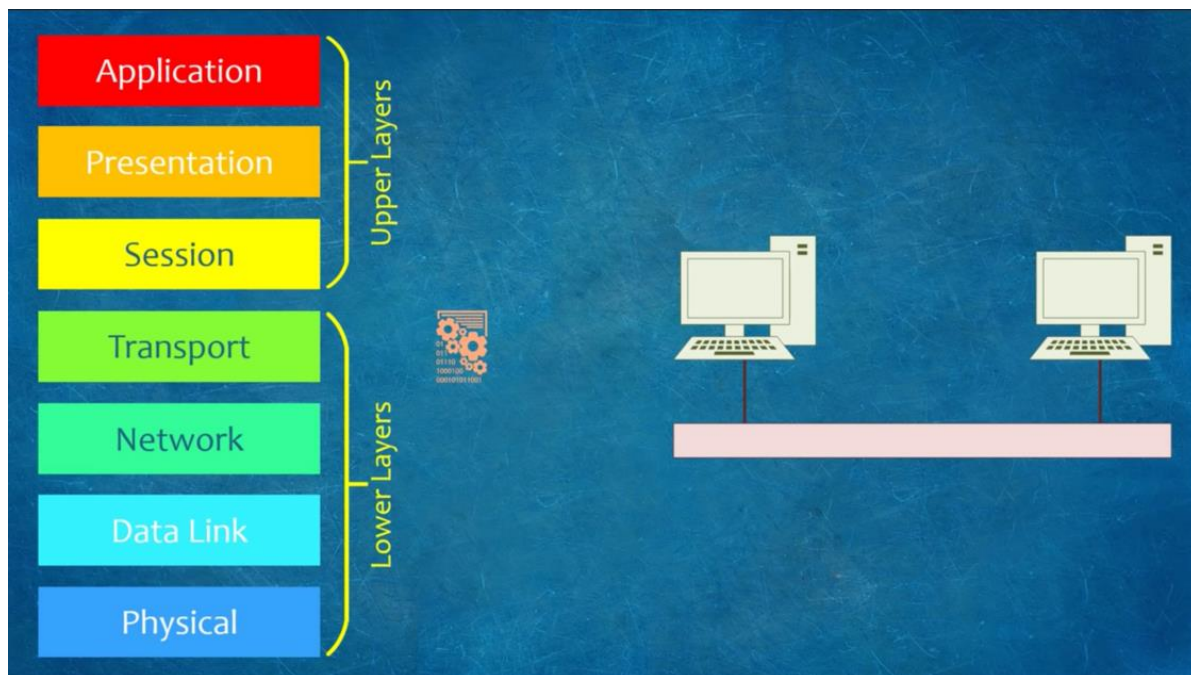
ISO International Organization for Standardization

• 5.3 网络层

- 网络层收到链路层发来的编译后的数据包，确认目标ip地址准确无误后，剥离「源目ip地址」，并将处理后的数据包交给传输层。【此时，已确定目标设备的ip地址】
- [源目ip地址] | [源目端口] | 数据内容

一、OSI模型

数据接收 傳輸層



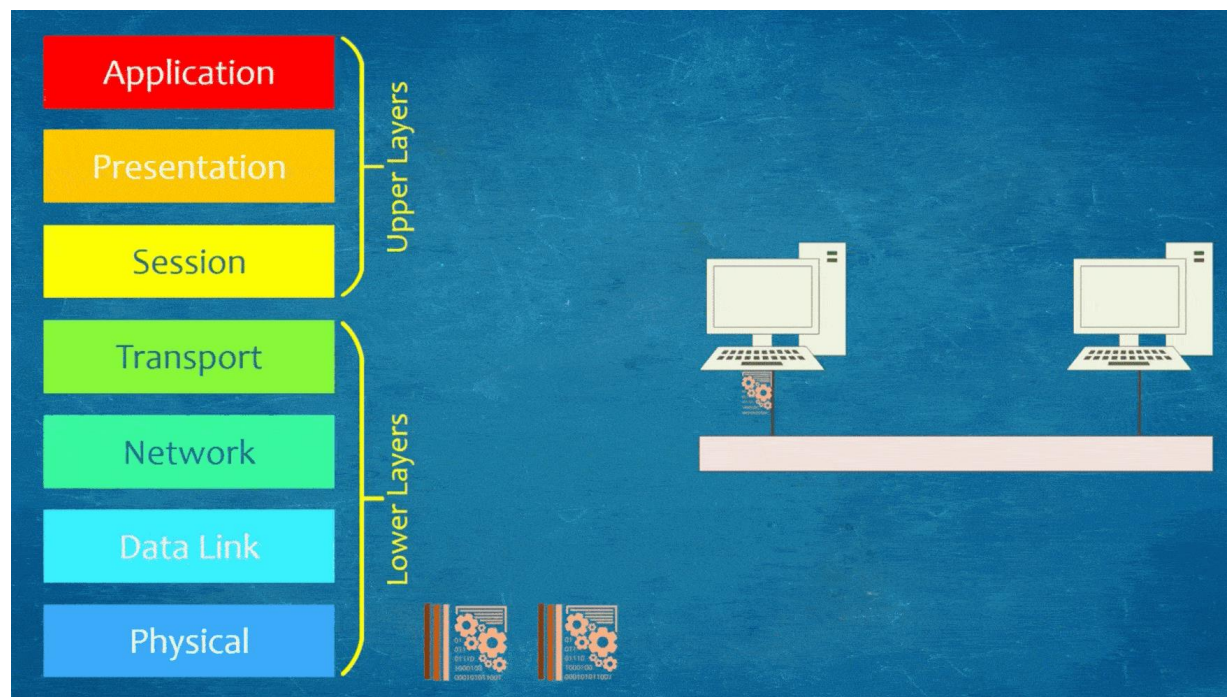
ISO International Organization for Standardization

• 5.4 传输层

- 传输层收到数据包后继续解封，查看「目标端口」是否畅通，确认无误后，剥离「源目端口信息」，并将处理后的数据交给会话层。【此时，已确定数据要传输到目标电脑上的什么应用】
- [源目端口] | 数据内容

一、OSI模型

数据接收 會話層/表示層/應用層



ISO International Organization for Standardization

- 5.5 会话层（數據內容）
 - 会话层拿到数据后，需要根据会话端口与相对应的应用建立关系，连接继续，将数据交给「表示层」。
- 5.6 表示层（數據內容）
 - 「表示层」收到数据后，需要对数据做进一步的解码处理，以确保应用能够识别这段数据内容，解码成功后，将数据交给应用层。
- 5.7 应用层（數據內容）
 - 「主机B」最终收到了「主机A」发过来的消息。

一、OSI模型

練習：把描述连到正确的OSI模型分层

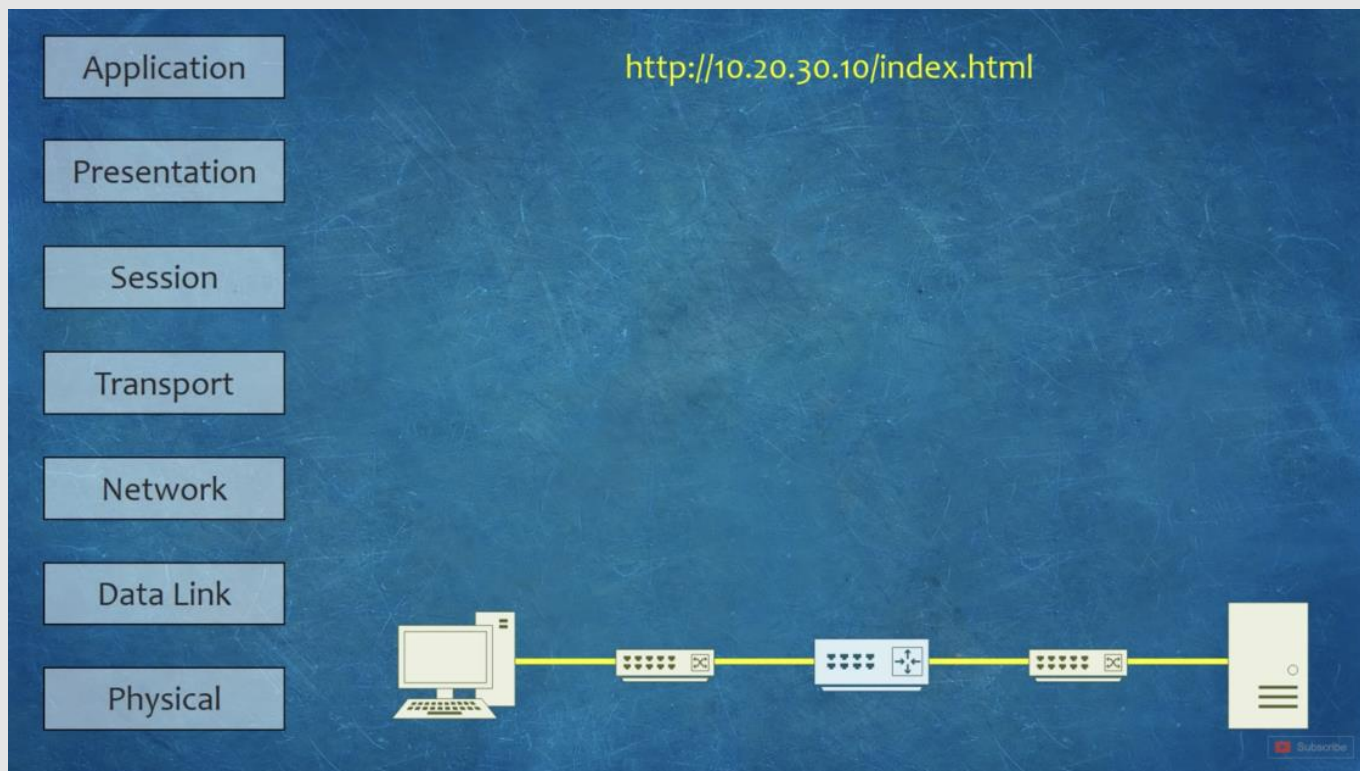
- 应用层 提供无错误数据
- 传输层 用于在介质上发送/接收数据的硬件
- 网络层 终端用户服务，例如Web浏览和文件传输
- 物理层 决定如何沿路径转发流量

一、OSI模型

練習：请阐述以下内容分别来自OSI模型的哪一层

- FTP（文件传输协议）
- Session Control Protocol（会话控制协议）
- JPEG（图像格式）
- GIF（图像格式）
- MPEG（图像格式）
- SMTP（电子邮件传输的协议）
- TIFF（图像格式）
- Telnet（隶属于TCP/IP协议族）
- Flow Control（流量控制）

二、HTTP请求范例



- 我们平时是如何来访问网页的？回答是使用「Web浏览器」（Google、火狐…），在浏览器上输入一个网站的地址，然后回车，这就发送了一个「HTTP请求」，这里请注意，这个请求是发给谁的？答案是「Web服务器」，大家可以理解为是一个放在异地的电脑主机，而这个主机里存放了很多网页文件。
- 我们输入不同的域名地址，访问到的是不同的主机，比如，我输入 `https://sspai.com`，这就是访问到了少数派家的主机，这个主机收到了我们的请求，发现我们想看少数派首页，因此，它就把首页给我们呈现出来了。
- 而 `https://sspai.com` 其实并不是地址的最终形态，它是经过处理的，便于我们记忆的网站地址，其真实的地址应该是一个ip地址，例如 `10.20.30.10`，而域名只是解析到ip地址上的一个名字，因此，当我们访问 `http://10.20.30.10/index.html` 的时候，是同样可以实现访问到少数派的首页的。

