第 4 篇 高级 TCP/IP 知识

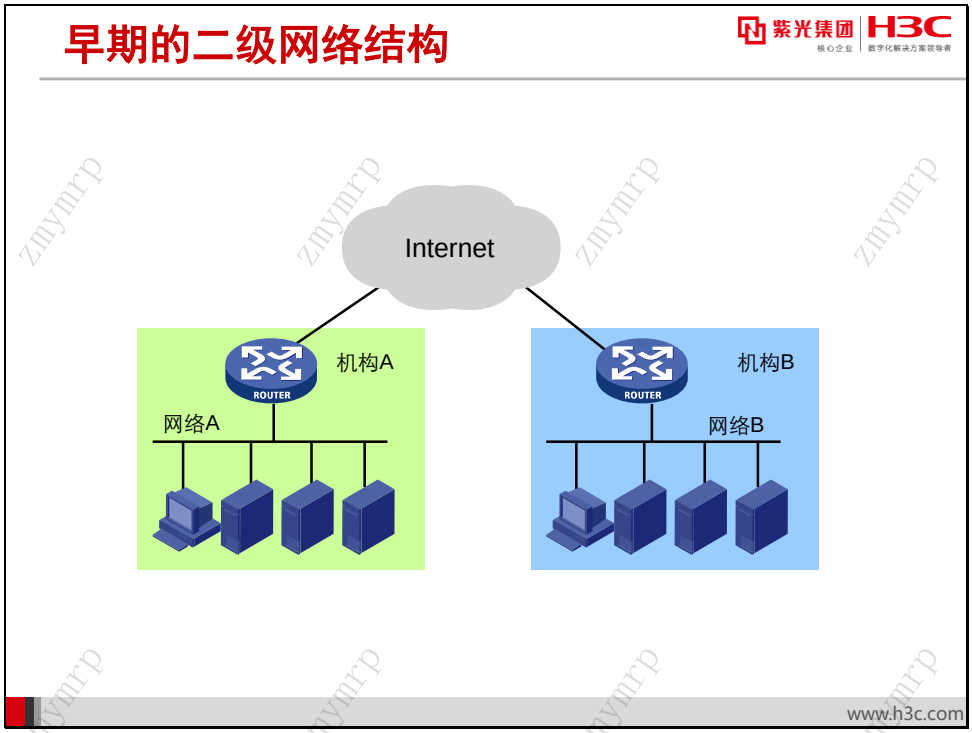
第16章 IP 子网划分

TCP/IP 网络用 IP 地址来标识各个节点，并且根据 IP 地址的类别（Class）进行 IP 地址分配。这种地址分配方法简单易用，但随着 Internet 容量及业务量的急速增长，这种方法表现出越来越多的弊端。

为了解决分类 IP 地址划分带来的地址浪费，就需要使用子网划分（Subnetting）的方法对地址进行有效利用。VLSM（Variable Length Subnet Mask，变长子网掩码）和 CIDR（Classless Inter-Domain Routing，无类域间路由）则可以进一步提高地址利用效率，而缓解地址数量不足的问题。在这几种技术中，子网划分是理解 VLSM 和 CIDR 的重要基础。本章后续将重点介绍子网划分的相关知识，并介绍 VLSM 及 CIDR 的基本概念。

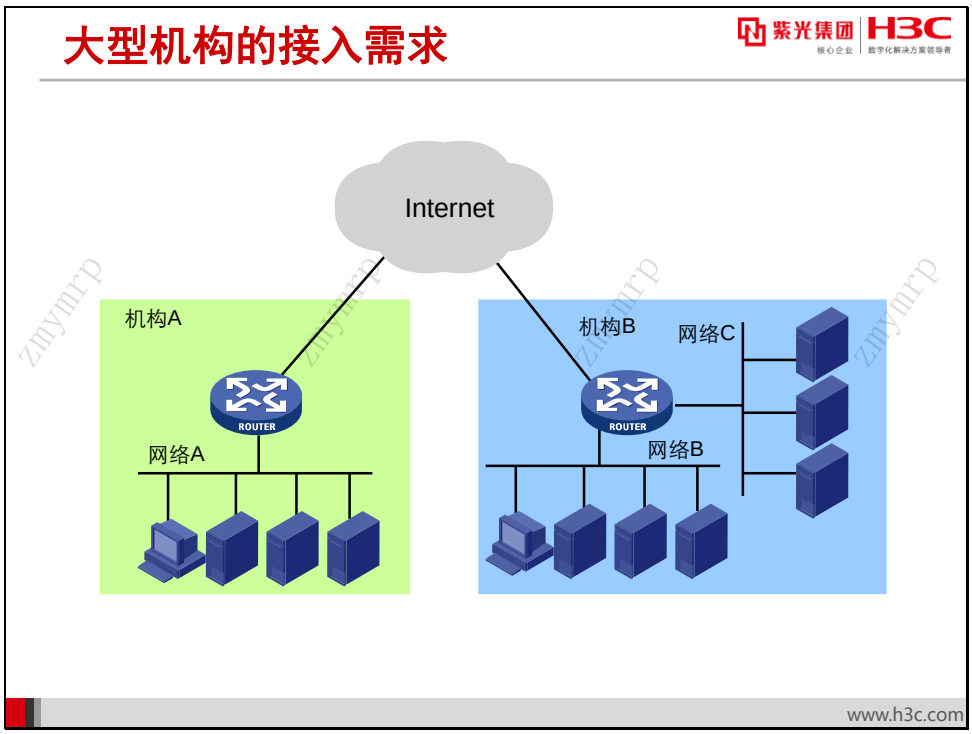
16.1 本章目标

16.2 IP子网划分的需求背景



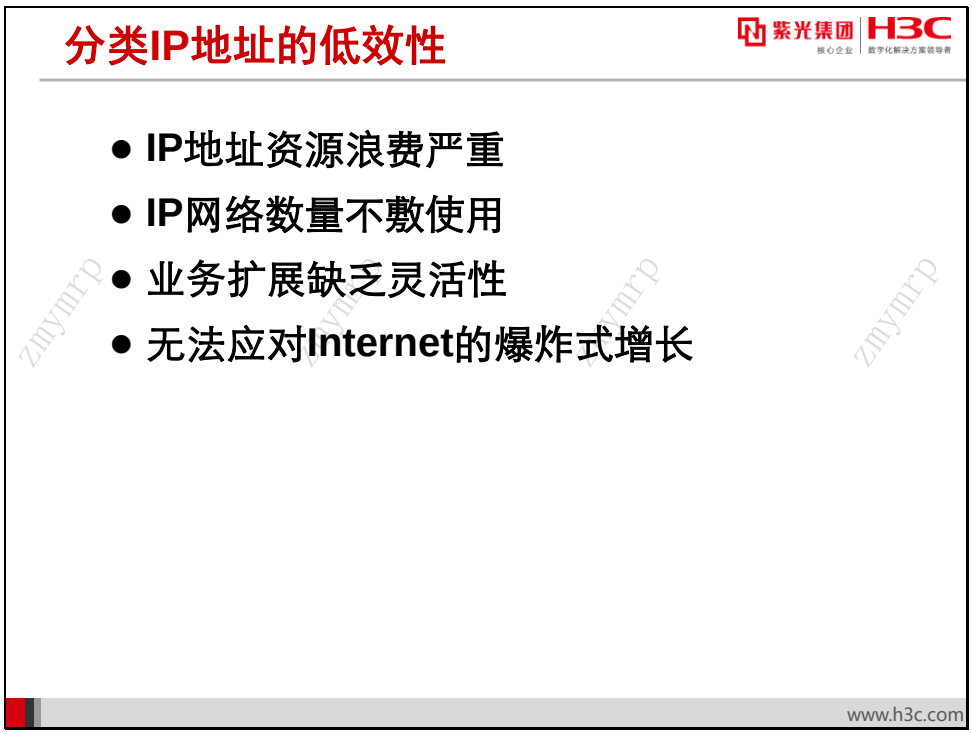
早期的 Internet 是一个简单的二级网络结构。接入 Internet 的机构由一个物理网络构成，该物理网络包括机构中需要接入 Internet 网络的全部主机。

自然分类法将 IP 地址划分为 A、B、C、D、E 类。每个 32 位的 IP 地址都被划分为由网络号和主机号构成的二级结构。为每个机构分配一个按照自然分类法得到的 Internet 网络地址，能够很好地适应满足当时的网络结构。



随着时间的推移，网络计算逐渐成熟，网络的优势被许多大型组织所认知，Internet 中出现了很多大型的接入机构。这些机构中需要接入的主机数量众多，单一物理网络容纳主机的数量有限，因此在同一机构内部需要划分多个物理网络。

早期解决这类大型机构接入 Internet 的方法是为机构内的每一个物理网络划分一个逻辑网络，即对每一个物理网络都分配一个按照自然分类法得到的 Internet 网络地址。



但这种“物理网络－自然分类 IP 网段”的对应分配方法存在严重问题：

- IP 地址资源浪费严重

举例来说，一个公司只有 1 个物理网络，其中需要 300 个 IP 地址。一个 C 类地址能提供 254 个主机 IP 地址，不满足需要，因此需要使用一个 B 类地址。1 个 B 类网络能提供 65534 个 IP 地址，网络中的地址得不到充分利用，大量的 IP 地址被浪费。

- IP 网络数量不敷使用

举例来说，一个公司拥有 100 个物理网络，每个网络只需要 10 个 IP 地址。虽然需要的地址量仅有 1000 个，但该公司仍然需要 100 个 C 类网络。很多机构都面临类似问题，其结果是，在 IP 地址被大量浪费的同时，IP 网络数量却不能满足 Internet 的发展需要。

- 业务扩展缺乏灵活性

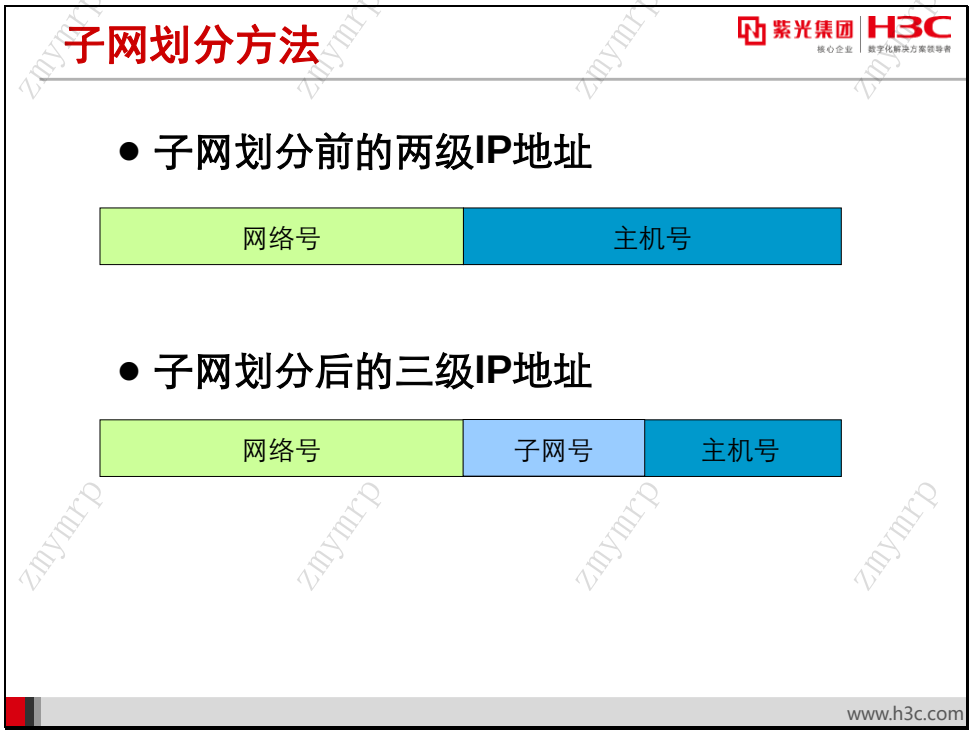
举例来说，一个公司拥有 1 个 C 类网络，其中只有 10 个地址被使用。该公司需要增加一个物理网络，就需要向 IANA 申请一个新的 C 类网络，在得到这个合法的 Internet 网络地址前，他们就无法部署这个网络接入 Internet。这显然无法满足企业发展的灵活性需求。

综上所述，仅依靠自然分类的 IP 地址分配方案，对 IP 地址进行简单的两层划分，无法应对 Internet 的爆炸式增长。

16.3 IP子网划分基础知识

上世纪 80 年代中期，IETF 在 RFC950 和 RFC917 中针对简单的两层结构 IP 地址所带来的日趋严重的问题提出了解决方法。这个方法称为子网划分（Subnetting）。

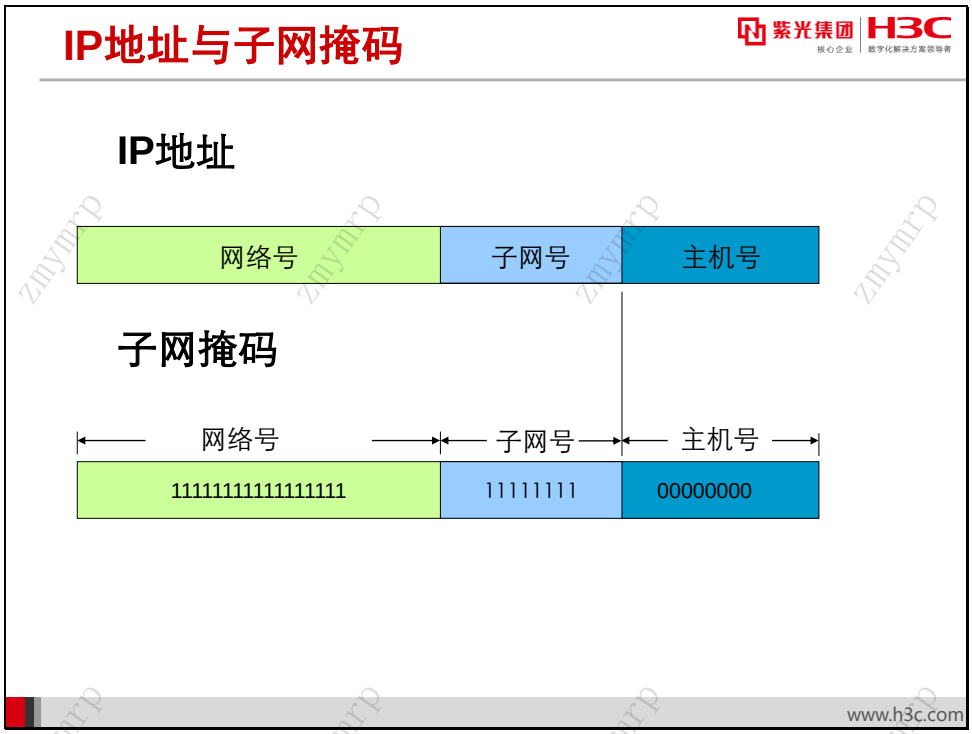
16.3.1 IP 子网划分的方法



普通两级结构的 IP 地址由网络号（network-number）和主机号（host-number）组成。划分子网的方法是从主机号（host-number）部分借用若干位作为子网号（subnet-number），剩余的位作为主机号（host-number）。于是两级的 IP 地址就变为三级的 IP 地址，包括网络号（network-number）、子网号（subnet-number）和主机号（host-number）。这样，拥有多个物理网络的机构可以将所属的物理网络划分为若干个子网。

子网划分属于本机构的内部事务。外部网络可以不必了解机构内由多少个子网组成，因为这个机构对外仍表现为一个没有划分子网的网络。从其他网络发送给本机构某个主机的数据，可以仍然根据原来的选路规则发送到本机构连接外部网络的路由器上。此路由器接收到 IP 数据包后再按网络号及子网号找到目的子网，将 IP 数据包交付给目的主机。要求路由器具备识别子网的能力。

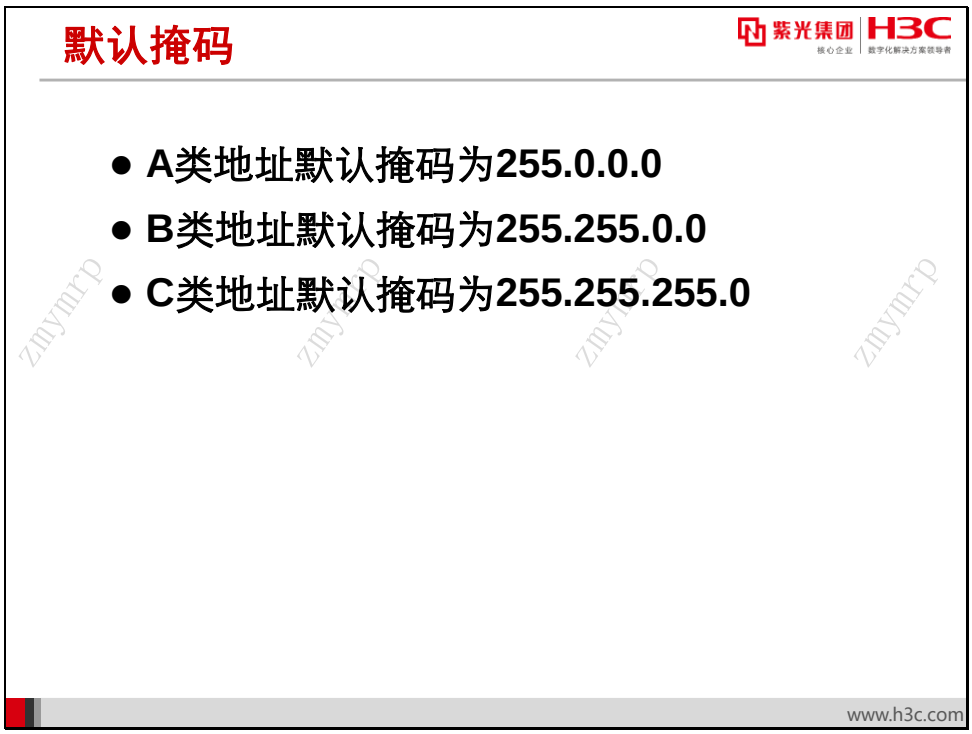
16.3.2 子网掩码



只根据 IP 地址本身无法确定子网号的长度。为了把主机号与子网号区分开，就必须使用子网掩码（subnet mask）。

子网掩码和 IP 地址一样都是 32 位长度，由一串二进制 1 和跟随的一串二进制 0 组成。子网掩码可以用点分十进制方式表示。与子网掩码中的 1 对应于 IP 地址中的网络号和子网号，子网掩码中的 0 对应于 IP 地址中的主机号。

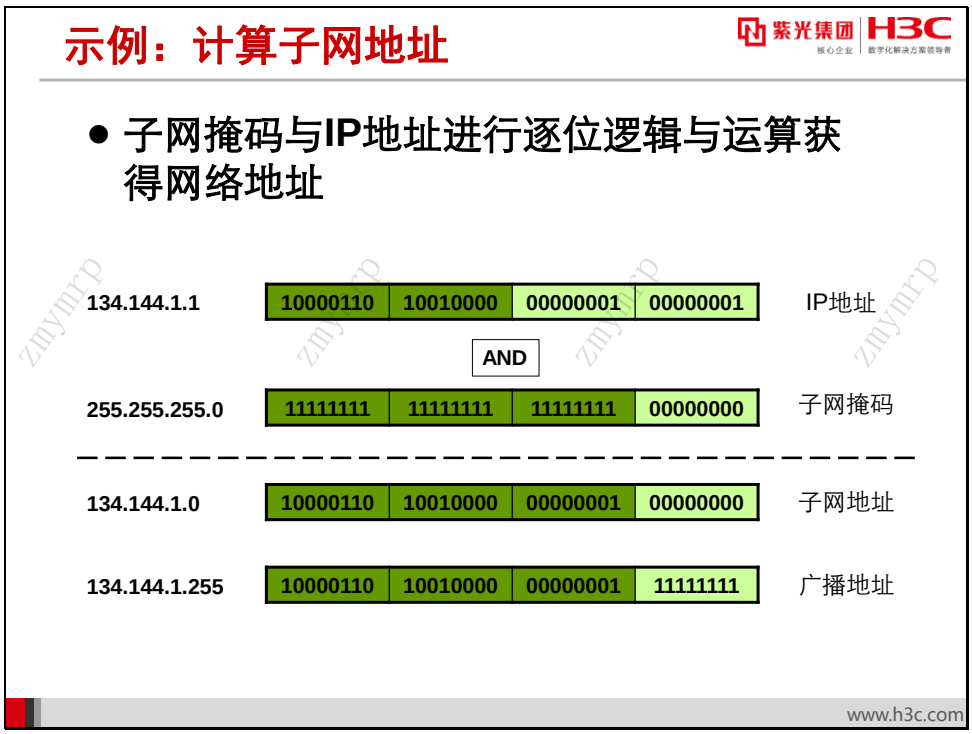
将子网掩码和 IP 地址进行逐位逻辑与运算，就能得出该 IP 地址的子网地址。



事实上，所有的网络都必须有一个掩码（address mask）。如果一个网络没有划分子网，那么该网络使用默认掩码：

- A 类地址的默认掩码为 255.0.0.0
- B 类地址的默认掩码为 255.255.0.0
- C 类地址的默认掩码为 255.255.255.0

将默认子网掩码和不划分子网的 IP 地址进行逐位逻辑与运算，就能得出该 IP 地址的网络地址。



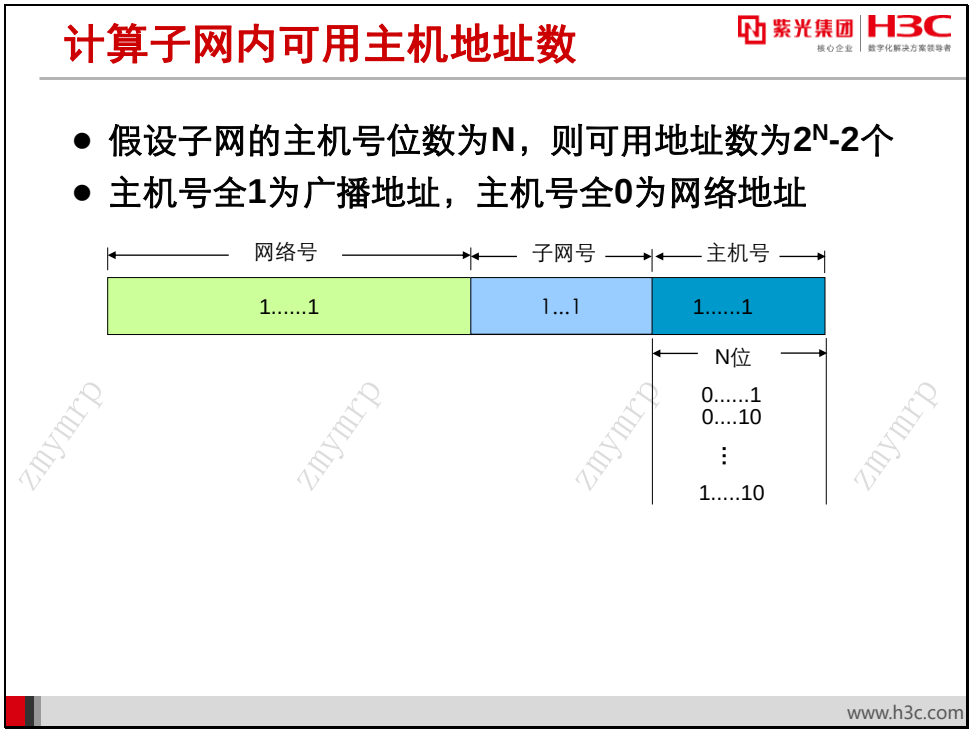
与普通掩码一样，通过子网掩码可以计算网络地址。将子网掩码和 IP 地址逐位进行逻辑与（AND）运算，计算的结果就是网络地址，在划分子网的情况下也称为子网地址。将子网地址的主机号全置位为 1，即可得到该子网的广播地址。

例如在图中，IP 地址 134.144.1.1 与子网掩码 255.255.255.0 进行与运算，得到其子网地址为 134.144.1.0。将主机号全置位为 1，得到该子网的广播地址为 134.144.1.255。

16.4 IP子网划分的常用计算

由于子网划分的出现，使得原本简单的 IP 地址规划和分配工作变得复杂起来。作为一个网络人员，你必须应该清楚的知道如何对你的网络进行子网划分，才能在满足网络应用需求的前提下合理高效地利用你手中的 IP 地址资源进行网络规划。

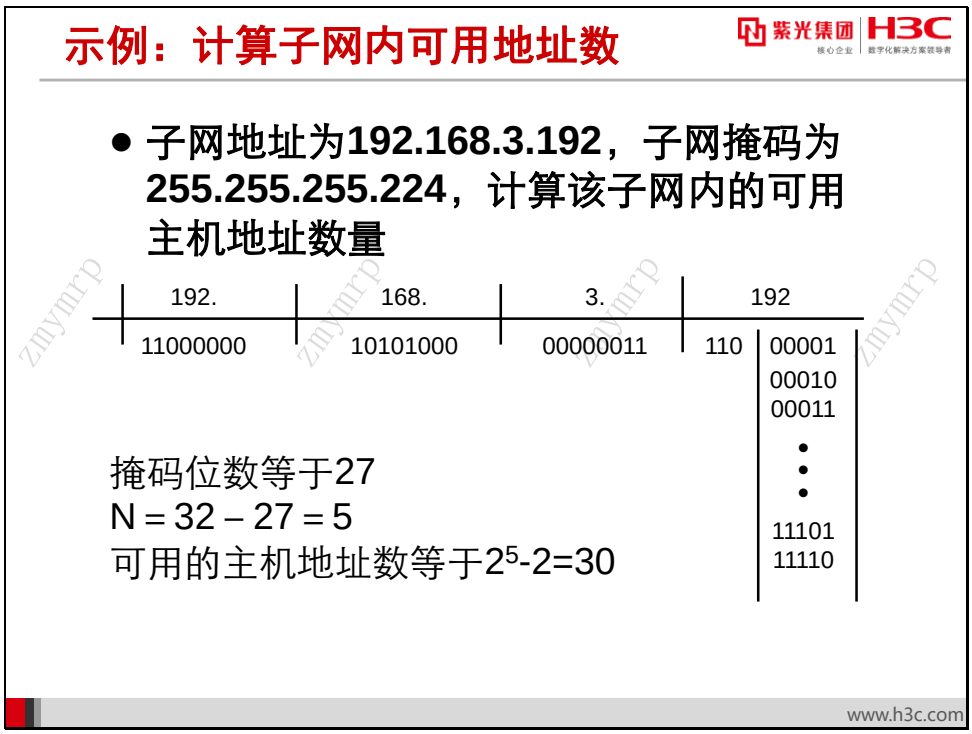
16.4.1 计算子网内可用地址数



计算子网内的可用主机数是子网划分计算中比较简单的一类问题，与计算 A、B、C 三类网络可用主机数的方法相同。

如果子网的主机号位数为 N bits，那么该子网中可用的主机数目为 2^N-2 个。减 2 是因为有两个主机地址不可用，即主机号为全为 0 和全 1。当主机号为全 0 时，表示该子网的网络地址；当主机号全为 1 时，表示该子网的广播地址。

要完成相关子网划分问题的计算，需要熟记 2 的 n 次幂的结果。因为计算过程中经常会进行二进制数与十进制数的相互转换，如果熟记这些结果的话将大大提高计算的速度。一般来讲熟记 2 的 1 到 10 次幂的结果在大多数的计算问题中就足够用了。



已知一个 C 类网络划分成子网后为 192.168.3.192，子网掩码为 255.255.255.224，计算该子网内可供分配的主机地址数量。

要计算可供分配的主机数量，就必须要知道主机号的位数。计算过程如下：

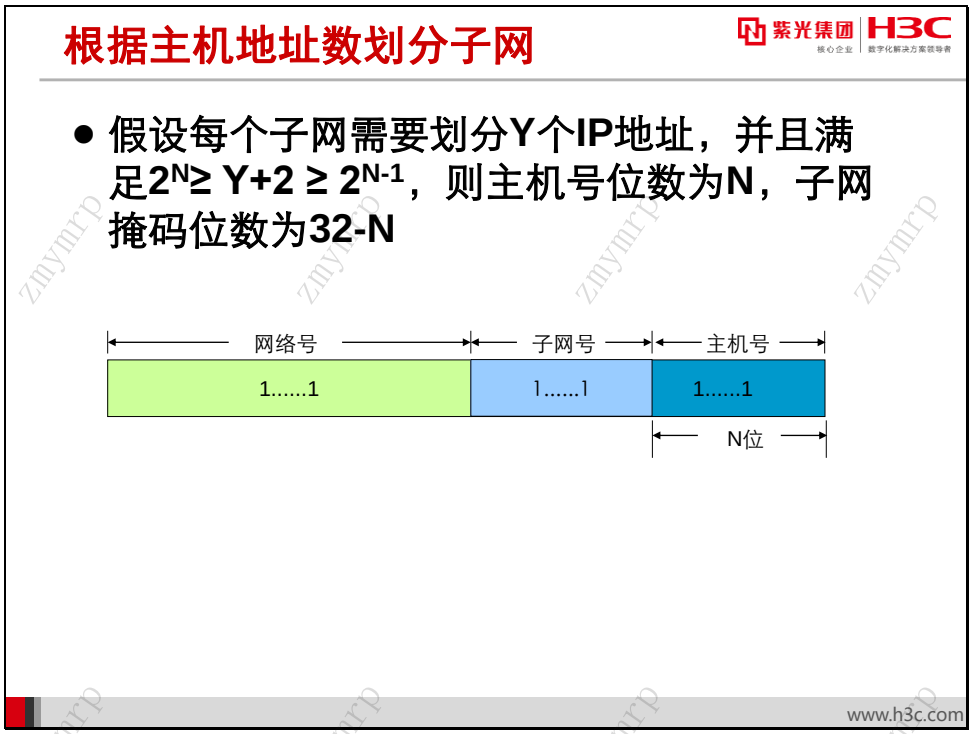
1) 计算掩码的位数。将十进制掩码 255.255.255.224 换算为二进制掩码 11111111.11111111.11111111.11100000，掩码的位数为 27

2) 计算主机号位数。主机号位数 N=32-27=5

3) 该子网可用的主机地址数量为 $2^N-2=2^5-2=30$ 个

这 30 个可用主机地址分别是：192.168.3.193、192.168.3.194、192.168.3.195……192.168.3.222。地址 192.168.3.192 为整个子网的地址，而 192.168.3.223 为这个子网的广播地址，都不能分配给主机使用。

16.4.2 根据主机地址数划分子网

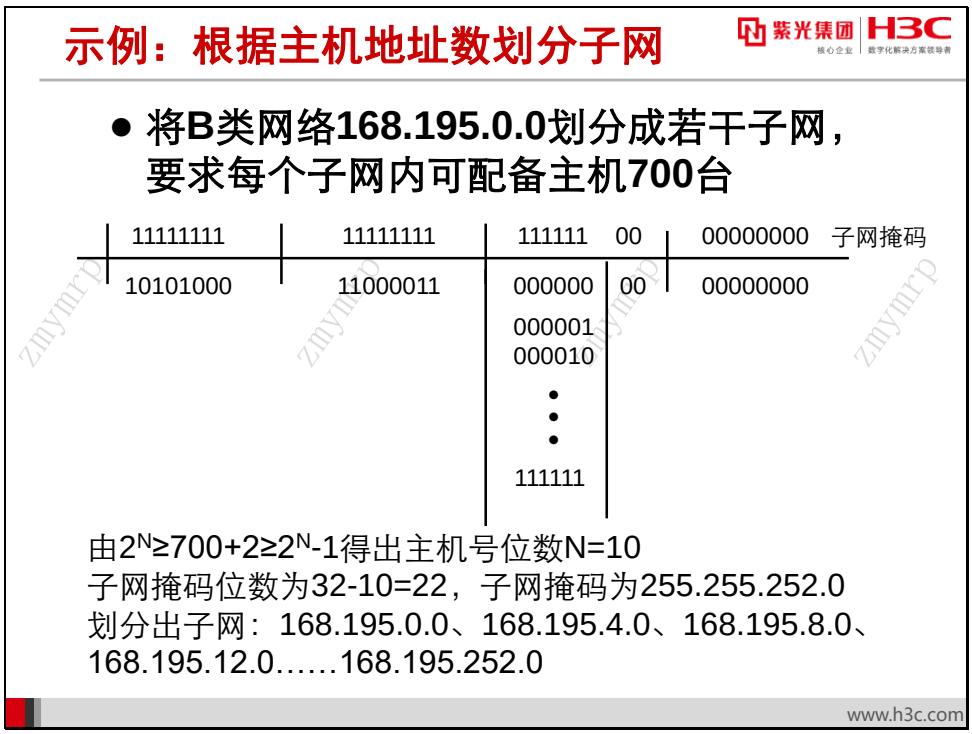


在子网划分计算中，有时需要在已知每个子网内需要容纳的主机数量的前提下，来划分子网。要想知道如何划分子网，就必须知道划分子网后的子网掩码，那么该问题就变成了求子网掩码。此类问题的计算方法总结如下：

1) 计算网络主机号的位数：假设每个子网需要划分出 Y 个 IP 地址，那么当 Y 满足公式 $2^N \geqslant Y+2 \geqslant 2^{N-1}$ 时，N 就是主机号的位数。其中 Y+2 是因为需要考虑主机号为全 0 和全 1 的情况

2) 计算子网掩码的位数：计算出主机号位数 N 后，可得出子网掩码位数为 **32-N**

3) 根据子网掩码的位数计算出子网号的位数 M。该子网就有 2^M 种划分法，具体的子网地址也可以很容易地算出



在本例中，需要将 B 类网络 168.195.0.0 划分成若干子网，要求每个子网内的主机数为 700 台。计算过程如下：

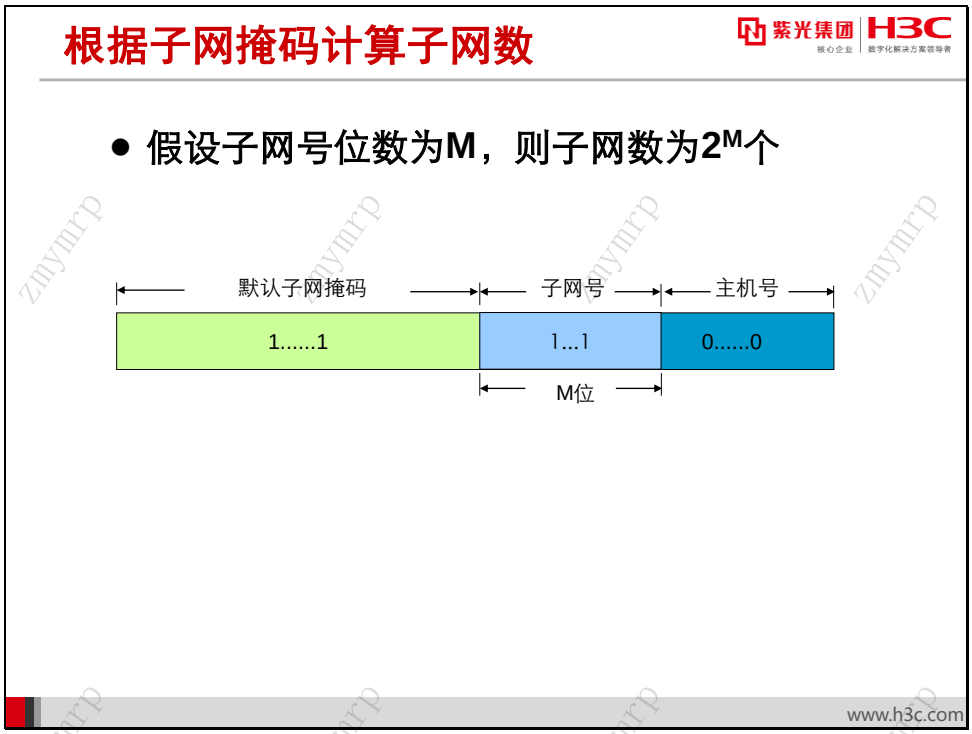
1) 按照例子中的子网划分要求，每个子网的主机地址数为 Y=700

2) 计算网络主机号。根据公式 $2^N \geqslant Y+2 \geqslant 2^{N-1}$ 计算出 N=10

3) 计算子网掩码的位数。子网掩码位数为 32-10=22，子网掩码为 255.255.252.0，二进制表示为 11111111.11111111.11111100.00000000

根据子网掩码位数可知子网号位数为 6。那么，该网络能划分成 2^6 个子网，这些子网分别是 168.195.0.0、168.195.4.0、168.195.8.0、168.195.12.0……168.195.252.0，子网掩码为 255.255.252.0。

16.4.3 根据子网掩码计算子网数



如果希望在一个网络中建立子网，就要在这个网络的默认掩码上增加若干位，形成子网掩码，这样就减少了用于主机地址的位数。加入到掩码中的位数决定了我们可以配置的子网数。

假设子网号的二进制位数（即子网掩码比默认掩码的位数增加的位数）为 M，那么可分配的子网数量为 2^M 个。

由此可见，对于特定网络来说，若使用位数较少的子网号，则获得的子网较少，而每个子网中可容纳的主机较多；反之，若使用位数较多的子网号，则获得的子网较多，而子网中可容纳的主机较少。因此可以根据网络中需要划分的子网数、每个子网中需要配置的主机数来选择合适的子网掩码。

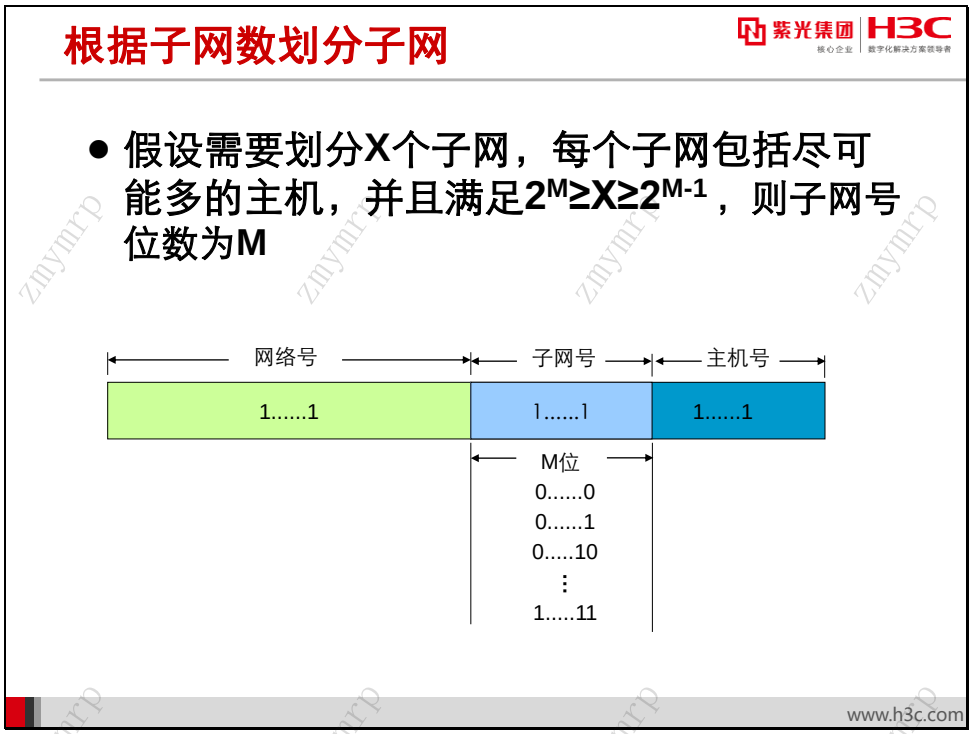
还应注意到，划分子网增加了灵活性，但却降低了 IP 地址的利用率，因为划分子网后主机号为全 0 或全 1 的 IP 地址不能分配给主机使用。

注意：

在 RFC950 规定的早期子网划分标准中，子网号不能为全 0 和全 1，所以子网数量应该为 2^M-2 个。但是在后期的 RFC1812 中，这个限制已经被取消了。

如无明确说明，在后续有关子网划分的计算中，都认为子网号可以为全 0 和全 1。

16.4.4 根据子网数划分子网

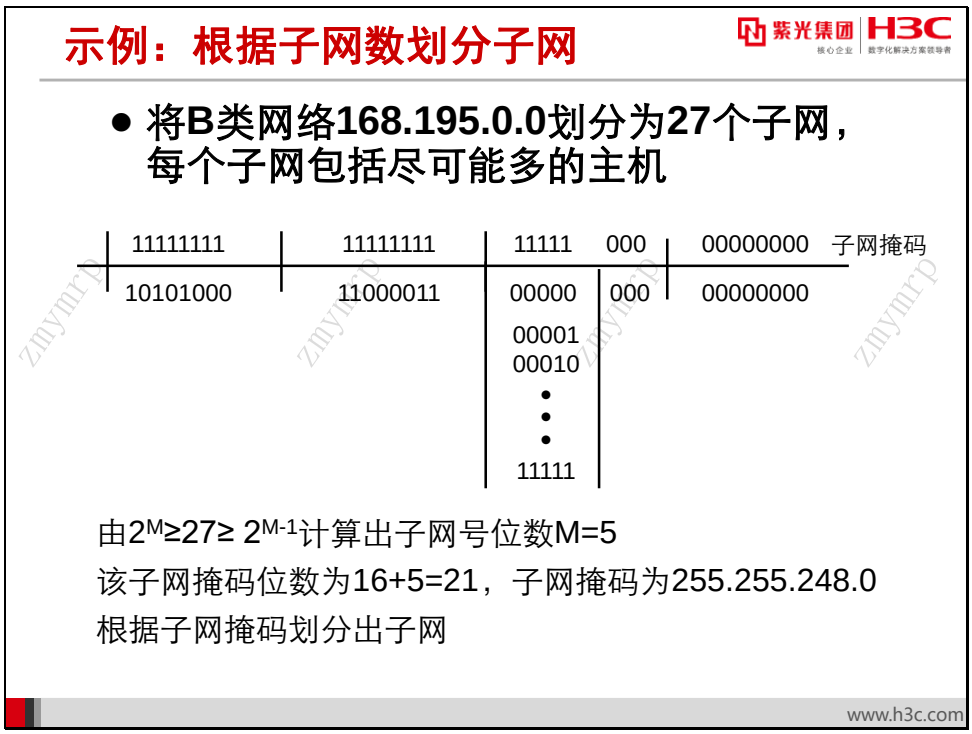


子网划分计算中，有时我们要在已知需要划分子网数量的前提下，来划分子网。当然，这类划分子网问题的前提是每个子网需要包括尽可能多的主机，否则该子网划分就没有意义了。因为，如果不要求子网包括尽可能多的主机，那么子网号位数可以随意划分成很大，而不是最小的子网号位数，这样就浪费了大量的主机地址。

比如，将一个 C 类网络 192.168.0.0 划分成 4 个子网，那么子网号位数应该为 2，子网掩码为 255.255.255.192。如果不考虑子网包括尽可能多的主机的话，子网号位数可以随意划分成大于 3、4、5，这样的话，主机号位数就变成 5、4、3，可用主机地址就大大地减少了。

同样，划分子网就必须得知道划分子网后的子网掩码，需要计算子网掩码。此类问题的计算方法总结如下：

1) 计算子网号的位数。假设需要划分 X 个子网，每个子网包括尽可能多的主机地址。那么当 X 满足公式 $2^M \geqslant X \geqslant 2^{M-1}$ 时，M 就是子网号的位数
2) 由子网号位数计算出网掩码，划分出子网



在本例中，需将 B 类网络 168.195.0.0 划分成 27 个子网，要求每个子包括尽可能多的主机。计算过程如下：

1) 按照例子中的子网划分要求，需要划分的子网数 X=27

2) 计算子网号的位数。根据公式 $2^M \geqslant X \geqslant 2^{M-1}$ 计算出 M=5

3) 计算子网掩码。子网掩码位数为 16+5=21，子网掩码为 255.255.248.0，二进制表示为 11111111.11111111.11111000.00000000

4) 由于子网号位数是 5，所以该 B 类网络 168.195.0.0 总共能划分成 2^5=32 个子网。这些子网是 168.195.0.0、168.195.8.0、168.195.16.0、168.195.24.0……168.195.248.0，子网掩码为 255.255.248.0。任意取其中的 27 个即可满足要求。

16.5 VLSM和CIDR

16.5.1 VLSM

虽然对网络进行子网划分的方法可以对 IP 地址结构进行有价值的扩充，但是仍然要受到一个基本的限制——整个网络只能有一个子网掩码。不论用户选择哪个子网掩码，都意味着各个子网内的主机数完全相等。不幸的是，在现实世界中，不同的组织对子网的要求是不一样的，希望一个组织把网络分成相同大小的子网是很不现实的。当在整个网络中一致地使用同一掩码时，在许多情况下会浪费大量主机地址。

针对这个问题，IETF 发布了标准文档 RFC1009。该文档规范了如何使用多个子网掩码划分子网。该标准规定，同一 IP 网络可以划分为多个子网并且每个子网可以有不同的大小。相对于原来的固定长度子网掩码技术，该技术称为 VLSM（Variable Length Subnet Mask，可变长子网掩码）。

VLSM 使网络管理员能够按子网的具体需要定制子网掩码，从而使一个组织的 IP 地址空间能够被更有效地利用。

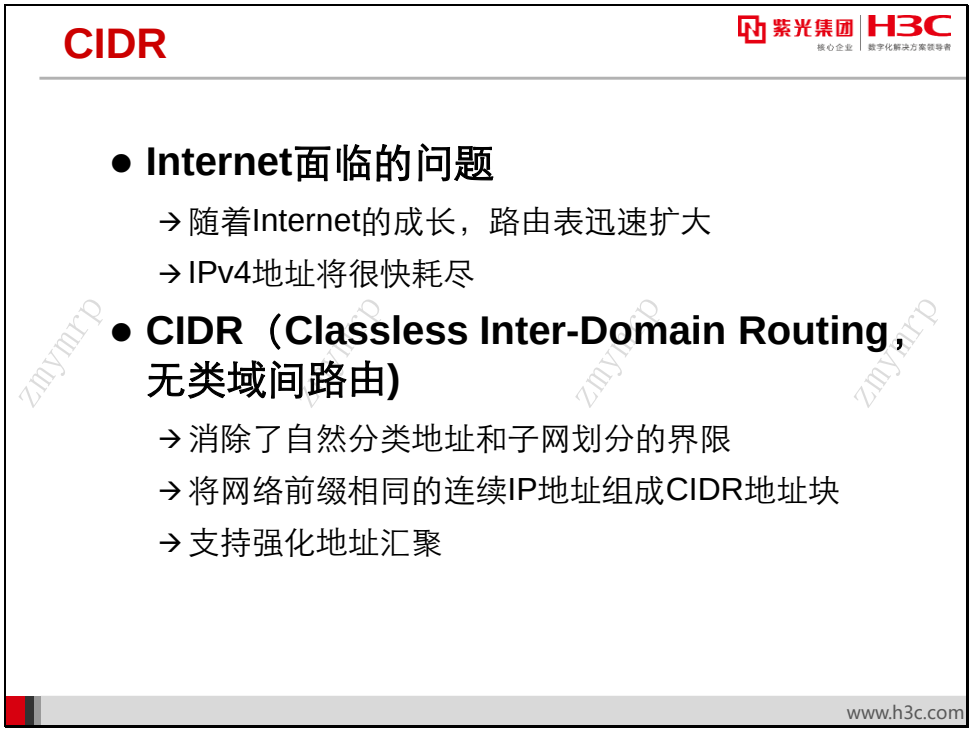
例如，假设某组织拥有一个 B 类网络，网络地址为 172.16.0.0，它使用 16 位的网络号。按照定长子网掩码的划分方法，该网络如果使用 6 位子网号，将会得到一个 22 位的子网掩码。整个网络可以划分为 64 个可用的子网，每个子网内有 1022 个可用的主机地址。

这种定长子网化策略对需要超过 30 个子网和每个子网内超过 500 台主机的组织是合适的。但是，如果这个组织由一个超过 500 台主机的大分支以及许多只有 40 至 50 台主机的小分支组

成，则大部分地址就被浪费了。每个分支即使只有 40 台主机，也将消耗一个有 1022 个主机地址的子网。显而易见，针对这样的组织，地址浪费现象是非常严重的。

解决这个矛盾的方法是允许对一个网络可以使用不同大小的子网掩码，对 IP 地址空间进行灵活的子网划分。考虑前面的例子，通过 VLSM 技术，网络管理员可以通过不同的子网掩码将网络切分为不同大小的部分。大的分支可以继续使用 22 位的子网掩码，然而小的分支可以使用 25 位或 26 位的子网掩码（126 个主机或 62 个主机）。这样，利用 VLSM 可以更好地避免 IP 地址的浪费。

16.5.2 CIDR



通过定长子网划分或可变长度子网划分的方法，在一定程度上解决了 Internet 在发展中遇到的困难。然而到 1992 年，Internet 仍然面临三个必须尽早解决的问题：

- B 类地址在 1992 年已经分配了将近一半，预计到 1994 年 3 月将全部分配完毕。
- Internet 主干网路由表中的路由条目数急剧增长，从几千个增加到几万个。
- IPv4 地址即将耗尽。

当时预计前两个问题将在 1994 年变得非常严重，因此 IETF 很快就研究出无分类编址的方法来解决前两个问题；而第三个问题由 IETF 的 IPV6 工作组负责研究。无分类编址是在 VLSM 基础上研究出来的，它的正式名字是无类域间路由 CIDR（Classless Inter-Domain Routing）。CIDR 在 RFC1517、RFC1518、RFC1519 及 RFC1520 中进行定义，现在 CIDR 已经成为 Internet 的标准协议。

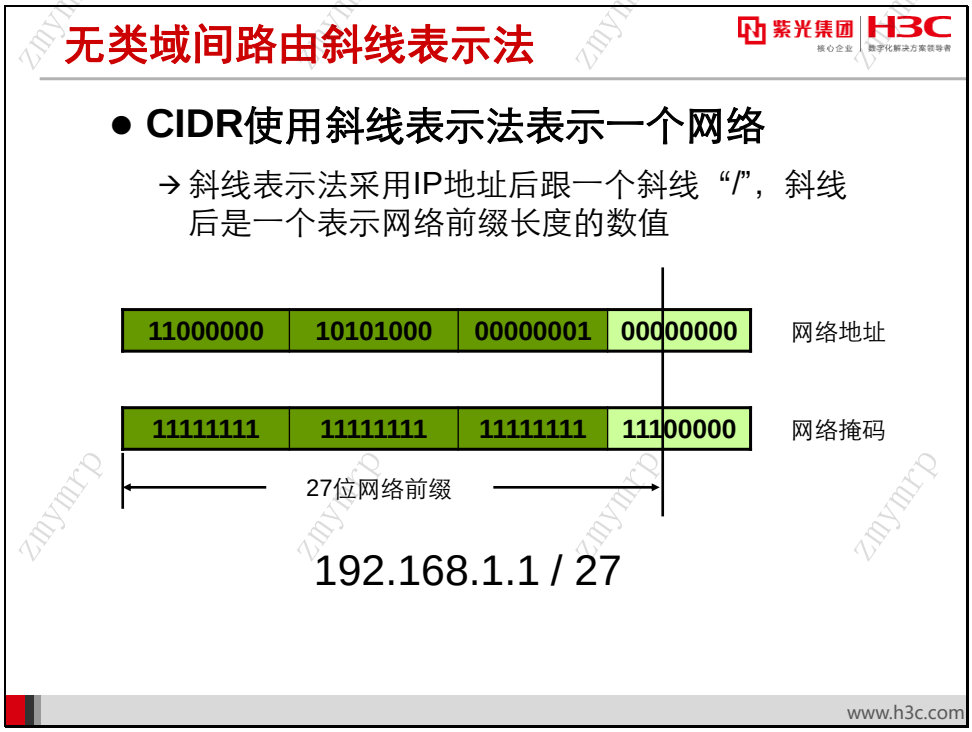
CIDR 消除了传统的自然分类地址和子网划分的界限，可以更加有效地分配 IPV4 的地址空间，在 IPv6 使用之前应对 Internet 的规模增长。

CIDR 不再使用“子网地址”或“网络地址”的概念，转而使用“网络前缀（network-prefix）”这个概念。与只能使用 8 位、16 位、24 位长度的自然分类网络号不同，网络前缀可以有各种长度，前缀长度由其相应的掩码标识。

CIDR 前缀既可以是一个自然分类网络地址，也可以是一个子网地址，也可以是由多个自然分类网络聚合而成的“超网”地址。所谓超网就是利用较短的网络前缀将多个使用较长网络前缀的小网络聚合为一个或多个较大的网络。例如，某机构拥有 2 个 C 类网络 200.1.2.0 和 200.1.3.0，而其需要在一个网络内部署 500 台主机，那么可以通过 CIDR 的超网化将这 2 个 C 类网络聚合为一个更大的超网 200.1.2.0，掩码为 255.255.254.0。

CIDR 可以将具有相同网络前缀的连续的 IP 地址组成 CIDR 地址块。一个 CIDR 地址块使用地址块的起始地址（前缀）和起始地址的长度（掩码）来定义。例如，某机构拥有 256 个 C 类网络 200.1.0.0、200.1.1.0……200.1.255.0，那么可以将这些地址合并为一个 B 类大小的 CIDR 地址块，其前缀为 200.1.0.0，掩码为 255.255.0.0。

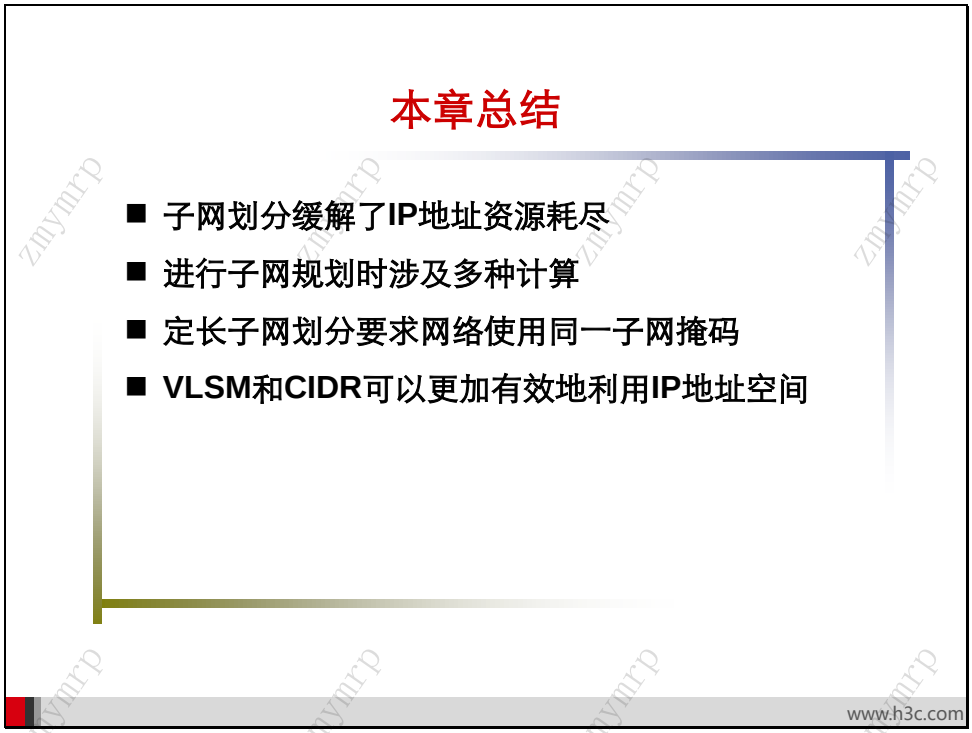
因为一个 CIDR 地址块可以表示很多个网络地址，所以支持 CIDR 的路由器可以利用 CIDR 地址块来查找目的网络，这种地址的聚合称为强化地址汇聚，它使得 Internet 的路由条目数大量减少。路由聚合减少了路由器之间路由选择信息的交互，从而提高了整个 Internet 的性能。



CIDR 使用“斜线表示法(slash notation)”表示一个网络，又称为 CIDR 记法。即在 IP 地址后面加上一个斜线“/”，然后写上网络前缀所占的位数，这个数值也就是子网掩码中为 1 的位数。这种表示方法也应用于子网掩码的表示。

例如，192.168.1.1/27 表示在 32 位的 IP 地址中，前 27 位表示网络前缀，而后面的 5 位表示主机号。

16.6 本章总结


本章总结
子网划分缓解了IP地址资源耗尽
进行子网规划时涉及多种计算
定长子网划分要求网络使用同一子网掩码
VLSM和CIDR可以更加有效地利用IP地址空间
www.h3c.com

第17章 DNS

在 TCP/IP 网络中，IP 地址是网络节点的标识。但是，IP 地址是点分十进制数，比较难于记忆。联想到在现实生活中，名字比身份证号码更容易被人记住，所以我们是否可以拿名字来标记某个网络节点呢？答案是肯定的。

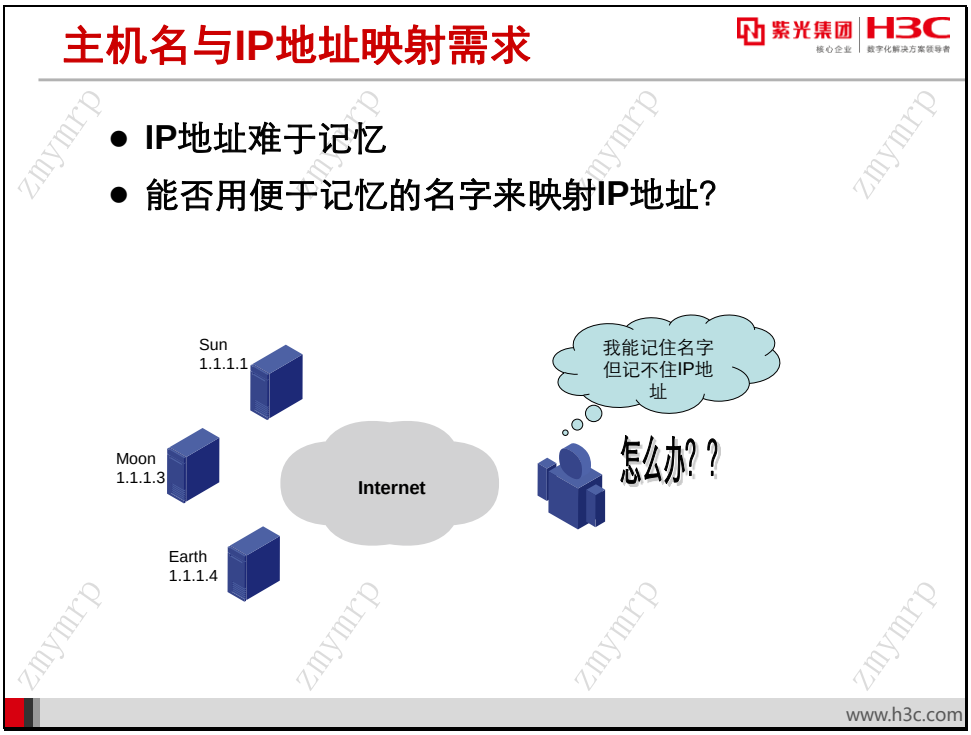
域名系统（DNS，Domain Name System）是一种用于 TCP/IP 应用程序的分布式数据库，提供域名与 IP 地址之间的转换。

本章将针对 DNS 展开学习和讨论，目标是使大家能够熟悉和掌握该协议的基础知识及工作方式。

17.1 本章目标

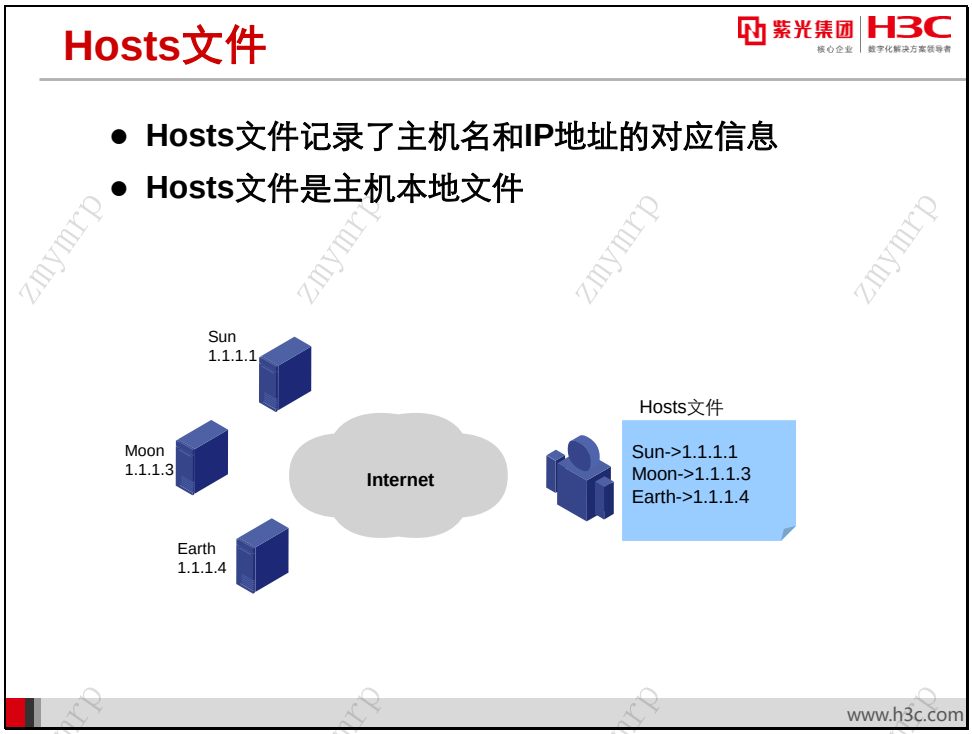
17.2 DNS域名系统概述

17.2.1 主机名与 IP 地址映射需求



在 TCP/IP 网络发展初期，人们直接使用 IP 地址来访问网络上的资源。随着网络规模的扩大和网络中所能提供服务的增加，需要记住越来越多的 IP 地址。但点分十进制的 IP 地址是很难记忆的。所以需要一种能够把 IP 地址与便于记忆的名字关联起来的方法，使人们只要能记住这些名字，就可以访问网络资源。