二、HTTP请求范例

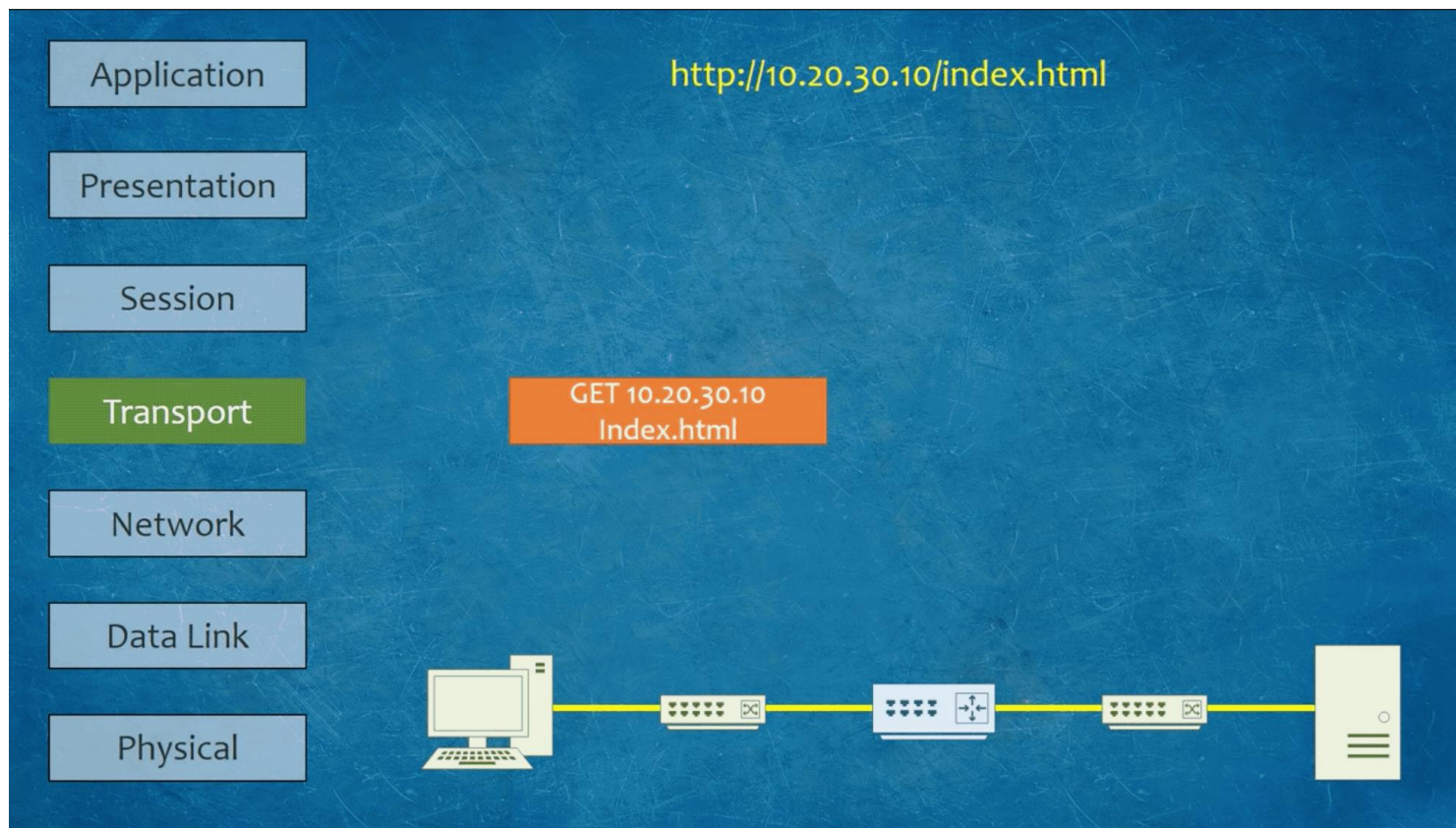
HTTP工作流程 上层协议



- HTTP工作流程 上层协议
- HTTP就是一个简单的浏览器请求，然后得到服务器的网页响应吗？当然不是，如果我们把HTTP请求放到OSI模型中，它甚至不能简单放入一层，尽管人们常说它应该是来自应用层的请求，但事实上它跨越了OSI模型的多层，这几层分别是「应用层」、「表示层」、「会话层」，我们也称这三层是一个「上层协议」。
- 当我们点击 `http://10.20.30.10/index.html` 的时候，它会创建要发送的请求，并确保正在发送的任何特殊字符均已正确编码，这里，HTTP并没有真正意义上的管理上层会话。

二、HTTP请求范例

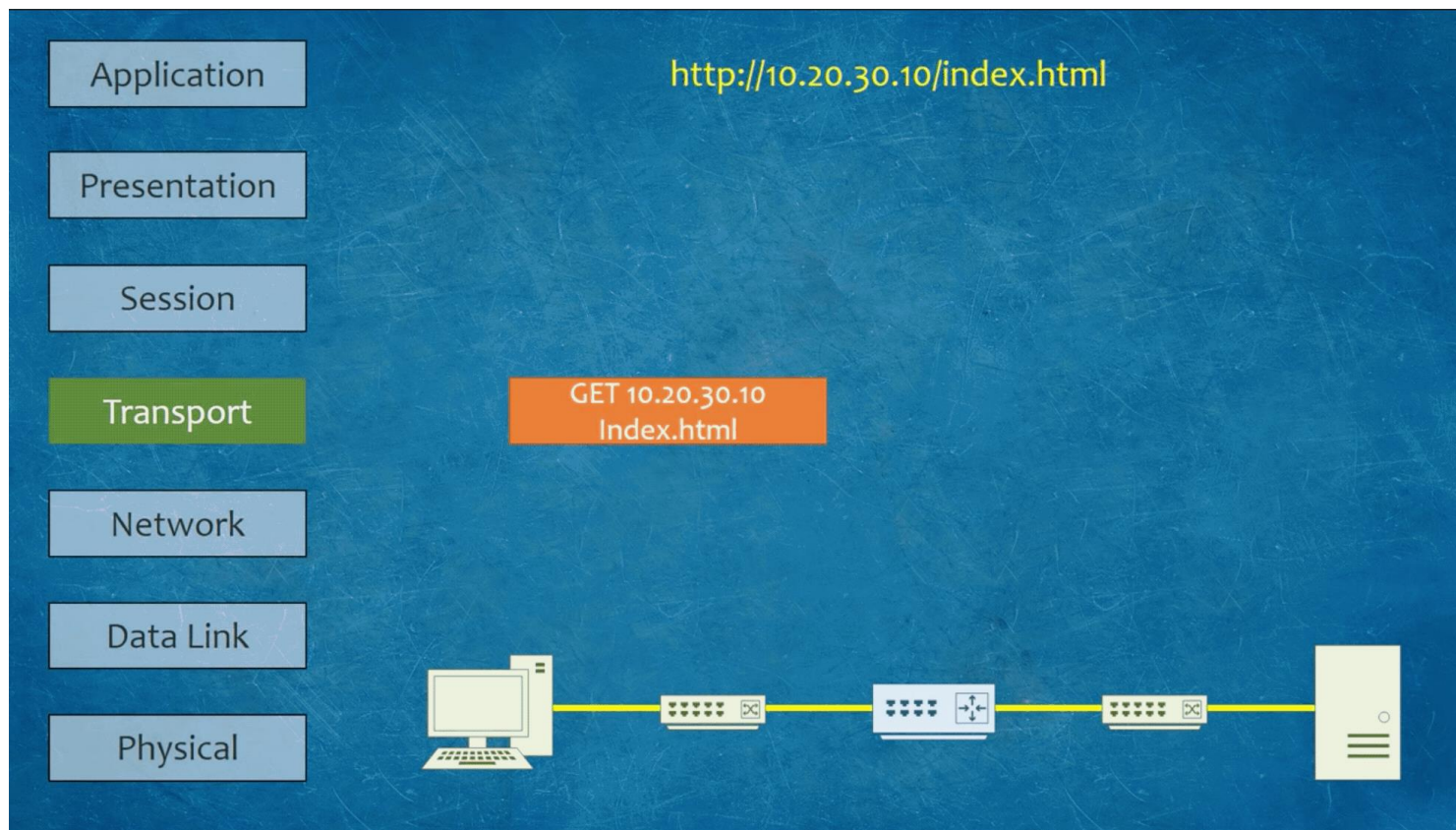
HTTP工作流程 傳輸層



- 2.1 【传输层】追加源目端口
- 在传输层，我们使用了一个称为「TCP」的协议。在传输层，数据被分割成小的片段，每个片段都有自己的源端口和目标端口。在应用层，数据是一个整体，我们不需要知道它被分割成多少片段。在传输层，我们只需要知道每个片段的源端口和目标端口。在应用层，我们只需要知道数据的源地址和目标地址。在传输层，我们只需要知道每个片段的源端口和目标端口。在应用层，我们只需要知道数据的源地址和目标地址。
- HTTP通常在服务器端使用80端口，而在客户端使用浏览器的端口。在传输层，我们使用「TCP」来建立连接。在应用层，我们使用「HTTP」来发送请求。在传输层，我们使用「TCP」来建立连接。在应用层，我们使用「HTTP」来发送请求。
- 这个源端口和目标端口就是一段附在请求后面的信息。在传输层，我们使用「TCP」来建立连接。在应用层，我们使用「HTTP」来发送请求。

二、HTTP请求范例

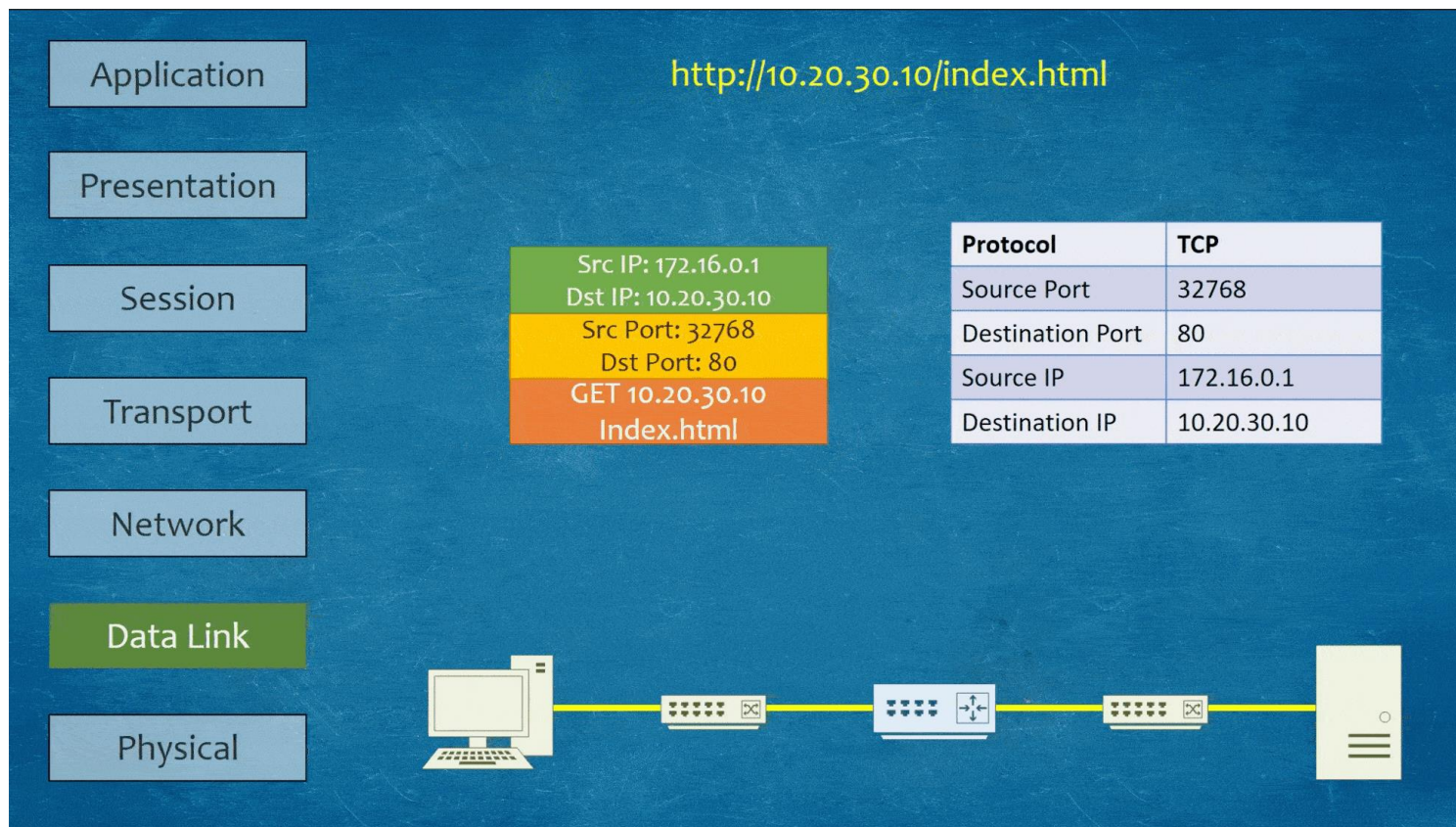
HTTP工作流程 網絡層



- 2.2【网络层】追加源目IP
- 信息到达网络层，我们需要使用IP地址协助寻址，在URL地址当中，我们可以获知该请求要发送到哪里，10.20.30.10 就是我们请求的目的地，此时的信息会追加「Dst IP: 10.20.30.10」，同时，为了让服务器知道请求者的IP地址，会同时追加一个源IP地址，也就是「Src IP: 172.16.0.1」，这个源IP地址并不固定，根据使用者的网络环境不同而不同。

二、HTTP请求范例

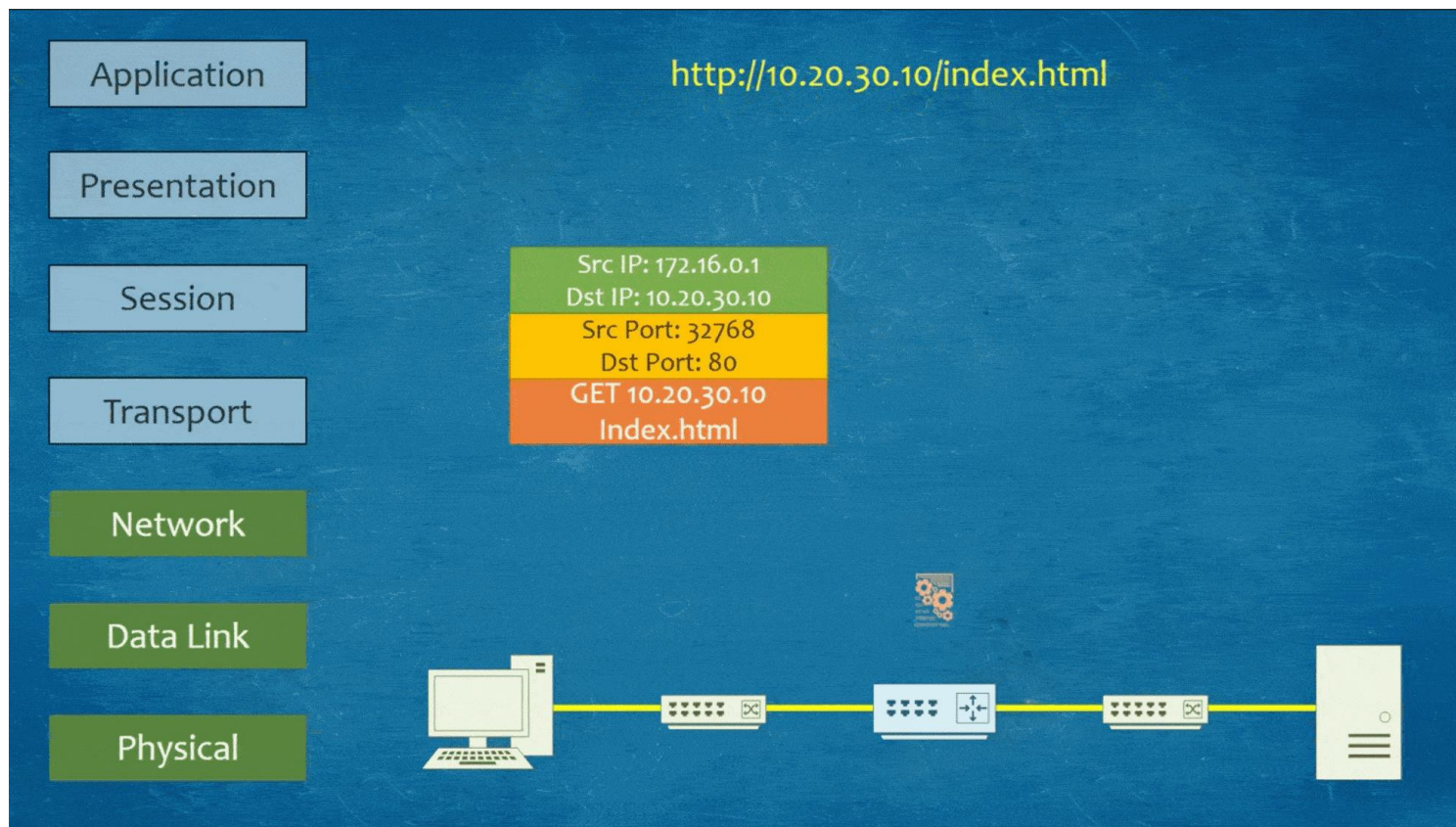
HTTP工作流程 數據鏈路層



- **【数据链路层】** 追加源目MAC地址
- 通过观察上图我们不难发现，客户端电脑和服务端并没有直接连接，它们位于不同的网络上，所以数据包时不能直接发送到服务器上的，若想要实现传输，则必须在数据链路层中将数据先发送给路由器，然后再让路由器帮我们转发数据。
- 因为我们有以太网协议，想要将数据从一个网段传递到另一个网段，就必须通过路由器，因此，我们需要在数据包上需要再追加客户端电脑网卡的MAC地址，也就是「Src MAC: AA:BB:CC:11:22:33」以及帮我们转发数据的路由器的MAC地址，也就是「Dst MAC: 2A:34:D2:11:23:A1」，需要注意的是，交换机并不需要MAC地址。当然，我们还在数据包末尾追加一段用于纠错的信息，这就是「CRC (Error Checking)」。

二、HTTP请求范例

HTTP工作流程 物理層

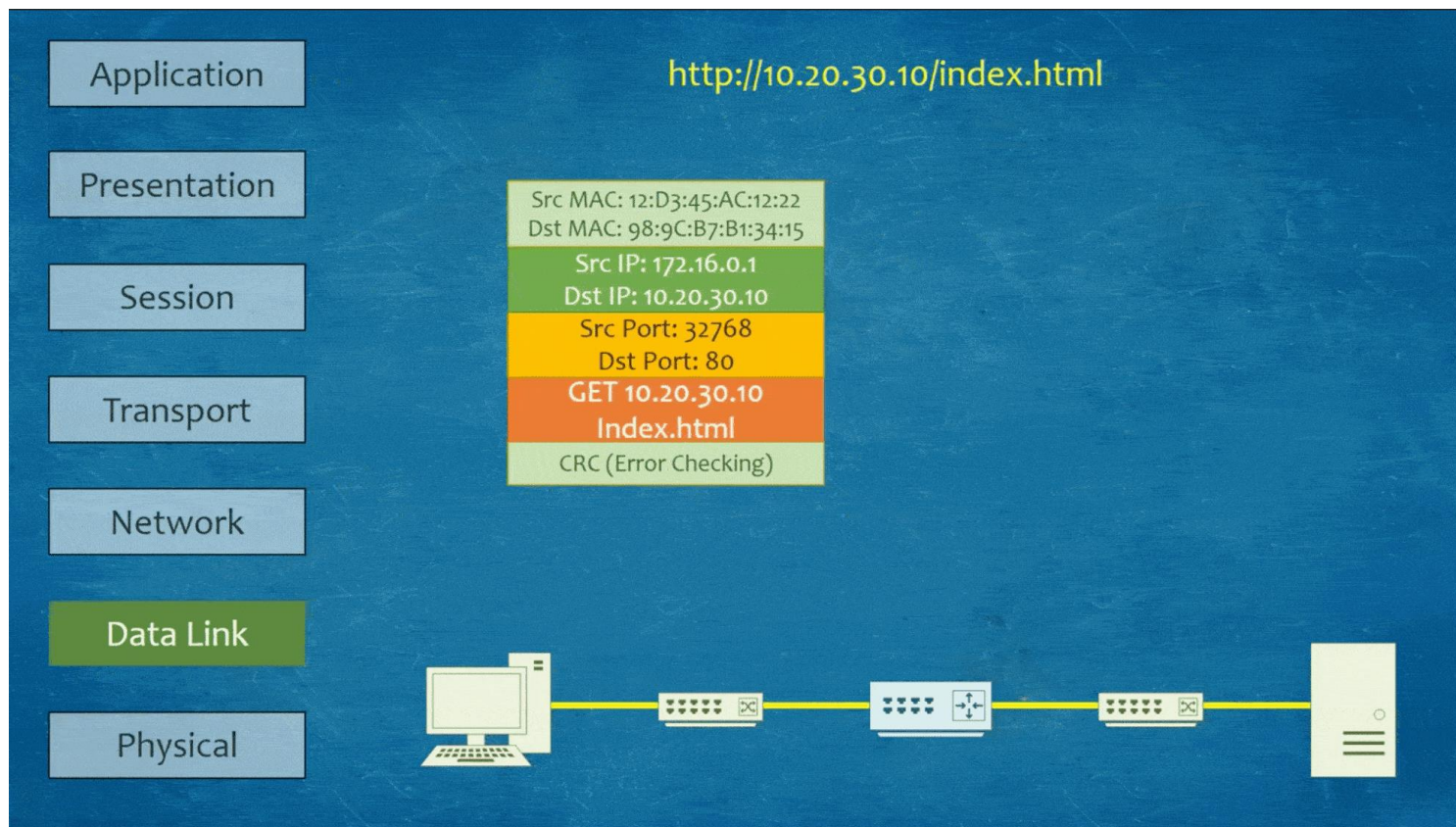


• 2.4 【物理层】数据发送至路由器

- 通过物理层将数据发送至路由器
- 最后，我们的客户端主机在追加好这些「标头信息」后，终于将数据发送到了路由器上，当路由器，此时，我们需要借助「网络层」「数据链路层」「物理层」共同协作，将数据包中的「源目MAC地址进行置换」，也就是将「源MAC地址」由原来「客户端主机的MAC地址」替换为该「路由器自己的MAC地址」，然后将「目的MAC地址」替换为「服务器的MAC地址」，这样一来，路由器就可以实现数据转发了。