17.2.2 Hosts 文件

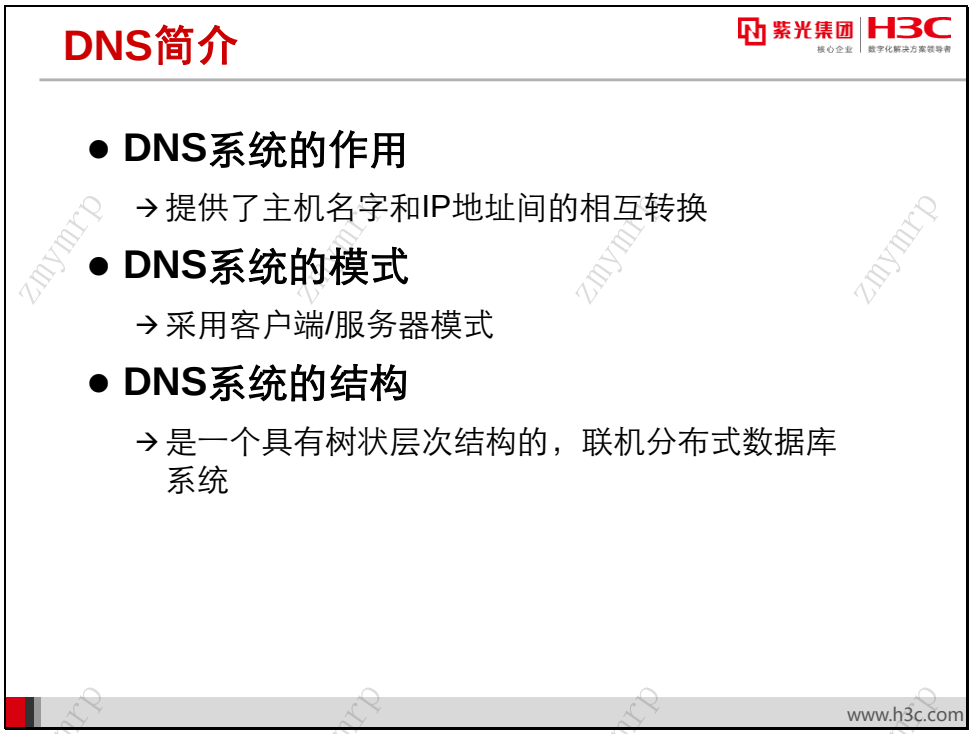


在 Internet 早期，网络中仅有几百台主机，那时的计算机使用一个叫做 Hosts 的文件来实现主机名与 IP 地址之间的映射。Hosts 文件中包括主机名和 IP 地址的对应信息。当一台主机需要通过主机名的方式访问网络上的另外一台主机时，它就会查看本地的 Hosts 文件，从文件中找到相对应的 IP 地址然后进行报文发送。如果在 Hosts 文件中没有关于那台主机名的相关信息，则主机访问将失败。

Hosts 文件是主机的本地文件，它的优点是查找响应速度快。它主要用来存储一些本地网络上的主机名与 IP 地址对应的信息。这样，主机在以主机名访问本地网络主机时，通过本地 Hosts 文件可以迅速获得相应 IP 地址。

每台主机的 Hosts 文件都需要手工单独更新，而且几乎没有自动配置。随着 Internet 规模快速增长，维护包含一个大量映射条目的文件的难度越来越大，而且在每台主机间进行经常同步更新几乎是一件不可能完成的任务。

17.2.3 DNS 简介



为了解决 Hosts 文件维护困难的问题，上世纪 80 年代 IETF 发布了域名系统（DNS，Domain Name System）。

DNS 域名系统主要解决了因特网上主机名与 IP 地址之间的相互转换，为用户实现多种网络资源的访问提供了必要条件。

DNS 系统采用客户端/服务器模式，DNS 客户端提出查询请求，DNS 服务器负责相应请求。DNS 客户端通过查询 DNS 服务器获得所需访问主机的 IP 地址信息，进而完成后续的 TCP/IP 通信过程。

DNS 系统是一个具有树状层次结构的，联机分布式数据库系统。1983 年因特网开始使用层次机构的命名树作为主机的名字，树状层次机构的主机名在管理，维护，扩展等方面具有更大的优势。DNS 系统也采用树状层次结构与之对应。

虽然从理论上讲 DNS 可以采用集中式设计，整个因特网只使用一台 DNS 服务器，这台 DNS 服务器包含 Internet 所有主机名与 IP 地址的映射关系。客户端简单地把所有咨询消息发送给这个惟一的名称服务器，该名称服务器则把响应消息返回给查询的主机。这种设计尽管具有诱人的简单性，但面对 Internet 大量的主机数量，而且仍在不断增长的主机数量，这种做法并不可取。集中式 DNS 系统存在的主要问题：

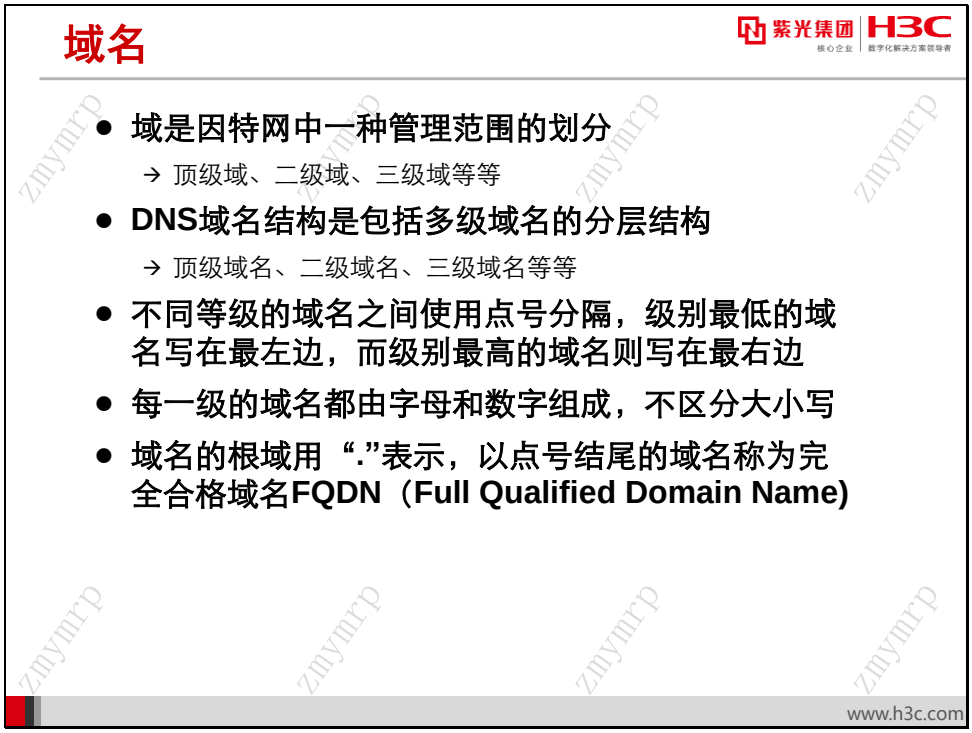
- **存在单点故障：**要是惟一的 DNS 服务器崩溃了，整个 DNS 服务也失效了。
- **服务处理的访问量巨大，性能不足：**单台 DNS 服务器将不得不处理所有 DNS 查询消息。

- **远距离访问，效率低下：**单台 DNS 服务器主机不可能在所有请求查询的客户主机附近。例如，假设 DNS 服务器在纽约，而所有的查询都来自于地球另一边的澳大利亚，可能要经过缓慢且拥塞的链路。这样造成的远距离访问将导致相当大的延迟。
- **系统维护工作量巨大：**单台 DNS 服务器将保存所有因特网主机的记录。这个集中式数据库不仅会相当巨大，而且不得不为每台新增的主机频繁更新。允许任何用户在集中式数据库中注册主机还存在身份认证与授权问题。

因此，因特网的 DNS 域名系统被设计成为一个联机分布式数据库系统，名字到 IP 地址的解析可以由若干个域名服务器共同完成。大部分的名字解析工作可以在本地的域名服务器上完成，效率很高。并且由于 DNS 使用分布式系统，即使单个服务器出现故障，也不会导致整个系统失效，消除了单点故障。

17.3 域名及域名结构

17.3.1 域名

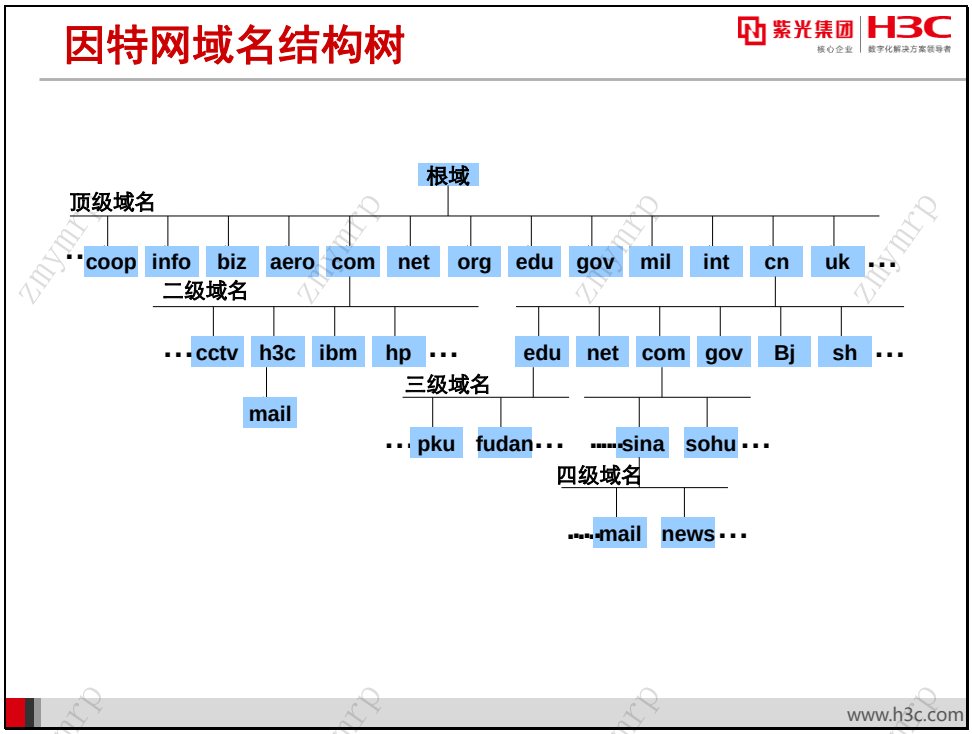


因特网早期使用的主机名是无等级概念的，其优点是短小精悍，记忆简单。但当因特网的联机主机数量急剧增加时，使用这种不划分等级的命名方法来管理一个规模巨大且不断变化的主机名集合是非常困难的。

DNS 域的本质是因特网中一种管理范围的划分，最大的域是根域，向下可以划分为顶级域、二级域、三级域、四级域等。相对应的域名是根域名、顶级域名、二级域名、三级域名等等。不同等级的域名之间使用点号分隔，级别最低的域名写在最左边，而级别最高的域名则写在最右边。如域名 www.abc.com 中，com 为顶级域名，abc 为二级域名，而 www 则表示二级域中的主机。

每一级的域名都由英文字母和数字组成，域名不区分大小写，但是长度不能超过 63 字节，一个完整的域名不能超过 255 个字节。根域名用“．”(点)表示。如果一个域名以点结尾，那么这种域名我们称之为完全合格域名（FQDN，Full Qualified Domain Name）。接入因特网的主机、服务器或其他网络设备都可以拥有一个唯一的完全合格域名。

17.3.2 因特网域名结构树



如上图所示，因特网的域名空间结构像是一棵倒过来的树。根域名就是树根，用点号表示。

根域名下属的顶级域名包括三大类：

- **国家顶级域名**

国家顶级域名采用 ISO3166 的规定。如：.CN 表示中国，.US 表示美国，.UK 表示英国等等。现在使用的国家顶级域名大约在 200 个左右。

- **国际顶级域名**

国际顶级域名采用.INT。国际性的组织可以在.INT 下注册。

- **通用顶级域名**

最早的顶级域名共有 6 个。分别为.COM 表示公司企业，.NET 表示网络服务机构，.ORG 表示非盈利组织，.EDU 表示教育机构(仅限美国)，.GOV 表示政府部门(仅限美国)，.MIL 表示军事部门(仅限美国)。随着因特网用户不断增加，从 2000 年 11 月起，因特网的域名管理机构 ICANN 又增加了七个通用顶级域名。分别为.AERO 用于航空运输业，.BIZ 用于公司和企业，.COOP 用于合作团体，.INFO 用于各种情况，.MUSEUM 用于博物馆，.NAME 用于个人，.PRO 用于自由职业者。

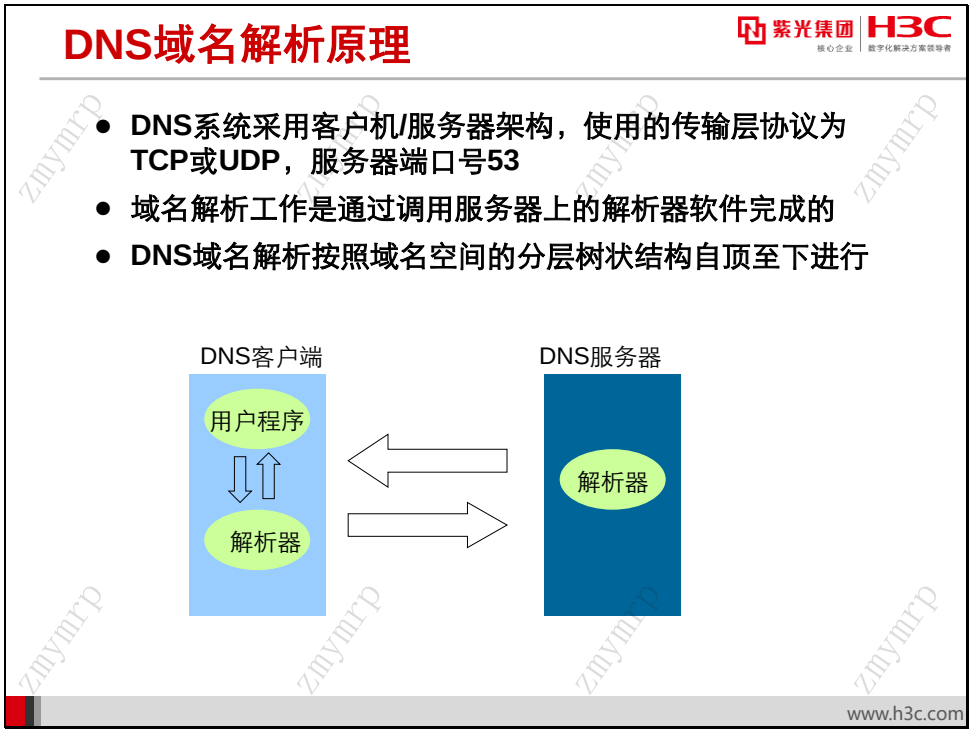
在顶级域下面注册的是二级域名，如图中在顶级域.COM 下的有二级域名 CCTV、H3C 等。国家顶级域名下注册的二级域名均由该国家自行确定。我国将二级域名划分为类别域名和行政域名两大类，类别域名如.COM，.EDU，.GOV 等分别代表不同的机构；行政域名如.BJ 表示北

京，.SH 表示上海，代表我国的各省、自治区及直辖市等。二级域名下面是三级域，四级域等。命名树上任何一个节点的域名就是将从该节点到最高层的域名串起来，中间以“.”分隔。

在域名结构中，节点在所属域中的标识可以相同，但域名必须唯一。例如图中 H3C 公司和新浪公司下都有一台主机的标识是 mail，但是两者的域名却是不同的，前者为 mail.h3c.com 而后者为 mail.sina.com.cn。

17.4 DNS域名解析

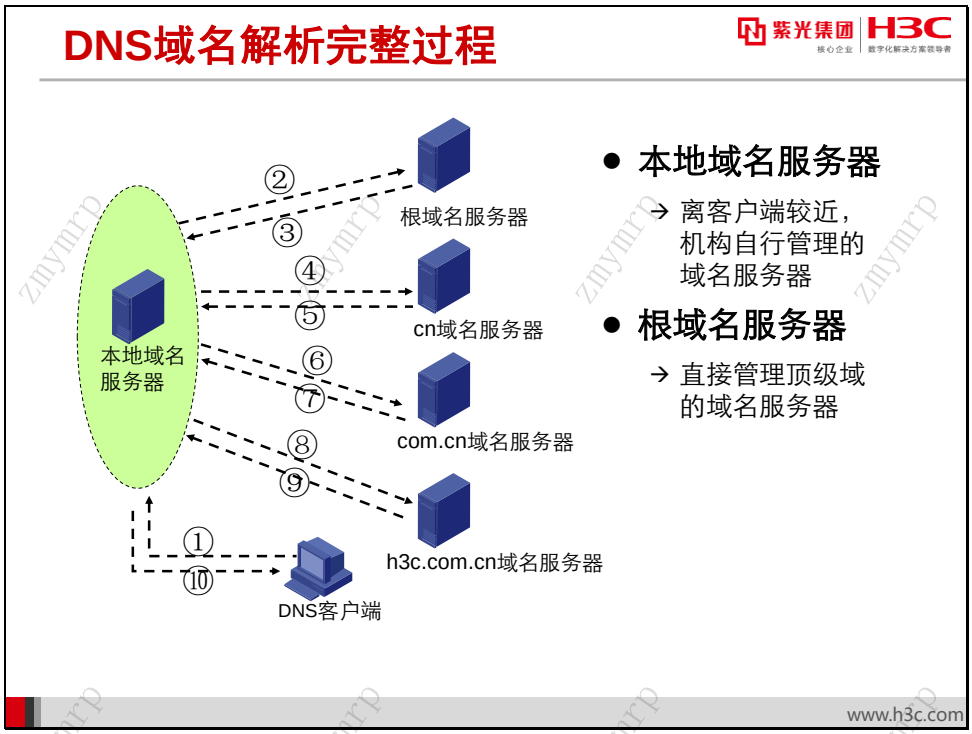
17.4.1 DNS 域名解析



将域名转换为对应的 IP 地址的过程称为域名解析。域名解析是 DNS 客户端上的用户程序调用解析器软件向 DNS 服务器发出请求，DNS 服务器调用安装在其上的解析器软件完成域名解析。DNS 采用客户端/服务器体系架构，使用 TCP 或 UDP 作为传输层协议，域名服务器使用 53 端口号侦听客户端发出的 DNS 查询请求。

DNS 域名解析是按照 DNS 分层结构的特点自顶向下进行的。然而，如果每一个域名解析都从根服务器开始，那么根服务器有可能无法承载因特网中大量的信息交互流量。在实际应用中，大多数域名解析都是由本地域名服务器在本地解析完成的。通过合理设定本地域名服务器，由本地域名服务器负责大部分的域名解析请求，可以很大程度上提高域名解析的效率。

17.4.2 DNS 域名解析过程



- **本地域名服务器**

每个因特网服务提供商或者一个大的网络机构，例如公司，大学等都可以拥有一台或多台可以自行管理的域名服务器，这类域名服务器称为本地域名服务器，有时也称为默认域名服务器。本地域名服务器一般离客户端较近。当一个 DNS 客户端发送 DNS 查询时，该查询首先被送往本地域名服务器，如果本地域名服务器数据库中存在对应的主机域名，本地域名服务器会立即将所查询的域名转换为 IP 地址返回客户端。在 Windows 操作系统的“Internet 协议（TCP/IP）属性”中设置的 DNS 服务器就是我们最常见的本地域名服务器。

- **根域名服务器**

通常根域名服务器用来管理顶级域，本身并不对域名进行解析，但它知道相关域名服务器的地址。在 DNS 解析过程中，当本地域名服务器的数据库中没有 DNS 客户端所查询的主机域名时，它会以 DNS 客户端的身份向某一个根域名服务器进行查询。根域名服务器收到本地域名服务器的查询后，会回应相关域名服务器的 IP 地址，本地域名服务器再向相关域名服务器发送查询请求。

上图中是一个完整的 DNS 域名解析示例。DNS 客户端进行域名 www.h3c.com.cn 的解析过程如下：

第1步：DNS 客户端向本地域名服务器发送请求，查询 www.h3c.com.cn 主机的 IP 地址；

第2步：本地域名服务器检查其数据库，发现数据库中没有域名为 www.h3c.com.cn 的主机，于是将此请求发送给根域名服务器；

第3步：根域名服务器查询其数据库，发现没有该主机记录，但是根域名服务器知道能够解析该域名的 cn 域名服务器的地址，于是将 cn 域名服务器的地址返回给本地域名服务器；

第4步：本地域名服务器向 cn 域名服务器查询 www.h3c.com.cn 主机的 IP 地址；

第5步：cn 域名服务器查询其数据库，发现没有该主机记录，但是 cn 域名服务器知道能够解析该域名的 com.cn 域名服务器的地址，于是将 com.cn 的域名服务器的地址返回给本地域名服务器；

第6步：本地域名服务器再向 com.cn 域名服务器查询 www.h3c.com.cn 主机 IP 地址；

第7步：com.cn 域名服务器查询其数据库，发现没有该主机记录，但是 com.cn 域名服务器知道能够解析该域名的 h3c.com.cn 域名服务器的 IP 地址，于是将 h3c.com.cn 的域名服务器的 IP 地址返回给本地域名服务器；

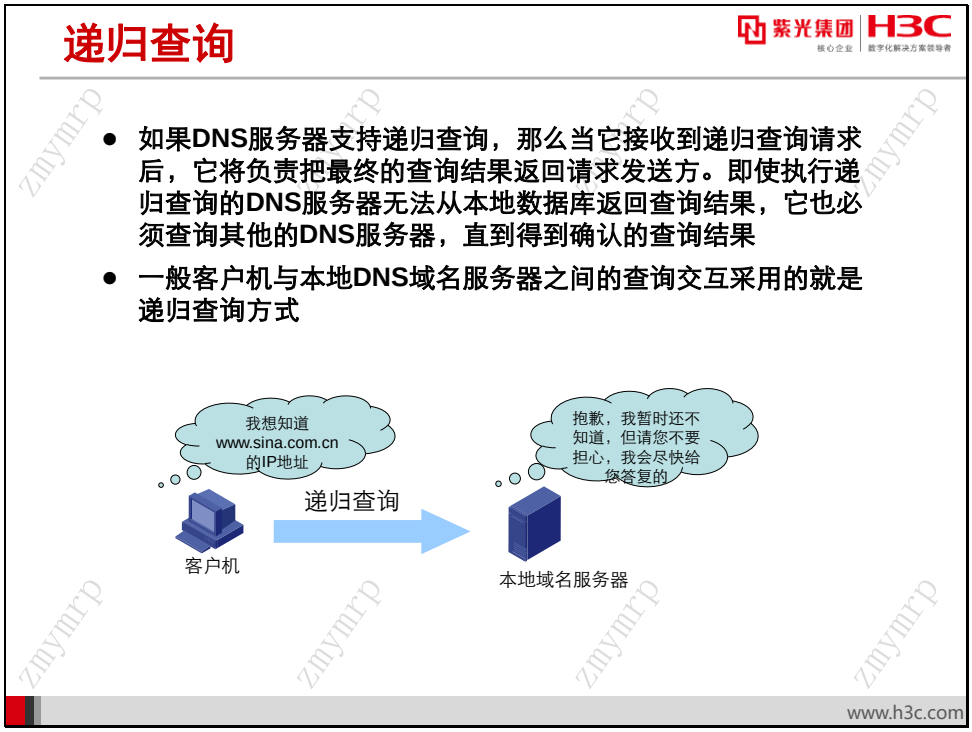
第8步：本地域名服务器向 h3c.com.cn 域名服务器发送查询 www.h3c.com.cn 主机的 IP 地址请求；

第9步：h3c.com.cn 域名服务器查询其数据库，发现有该主机记录，于是给本地域名服务器返回“www.h3c.com.cn”所对应的 IP 地址；

第10步：最后本地域名服务器将 www.h3c.com.cn 的 IP 地址返回给客户端。至此，整个解析过程完成。

17.5 DNS查询方式

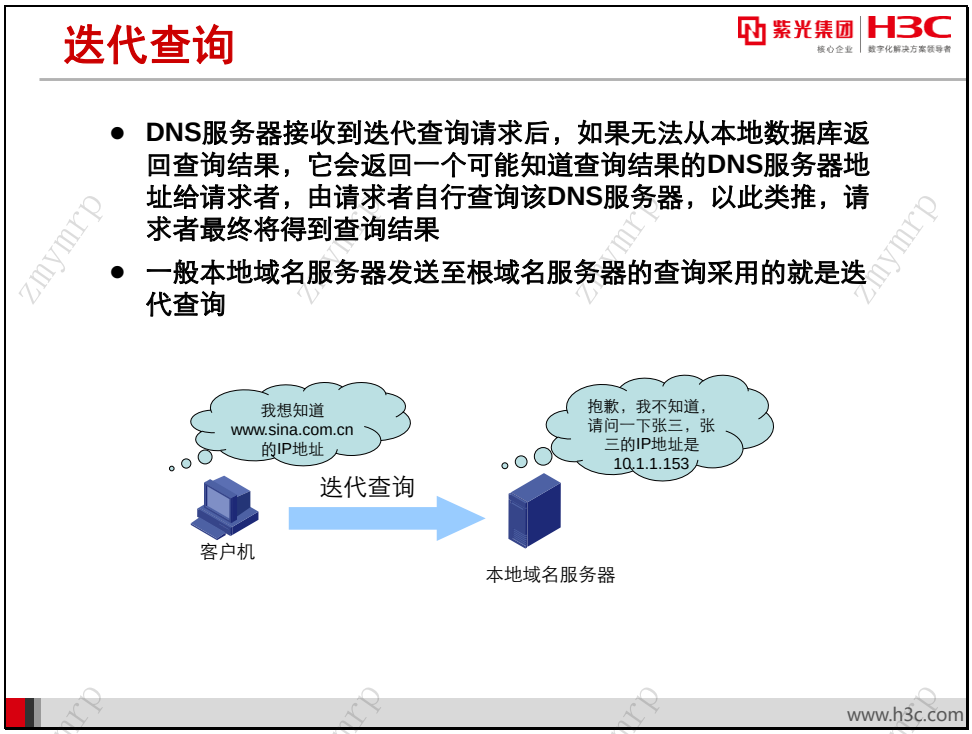
17.5.1 递归查询



DNS 域名解析包括两种查询方式：一种为递归查询；另外一种为迭代查询。

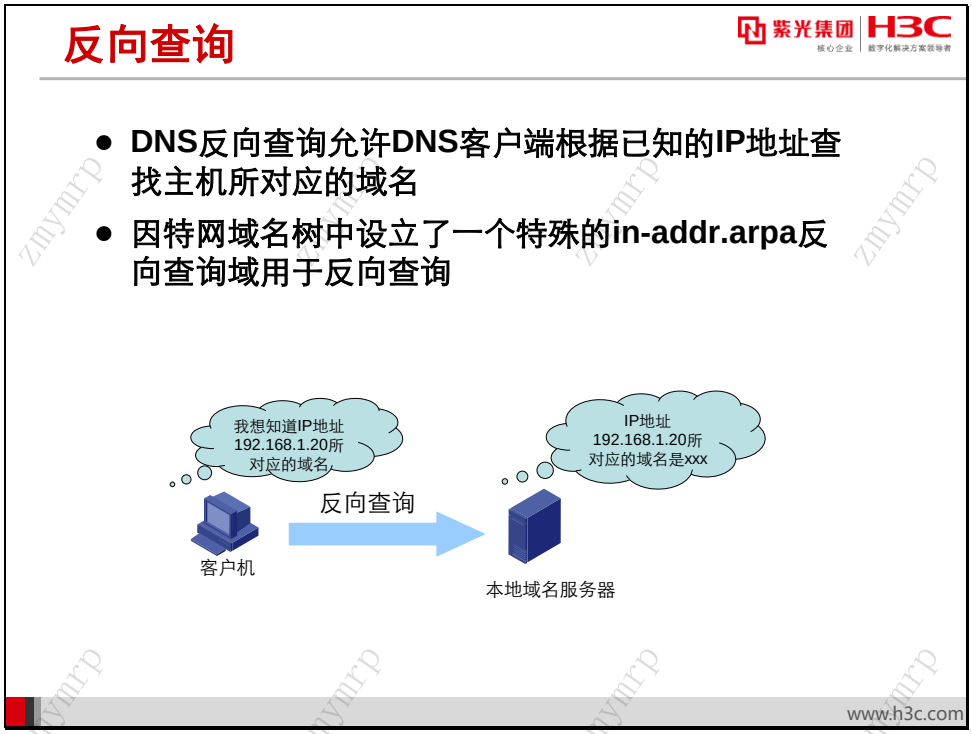
对于一个递归查询，DNS 服务器如果不能直接回应解析请求，它将以 DNS 客户端的方式继续请求其他 DNS 服务器，直到查询到该主机的域名解析结果。回复结果可以是该主机的 IP 地址或者是该域名无法解析。不论是哪种结果，DNS 服务器都将把结果返回给客户端。一个递归查询很好的例子是，当本地域名服务器接受了客户端的查询请求时，本地域名服务器将力图代表客户端来找到答案，而在本地域名服务器执行所有工作的时候，客户端只是等待，直到本地域名服务器将最终查询结果返回该客户端。

17.5.2 迭代查询



在迭代解析方式中，如果服务器查不到相应的记录，会向客户端返回一个可能知道结果的域名服务器地址，由客户端继续向新的服务器发送查询请求。迭代查询的很好例子是一台本地域名服务器发送请求到根服务器。当某个企业的本地域名服务器向根域名服务器提出查询，根域名服务器并不一定代表本地域名服务器来担当起回答查询的责任，它会指引本地域名服务器到另一台域名服务器进行查询。例如，当根服务器被要求查询 www.h3c.com.cn 的地址时，根域名服务器不会到 h3c.com.cn 域名服务器查询 www 主机的地址，它只是给本地域名服务器返回一个提示，告诉本地域名服务器到 cn 域名服务器去继续查询和得到结果。所以，对域名服务器的迭代查询只能得到一个提示，再继续查询。

17.6 DNS反向查询



在 DNS 查询中，客户端希望知道域名所对应的 IP 地址，这种查询我们称为正向查询。大部分的 DNS 查询都是正向查询。

与正向查询相对应，有一种查询是反向查询。它允许 DNS 客户端根据已知的 IP 地址查找主机所对应的域名。反向查询采取问答形式进行，如“您能告诉我使用 IP 地址 192.168.1.20 的计算机的 DNS 名称吗？”