二、HTTP请求范例

HTTP工作流程 服务器端数据处理



- 2.5 服务器端数据处理
 - 服务器通过链路层收到由路由器转发过来的数据信息，并检查是否有错误，若没有错误则会将「源目MAC地址」及「CRC错误校验」信息剥离，并将信息交给网络层。
 - 网络层收到后，确定源目ip地址是否正确，确保信息准确无误后，再将信息转发给传输层。
 - 传输层，则确定源目端口信息是否准确，然后将数据交给上三层，让应用根据请求地址寻找指定页面，并用脚本获取URL对应参数进行后续处理

補充知識 詳解 OSI七層協議

- 如何记住所有层的名字呢？很简单。
 - 请不要把暗号告诉任何人 (Please Do Not Tell the Secret Password to Anyone) 。
 - Please | 物理层 (Physical Layer)
 - Do | 数据链路层 (Data Link Layer)
 - Not | 网络层 (Network Layer)
 - Tell (the) | 传输层 (Transport Layer)
 - Secret | 会话层 (Session Layer)
 - Password (to) | 表示层 (Presentation Layer)
 - Anyone | 应用层 (Application Layer)
- 虽然某些技术（比如协议）在逻辑上比起其它层来说可能“属于”某一层，但并非所有的技术都完全契合 OSI 模型中的单个层。例如，以太网 (Ethernet)、802.11 (Wifi) 和地址解析协议 (ARP, Address Resolution Protocol) 程序在不只一层上工作。
- OSI 只是一个模型，一个工具，并不是一组规则。

補充知識

TCP/IP 是由 OSI 七層協定簡化而來

OSI七层网络模型	TCP/IP四层概念模型	对应网络协议
应用层 (Application)	应用层	HTTP、TFTP, FTP, NFS, WAIS、SMTP
表示层 (Presentation)		Telnet, Rlogin, SNMP, Gopher
会话层 (Session)		SMTP, DNS
传输层 (Transport)	传输层	TCP, UDP
网络层 (Network)	网络层	IP, ICMP, ARP, RARP, AKP, UUCP
数据链路层 (Data Link)	数据链路层	FDDI, Ethernet, Arpanet, PDN, SLIP, PPP
物理层 (Physical)		IEEE 802.1A, IEEE 802.2到IEEE 802.11

補充知識 OSI 第一层

- 节点（设备）和网络硬件。设备包括集线器、中继器、路由器、计算机、打印机，等等。这些设备内的硬件包括天线、放大器、网卡（NIC, Network Interface Card），等等。
- 设备接口机制。电缆如何连接到某个设备，以及连接到设备上的哪个地方（电缆连接器和设备插座）？连接器的大小和形状如何，它有多少个引脚？决定引脚处于活动状态还是非活动状态的东西是什么？
- 功能和程序逻辑。连接器中每个引脚的功能是什么——发送还是接收？决定事件顺序，以便节点能够开始与第二层上的另一个节点通信的程序逻辑是什么？
- 电缆协议和规范。以太网（CAT）、USB、数字用户线（DSL, Digital Subscriber Line）等。规范包括最大电缆长度、调制技术、无线电规范、线路编码和位同步（下文还有更多）。
- 电缆类型。选择有屏蔽或非屏蔽双绞线、非双绞线、同轴电缆等。从这里了解更多电缆类型。
- 信号类型。基带一次一个比特流，就像铁路一样——只支持单向。宽带同时包含多个比特流，就像双向高速公路一样。
- 信号传输方法（可能是有线的或无线的）。选择包括电（以太网）、光（光纤网络、光纤）、无线电波（802.11 WiFi, a/b/g/n/ac/ax 变种或蓝牙）。如果是无线的话，则要考虑频率：2.5 GHz 还是 5 GHz。如果是有线或以太网的话，则还要考虑网络标准，例如 100BASE-T 和相关标准。

補充知識 OSI 第一层

- 如何排查第一层中的问题？这里是第一层中要当心的一些问题：
 - 电缆失效，例如电线损坏或连接器损坏
 - 网络硬件设备故障，例如电路损坏
 - 东西正被拔出（我们都遇到过……）
- 如果第一层出了问题，第一层以上的任何东西都不会正常工作。
- 第一层包含的是基础设施，它让网络通信变成可能。
- 它定义了用于激活、维护和停用网络设备之间的物理连接的电气、机械、程序和功能规范。——来源
- 有趣的事实：深海通信电缆在全世界传输数据。这张地图会让你大开眼界：
<https://www.submarinecablemap.com/>。

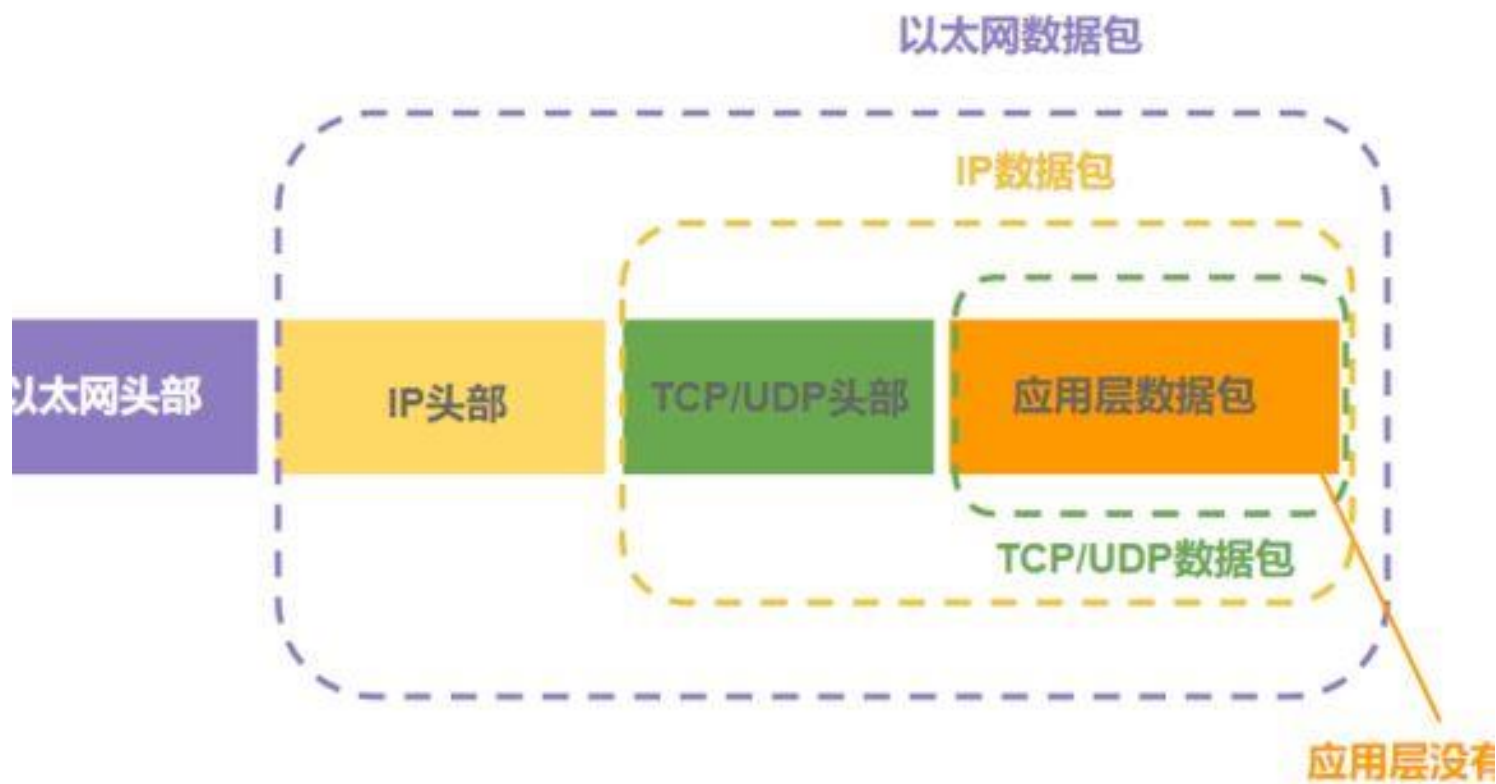
1.1 通过光缆、电缆、无线电波等方式将设备连接起来组网

1.2 两个不同局域网(移动、电信)通信，需要ISP互联网服务供应商的物理连接

1.3 作用：传送比特流 0和1

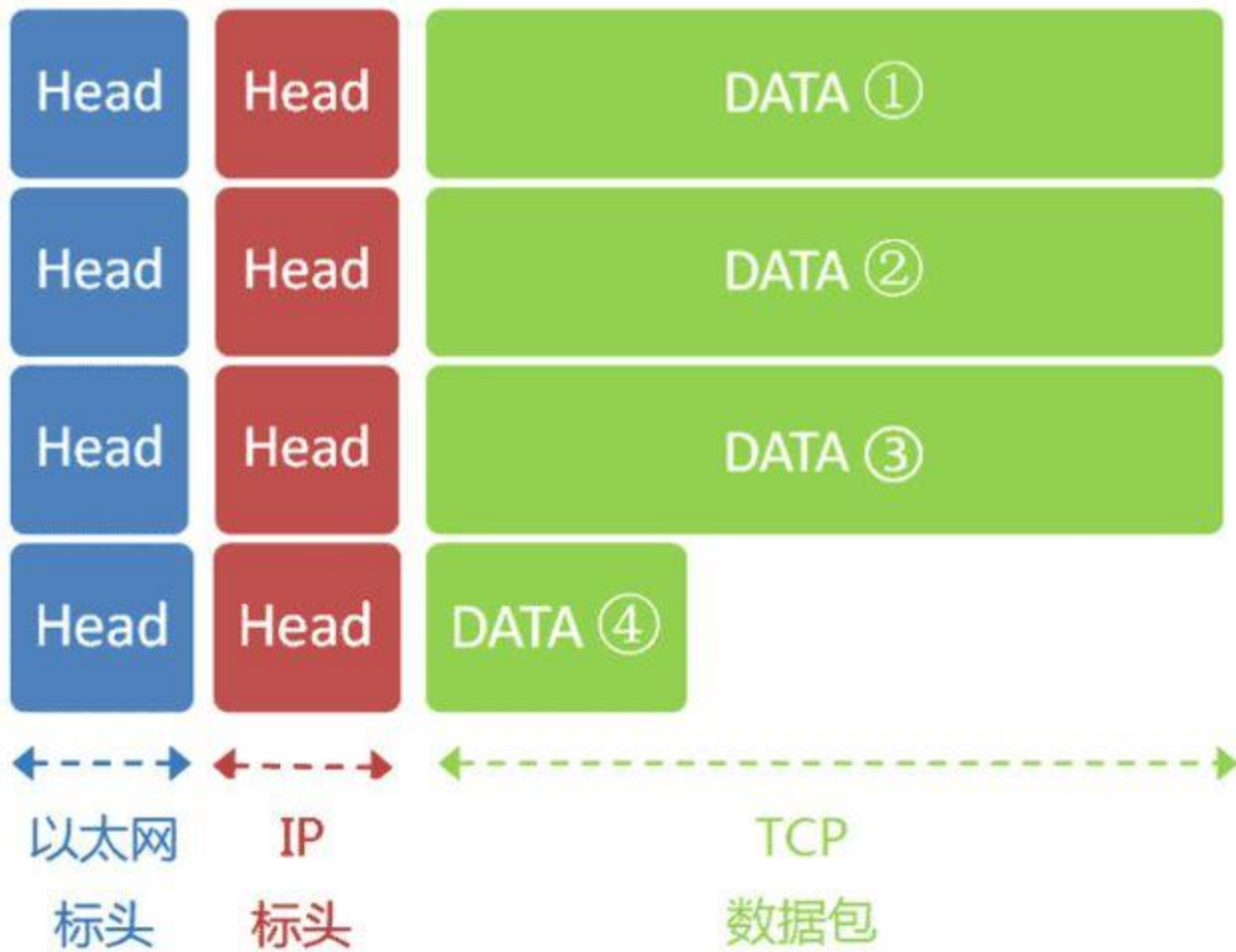
1 物理层

OSI 第二层



- 第二层是 数据链路层。它定义了数据的传输格式、可以在节点间流动的数据量大小、数据流动可以持续的时长，以及在流中检测到错误时应采取的措施。
- 使用更加正式的技术术语描述如下：
 - 线路规划。谁应该交流多久？节点传输信息的时间应该持续多久？
 - 流量控制。应该传输的数据量是多少？
 - 错误控制-检测和校正。从电尖峰脉冲到卑鄙的连接器，所有的数据传输方法都有可能出错。一旦第二层的技术告知网络管理员有关第二层问题或第一层的问题，系统管理员就能为后续几层纠正那些错误。第二层主要关心的是错误检测，而不是错误校正。

OSI 第二层



由于以太网数据包的数据部分，最大长度为1500字节，当IP包过大时，会分割下来，但是每个分割包的头部都一样

- 第二层是 数据链路层。它定义了数据的传输格式、可以在节点间流动的数据量大小、数据流动可以持续的时长，以及在流中检测到错误时应采取的措施。
- 第二层内有两个截然不同的子层：
 - 介质访问控制（MAC, Media Access Control）：MAC 子层负责分配硬件标识号，这个标识号被称为 MAC 地址，它能够唯一标识网络上的各个设备。两个设备不应该有相同的 MAC 地址。MAC 地址在硬件制造时就分配好了，位于网卡当中，大多数网络都会自动对其进行识别。交换机会跟踪网络上所有的 MAC 地址。在这里和这里了解更多有关 MAC 地址的信息，在这里进一步了解网络交换机。
 - 逻辑链路控制（LLC, Logical Link Control）：LLC 子层处理帧的寻址以及流量控制。速度取决于两个节点之间的链路，例如以太网或 Wifi。

OSI 第二层

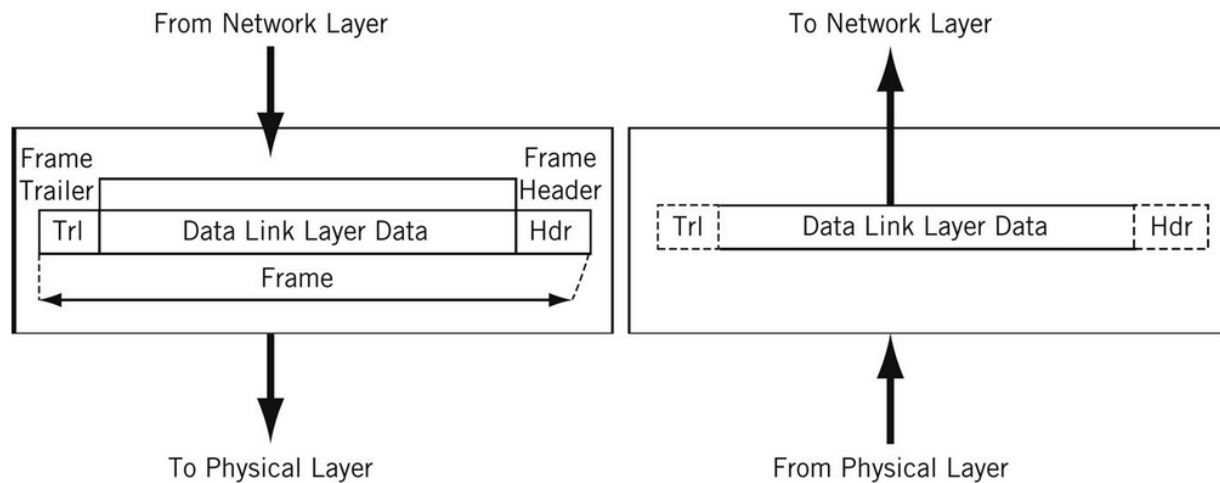
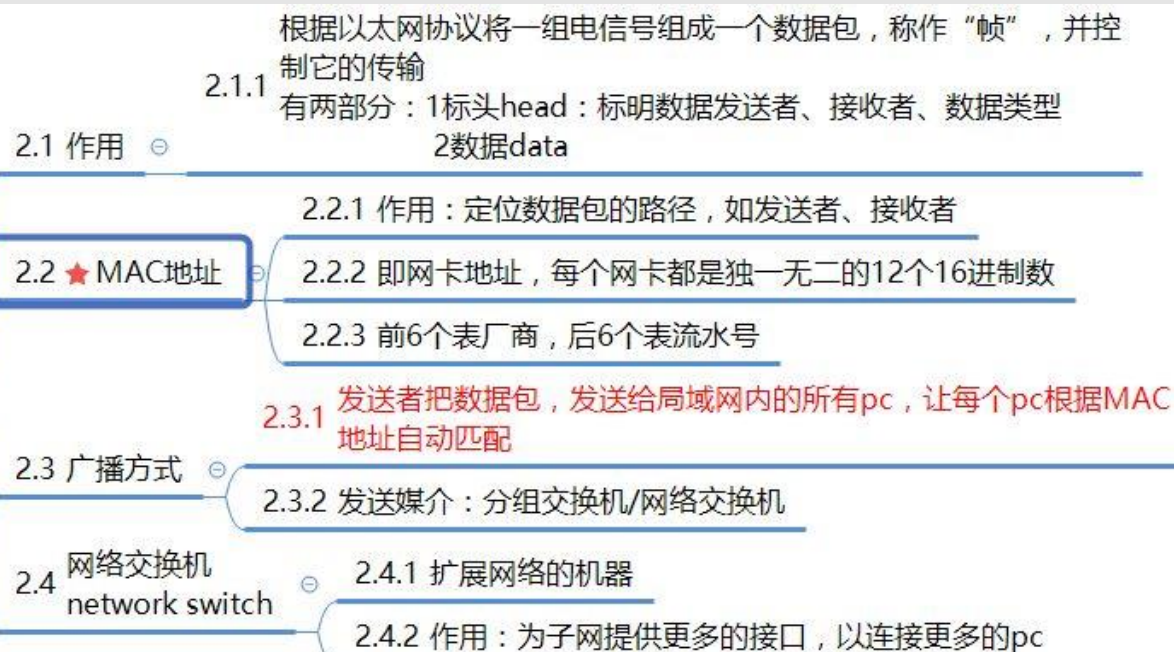


FIGURE 1.13 The data link layer, showing that data link layer frames have both header and trailer.

- 第二层的数据单元是 帧 (frame)。
- 每一帧都包括一个帧头、主体和一个帧尾：
 - 帧头：通常包括源节点和目的节点的 MAC 地址。
 - 主体：由要传输的比特组成。
 - 帧尾：包括错误检测信息。当检测到错误时，根据实现或网络的配置或协议，帧可能被丢弃，或者错误会被报告给上面的层，用于进一步错误校正。例如，错误检测机制的有循环冗余校验 (CRC, Cyclic Redundancy Check) 和帧校验序列 (FCS, Frame Check Sequence)。

OSI 第二层

2 数据链路层



- 如何排查 OSI 第二层中的问题
 - 可能在第一层上发生的所有问题
 - 两个节点间的连接（会话）不成功
 - 成功建立但又间歇性失败的会话
 - 帧冲突
- 数据链路层允许局域网内的各节点彼此相互通信。这一层建立了线路规划、流量控制和错误控制的基础。

OSI 第三层

Destination	Subnet mask	Interface
128.75.43.0	255.255.255.0	Eth0
128.75.43.0	255.255.255.128	Eth1
192.12.17.5	255.255.255.255	Eth3
default		Eth2

- 第三层是网络层。就是在网
络中，或节间通过路由器和信
息传递的。路由器是在网
络中，或节间通过路由器和信
息传递的。
- 路由器的主要功能——
它是由数据包组成的。数
据包是由数据包组成的。
- 路由器不仅通过连接到网络
服务提供商（ISPs，
Internet Service
Providers）提供因特网访问，
还跟踪着在网络中的每一个
（记住交换机的跟踪的是一个
网络中所有的 MAC 地址），及
它在这些网络中的数据包的
不同路径。
- 路由器将所有的地址和路由
信息都保存在路由表中。

OSI 第三层

Destination	Subnet mask	Interface
128.75.43.0	255.255.255.0	Eth0
128.75.43.0	255.255.255.128	Eth1
192.12.17.5	255.255.255.255	Eth3
default		Eth2

- 第三层的数据单元是 数据包 (data packet)。通常，每个数据包都包含一个帧 加上 一个 IP 地址信息的包装。换句话说，帧被第三层的地址信息封装了。
- 数据包中传输的数据有时也被称为 负载 (payload)。每个包都拥有到达目的地所需的一切，但是它能不能成功抵达就是另外一回事儿了。
- 第三层上的传输是无连接的、尽力而为的——除了将流量发往它应该去的地方，它们不会做任何事。更多与数据传输有关的协议在第四层。
- 节点一旦连接到因特网，它就会被赋予一个因特网协议 (IP, Internet Protocol) 地址，它看起来要么像 172.16.254.4 (IPv4 地址)，要么像 2001:0db8:85a3:0000:0000:8a2e:0370:7334 (IPv6 地址)。路由器在它们的路由表中使用 IP 地址。
- IP 地址通过地址解析协议 (ARP, Address Resolution Protocol) 与物理节点的 MAC 地址相关联，ARP 用节点对应的 IP 地址解析 MAC 地址。
- ARP 通常被认为是第二层的一部分，但是由于 IP 地址在第三层以下都不存在，所以 ARP 也是第三层的一部分。