为了实现反向查询，在 DNS 标准中定义了特殊域 in-addr.arpa 域，并保留在 Internet 域名空间中，以便提供切实可靠的方式执行反向查询。为了创建反向域名空间，in-addr.arpa 域中的子域是按照点分十进制表示法编号的 IP 地址的相反顺序构造的。

为什么不能使用正向的 IP 地址顺序构造 in-addr.arpa 域呢？这主要与 DNS 域名构成的层级关系有关。在 DNS 域名空间树结构中，根域是域名树的树根，最接近树根的是顶级域，依次向下是二级域，三级域，四级域……。我们可以看出，越靠近树根域的范围越大，越远离树根域的范围越小，换句话说也就是越具体。因此对于域名 www.abc.com.cn，从左往右看时呈现了一种范围从小到大，逐层包容的关系。

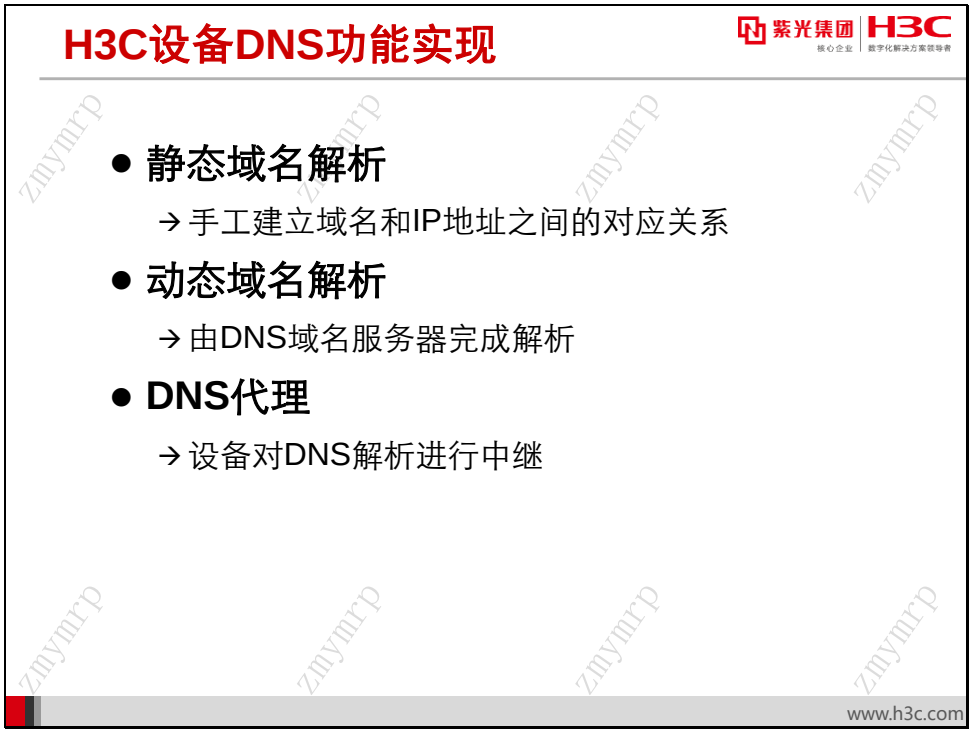
对于 IP 地址的构成方式，从左向右看，首先是网络地址部分，然后才是具体的主机部分，包容关系与域名的构成方式恰恰相反。在反向查找中，其实也是把 IP 地址作为特殊的域名对待，因此需要把 IP 地址的 4 个八位字节倒置排列形成特殊的 in-addr.arpa 域。

在 DNS 中建立的 in-addr.arpa 域中使用一种称为指针类型的资源记录。这种资源记录用于反向查找区域中创建映射，它一般对应于其正向查找区域中某一主机的 DNS 主机名。与正向

查找一样，如果所查询的反向名称不能从 DNS 服务器应答，则同样的 DNS 查询（递归或迭代）过程可用来查找对反向查找区域具有绝对权威且包含查询名称的 DNS 服务器。

17.7 H3C设备DNS特性及配置

17.7.1 H3C 设备 DNS 特性



目前在 H3C 系列产品上，能够实现如下与 DNS 相关的功能：

- **静态域名解析**

静态域名解析就是在设备上手工建立域名和 IP 地址之间的对应关系。当用户使用域名进行某些应用（如 Telnet）时，系统查找静态域名解析表，从中获取指定域名对应的 IP 地址。

- **动态域名解析**

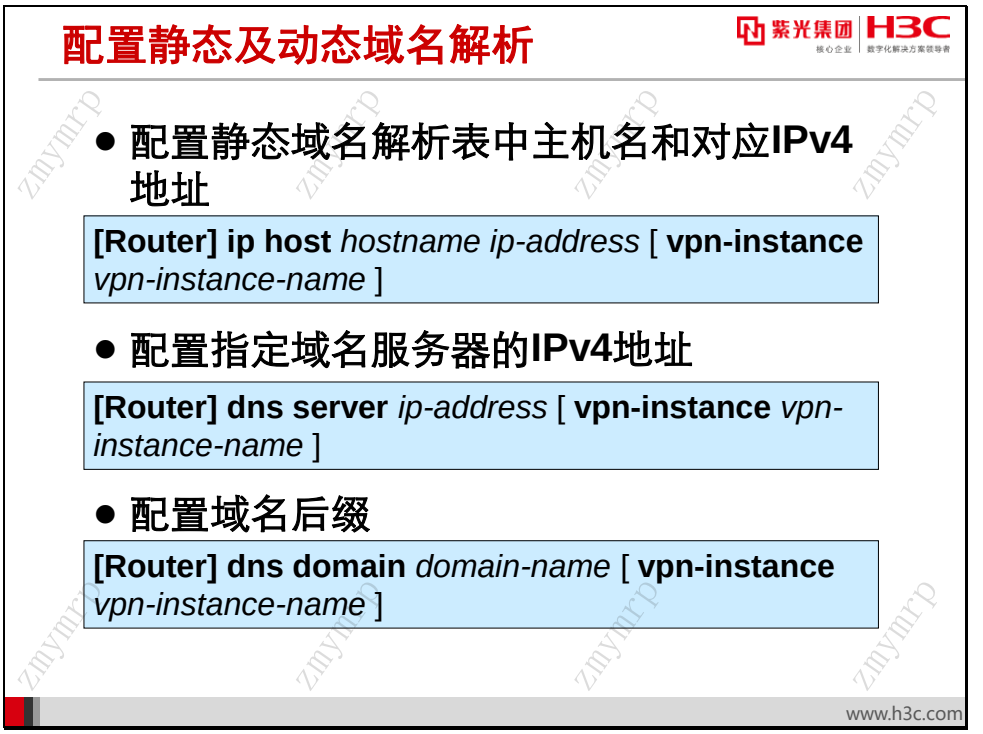
动态域名解析就是由设备查询 DNS 域名服务器，由 DNS 域名服务器完成解析。动态域名解析支持域名后缀列表功能。用户可以预先设置一些域名后缀，在域名解析的时候，用户只需要输入域名的部分字段，系统会自动把输入的域名加上不同的后缀进行解析。举例说明，用户想查询域名 h3c.com，那么可以先在后缀列表中配置 com，然后输入 h3c 进行查询，系统会自动将输入的域名与后缀连接成 h3c.com 进行查询。

- **DNS 代理**

DNS 代理（DNS proxy）用来在 DNS 客户端和 DNS 服务器之间转发 DNS 请求和应答报文，代替 DNS 客户端进行域名解析。局域网内的 DNS 客户端把 DNS 代理当作 DNS 服务器，将 DNS 请求报文发送给 DNS 代理。DNS 代理将该请求报文转发到正确的 DNS 服务器，并将 DNS 服务器的应答报文返回给 DNS 客户端，从而实现域名解析。使用 DNS 代理功能后，当

DNS 服务器的地址发生变化时，只需改变 DNS 代理上的配置，无需改变局域网内每个 DNS 客户端的配置，从而简化了网络管理。

17.7.2 配置静态和动态域名解析



在路由器上可以配置 DNS 静态域名解析，能够加快解析速度。配置的要点是建立域名和 IP 地址之间的映射表。

在系统视图下，配置静态解析表中主机名和对应的 IPv4 地址：

ip host *hostname ip-address* [**vpn-instance** *vpn-instance-name*]

参数 **vpn-instance** *vpn-instance-name* 是指配置命令是对指定的 VPN 有效。

如果网络中有域名服务器，可以在路由器上配置 DNS 动态域名解析。

在系统视图下，指定域名服务器的 IPv4 地址：

dns server *ip-address* [**vpn-instance** *vpn-instance-name*]

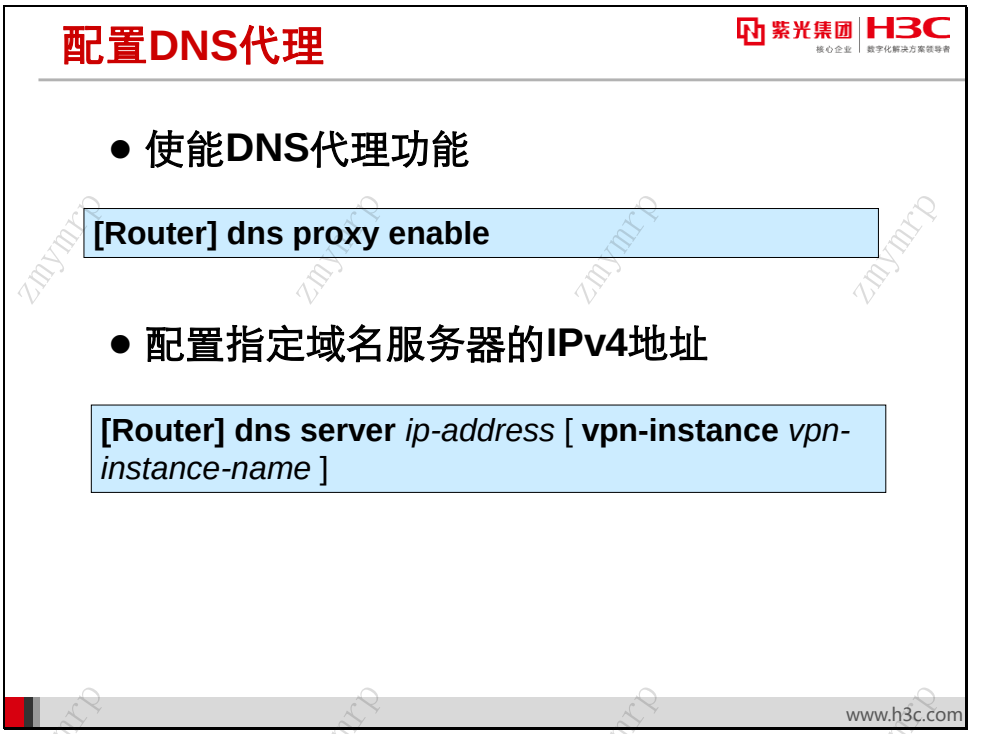
配置完成后，如果用户使用域名进行某些应用，如 Ping 一个主机名时，路由器将向所指定的域名服务器发出 DNS 解析请求，请求域名所对应的 IPv4 地址。

为了使用方便，可以在系统视图下配置域名后缀列表：

dns domain *domain-name* [**vpn-instance** *vpn-instance-name*]

配置完成后，路由器发送 DNS 解析请求时会自动把域名加上所配置的后缀。

17.7.3 配置 DNS 代理



如果 DNS 服务器和客户端不在同一个子网内，可以在路由器上使能 DNS 代理。配置步骤如下：

第1步：在系统视图下使能 DNS 代理功能

dns proxy enable

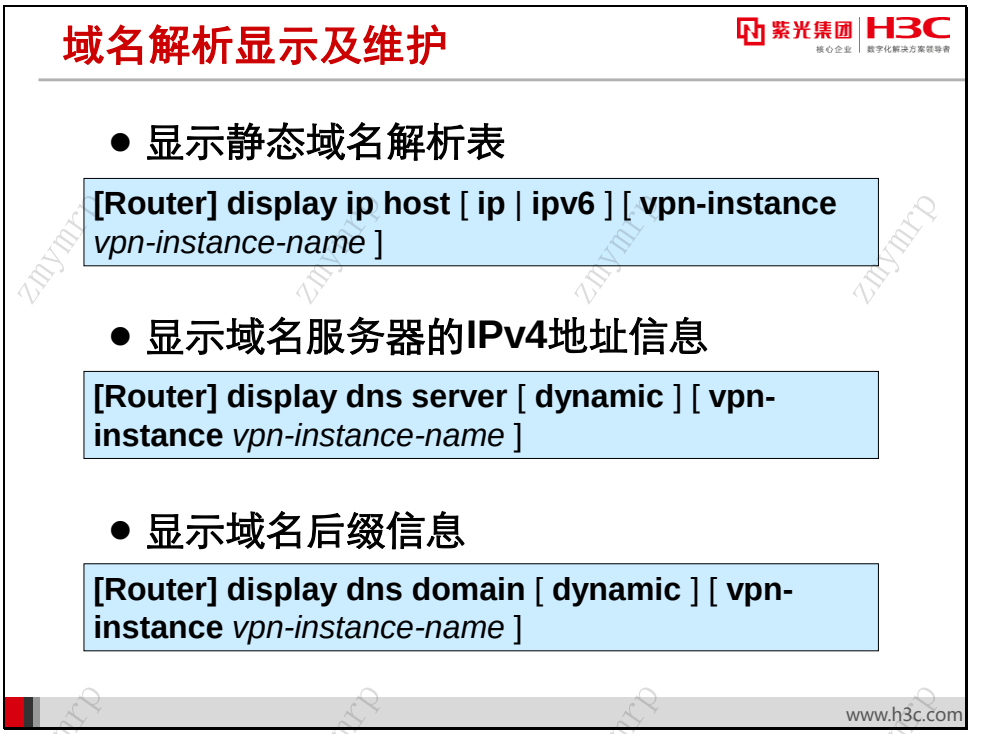
只有使能 DNS 代理功能后，路由器才能中继客户端与 DNS 服务器之间的交互报文。

第2步：在系统视图下指定域名服务器的 IPv4 地址

dns server *ip-address* [**vpn-instance** *vpn-instance-name*]

配置完成后，路由器作为 DNS 代理，将客户端发出请求报文转发到所配置的 DNS 服务器，并将 DNS 服务器的应答报文返回给 DNS 客户端，完成跨越子网的域名解析。

17.7.4 域名解析显示及维护

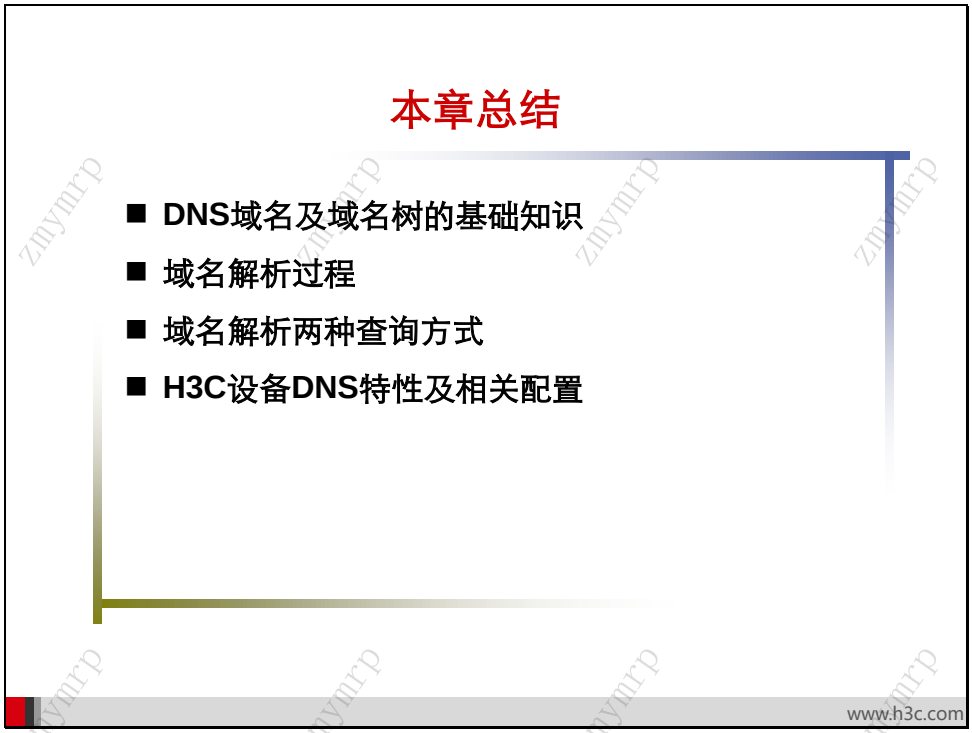


- **域名解析显示及维护**

在完成上述配置后，在任意视图下执行 display 命令可以显示域名解析配置后的运行情况，通过查看显示信息验证配置的效果。

操作	命令
显示静态域名解析表	**display ip host host [ip | ipv6] [vpn-instance** *vpn-instance-name* **]**
显示域名服务器的IPv4地址信息	**display dns server [dynamic] [vpn-instance** *vpn-instance-name* **]**
显示域名后缀信息	**display dns domain [dynamic] [vpn-instance** *vpn-instance-name* **]**

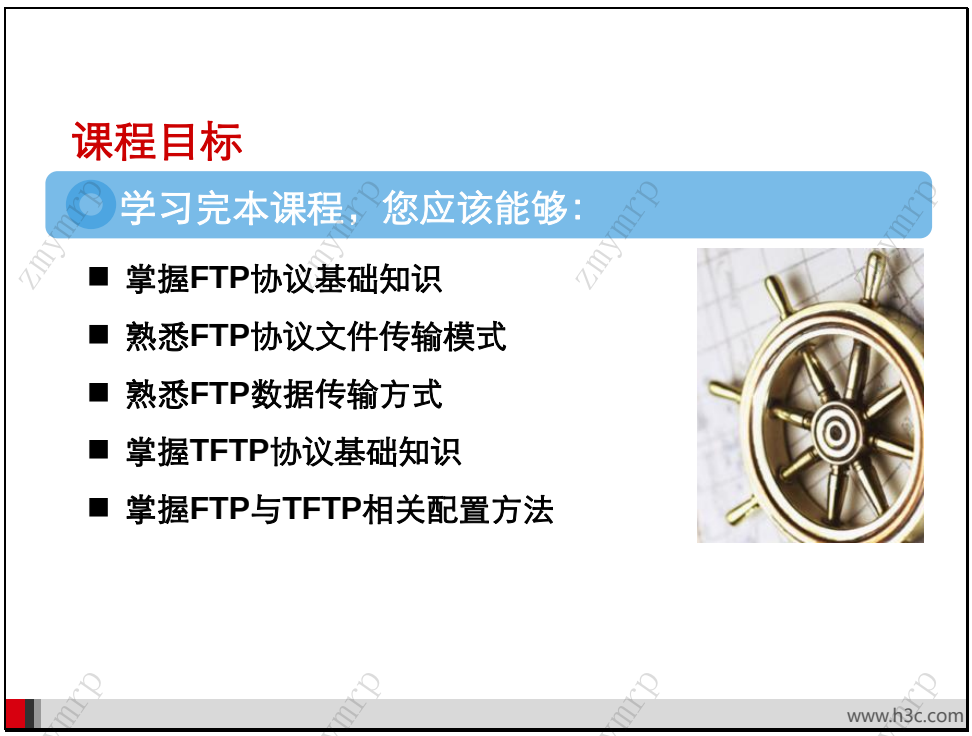
17.8 本章总结


本章总结
DNS域名及域名树的基础知识
域名解析过程
域名解析两种查询方式
H3C设备DNS特性及相关配置
www.h3c.com

第18章 文件传输协议

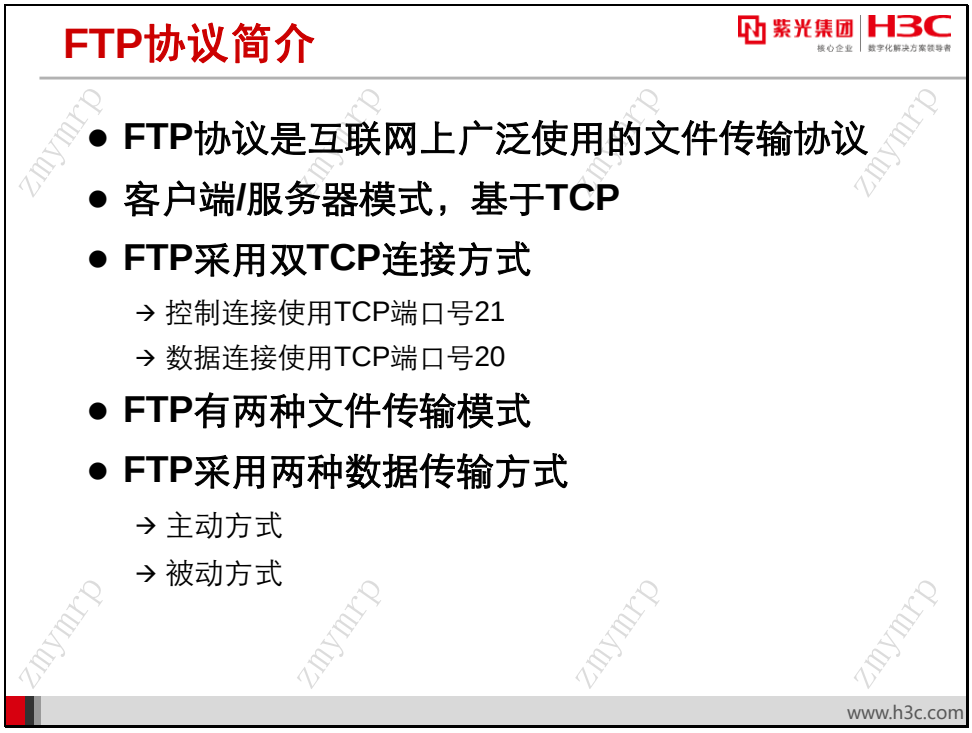
在互联网中我们经常需要在远端主机和本地服务器之间传输文件，文件传输协议提供的应用服务满足了我们的这种需求。FTP（File Transfer Protocol）是互联网上文件传输的标准协议，FTP 使用 TCP 作为传输协议，支持用户的登录认证及访问权限的设置。互联网上另一种常用的文件传输协议是 TFTP（Trivial File Transfer Protocol）协议，TFTP 是一种简单的文件传输协议，不支持用户的登录认证，也不具备复杂的命令。TFTP 使用 UDP 作为传输协议，并具有重传机制。接下来我们将对这两种传输协议进行介绍。

18.1 本章目标



18.2 FTP协议

18.2.1 FTP 协议介绍



FTP（File Transfer Protocol，文件传输协议）用于在远端服务器和本地主机之间传输文件，是 IP 网络上传输文件的通用协议。在万维网（WWW，World Wide Web）出现以前，用户使用命令行方式传输文件，最通用的应用程序就是 FTP。虽然目前大多数用户在通常情况下选择使用 Email 和 Web 传输文件，但是 FTP 仍然有着比较广泛的应用。

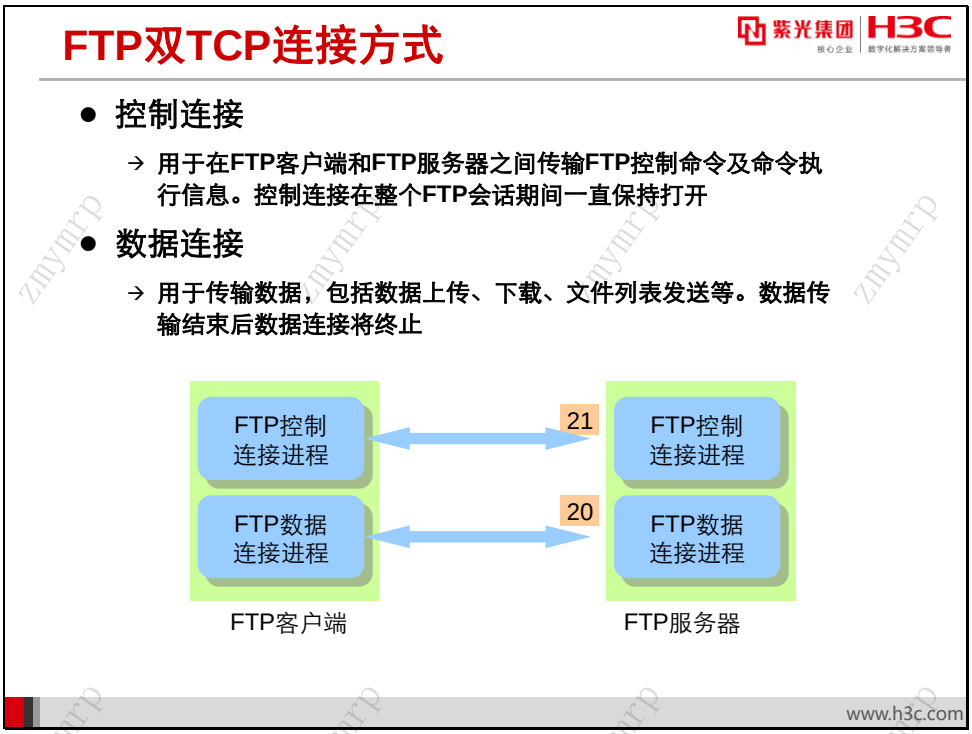
FTP 采用客户端/服务器的设计模式，承载在 TCP 协议之上。FTP 功能强大，拥有丰富的命令集。FTP 支持对登录服务器的用户名和口令进行验证，可以提供交互式的文件访问，允许客户指定文件的传输类型，并且可以设定文件的存取权限。

FTP 采用双 TCP 连接方式，使用 TCP 协议的端口 20 和 21，端口 20 用于数据连接，端口 21 用于控制连接。

为了确保文件能够准确无误的传送到对方，而不会引起格式上的误解，FTP 使用了两种文件传输模式，来传送不同类型的文件。

在 FTP 数据连接过程中，有两种数据传输方式：主动方式和被动方式。

18.2.2 FTP 双连接方式



通过 FTP 进行文件传输时，需要在服务器和客户端之间建立两个 TCP 连接：FTP 控制连接和 FTP 数据连接。

FTP 服务器启动后，FTP 服务打开 TCP 端口号 21 作为侦听端口，等待客户端的连接。客户端随机选择一个 TCP 端口号作为控制连接的源端口，主动发起对 FTP 服务器端口号 21 的 TCP 连接。控制连接建立后，FTP 客户端和 FTP 服务器之间通过该连接交互 FTP 控制命令和命令执行的应答信息。FTP 控制连接在整个 FTP 会话过程中一直保持打开。

FTP 的数据连接主要有三大用途：

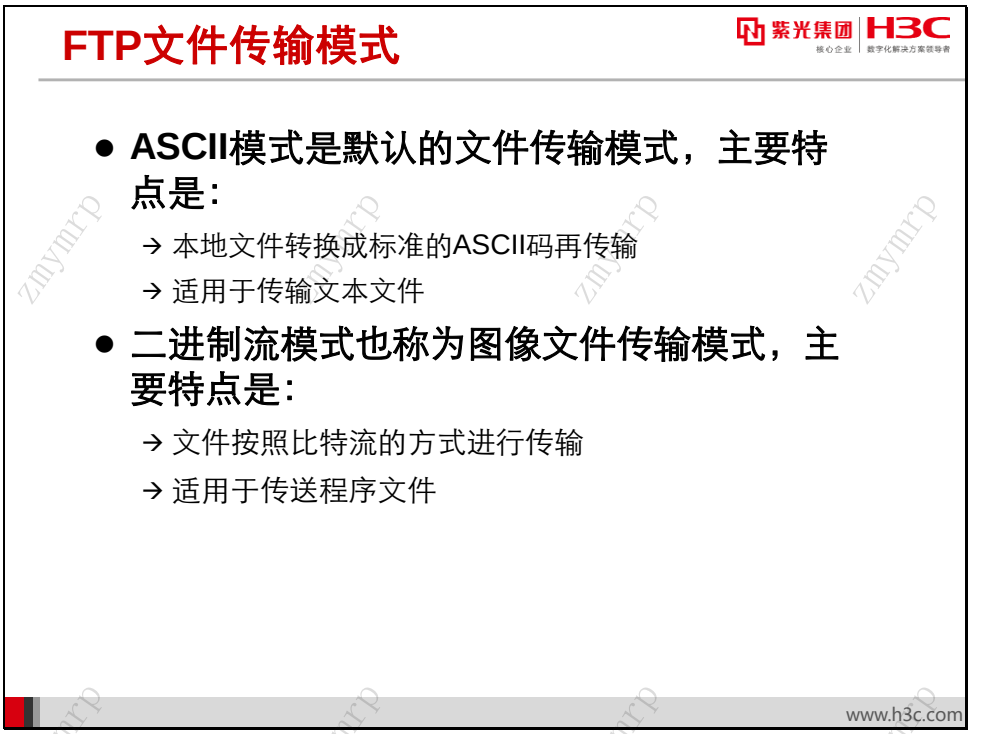
- 客户端向服务器发送文件
- 服务器向客户端发送文件
- 服务器向客户端发送文件列表

数据连接是实际用于传输文件的逻辑通道。FTP 客户端通过控制连接发出文件传输的请求，服务器的控制进程接收到传输请求后创建数据传输进程和数据连接。

FTP 服务器把文件列表通过数据连接发送到客户端，而不是在控制连接上使用多行应答。这样的好处是避免了行的有限性对文件列表大小的限制，并且用户可以把文件列表以文件的方式保存，而不仅仅只在终端上显示出来。

服务器和客户端在需要传输数据的时候建立数据连接，数据传输完毕后终止。数据连接在整个 FTP 会话过程中不需要一直存在。

18.2.3 FTP 文件传输模式



对于相同的一个文件，不同的操作系统可能会有不同的存储表达方式。为了在不同操作系统之间进行文件传输，所以 FTP 协议定义了不同的文件传输模式，目前使用最广泛的文件传输模式包括：