OSI 第三层

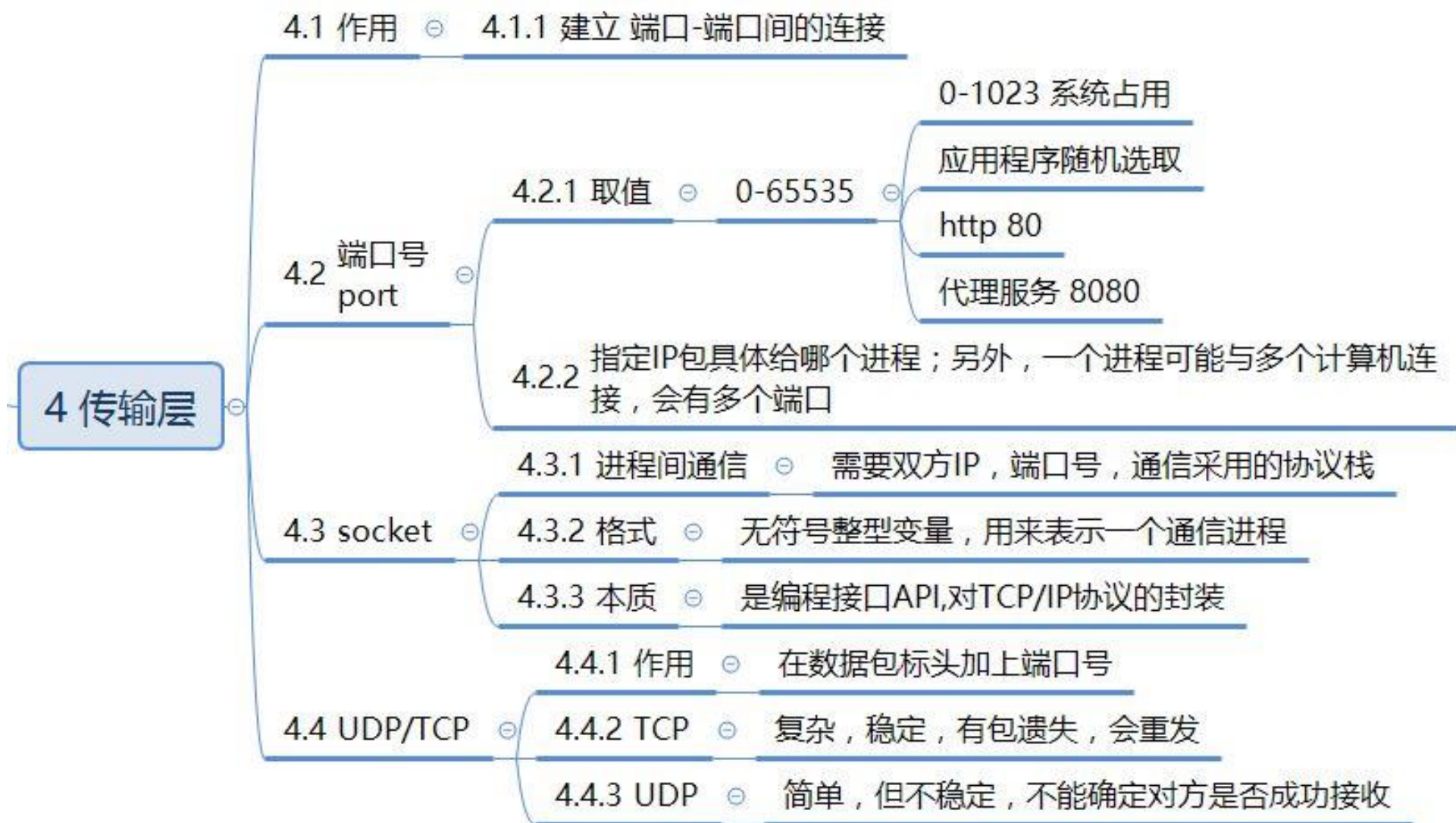
Destination	Subnet mask	Interface
128.75.43.0	255.255.255.0	Eth0
128.75.43.0	255.255.255.128	Eth1
192.12.17.5	255.255.255.255	Eth3
default		Eth2

- 如何排查第三层中的问题？这里是第三层中要当心的一些问题：
 - 所有可能在之前各层中出现的问题 :)
 - 路由器或其它节点故障或无功能
 - IP 地址配置不正确
- 很多第三层问题的答案都要求使用像 ping、trace、show ip route 或 show ip protocols 这样的命令行工具。在这里了解更多与有关一至三层问题排查的信息。
- 第三层允许节点连接到因特网并跨越不同网络发送数据。

3 网络层

- 3.1 作用
 - 3.1.1 建立 主机-主机的连接
- 3.2 引入新的地址模式-IP地址/网络地址，可以区分哪些pc是在同一子网（局域网）内
- 3.3 IP类型分类
 - 3.3.1 IPv4
 - 32个二进制，4字节*8位
 - 前24位表网络
 - 同一子网下，必须相同
 - 后8位表主机
 - 3.3.2 IPv6
 - 128个二进制，8字节*16位
- 3.4 子网掩码 subnet mask
 - 3.4.1 通过and运算判断是否在同一子网下
ip and 255.255.255.0
- 3.5 路由 routing
 - 3.5.1 作用
 - 通过网络把数据从原地址到目标地址
 - 引导分组传送，经过一些中间节点后，到达目的地
 - 3.5.2 实现方法
 - 定义一条路径，经过因特网把包发送到目的地，但不指定完全路径，只定义两个网关之间的路径段
 - 3.5.3 路由器 router
 - 连接两个或多个网络并实现路由功能的机器；可看作配有多个网卡的专用电脑，让网卡接入不同的网络中
 - 3.5.4 网关 gateway
 - 网络层使用的路由器，通常指路由器的IP
 - 3.5.5
 - ex:A:发送地址 B：接收地址
 - 1.若在同一局域网：通过广播方式即可找到
 - 2.若在不同局域网：
 - a. A先将包根据网关添加路由器/主机地址，通过交换机的广播方式发给主机
 - b. A的主机将数据包传送给B所在的主机，再由主机根据MAC广播给B 完成
 - 注：IP包不断被路由封装和拆开（添加/删除地址）
- 3.6 ARP协议
 - 3.6.1 作用
 - 局域网内IP和MAC地址的对应关系
 - 3.6.2 作用位置
 - 介于数据链路层和网络层之间
 - 3.6.3 局限
 - 仅限于ipv4，ipv6用Neighbor Discovery Protocol替代

OSI 第四层



OSI 第四层

- 第四层是 传输层。在这一层，会深入探讨了两个节点之间连接的具体细节，以及信息是如何在它们之间进行传输的。第四层建立在第二层的功能之上——线路规划、流量控制和错误控制。
- 这一层也负责数据包的分段，或者说数据包如何被拆分成小片并发往整个网络。
- 不像上一层，第四层也理解整个消息，而不只是每个独立的数据包的内容。根据对整个消息的理解，第四层不再一次性发送所有数据包，从而管理网络拥塞。
- 第四层的数据单元有好几个不同的名字，对于 TCP 而言，数据单元是数据包。对于 UDP 而言，包被称为数据报（datagram）。
- 第四层中最有名的两个协议是传输控制协议（TCP, Transmission Control Protocol）和用户数据报协议（UDP, User Datagram Protocol）。

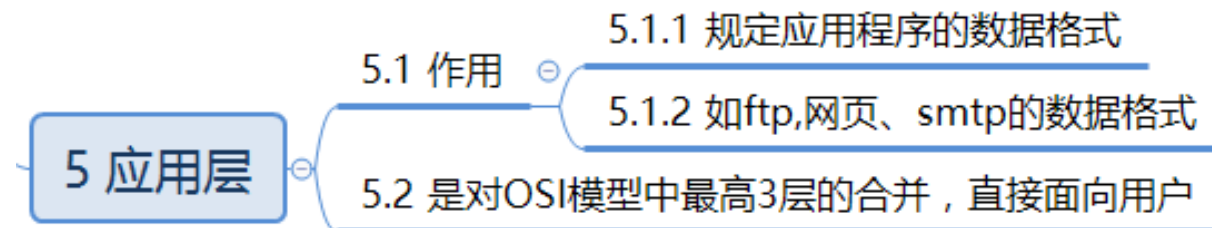
OSI 第四层

- TCP 是一个面向连接的协议，优先保证的是数据的质量而不是速度。
- TCP 显式地与目的节点建立连接，并要求在数据传输时进行源节点与目的节点之间的握手操作。握手能够确认数据已经被接收。如果目的节点没有收到所有的数据，TCP 就会要求进行重传。
- TCP 也会确保数据包以正确的顺序交付或者重组。从这里了解更多有关 TCP 的信息。
- UDP 是一个无连接的协议，优先保证速度而不是数据的质量。UDP 不要求进行握手，这也正是它被称为无连接的原因。
- 因为 UDP 不必等待确认，所以它可以以更快的速度发送数据，但并非所有的数据都能成功传输，我们也不会知道哪些数据传输失败了。
- 如果信息被拆分成多个数据报，除非这些数据报都包含一个序列号，否则 UDP 无法确保以正确的顺序重组数据包。
- TCP 和 UDP 都将数据发往网络设备上的特定端口，这些网络设备都有自己的 IP 地址。IP 地址和端口号的组合被称为套接字（socket）。

OSI 第四层

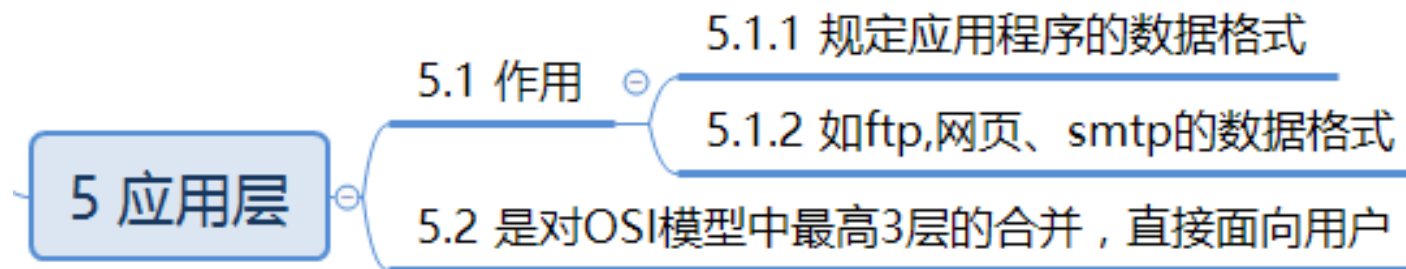
- 如何排查 OSI 第四层中的问题？这里是第四层中要当心的一些问题：
 - 所有可能在之前各层中出现的问题 :)
 - 被封锁的端口——检查你的访问控制列表（ACL, Access Control List）和防火墙
 - 服务质量（QoS, Quality of Service）设置。QoS 是路由器/交换机的一个功能，可以对流量进行优先级排序，并且它们真的可以把事情搞砸。
- 传输层通过将消息分割成多个数据包提供端到端的消息传输，支持面向连接的和无连接的通信。