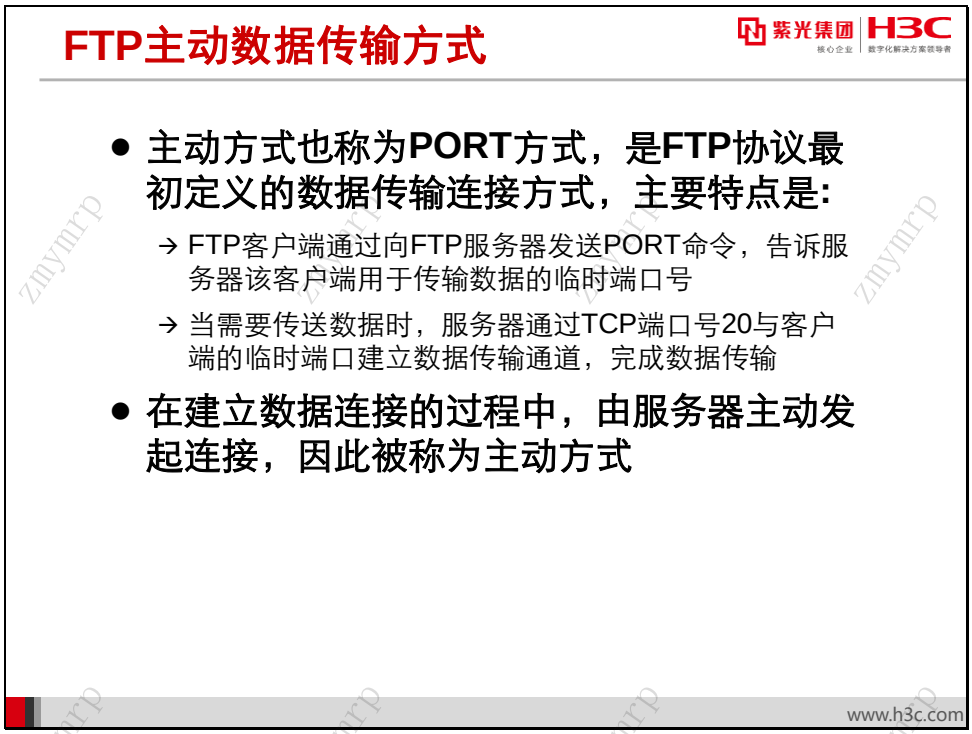
- **ASCII 模式**

ASCII 模式是默认的文件传输模式。发送方把本地文件转换成标准的 ASCII 码，然后在网络中传输；接收方收到文件后，根据自己的文件存储表达方式而把它转换成本地文件。ASCII 文件传输模式通常适用于传输文本文件。

- **二进制流模式**

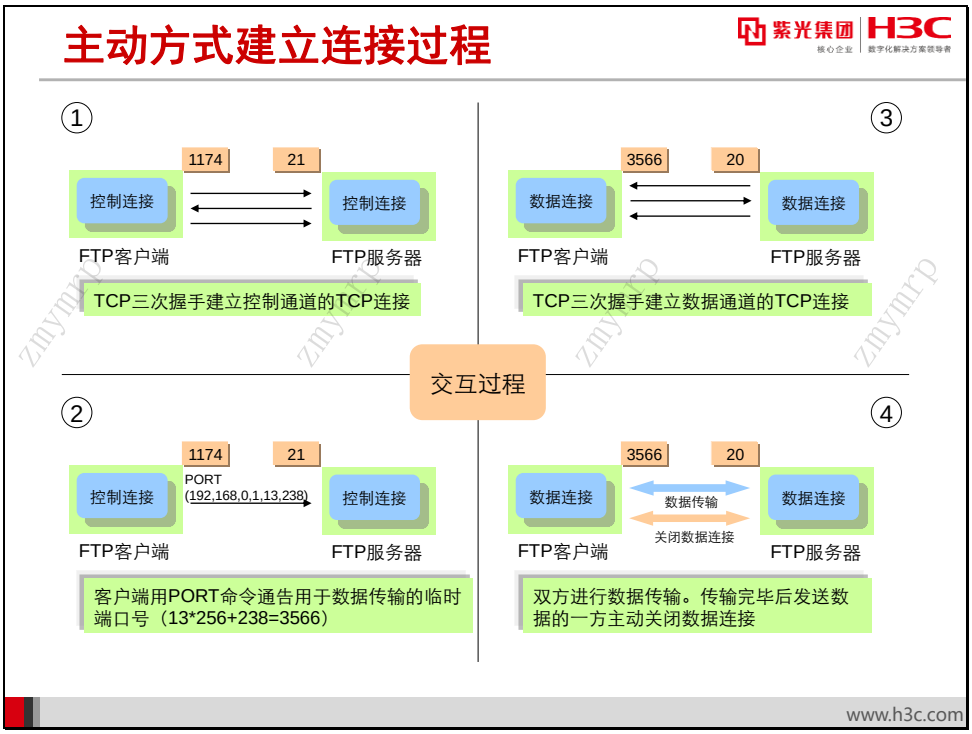
二进制流模式也称为图像文件传输模式。发送方不做任何转换，把文件按照比特流的方式进行传输。二进制文件类型通常适用于传送程序文件。

18.2.4 FTP 主动数据传输方式



FTP 主动传输方式也称为 PORT 方式，是 FTP 协议最初定义的数据传输方式。采用主动方式建立数据连接时，FTP 客户端会通过 FTP 控制连接向 FTP 服务器发送 PORT 命令，PORT 命令携带如下格式的参数（A1，A2，A3，A4，P1，P2），其中 A1，A2，A3，A4 表示需要建立数据连接的主机 IP 地址，而 P1 和 P2 表示客户端用于传输数据的临时端口号，临时端口号的数值为 256*P1+P2。当需要传送数据时，服务器通过 TCP 端口号 20 与客户端提供的临时端口建立数据传输通道，完成数据传输。在整个过程中，由于服务器在建立数据连接时主动发起连接，因此被称为主动模式。

如果客户端处于防火墙内部，主动方式可能会遇到问题。因为客户端提供的端口号是随机的，防火墙并不知道。而为了安全起见，通常防火墙只会允许外部主机访问部分内部已知端口，阻断对内部随机端口的访问，从而造成无法建立 FTP 数据连接。



FTP 主动方式建立连接的过程如下：

- **阶段一：建立控制通道 TCP 连接**

1) FTP 客户端以随机端口（图中是 1174）作为源端口，向 FTP 服务器的 TCP 端口 21 发送一个 TCP SYN 报文，开始建立 TCP 连接；

2) FTP 服务器收到 SYN 报文后发送 SYN ACK 报文给客户端，源端口为 TCP 端口 21，目的端口为 FTP 客户端使用的随机端口 1174；

3) FTP 客户端收到 FTP 服务器发送的 SYN ACK 报文后，向 FTP 服务器回送一个 ACK 报文，完成 TCP 三次握手，建立 FTP 控制连接。

- **阶段二：主动方式参数传递**

4) 当 FTP 客户端希望请求文件列表或者需要同服务器进行文件传输时，FTP 客户端会通过已经建立好的控制通道向服务器发送 PORT 命令，命令中包含了自己的 IP 地址和端口号。在图中，IP 地址是 192.168.0.1，端口号是 13*256+238=3566。

- **阶段三：建立数据通道 TCP 连接**

5) FTP 服务器向 FTP 客户端发送一个 SYN 报文，主动建立 TCP 连接。通信的源端口为 FTP 服务器的 TCP 端口号 20，目的端口为客户端在 PORT 命令中发送给服务器的端口号 3566；

6) FTP 客户端以端口号 3566 为源端口，20 为目的端口向 FTP 服务器发送一个 SYN ACK 报文；

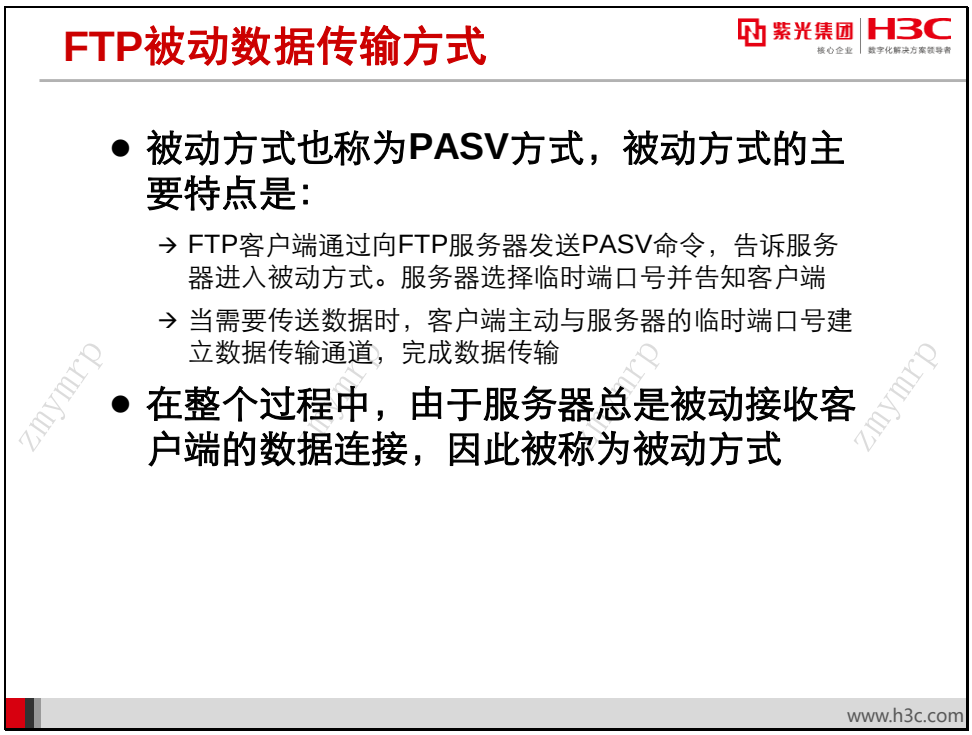
7) FTP 服务器端向 FTP 客户端发送一个 ACK 报文，完成 TCP 三次握手，建立数据通道的 TCP 连接。

- **阶段四：数据传输**

8) 数据通道连接建立后，FTP 客户端与 FTP 服务器利用该通道进行数据的传输；

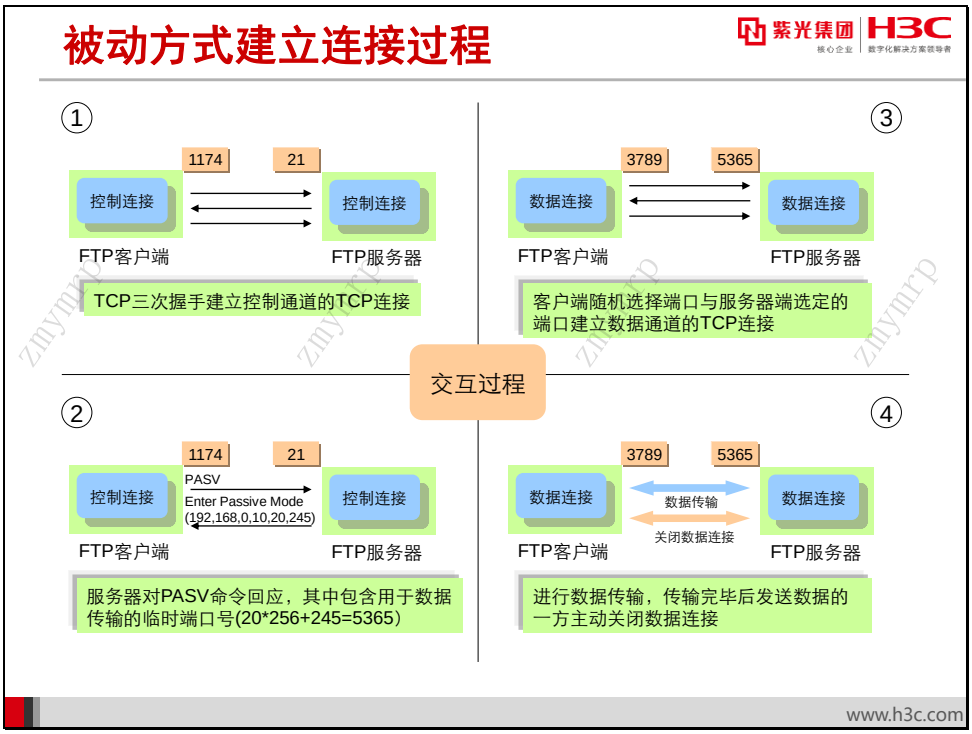
9) 数据传输完毕后，由发送数据的一方发送 FIN 报文，关闭这条数据连接。如果 FTP 客户端需要打开新的数据连接，则可以通过控制通道发送相关命令再次建立新的数据传输通道。

18.2.5 FTP 被动数据传输方式



被动方式也称为 PASV 方式。FTP 控制通道建立后，希望通过被动方式建立数据传输通道的 FTP 客户端会利用控制通道向 FTP 服务器发送 PASV 命令，告诉服务器进入被动方式传输。服务器选择临时端口号并告知客户端，一般采用如下形式命令：Entering Passive Mode（A1，A2，A3，A4，P1，P2），其中 A1，A2，A3，A4 表示服务器的 IP 地址，P1，P2 表示服务器的临时端口号，数值为 256*P1+P2。当需要传送数据时，客户端主动与服务器的临时端口建立数据传输通道，并完成数据传输。在整个过程中，由于服务器总是被动接收客户端的数据连接，因此被称为被动方式。

采用被动方式时，两个连接都由客户端发起。一般防火墙不会限制从内部的客户端发出的连接，所以这样就解决了在主动方式下防火墙阻断外部发起的连接而造成无法进行数据传输的问题。



FTP 被动方式建立连接的过程如下：

- **阶段一：建立控制通道 TCP 连接**

1) FTP 客户端以随机选择的临时端口号（图中是 1174）作为源端口向 FTP 服务器 TCP 21 端口发送一个 TCP SYN 报文，开始建立 TCP 连接；

2) FTP 服务器收到 SYN 报文后发送 SYN ACK 报文给客户端，源端口为 TCP 21 端口，目的端口为 FTP 客户端使用的随机端口号 1174；

3) FTP 客户端收到 FTP 服务器发送的 SYN ACK 报文后，向 FTP 服务器回送一个 ACK 报文，完成 TCP 三次握手建立 FTP 控制连接。

- **阶段二：被动方式参数传递**

4) 当 FTP 客户端希望请求文件列表或者需要同服务器进行文件传输时，FTP 客户端会通过已经建立好的控制通道向服务器发送 PASV 命令，告诉服务器进入被动模式。服务器对客户端的 PASV 命令应答，应答中包含了服务器的 IP 地址和一个临时端口信息。在图中，IP 地址是 192.168.0.10，端口号是 20*256+245=5365。

- **阶段三：建立数据通道 TCP 连接**

5) 此时，FTP 客户端已经得知 FTP 服务器使用的临时端口号是 5365。FTP 客户端以随机选择的临时端口号（图中是 3789）作为源端口，向 FTP 服务器的端口 5365 发送一个 SYN 报文，主动建立 TCP 连接；

6) FTP 服务器端发送 SYN ACK 给 FTP 客户端，目的端口为客户端自己选择的端口 3789，源端口为 5365；

7) FTP 客户端向 FTP 服务器端发送 ACK 消息，完成 TCP 三次握手，建立数据通道的 TCP 连接。

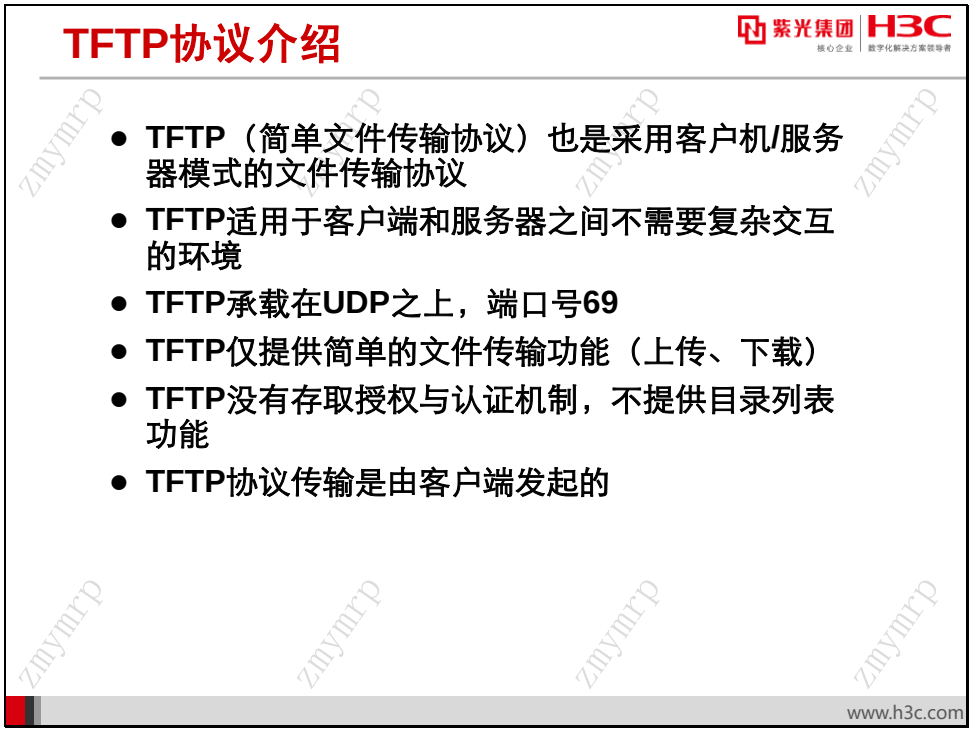
- **阶段四：数据传输**

8) 数据通道连接建立后，FTP 客户端与 FTP 服务器利用该通道进行数据的传输；

9) 数据传输完毕后，由发送数据的一方发送 FIN 报文,关闭这条数据连接。如果 FTP 客户端需要打开新的数据连接，则可以通过控制通道发送相关命令再次建立新的数据传输通道。

18.3 TFTP协议

18.3.1 TFTP 协议介绍



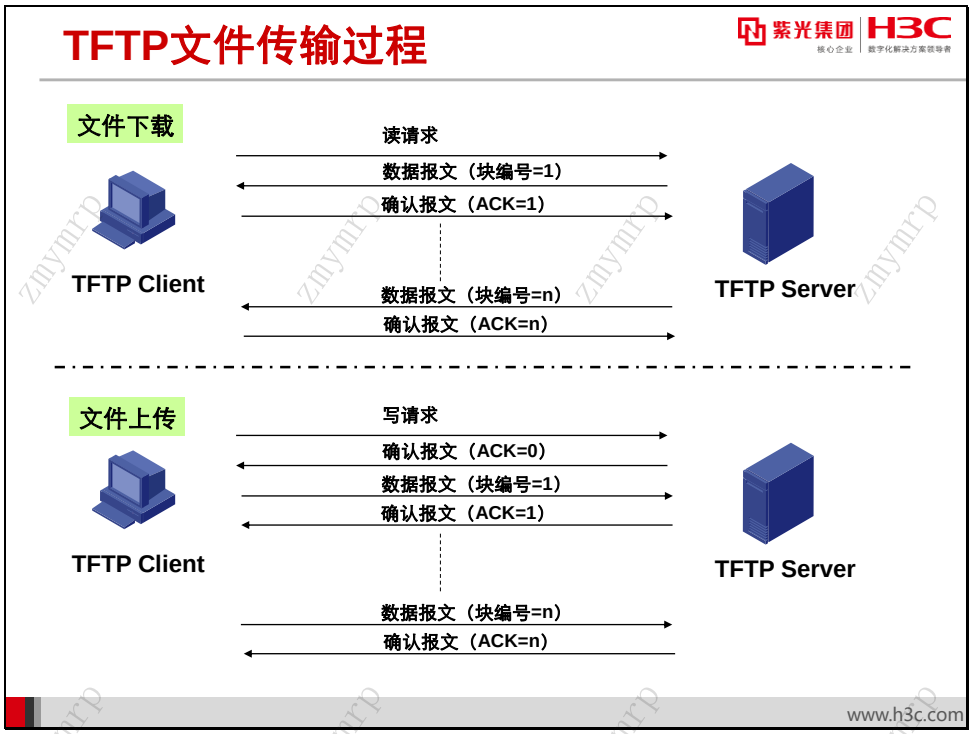
TFTP（Trivial File Transfer Protocol，简单文件传输协议）也是用于在远端服务器和本地主机之间传输文件的，相对于 FTP，TFTP 没有复杂的交互存取接口和认证控制，适用于客户端和服务器之间不需要复杂交互的环境。

TFTP 采用客户端/服务器设计方式，承载在 UDP 协议上，TFTP 服务器使用众所周知的端口号 69 侦听 TFTP 连接。由于 UDP 本身不能提供可靠的数据传输，因此 TFTP 使用自己设计的超时重传机制确保数据正确传送。TFTP 只能提供简单的文件传输能力，包括文件的上传和下载。TFTP 也不像 FTP 那样拥有一个庞大的命令集，不支持文件目录列表功能，也不能对用户的身份进行验证和授权。

TFTP 协议传输是由客户端发起的。当需要下载文件时，由客户端向 TFTP 服务器发送读请求包，然后从服务器接收数据，并向服务器发送确认；当需要上传文件时，由客户端向 TFTP 服务器发送写请求包，然后向服务器发送数据，并接收服务器的确认。

与 FTP 类似，TFTP 传输文件有两种模式：netascii 模式和 octet 模式。octet 传输模式对应于 FTP 中的二进制流模式，用于传输程序文件；netascii 模式对应于 FTP 中的 ASCII 模式，用于传输文本文件。

18.3.2 TFTP 文件传输过程



TFTP 进行文件传输时，将待传输文件看成由多个连续的文件块组成。每一个 TFTP 数据报文中包含一个文件块，同时对应一个文件块编号。每次发完一个文件块后就等待对方的确认，确认时应指明所确认的块编号。发送方发完数据后如果在规定的时间内收不到对端的确认那么发送方就要重新发送数据。发送确认的一方如果在规定时间内没有收到下一个文件块数据，则重发确认报文。这种方式可以确保文件的传送不会因某一数据的丢失而失败。

每次 TFTP 发送的数据报文中包含的文件块大小固定为 512 字节，如果文件长度恰好是 512 字节的整数倍，那么在文件传送完毕后，发送方还必须在最后发送一个不包含数据的数据报文，用来表明文件传输完毕。如果文件长度不是 512 字节的整数倍，那么最后传送的数据报文所包含的文件块肯定小于 512 字节，这正好作为文件结束的标志。

TFTP 的文件传输过程以 TFTP 客户端向 TFTP 服务器发送一个读请求或写请求开始。读请求表示 TFTP 客户端需要从 TFTP 服务器下载文件，写请求表示客户端需要向服务器上传文件。

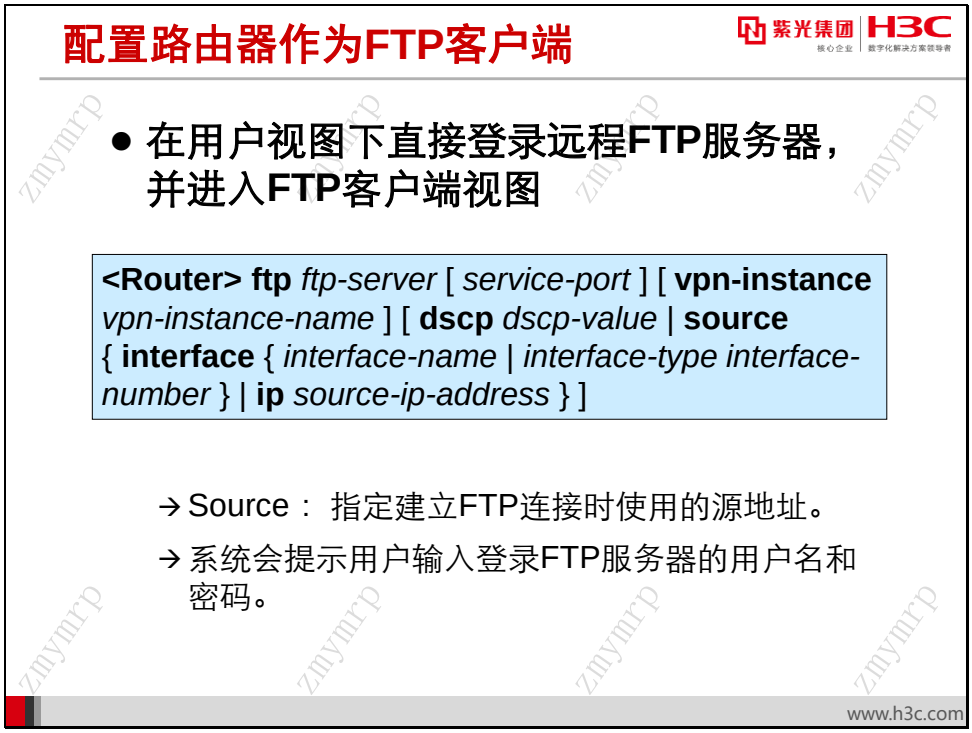
TFTP 客户端需要从 TFTP 服务器下载文件时，会向 TFTP 服务器发送一个读请求报文，包含需要下载的文件名信息和文件传输的模式（netascii 或 octet）。如果这个文件可以被客户端下载，那么服务器回应一个数据报文，报文中包括文件的第一个文件块，块编号为 1。客户端收到块编号为 1 的数据报文后，返回一个确认报文，报文中的块编号为 1。服务器收到确认后继续发送块编号为 2 的数据报文，客户端回应块编号为 2 的确认报文。这个过程周而复始，直至文件全部传输完毕。除了最后一个数据报文可含有不足 512 字节的数据，其他每个数据报

文均含有 512 字节的数据。当客户端收到一个不足 512 字节的数据报文后，就知道它收到了最后一个数据分组。

TFTP 客户端需要向 TFTP 服务器上传文件时，会向 TFTP 服务器发送一个写请求报文，包含需要在服务器上保存的文件名信息和文件传输模式（netascii 或 octet）。如果这个文件可以被客户端上传，那么服务器回应一个块编号为 0 的确认报文。客户端继续发送块编号为 1 的数据报文，服务器返回块编号为 1 的确认报文。然后客户端继续发送块编号为 2 的数据报文，服务器返回块编号为 2 的确认报文。以此类推，直至文件全部传输完毕。

18.4 配置FTP与TFTP

18.4.1 FTP 客户端配置方法



路由器可以作为 FTP 客户端，建立设备与远程 FTP 服务器的连接，访问远程 FTP 服务器上的文件。

在用户视图下用以下命令来登录远程 FTP 服务器，并进入 FTP 客户端视图：

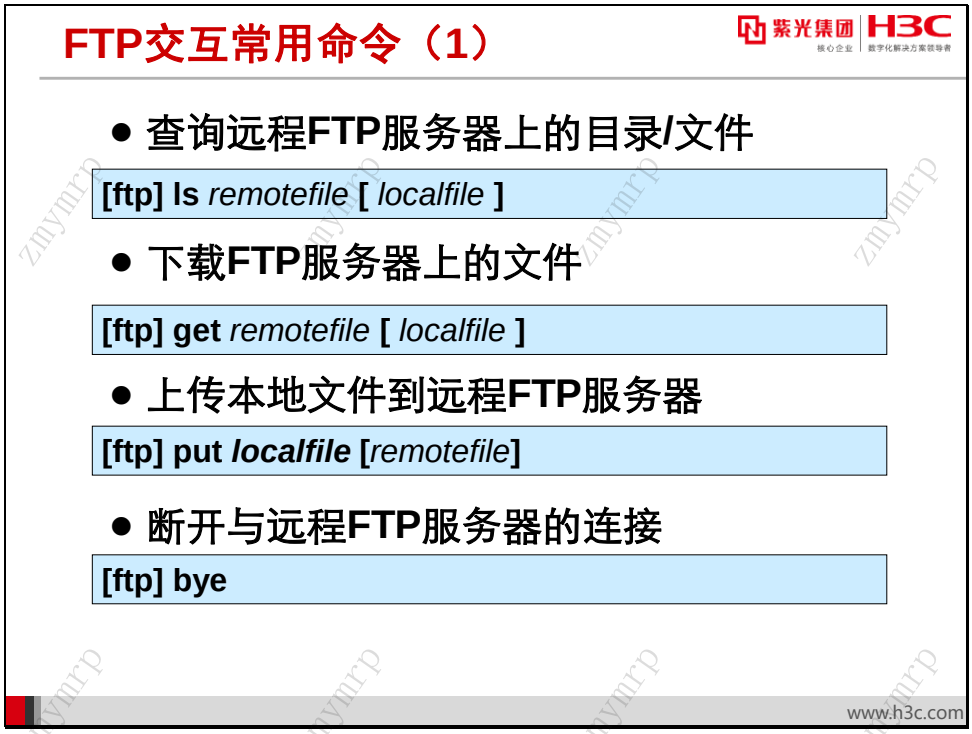
ftp *ftp-server* [*service-port*] [**vpn-instance** *vpn-instance-name*] [**dscp** *dscp-value* | **source** { **interface** { *interface-name* | *interface-type interface-number* } | **ip** *source-ip-address* }]

命令中常见参数含义如下：

- *ftp-server*：FTP 服务器的主机名或 IP 地址。
- *service-port*：远端设备提供 FTP 服务的 TCP 端口号，取值范围为 0～65535，缺省值为 21。
- **vpn-instance** *vpn-instance-name*：指定 FTP 服务器所属的 VPN。
- **source** { **interface** *interface-type interface-number* | **ip** *source-ip-address* }：指定建立 FTP 连接时使用的源地址

在上述命令中，如果不指定任何参数，则只进入 FTP 客户端视图，不登录 FTP 服务器。

如果指定参数，系统会提示用户输入登录 FTP 服务器的用户名和密码。如果用户名和密码正确，则登录成功，并进入 FTP 客户端视图；否则，登录失败。



在登录到 FTP 服务器后，通常会查看服务器上的目录和文件，以确定需要下载的文件名。在 FTP 视图下查看 FTP 服务器上目录和文件：

ls *remotefile* [*localfile*]

然后可以指定所需要下载的文件和下载到本地后的文件名。其命令如下：

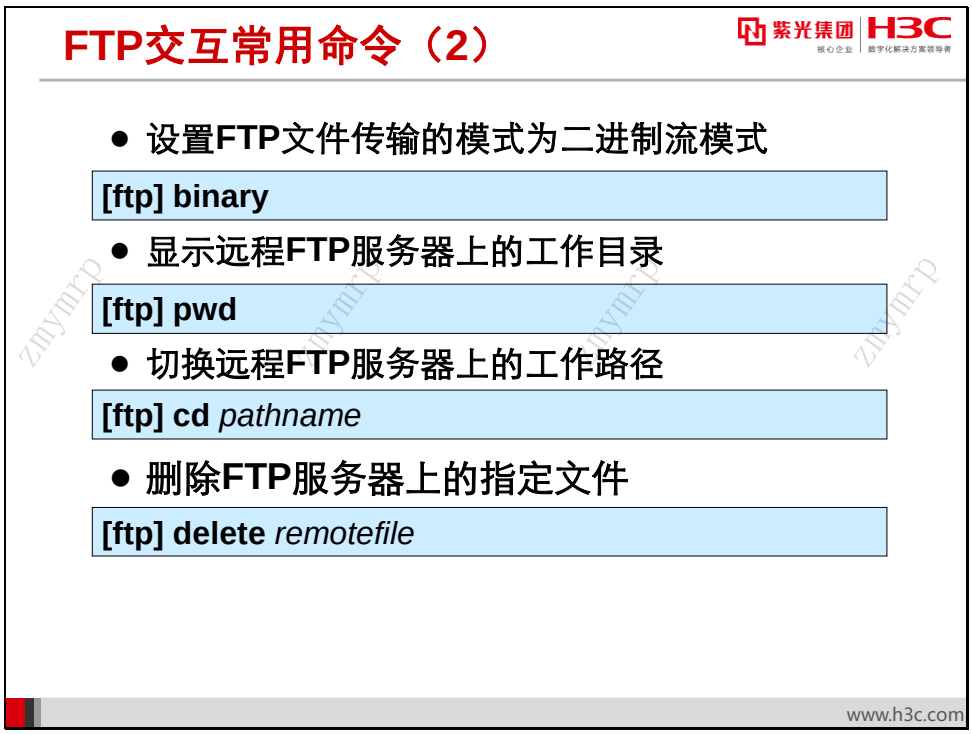
get *remotefile* [*localfile*]