OSI 第五层



- 第五层是 会话层，负责建立、维持和终止会话。
- 会话建立在两个网络应用之间，是双方商定好的连接。第五层不需要保留节点的概念，因为它是之前各层抽象出来的（关心）的概念。
- 所以会话是一个建立在两个特定的用户应用之间的连接，其中有一些重要的概念需要考虑：
 - 客户端与服务器模型：请求信息的应用被称为客户端，拥有被请求信息的应用被称为服务器。
 - 请求与响应模型：在建立会话的过程和会话期间，不断有来回的信息请求，还有包含被请求信息的响应或者是“嘿，我没有你要的东西”。
- 会话持续的时间可以非常短，也可以非常长，有时会话也可能会失败。
- 根据所采用的协议，会话可能会启动各种故障解决程序。根据所使用的应用程序/协议/硬件，会话可能支持单工，半双工或全双工模式。
- 第五层中协议的例子有网络基本输入输出系统（NetBIOS, Network Basic Input Output System）和远程过程调用协议（RPC, Remote Procedure Call Protocol）等等。
- 从这里往上（第五层及以上），网络关注的是与用户应用程序建立连接以及如何向用户展示数据。

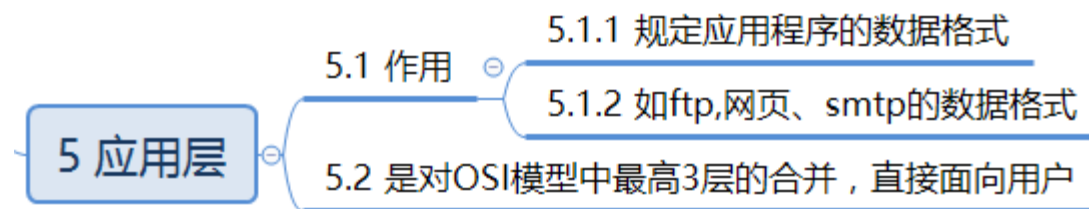
OSI 第五层



- 如何诊断 OSI 第五层中的问题？这里是第五层中需要当心的一些问题：
 - 服务器不可用
 - 服务器未被正确地配置，例如 Apache 或 PHP 配置
 - 会话故障——断连、超时，等等
- 会话层负责初始化、维持并终止两个用户应用程序之间的连接。它响应来自表示层的请求，并向传输层发起请求。

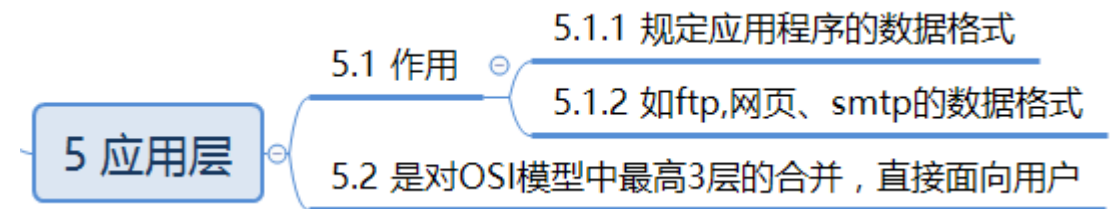
OSI 第六层

- 第六层是 表示层，负责数据的格式，比如字符编码与转换，以及数据加密。
- 托管用户应用程序的操作系统通常包含第六层中的程序，这个功能并不总是被网络协议实现。
- 第六层确保第七层中的用户程序可以成功地消费数据，当然还有最终数据的展示。
- 有三种数据格式化方法需要注意：
 - 美国信息交换标准代码（ASCII, American Standard Code for Information Interchange）：这个七位编码技术是字符编码中使用最广泛的标准。ASCII 的一个超集是 ISO-885901，它提供了西欧语言所必需的大多数字符。
 - 扩充的二进制编码的十进制交换码（EBDCIC, Extended Binary-Coded Decimal Interchange Code）：由 IBM 设计，用于大型机。此编码与其他字符编码方法不兼容。
 - 万国码（Unicode）：可以使用 32 位，16 位或 8 位字符的字符编码，它尝试容纳所有已知的字母。
- 如何诊断 OSI 第六层中的问题
 - 驱动程序不存在或损坏
 - 操作系统用户访问级别不正确
- 表示层负责格式化与加密数据

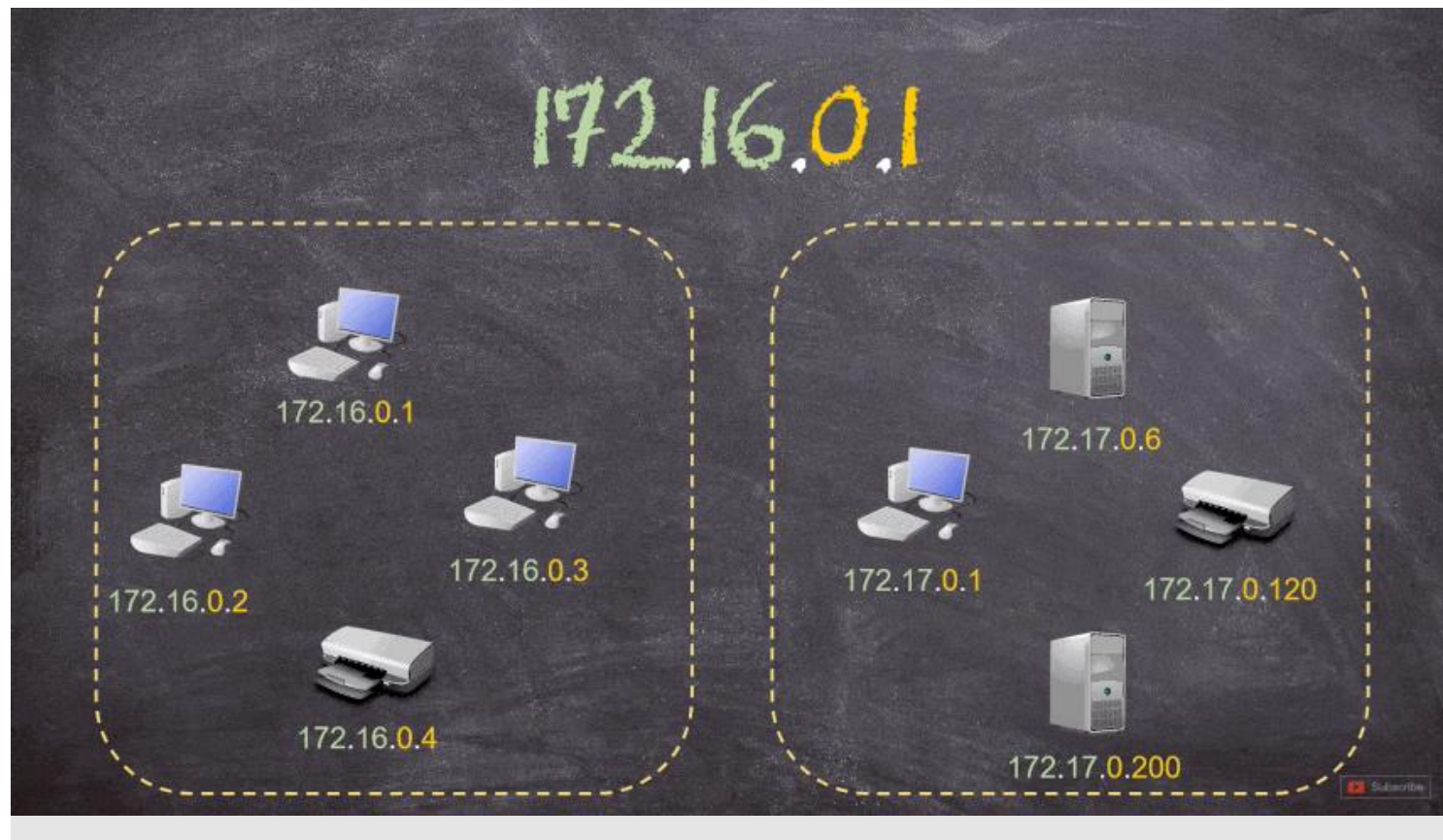


OSI 第七层 应用层。

- 第七层是 顾名思义，这一层最终负责支持用户程序使用的服务。应用程序包括安装在操作系统中的软件程序，比如因特网浏览器（例如 Firefox）或文字处理程序（例如 Microsoft Word）。
- 应用程序可以在后台执行专门的网络功能，也可以要求第七层中专门的服务。
- 例如专门创建电子邮件程序，它在网络上运行并利用第七层中网络功能（比如电子邮件协议）。
- 应用程序也可以控制用户交互，比如安全检查（例如 MFA）、识别两名参与者的身份、初始化信息交换等。
- 这一层中运行的协议包括文件传输协议（FTP, File Transfer Protocol）、安全壳协议（SSH, Secure Shell）、简单邮件传输协议（SMTP, Simple Mail Transfer Protocol）、因特网消息访问协议（IMAP, Internet Message Access Protocol）、域名服务（DNS, Domain Name Service）和超文本传输协议（HTTP, Hypertext Transfer Protocol）。
- 虽然这些协议中的每一个都服务于不同的功能，运行的方式也各不相同，但从较高的层次看，它们都促进了信息的交流。
- 如何诊断 OSI 第七层中的协议
 - 所有之前各层中的问题
 - 软件应用程序配置不正确
 - 用户操作失误（我们都遇到过……）
- 应用层拥有用户应用程序运行所需的服务和功能，不包括应用程序本身。



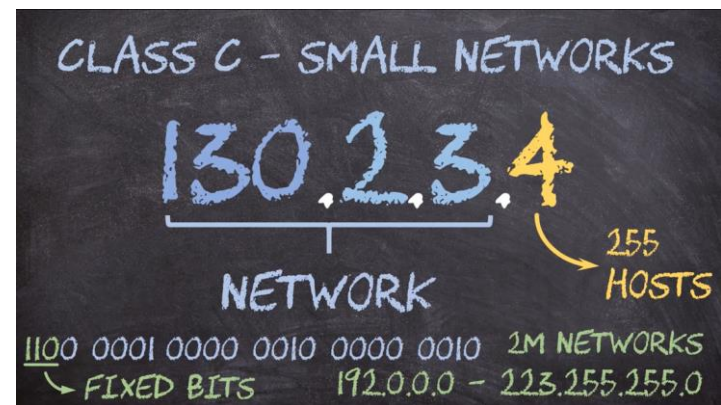
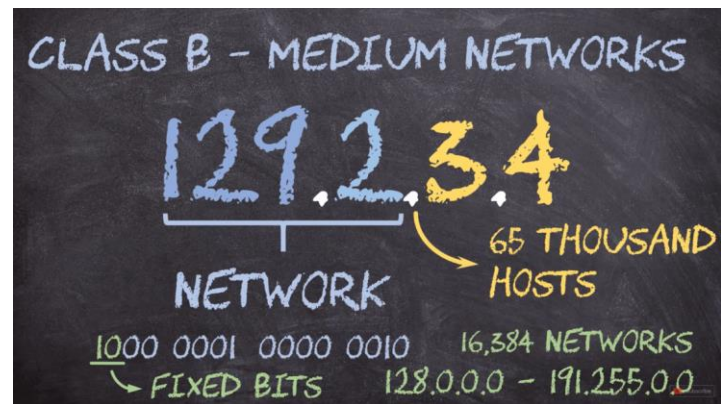
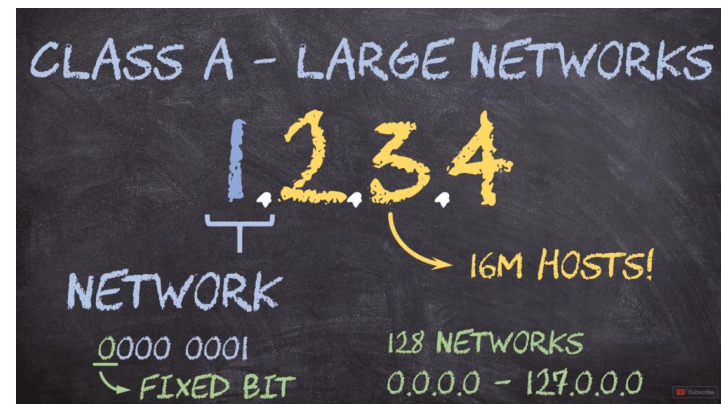
補充知識 IP



- IP 地址可以代表「两个地址」，一个是 Host Address 也就是主机地址，另一个是 Network Address 也就是网络地址
- 我们拿 172.16.0.1 这个 IP 地址举例：其中 172.16 是网络地址，而后面的 0.1 是主机地址
- 观察上图，我们可以发现，左侧以 172.16 开头的主机在同一个网络中，右侧以 172.17 开头的主机在另一个网络中。这是两个不同的网络，它们之间不能直接进行通信！但是在同一网络中的设备可以进行通信，如果你有新的设备加入，只需要将地址设置为 172.16 或 172.17 即可。如果想要实现两个网段之间的通信，则需要借助路由器来实现。

補充知識 IP

- A 类 IP 地址第一个八位位组是网络空间，后三个八位位组是主机空间；虽然拥有很少的网络空间，但是却拥有庞大的主机空间。这里需要注意的是，A 类地址八位位组中的第一位始终为 0 固定位，剩下的 7 位可供我们进行分配，因此，我们只有 0~127 即 128 个网络空间可供分配，网络空间从 0.0.0.0 ~ 127.0.0.0。
- 另外，由于第一个八位位组以 0 和 127 开头的网络空间是要用作保留的，所以实际可供分配网络空间为 126 个，其网络空间 IP 地址为 1.0.0.0 ~ 126.0.0.0。每个空间可供分配主机为 16777216 台（ $2^{24}=16777216$ ）。
- B 类网络适用于中等数量主机的情况，前两个八位位组用于网络空间划分，后面两个则用于主机地址划分，网络的前两个位始终为 1 和 0（10000000）即 128，可供我们设置的有 14 位，也就是 16384 个网络空间（ $2^{14}=16384$ ），其网络空间 IP 为 128.0.0.0 ~ 191.255.0.0，每个网络空间可以拥有 65536 台主机（ $2^{16}=65536$ ）。
- C 类网络属于小型网络，这类网络 IP 地址的前三个数字为网络地址，最后一个数字是主机地址，且第一个八位位组的前三位固定位 110（11000000）即 192，可供我们设置的有 21 位，也就是 2097152 个网络空间（ $2^{21}=2097152$ ），其网络空间 IP 为 192.0.0.0 ~ 223.255.255.0，每个网络空间可以有 256（ 2^8 ）台主机。



練習

- 请问下面的 IP 分别属于第几类 IP 地址，请将答案写在 IP 地址后方。

- 9.4.3.47
- 203.42.62.1
- 103.88.77.22
- 151.10.13.55
- 222.127.16.4

- 请问下面的IP地址哪一部分是主机地址？哪一部分是网络地址？

- 9.4.3.47
- 203.42.62.1
- 103.88.77.22
- 151.10.13.55
- 222.127.16.4

補充知識 IP-子網掩碼

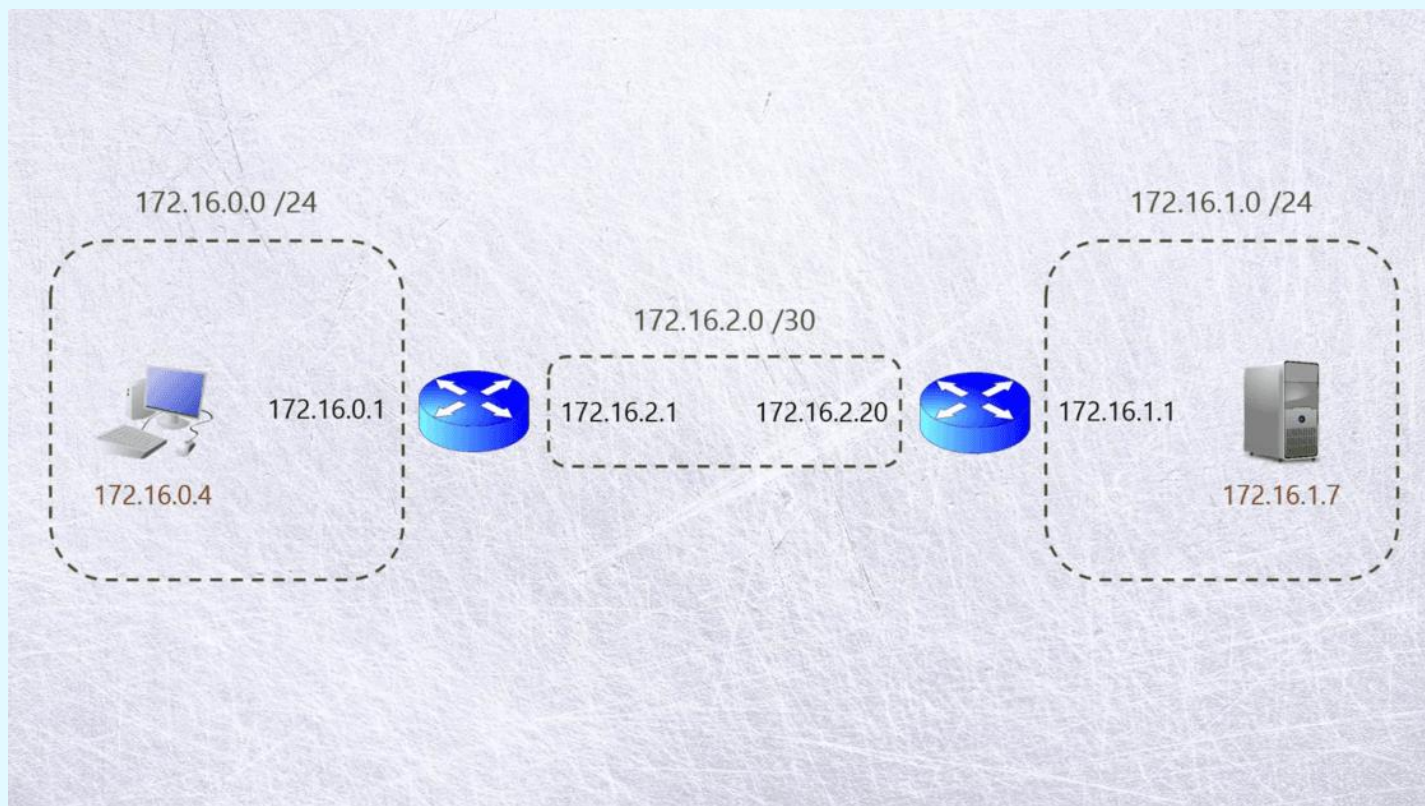
172.16.3.4

- 子网掩码
- 子网掩码同样由 4 个八位位组组成，与 IP 地址如出一辙；如图，255.255.0.0 就是 172.16.3.4 的子网掩码。
- 子网掩码的每一位都与 IP 地址对齐，例如 172 与第一个 255 对齐，16 与第二个 255 对齐，通过图中我们可以发现，子网掩码设置为 255 的位，即为网络地址，设置位 0 的位，即为主机地址。
- 我们可以通过子网掩码很快的得知 IP 地址属于哪一类网络。
- 若子网掩码设置为 255.0.0.0 则为 A 类网络；255.255.0.0 则为 B 类网络；255.255.255.0 则为 C 类网络。

練習

- 如果，我们拥有一个 B 类地址，172.16.0.0，子网掩码为 255.255.0.0，我们使用子网划分将子网掩码设置为 255.255.240.0，请问：
- 问：我们此时拥有多少个子网空间？
 - 答：16 个，172.16.0.0 打开了八位位组的前 16 位，而 (255.255.240.0) 打开了八位位组的前 20 位， $20-16=4$ 位，这是子网的另外 4 位，因此 2 的 4 次方=16，因此，我们此时拥有16个子网空间。
- 问：每个子网空间可以划分多少 IP 地址？（例如：176.16.0.1 ~176.16.15.255）
 - 答：每个子网有 $16 * 256 = 4096$ 个 IP 地址。

練習



- 问：为什么网段 A 中的设备无法通过路由器与网段 B 进行通信？
- 答：问题的重点在中间路由器所在的网段，通过 `172.16.2.0/30` 得知，八位位组的前30位都打开了，因此只留下 2 位允许划分空间，根据 $2^2=4$ 可以确定该网段之划分了 4 个 IP 地址 `172.16.2.0~172.16.2.3`。右侧的路由器 IP 为 `172.16.2.20`，很明显不在正确范围内

補充知識 双绞线

线缆名称	线缆介绍	线缆速率
CAT-1	以往用在传统电话的网络线路	无
CAT-2	以往用在令牌环网络	4 Mbit/s
CAT-3	提供16MHz的带宽，曾经常用在 10 Mbit/s 以太网网络	10 Mbit/s
CAT-4	提供20MHz的带宽，曾经常用在 16 Mbit/s 的令牌环网	16 Mbit/s
CAT-5	提供100MHz的带宽，目前常用在快速以太网（100 Mbit/s）中	100 Mbit/s
CAT-5e	提供125MHz的带宽，目前常用在快速以太网及千兆以太网（1000Mbit/s）中	1000 Mbit/s
CAT-6	提供250MHz的带宽，比CAT-5与CAT-5e高出一倍半	2500 Mbit/s
CAT-6A	提供500MHz的带宽，使用在万兆以太网（10 Gbit/s）中	10 Gbit/s
CAT-7	设计供以600MHz频率传输信号	未定



Cat5e



Cat6



Cat6a



Cat7

- 双绞线
- 我们常见的网线都是由四对电线组成的，而在早期的标准中，其线缆并不一定是由四对组成的（例如：10BASE-T和100BASE-T都是由两对绞线组成），但是如果想要让传输速度达到1Gbps或10Gbps，就必须使用全部四对电线才可以。