也可以上传本地文件到远程 FTP 服务器上。命令如下：

put *localfile* [*remotefile*]

在下载完成后，在 FTP 视图下用命令断开与 FTP 服务器之间的连接。

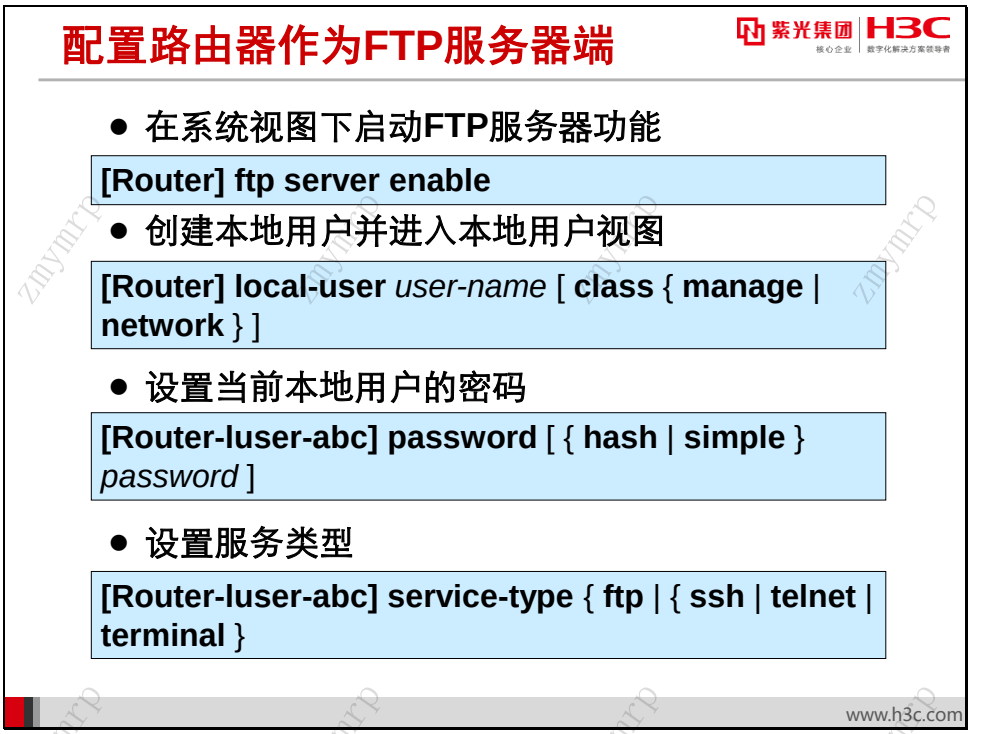
bye



其它会经常使用的命令有：

命令	操作
binary	设置FTP文件传输的模式为二进制流模式
pwd	显示远程FTP服务器上的工作目录
cd *pathname*	切换远程FTP服务器上的工作路径
delete *remotefile*	删除FTP服务器上的指定文件

18.4.2 FTP 服务器端配置



当路由器作为 FTP 服务器时，可进行如下配置。

第1步：在系统视图下启动 FTP 服务器功能；

ftp server enable

因为缺省情况下，FTP 服务器功能处于关闭状态，所以必须使能 FTP 服务。

第2步：创建本地用户并设置相应的密码、服务类型等参数。

创建本地用户并进入本地用户视图：

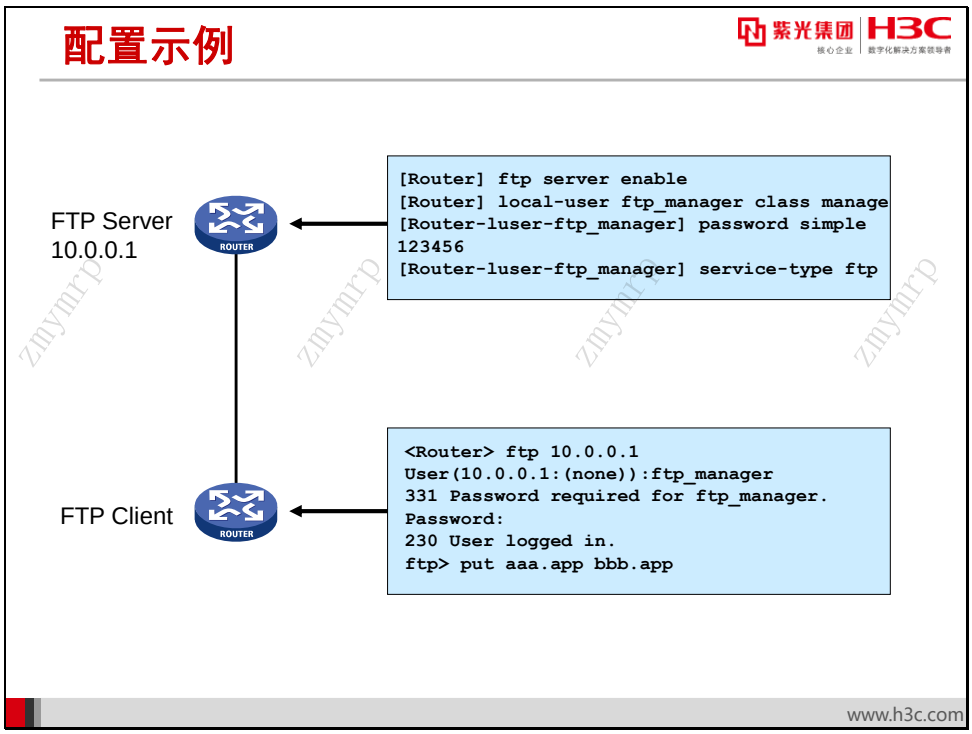
local-user *user-name* [**class** { **manage** | **network** }]

在本地用户视图下设置当前本地用户的密码：

password [{ **hash** | **simple** }*password*

在本地用户视图下设置服务类型：

service-type { **ftp** | { **ssh** | **telnet** | **terminal** } }



上图为配置路由器作为服务器端和客户端的示例。图中作为 FTP 服务器的路由器接口 IP 地址是 10.0.0.1。

配置路由器作为 FTP 服务器端：

```
[Router] ftp server enable
[Router] local-user ftp_manager class manage
[Router-luser-ftp_manager] password simple 123456
[Router-luser-ftp_manager] service-type ftp
```

所配置的用户名为 ftp_manager，密码为 123456，格式为未加密的明文，FTP 服务的目录为缺省目录。

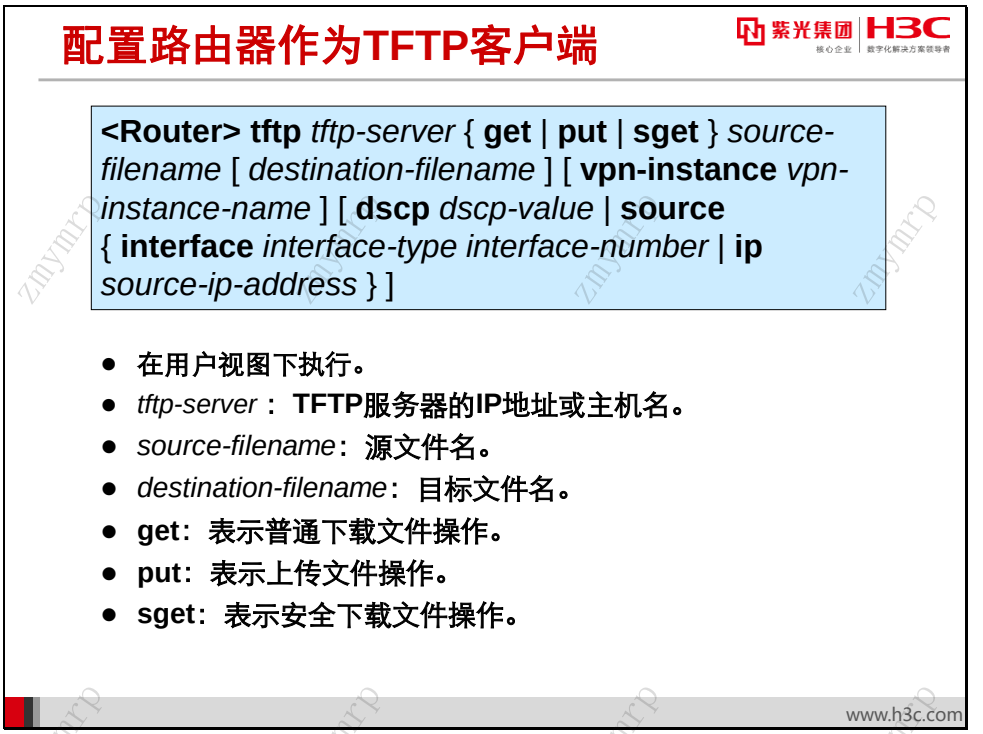
在客户端登录 FTP 服务器端，如下：

```
<Router> ftp 10.0.0.1
User(10.0.0.1:(none)):ftp_manager
331 Password required for ftp_manager.
Password:
230 User logged in.
```

然后上传本地文件 aaa.app 至服务器端，并在服务器端修改其文件名为 bbb.app。

```
ftp> put aaa.app bbb.app
```

18.4.3 TFTP 客户端配置



当路由器作为 TFTP 客户端时，可以把本地文件上传到 TFTP 服务器，还可以从 TFTP 服务器下载文件到本地。下载又分为两种：

- 普通下载。在这种方式下，设备将获取的远端文件直接写到存储设备中。这样如果是覆盖性下载系统文件，而且下载失败（如网络断开等原因），则原系统文件已被删除，设备将无法正常启动。
- 安全下载。在这种方式下，设备将获取的远端文件先保存到内存中，等用户文件全部接收完毕，才将它写到存储设备中。这样如果系统文件下载失败（如网络断开等原因），因为原有的系统文件没有被覆盖，设备仍能够启动。这种方法安全系数较高，但需要较大的内存空间。

当操作启动文件或配置文件等重要文件时，建议采用安全下载。

配置路由器作为 TFTP 客户端的命令为：

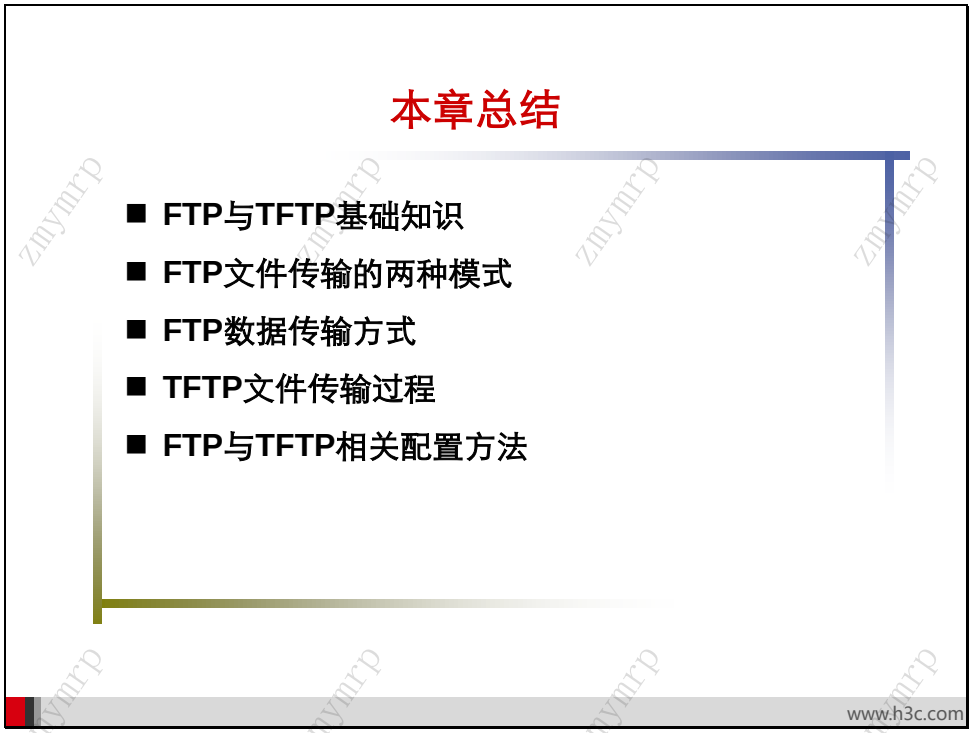
tftp *tftp-server* { **get** | **put** | **sget** } *source-filename* [*destination-filename*] [**vpn-instance** *vpn-instance-name*] [**dscp** *dscp-value* | **source** { **interface** *interface-type interface-number* | **ip** *source-ip-address* }]

其中参数含义如下：

- *tftp-server*：TFTP 服务器的 IP 地址或主机名。
- *source-filename*：源文件名。

- *destination-filename*：目标文件名。
- **vpn-instance** *vpn-instance-name*：指定 TFTP 服务器所属的 VPN。
- **dscp** *dscp-value*：指定设备发送的 TFTP 报文中携带的 DSCP 优先级的取值，取值范围为 0～63，缺省值为 0。
- **get**：表示普通下载文件操作。
- **put**：表示上传文件操作。
- **sget**：表示安全下载文件操作。
- **ip** *source-ip-address*：当前 TFTP 客户端发送报文所使用的源 IP 地址。此地址必须是设备上已配置的 IP 地址。
- **interface** *interface-type interface-number*：当前 TFTP 客户端传输使用的源接口，包括接口类型和接口编号。此接口下配置的主 IP 地址即为发送报文的源地址。

18.5 本章总结

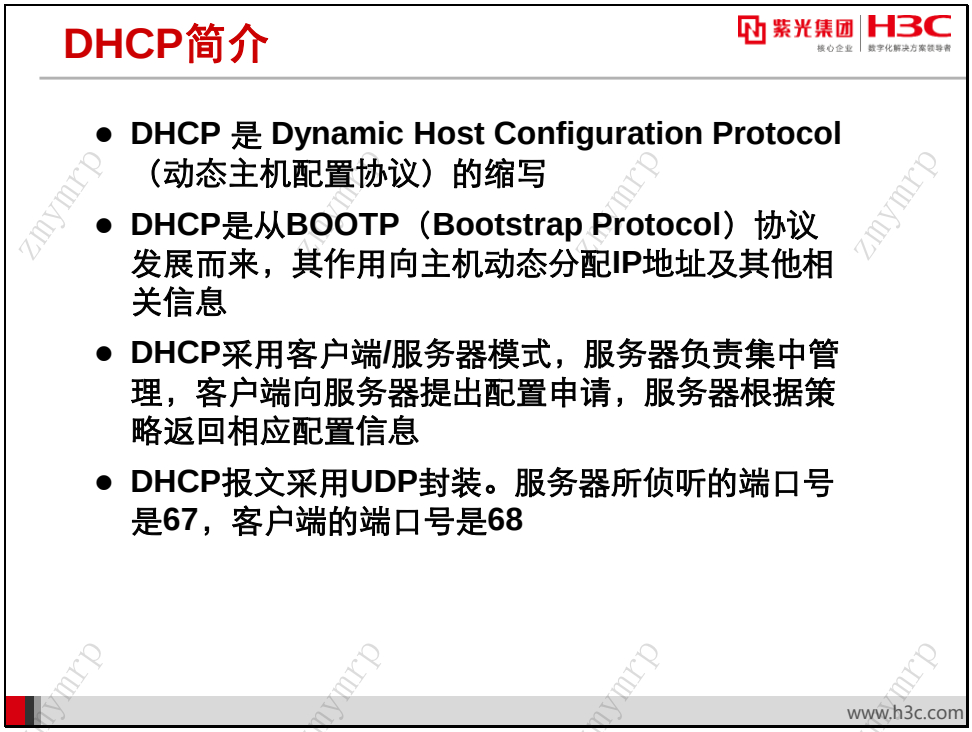

本章总结
FTP与TFTP基础知识
FTP文件传输的两种模式
FTP数据传输方式
TFTP文件传输过程
FTP与TFTP相关配置方法
www.h3c.com

第19章 DHCP

在 TCP/IP 网络中，主机和网络设备需要具有 IP 地址、掩码等信息才能正常的工作。在小型网络中，管理员可以手工为每台主机设定 IP 地址、掩码、网关等。当网络规模增大时，管理员所需配置工作量也相应增大，且手工配置容易造成 IP 地址配置错误。DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）的作用是为局域网中的每台计算机自动分配 TCP/IP 信息，包括 IP 地址、子网掩码、网关，以及 DNS 服务器等。其优点是终端主机无须配置、网络维护方便。本章主要讲述了 DHCP 协议的特点及其原理，并介绍了 DHCP 中继，最后介绍如何在 H3C 路由器上配置 DHCP 服务。

19.1 本章目标

19.2 DHCP简介



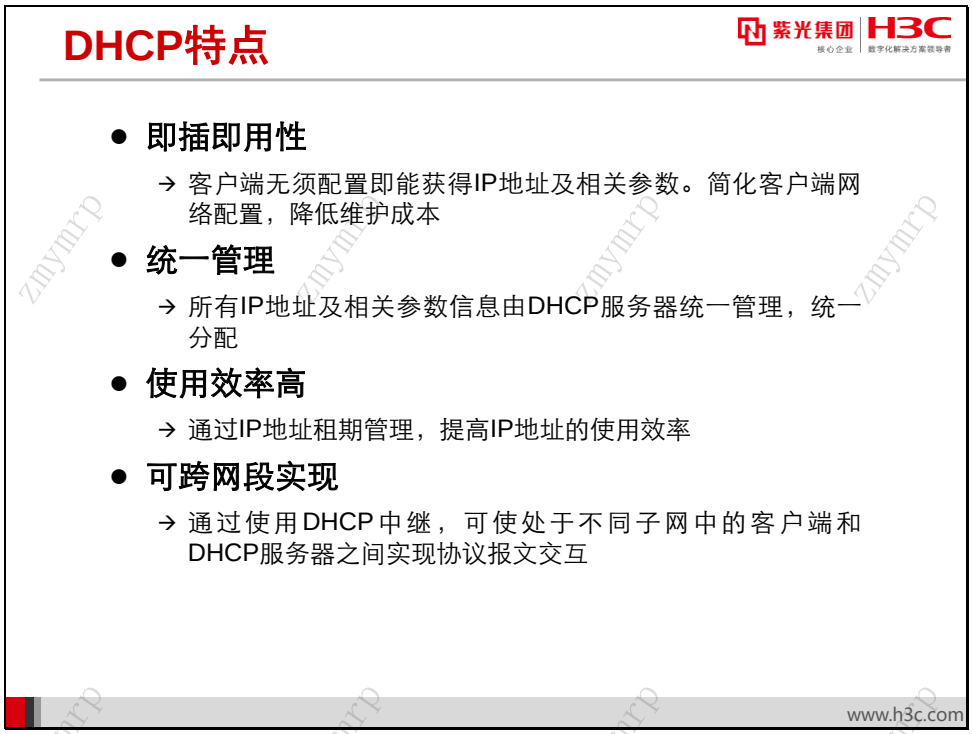
DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）的前身是 BOOTP（Bootstrap Protocol）。早期网络中大量使用无盘工作站，这种工作站没有硬盘，启动时从 BOOTROM 初始化系统并连接到网络上，BOOTP 协议可以自动地为这些主机设定 TCP/IP 环境。但 BOOTP 协议有一个缺点：在设定前必须事先获得客户端的硬件地址，而且硬件地址与 IP 地址的对应是静态绑定的。换而言之，BOOTP 没有“动态性”，在有限的 IP 资源环境中，使用 BOOTP 会造成 IP 地址极大的浪费。

DHCP 可以说是 BOOTP 的增强版本，能够动态地为主机分配 IP 地址，并设定主机的其他信息，例如缺省网关、DNS 服务器地址等。而且 DHCP 完全向下兼容 BOOTP，BOOTP 客户端也能够在 DHCP 的环境中良好运行。

DHCP 运行在客户端/服务器模式，服务器负责集中管理 IP 配置信息（包括 IP 地址、子网掩码、缺省网关、DNS 服务器地址等）。客户端主动向服务器提出请求，服务器根据策略返回相应配置信息；客户端使用从服务器获得的配置信息进行数据通讯。

DHCP 协议报文采用 UDP 方式封装。DHCP 服务器所侦听的端口号是 67，客户端的端口号是 68。

19.2.1 DHCP 协议特点



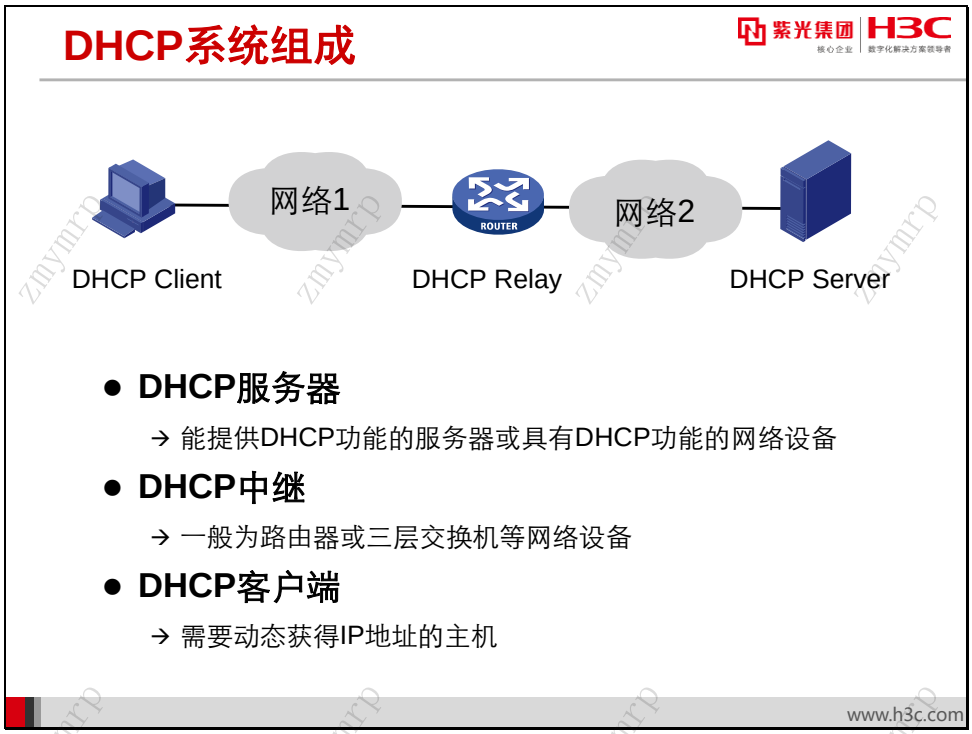
在一个通过 DHCP 实现 IP 地址分配和管理的网络中，DHCP 客户端无需配置即可自动获得所需要的网络参数，网络管理人员和维护人员的工作压力得到了很大程度上的减轻。

DHCP 服务器对客户端的所有配置信息进行统一管理。通过监听 DHCP 客户端的请求消息 DHCP 服务器给予相应的回复，回应给客户端的配置信息包括 IP 地址、子网掩码、缺省网关等参数。

DHCP 可以设定所分配 IP 地址资源的使用期限。使用期限到期后的 IP 地址资源可以由 DHCP 服务器进行回收。相比 BOOTP 协议，DHCP 可以更加有效的利用 IP 地址资源。

通常情况下，DHCP 采用广播方式实现报文交互，DHCP 服务仅局限在本地网段。如果需要跨本地网段实现 DHCP，需要使用 DHCP 中继技术实现。

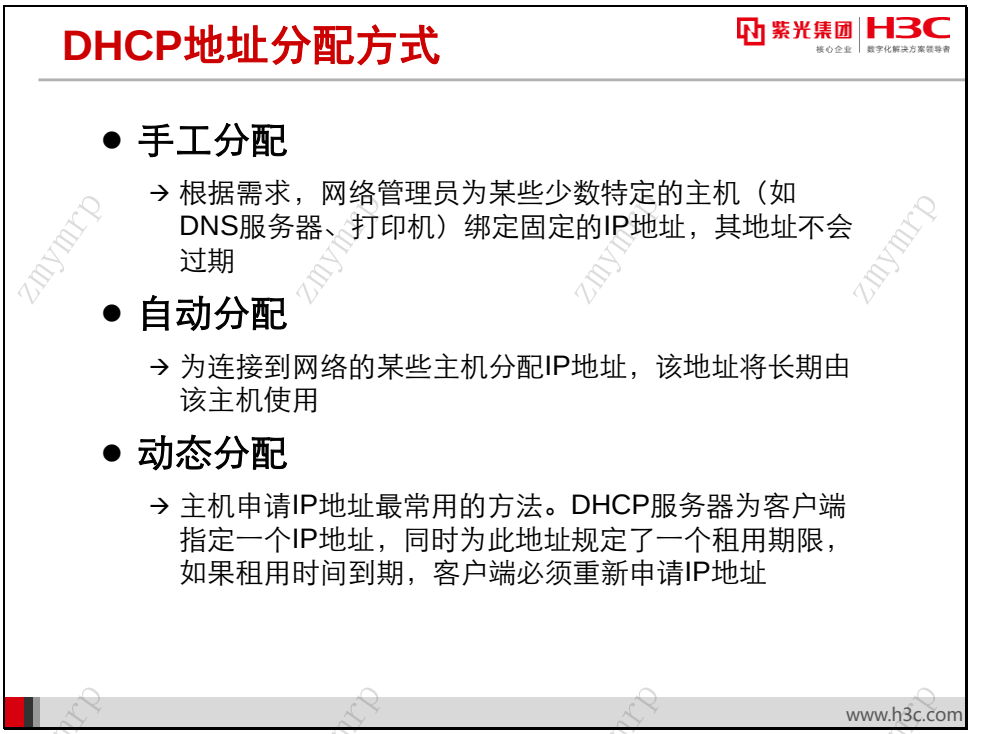
19.2.2 DHCP 系统组成



在 DHCP 的讨论中最常见的多的几个术语包括：

- **DHCP 服务器**：DHCP 服务器提供网络设置参数给 DHCP 客户端，通常是一台能提供 DHCP 服务功能的服务器或网络设备。
- **DHCP 中继**：在 DHCP 服务器和 DHCP 客户端之间转发跨网段 DHCP 报文的设备，通常是网络设备。
- **DHCP 客户端**：DHCP 客户端通过 DHCP 服务器来获取网络配置参数，通常是一台主机或网络设备。

19.2.3 DHCP 地址分配方式



针对客户端的不同需求，DHCP 提供三种 IP 地址分配方式：

- **手工分配**

由管理员为少数特定 DHCP 客户端（如 DNS、WWW 服务器、打印机等）静态绑定固定的 IP 地址。通过 DHCP 服务器将所绑定的固定 IP 地址分配给 DHCP 客户端。此 IP 地址永久被该客户端使用，其它主机无法使用。

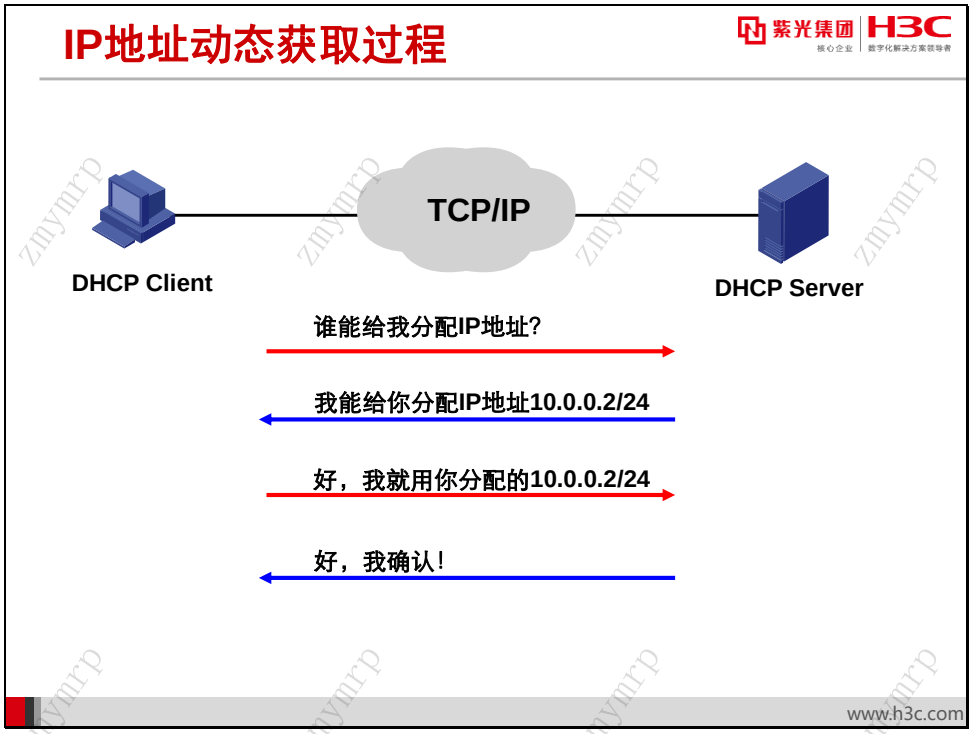
- **自动分配**

DHCP 服务器为 DHCP 客户端动态分配租期为无限长的 IP 地址。只有客户端释放该地址后，该地址才能被分配给其他客户端使用。

- **动态分配**

DHCP 服务器为 DHCP 客户端分配具有一定有效期限的 IP 地址。如果客户端没有及时续约，到达使用期限后，此地址可能会被其它客户端使用。绝大多数客户端得到的都是这种动态分配的地址。

19.3 IP地址动态获取过程



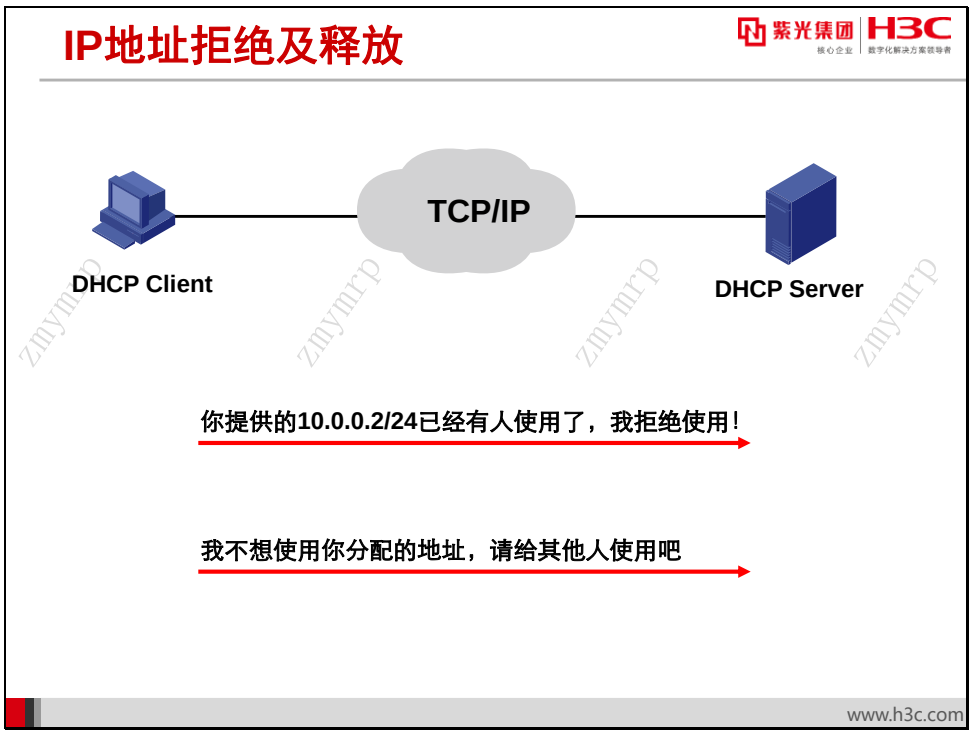
DHCP 客户端从 DHCP 服务器动态获取 IP 地址，主要通过四个阶段进行：

1) 发现阶段，即 DHCP 客户端寻找 DHCP 服务器的阶段。客户端以广播方式发送 DHCP-DISCOVER 报文。

2) 提供阶段，即 DHCP 服务器提供 IP 地址的阶段。DHCP 服务器接收到客户端的 DHCP-DISCOVER 报文后，根据 IP 地址分配的优先次序选出一个 IP 地址，与其他参数一起通过 DHCP-OFFER 报文广播发送给客户端。

3) 选择阶段，即 DHCP 客户端选择 IP 地址的阶段。如果有多台 DHCP 服务器向该客户端发来 DHCP-OFFER 报文，客户端只接受第一个收到的 DHCP-OFFER 报文，然后以广播方式发送 DHCP-REQUEST 报文，该报文中包含 DHCP 服务器在 DHCP-OFFER 报文中分配的 IP 地址。

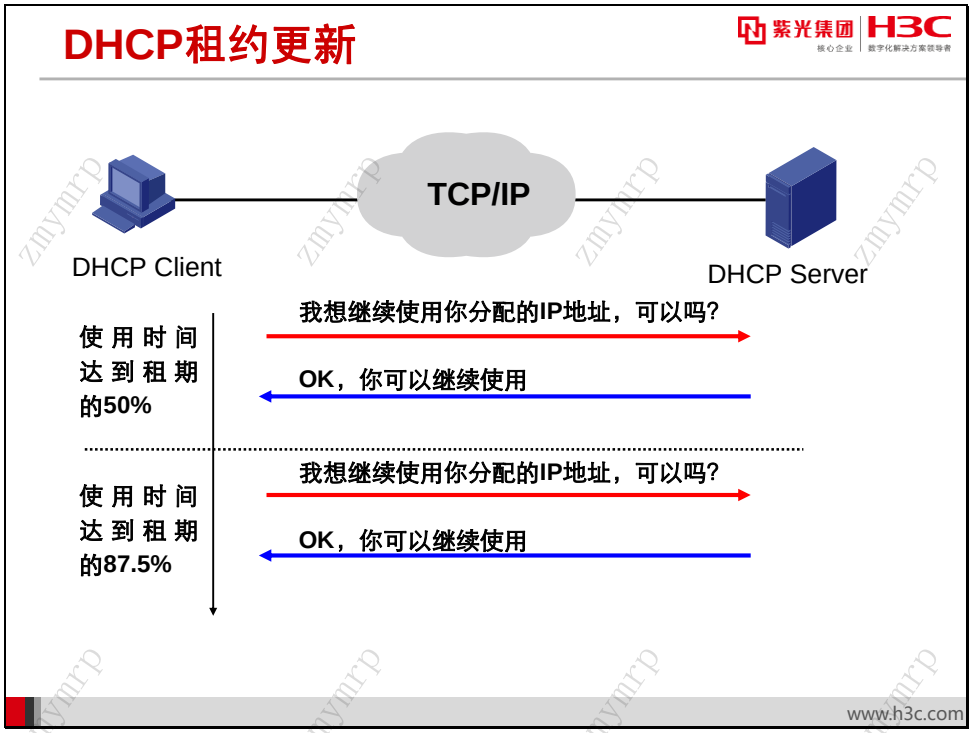
4) 确认阶段，即 DHCP 服务器确认 IP 地址的阶段。DHCP 服务器收到 DHCP 客户端发来的 DHCP-REQUEST 报文后，只有 DHCP 客户端选择的服务器会进行如下操作：如果确认将地址分配给该客户端，则返回 DHCP-ACK 报文；否则返回 DHCP-NAK 报文，表明地址不能分配给该客户端。



当 DHCP 客户端收到 DHCP 服务器包含配置参数的 DHCP Ack 报文后，会发送免费 ARP 报文进行探测，目的地址为 DHCP 服务器指定分配的 IP 地址，如果探测到此地址没有被使用，那么 DHCP 客户端就会使用此地址并且配置完毕。如果 DHCP 客户端探测到地址已经被分配使用，DHCP 客户端会发送给 DHCP 服务器 DHCP Decline 报文，表明拒绝使用该地址，并且重新开始 DHCP 进程。

当 DHCP 客户端选择放弃它的 IP 地址或租期时，它将向 DHCP 服务器发送 DHCP Release 报文。

19.4 DHCP租约更新

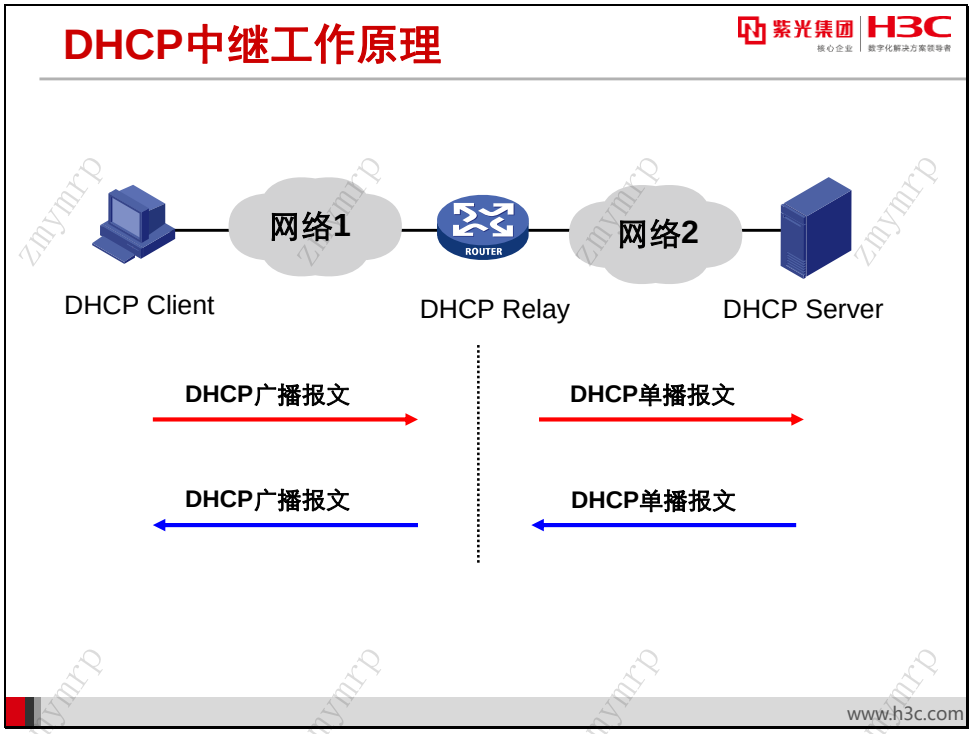


如果采用动态地址分配方式，则 DHCP 服务器分配给客户端的 IP 地址有一定的租借期限，当租借期满后服务器会收回该 IP 地址。如果 DHCP 客户端希望继续使用该地址，需要更新 IP 地址租约。

在 DHCP 客户端的 IP 地址租约期限达到一半时间时，DHCP 客户端会向为它分配 IP 地址的 DHCP 服务器单播发送 DHCP-REQUEST 报文，以进行 IP 租约的更新。如果客户端可以继续使用此 IP 地址，则 DHCP 服务器回应 DHCP-ACK 报文，通知 DHCP 客户端已经获得新 IP 租约；如果此 IP 地址不可以再分配给该客户端，则 DHCP 服务器回应 DHCP-NAK 报文，通知 DHCP 客户端不能获得新的租约。

如果在租约的一半时间进行的续约操作失败，DHCP 客户端会在租约期限达到 7/8 时，广播发送 DHCP-REQUEST 报文进行续约。DHCP 服务器的处理方式同上，不再赘述。

19.5 DHCP中继介绍



由于在 IP 地址动态获取过程中采用广播方式发送报文，因此 DHCP 只适用于 DHCP 客户端和服务器处于同一个子网内的情况。为进行动态主机配置，需要在所有网段上都设置一个 DHCP 服务器，这显然是很不经济的。

DHCP 中继功能的引入解决了这一难题。客户端可以通过 DHCP 中继与其他子网中 DHCP 服务器通信，最终获取到 IP 地址。这样，多个网络上的 DHCP 客户端可以使用同一个 DHCP 服务器，既节省了成本，又便于进行集中管理。

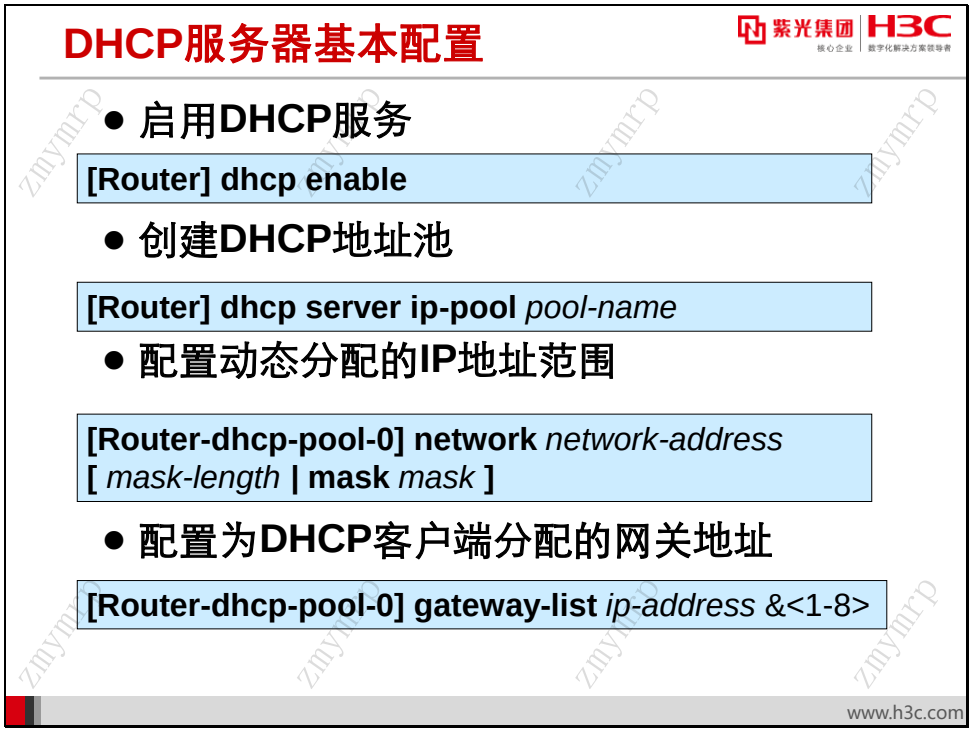
DHCP 中继的工作原理：

1) 具有 DHCP 中继功能的网络设备收到 DHCP 客户端以广播方式发送的 DHCP-DISCOVER 或 DHCP-REQUEST 报文后，根据配置将报文单播转发给指定的 DHCP 服务器。

2) DHCP 服务器进行 IP 地址的分配，并通过 DHCP 中继将配置信息广播发送给客户端，完成对客户端的动态配置。

19.6 DHCP服务器配置

19.6.1 DHCP 服务器基本配置



在大型网络中，客户端通常由专门的 DHCP 服务器分配 IP 地址。在小型网络中，可以在路由器上启用 DHCP 服务，使路由器具有 DHCP 服务器功能，从而给客户端分配地址及相关参数。

在路由器上配置 DHCP 服务器的步骤如下：

第1步：在系统视图下启用 DHCP 服务

dhcp enable

只有启用 DHCP 服务后，其它相关的 DHCP 配置才能生效。

第2步：在系统视图下创建 DHCP 地址池

dhcp server ip-pool *pool-name*

第3步：在 DHCP 地址池视图下配置地址范围

network *network-address* [*mask-length* | **mask** *mask*]

通常情况下，采用动态地址分配方式进行地址分配。对于采用动态地址分配方式的地址池，需要配置该地址池可分配的地址范围，地址范围的大小通过掩码来设定。

第4步：在 DHCP 地址池视图下配置为 DHCP 客户端分配的网关地址

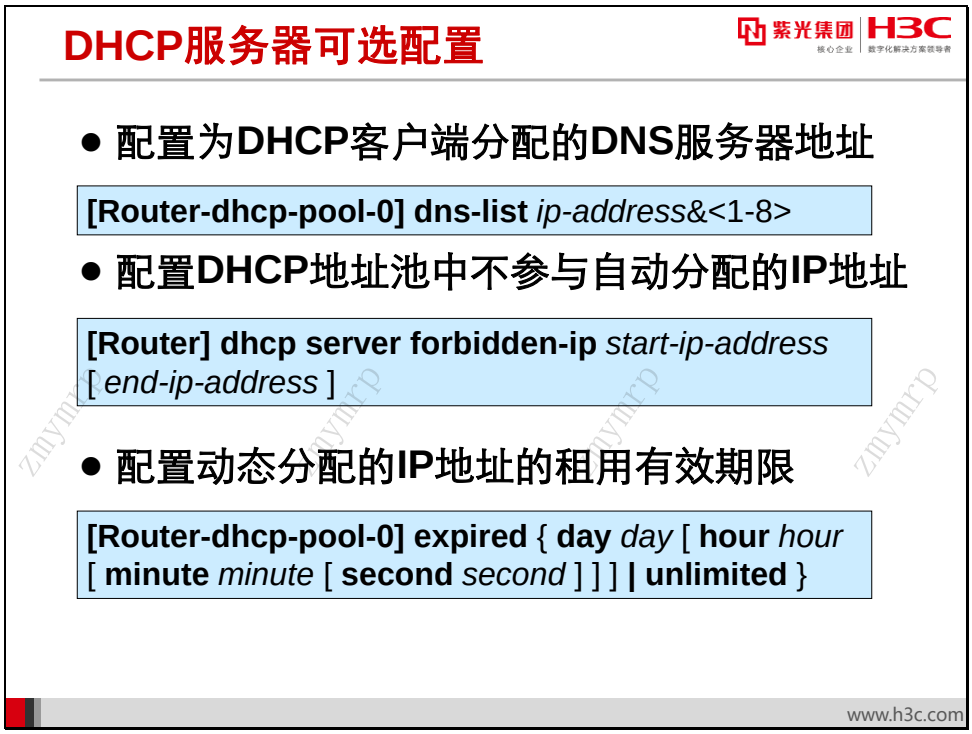
gateway-list *ip-address* &<1-8>

DHCP 客户端访问本子网以外的服务器或主机时，数据必须通过网关进行转发。DHCP 服务器可以在为客户端分配 IP 地址的同时指定网关的地址。

参数&<1-8>表示最多可以输入 8 个 IP 地址，每个 IP 地址之间用空格分隔。

通过以上配置，在客户端发送 DHCP 协议报文给 DHCP 服务器后，服务器会给客户端分配地址池里所配置的地址，并分配所指定的网关。

19.6.2 DHCP 服务器可选配置



通过域名访问 Internet 上的主机时，需要将域名解析为 IP 地址，这是通过 DNS（Domain Name System，域名系统）实现的。为了使 DHCP 客户端能够通过域名访问 Internet 上的主机，DHCP 服务器应在为客户端分配 IP 地址的同时指定 DNS 服务器地址。

在 DHCP 地址池视图下，配置为 DHCP 客户端分配的 DNS 服务器地址：

dns-list *ip-address*&<1-8>

DHCP 服务器在分配地址时，需要排除已经被占用的 IP 地址（如网关、DNS 服务器等）。否则，同一地址分配给两个客户端会造成 IP 地址冲突。

在系统视图下，配置 DHCP 地址池中哪些 IP 地址不参与自动分配：

dhcp server forbidden-ip *start-ip-address* [*end-ip-address*]

DHCP 服务器在分配地址时，可以指定所分配给客户端的地址租用期限。

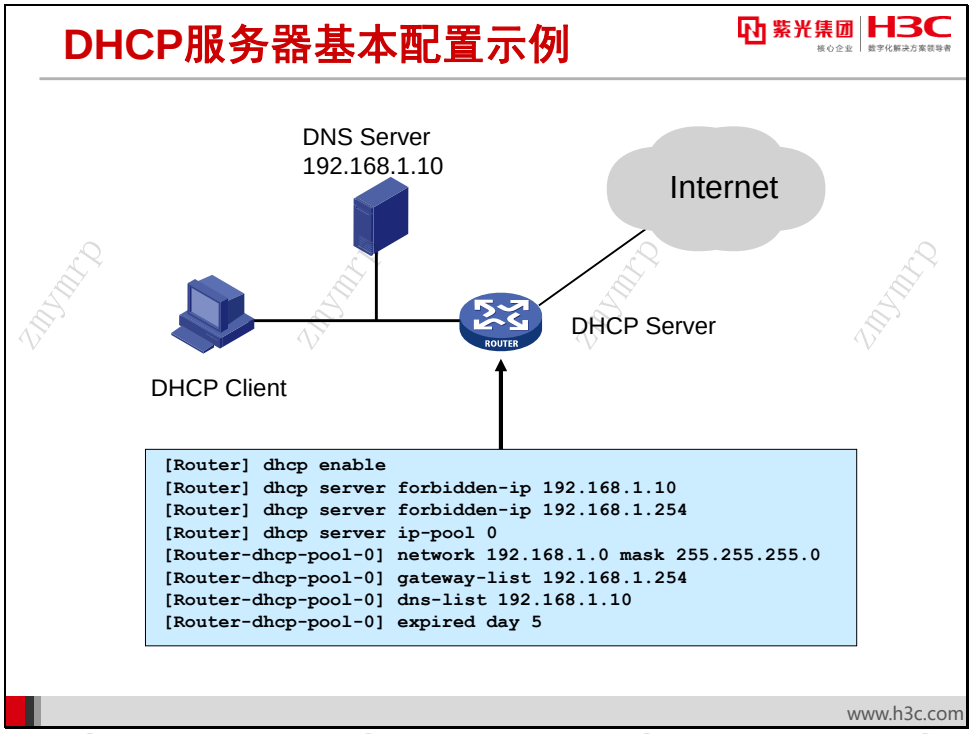
在 DHCP 地址池视图下，配置为 DHCP 客户端分配的 IP 地址的租用期限：

expired { **day** *day* [**hour** *hour* [**minute** *minute*] [**second** *second*]] | **unlimited** }

参数说明：

- **day** *day*：天数，取值范围为 0～365。
- **hour** *hour*：小时数，取值范围为 0～23。
- **minute** *minute*：分钟数，取值范围为 0～59。
- second *second*：指定租约过期的秒数，*second* 取值范围为 0～59。
- **unlimited**：有效期限为无限长。

19.6.3 DHCP 服务器配置示例



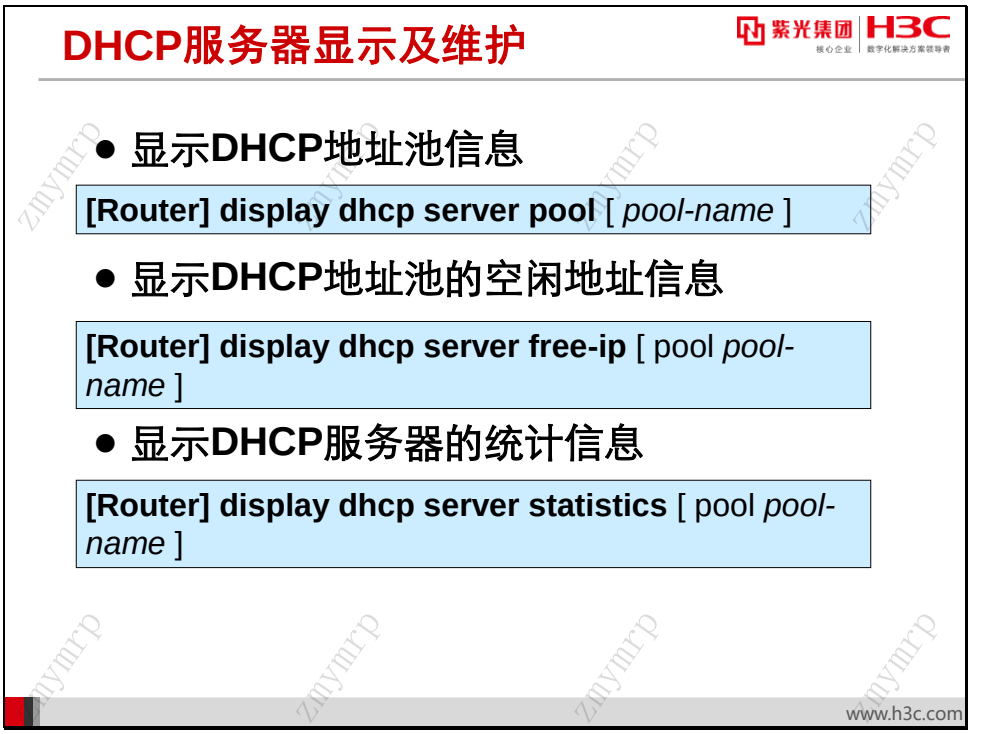
上图是在路由器上运行 DHCP 服务器的基本配置示例。图中路由器启用 DHCP 服务器功能，所使用的 DHCP 地址池是 192.168.1.0/24，池中地址的租用期限是 5 天。地址池中有 2 个地址不能被自动分配：地址 192.168.1.10 被 DNS 服务器固定使用，192.168.1.254 被分配给客户端作为网关。

配置路由器：

```
[Router] dhcp enable
[Router] dhcp server forbidden-ip 192.168.1.10
[Router] dhcp server forbidden-ip 192.168.1.254
[Router] dhcp server ip-pool 0
[Router-dhcp-pool-0] network 192.168.1.0 mask 255.255.255.0
[Router-dhcp-pool-0] gateway-list 192.168.1.254
```

```
[Router-dhcp-pool-0] dns-list 192.168.1.10
[Router-dhcp-pool-0] expired day 5
```

19.6.4 DHCP 服务器的显示及维护

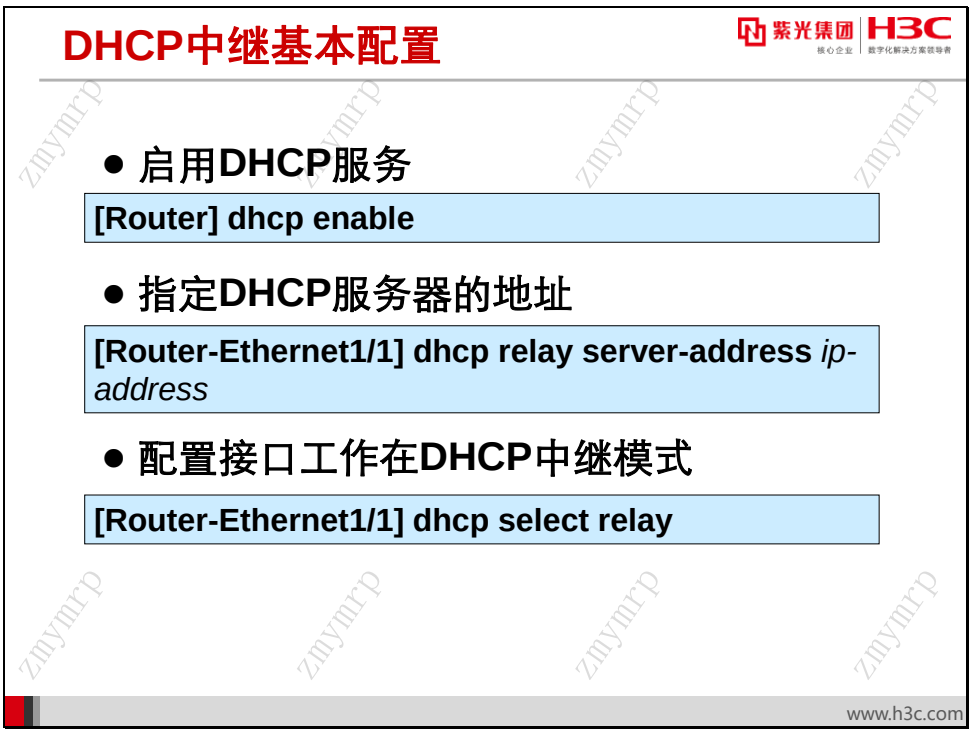


在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 DHCP 服务器的运行情况，通过查看显示信息验证配置的效果。图中为较常用的查看 DHCP 服务器的命令。

操作	命令
显示 DHCP 地址池的信息	**display dhcp server pool** [*pool-name*]
显示DHCP地址池的可用地址信息	**display dhcp server free-ip**
显示DHCP服务器的统计信息	**display dhcp server statistics** [pool *pool-name*]

19.7 DHCP中继配置

19.7.1 DHCP 中继基本配置



客户端和 DHCP 服务器如果不在同一个子网内，就需要用到 DHCP 中继功能来使网络设备转发 DHCP 协议报文。

在路由器上配置 DHCP 中继的步骤如下：

第1步：在系统视图下启用 DHCP 服务

dhcp enable

只有启用 DHCP 服务后，其它相关的 DHCP 配置才能生效。

第2步：在接口视图下指定 DHCP 服务器的地址

dhcp relay server-address *ip-address*

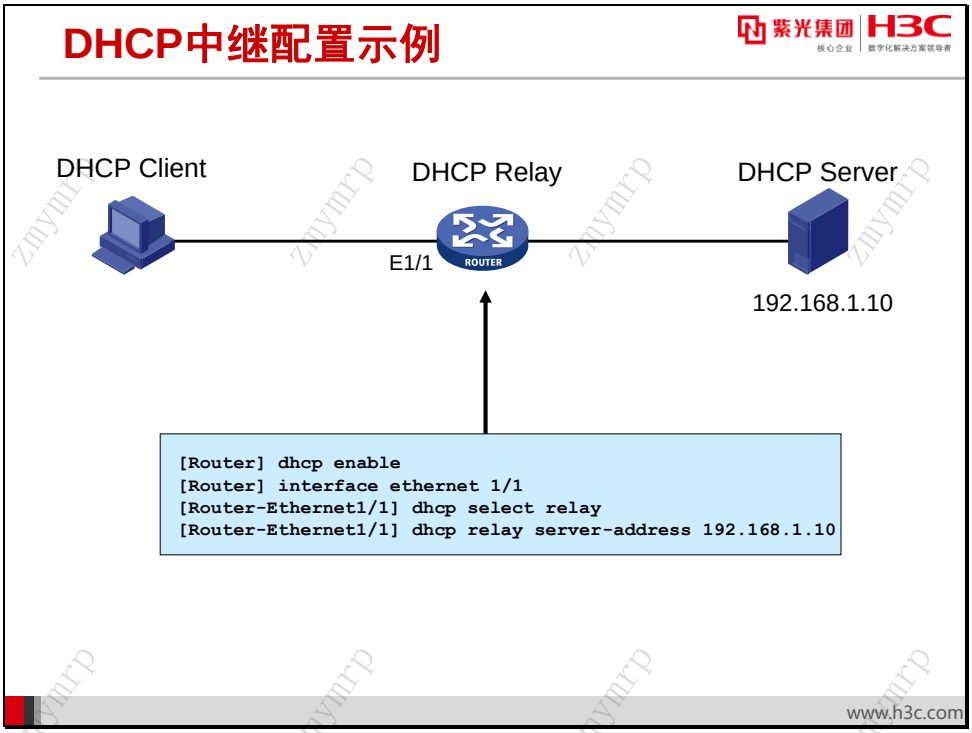
为了提高可靠性，可以在一个网络中设置多个DHCP服务器。DHCP中继上配置多个DHCP服务器后，DHCP 中继会将客户端发来的 DHCP 报文转发给所有的服务器。

第3步：在接口视图下配置接口工作在 DHCP 中继模式

dhcp select relay

缺省情况下，启用 DHCP 服务后，接口工作在 DHCP 服务器模式。所以需要使用上述命令来使接口工作在中继模式。

通过以上配置，路由器在收到客户端发出的 DHCP 协议报文后，会以单播形式转发给指定的 DHCP 服务器。



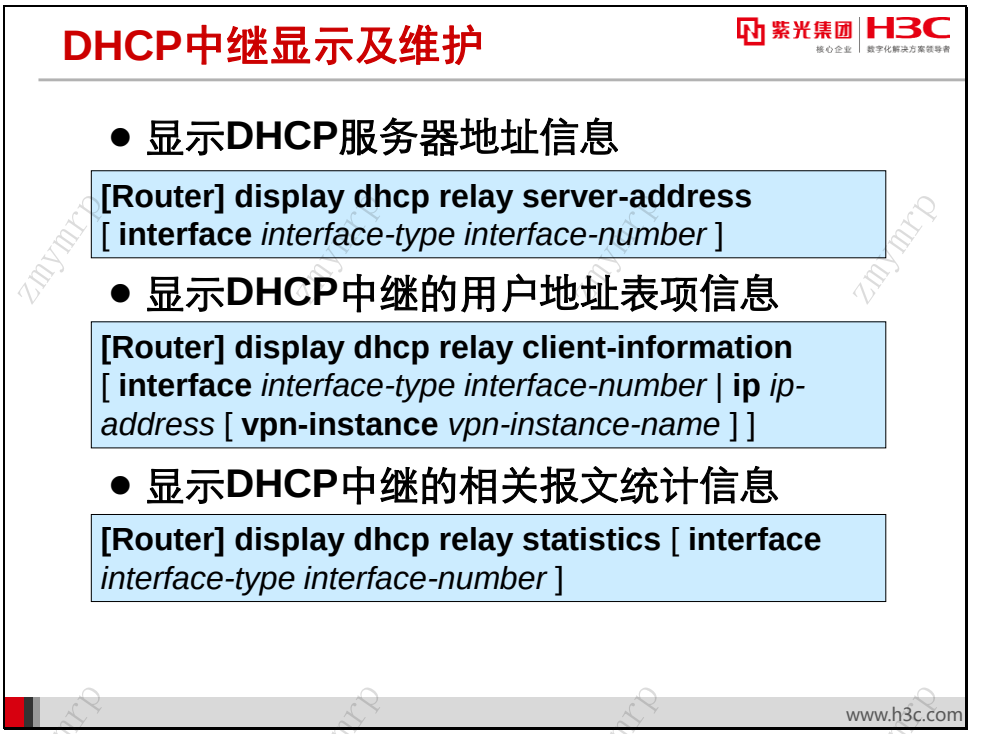
上图是在路由器上运行 DHCP 中继的配置示例。图中路由器的接口 E1/1 连接到客户端，DHCP 服务器的 IP 地址是 192.168.1.10。

配置路由器：

```
[Router] dhcp enable
[Router] interface ethernet 1/1
[Router-Ethernet1/1] dhcp select relay
[Router-Ethernet1/1] dhcp relay server-address 192.168.1.10
```

配置完成后，路由器能够中继客户端与服务器之间的 DHCP 协议报文交互。

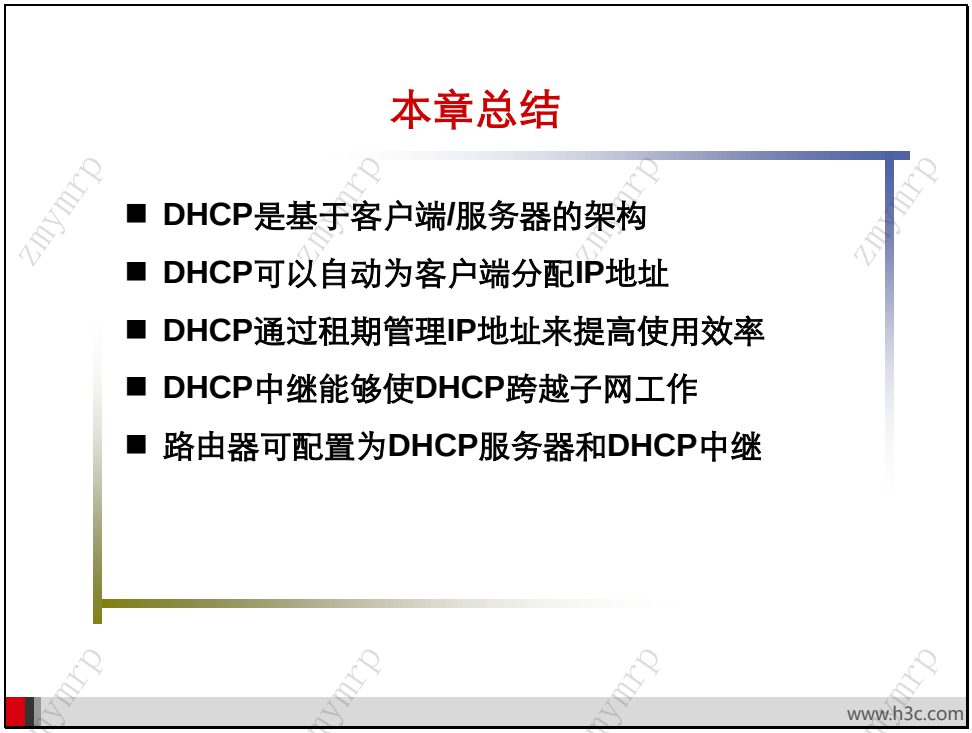
19.7.2 DHCP 中继的显示与维护



在任意视图下执行 display 命令可以显示配置后 DHCP 中继的运行情况，通过查看显示信息验证配置的效果。图中为常用的查看 DHCP 中继信息的命令。

操作	命令
显示工作在DHCP中继模式的接口上指定的DHCP服务器地址信息	**display dhcp relay server-address** [**interface** *interface-type interface-number*]
显示DHCP中继的用户地址表项信息	**display dhcp relay client-information** [**interface** *interface-type interface-number* | **ip** *ip-address* [**vpn-instance** *vpn-instance-name*]]
显示DHCP中继的相关报文统计信息	**display dhcp relay statistics** [**interface** *interface-type interface-number*]

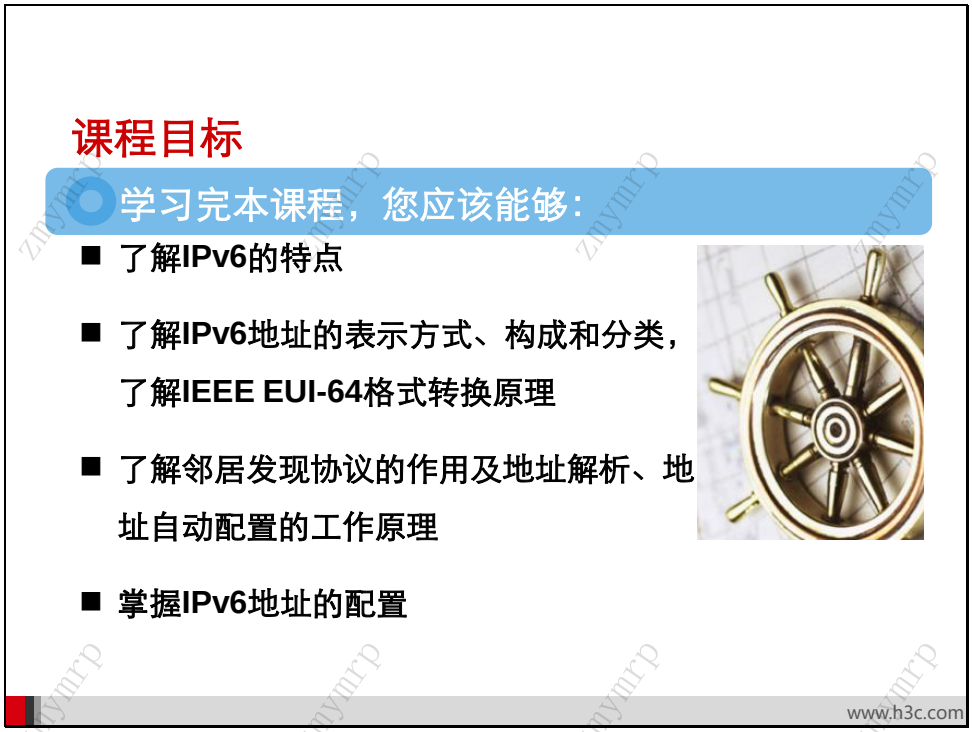
19.8 本章总结


本章总结
DHCP是基于客户端/服务器的架构
DHCP可以自动为客户端分配IP地址
DHCP通过租期管理IP地址来提高使用效率
DHCP中继能够使DHCP跨越子网工作
路由器可配置为DHCP服务器和DHCP中继
www.h3c.com

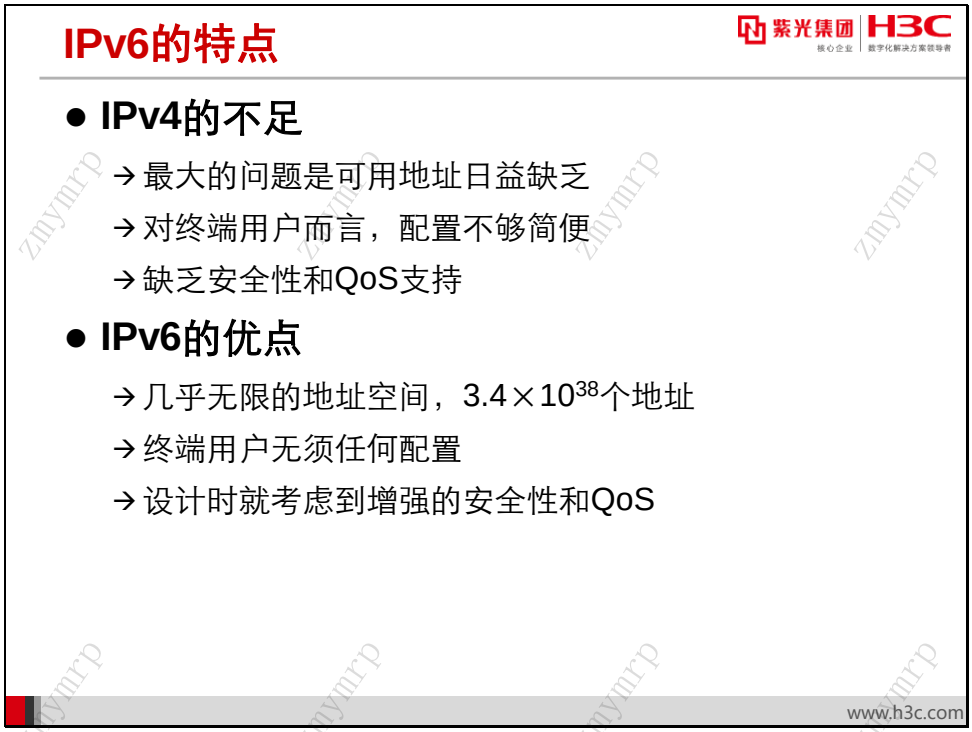
第20章 IPv6 基础

IPv4（Internet Protocol version 4，因特网协议版本 4）是目前因特网所使用的网络层协议。自 20 世纪 80 年代初以来，IPv4 一直在因特网上良好稳定的运行着。但是，IPv4 协议设计之初是为几百台计算机组成的小型网络而设计的，随着因特网及其所提供的服务突飞猛进的发展，IPv4 已经暴露出一些不足之处。IPv6（Internet Protocol Version 6，因特网协议版本 6），也称为 IPng（IP Next Generation，下一代因特网协议），是 IETF（Internet Engineering Task Force，因特网工程任务组）设计的一套因特网协议规范，是 IPv4 的升级版本。IPv6 与 IPv4 最大的区别是，IP 地址的长度从 32 位增加到 128 位。除此之外，IPv6 还在安全性、QoS 等方面进行了增强，并设计了全新的邻居发现协议来实现地址解析、地址自动配置等功能。

20.1 本章目标



20.2 IPv6的特点



实践证明 IPv4 是一个非常成功的协议，它本身也经受住了因特网从最初数目很少的计算机发展到目前上亿台计算机互联的考验。但是，IPv4 协议也不是十全十美的，随着因特网规模的快速扩张，逐渐地暴露出了一些问题。其中最严重的问题是 IPv4 可用地址日益缺乏。

数据显示，截至 2007 年 4 月，整个 IPv4 的可用地址空间只剩下 18%的地址空间没有被分配。而近十年来，Internet 爆炸式增长与使用 IP 地址的 Internet 服务与应用设备（如 PDA、家庭与小型办公室网络、IP 电话与无线服务等）的大量涌现，加快了 IPv4 地址的消耗速度。从 2000 年到 2007 年，亚洲的因特网用户增长了 1.5 倍，非洲增长了 5 倍多，拉美和中东增长了 3 倍多，欧洲也增长了近 1 倍。对于除美国以外的其它地区来说，对 IPv4 地址的需求便更加紧张。预计全世界使用因特网的用户达到世界人口的 20%时，IPv4 地址将严重紧缺，从而限制 IP 技术应用的进一步发展。

另外，对于终端用户来说，IPv4 的配置不够简便。终端用户需要给网络接口卡手工配置地址或指定其使用 DHCP 服务自动获得地址，这给一些没有网络知识的用户造成了不便。同时，IPv4 协议中还存在诸如安全性差、QoS 功能弱等其他问题。

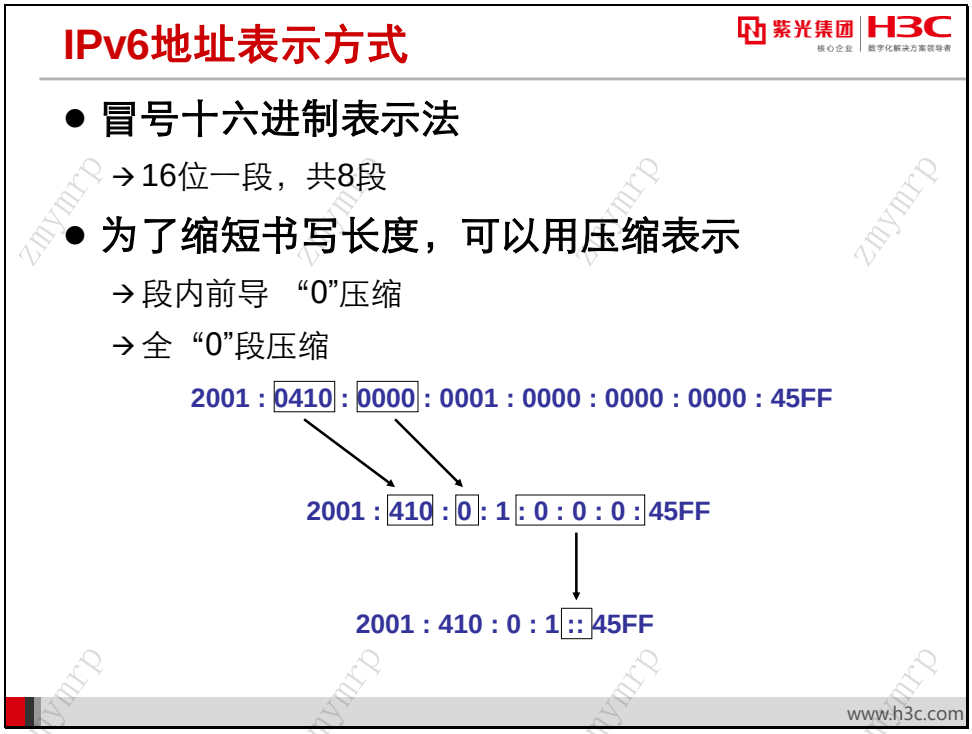
所以，因特网工程任务组 IETF 在 20 世纪 90 年代开始着手 IPng 的制定工作，IPv6 由此应运而生。

IPv6 协议最大的特点是几乎无限的地址空间。IPv4 地址的位数是 32 位，但在 IPv6 中，地址的位数增长了 4 倍，达到 128 位。所以，IPv6 地址空间大得惊人。IPv4 中，理论上可编址的节点数是 2 的 32 次方，也就是 4294967296，按照目前的全世界人口数，大约每 3 个人

有 2 个 IPv4 地址。而 IPv6 的 128 位长度的地址意味着 3.4×10^{38} 个地址！世界上的每个人都可以拥有 5.7×10^{28} 个 IPv6 地址！这个地址量是非常巨大的。有个夸张的说法——可以为地球上的每一粒沙子都分配一个 IPv6 地址。

同时， IETF 在制定 IPv6 时，还考虑到了在 IPv6 中需要解决其它一些 IPv4 协议中存在的问题，如前文提到的配置不够简便、安全性差、QoS 功能弱等等，从而使协议本身能够适应目前网络的发展需要。

20.3 IPv6地址



在 IPv4 中，地址是用 192.168.1.1 这种点分十进制方式来表示的。但在 IPv6 中，地址共有 128 位，如果再用十进制表示的话就太长了。所以，IPv6 采用冒号十六进制表示法来表示地址。

IPv6 地址的 128 位被分成 8 段，每 16 位为一段，每段被转换为一个 4 位十六进制数，并用冒号隔开。例如：1234:5678:90AB:CDEF:ABCD:EF01:2345:6789。

为了尽量缩短地址的书写长度，IPv6 地址可以采用压缩方式来表示。在压缩时，有以下几个规则：

- 每段中的前导 0 可以去掉，但保证每段至少有一个数字

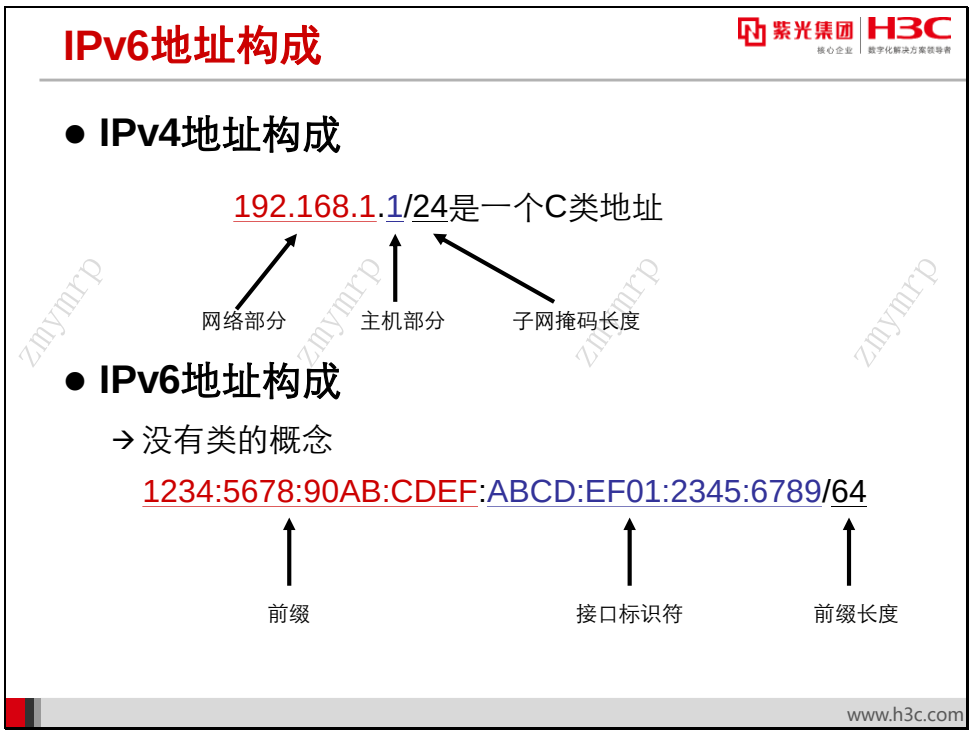
  如：2001:0410:0000:0001:0000:0000:0000:45FF 就可以压缩为 2001:410:0:1:0:0:0:45FF。

  但有效 0 不能被压缩。所以上述地址不能压缩为 2001:41:0:1:0:0:0:45FF 或 21:410:0:1:0:0:0:45FF。

- 一个或多个连续的段内各位全为 0 时，可用::（双冒号）压缩表示，但一个 IPv6 地址中只允许有一个双冒号（::）

  如：2001:0410:0000:0001:0000:0000:0000:45FF 就可以压缩为 2001:410:0:1::45FF 或 2001:410::1:0:0:0:45FF。

  但不允许多个::存在于一个地址中。所以上述地址不能压缩成 2001:410::1::45FF。



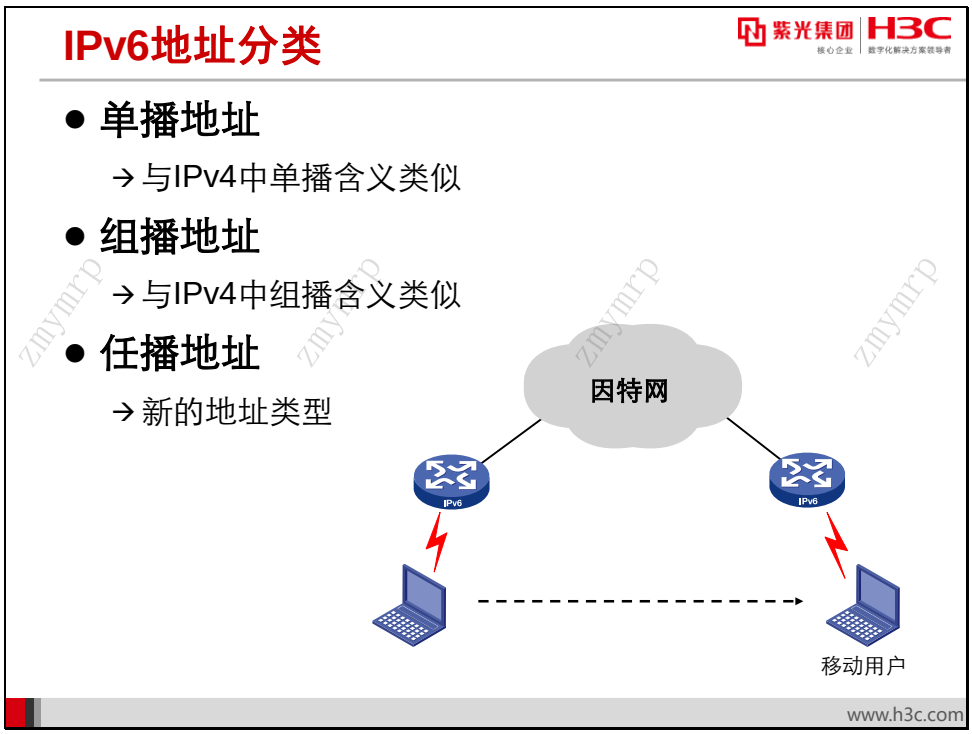
IPv6 取消了 IPv4 的网络号、主机号和子网掩码的概念，代之以前缀、接口标识符、前缀长度；IPv6 也不再有 IPv4 地址中 A 类、B 类、C 类等地址分类的概念。

前缀：前缀的作用与 IPv4 地址中的网络部分类似，用于标识了这个地址属于哪个网络。

接口标识符：与 IPv4 地址中的主机部分类似，用于标识了这个地址在网络中的具体位置。

前缀长度：作用类似于 IPv4 地址中的子网掩码，用于确定地址中哪一部分是前缀，哪一部分是接口标识符。

例如，地址 1234:5678:90AB:CDEF:ABCD:EF01:2345:6789/64，/64 表示此地址的前缀长度是 64 位，所以此地址的前缀就是 1234:5678:90AB:CDEF，接口标识符就是 ABCD:EF01:2345:6789。



IPv4 地址包括单播、组播、广播等几种类型。与其类似，IPv6 地址也有不同类型，包括：单播地址、组播地址和任播地址。IPv6 地址中没有广播地址，在 IPv4 协议中某些需要用到广播地址的服务或功能，IPv6 协议中都用组播地址来完成。

- 单播地址

用来唯一标识一个接口，类似于 IPv4 的单播地址。单播地址只能分配给一个节点上的一个接口，发送到单播地址的数据报文将被传送给此地址所标识的接口。

IPv6 单播地址根据其作用范围的不同，又可分为链路本地地址、站点本地地址、全球单播地址等；还包括一些特殊地址，如未指定地址和环回地址。

- 组播地址

用来标识一组接口，类似于 IPv4 的组播地址。多个接口可配置相同的组播地址，发送到组播地址的数据报文被传送给此地址所标识的所有接口。

IPv6 组播地址的范围是 FF00::/8。

- 任播地址

任播地址是 IPv6 中特有的地址类型，也用来标识一组接口。但与组播地址不同的是，发送到任播地址的数据报文被传送给此地址所标识的一组接口中距离源节点最近的一个接口。比如，移动用户在使用 IPv6 协议接入因特网时，根据地理位置的不同，接入距离用户最近的一个接收站。

任播地址是从单播地址空间中分配的，并使用单播地址的格式。仅看地址本身，节点是无法区分任播地址与单播地址的。所以，必须在配置时明确指明它是一个任播地址。

地址类型		IPv6前缀标识
单播地址	未指定地址	::/128
	环回地址	::1/128
	链路本地地址	FE80::/10
	站点本地地址	FEC0::/10
	全球单播地址	2000::/3
组播地址		FF00::/8
任播地址		从单播地址空间中进行分配，使用单播地址的格式

表中列出了常用的 IPv6 单播地址、组播地址、任播地址的类型及格式。

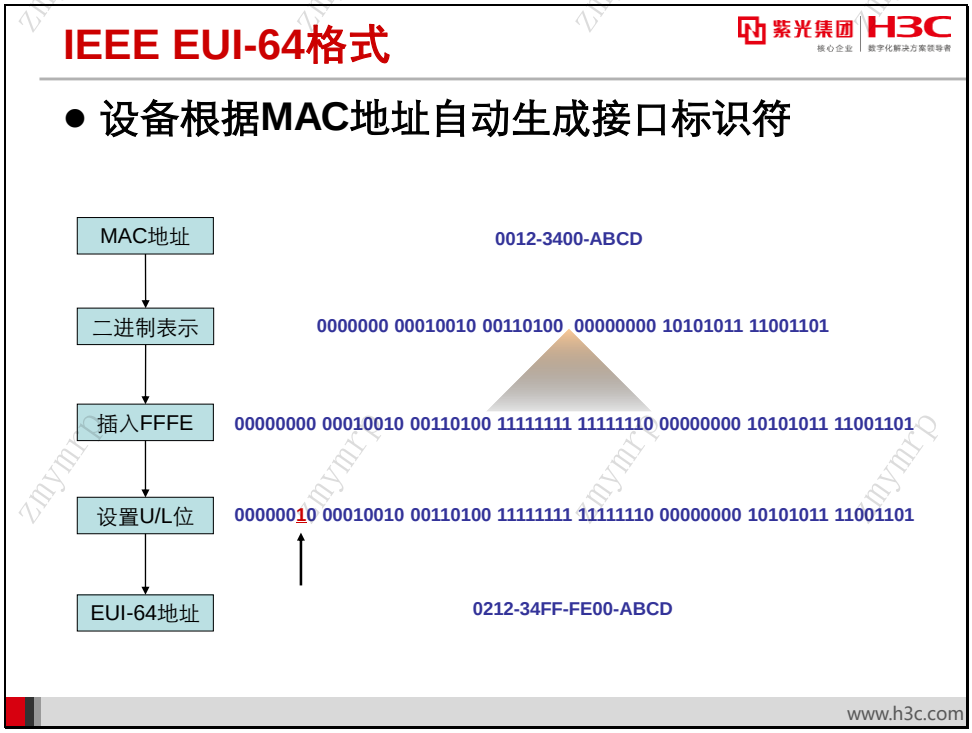
- 未指定地址：地址“::”称为未指定地址，不能分配给任何节点。在节点获得有效的 IPv6 地址之前，可在发送的 IPv6 报文的源地址字段填入该地址，表示目前暂无地址。未指定地址不能作为 IPv6 报文中的目的地址。
- 环回地址：单播地址 0:0:0:0:0:0:0:1（简化表示为::1）称为环回地址，不能分配给任何物理接口。它的作用与 IPv4 中的环回地址 127.0.0.1 相同，节点可通过给自己发送 IPv6 报文而测试协议是否工作正常。
- 链路本地地址：用于链路本地节点之间的通信。在 IPv6 中，以路由器为边界的一个或多个局域网段称之为链路。使用链路本地地址作为目的地址的数据报文不会被转发到其他链路上。其前缀标识为 FE80::/10。
- 站点本地地址：与 IPv4 中的私有地址类似。使用站点本地地址作为目的地址的数据报文不会被转发到本站点（相当于一个私有网络）外的其它站点。其前缀标识为 FEC0::/10。站点本地地址在实际应用中很少使用。
- 全球单播地址：与 IPv4 中的公有地址类似。全球单播地址由 IANA（Internet Assigned Numbers Authority，Internet 地址分配机构）负责进行统一分配。全球单播地址前缀标识为 2000::/3。
- 组播地址：地址标识为 FF00::/8。常用的预留组播地址有 FF02::1（链路本地范围所有节点组播地址）、FF02::2（链路本地范围所有路由器组播地址）等。

另外，还有一类组播地址：被请求节点（Solicited-Node）地址。该地址主要用于获取同一链路上邻居节点的链路层地址及实现重复地址检测。每一个单播或任播 IPv6 地址都有一个对应的被请求节点地址。其格式为：

FF02:0:0:0:0:1:FFXX:XXXX

其中，FF02:0:0:0:0:1:FF 为 104 位固定格式；XX:XXXX 为单播或任播 IPv6 地址的后 24 位。

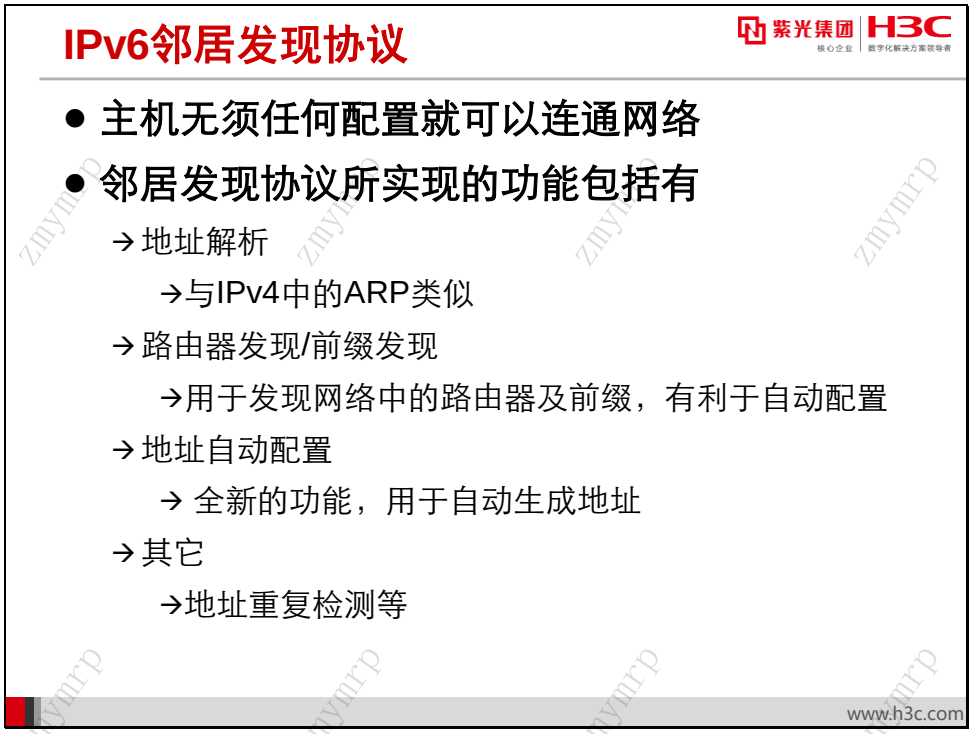
- 任播地址：任播地址与单播地址没有区别，是从单播地址空间中分配的。



构成 IPv6 单播地址的接口标识符用来在网络中唯一标识一个接口。目前 IPv6 单播地址基本上都要求接口标识符为 64 位。在 IPv6 协议中，接口标识符可以由管理员配置，也可以由设备自动生成。自动生成的好处是用户无须配置地址，降低了网络部署难度。如果由设备自动生成接口标识符，则需要符合 IEEE EUI-64 格式规范。

IEEE EUI-64 格式的接口标识符是从接口的链路层地址（MAC 地址）变化而来的。IPv6 地址中的接口标识符是 64 位，而 MAC 地址是 48 位，因此需要在 MAC 地址的中间位置（从高位开始的第 24 位后）插入十六进制数 FFFE（1111111111111110）。为了确保这个从 MAC 地址得到的接口标识符是唯一的，还要将第 7 位（U/L 位）设置为“1”。最后得到的这组数就作为 EUI-64 格式的接口标识符。

20.4 邻居发现协议



IPv6 邻居发现协议是 IPv6 中一个非常重要的协议。它实现了一系列功能，包括地址解析、路由器发现/前缀发现、地址自动配置、地址重复检测等。

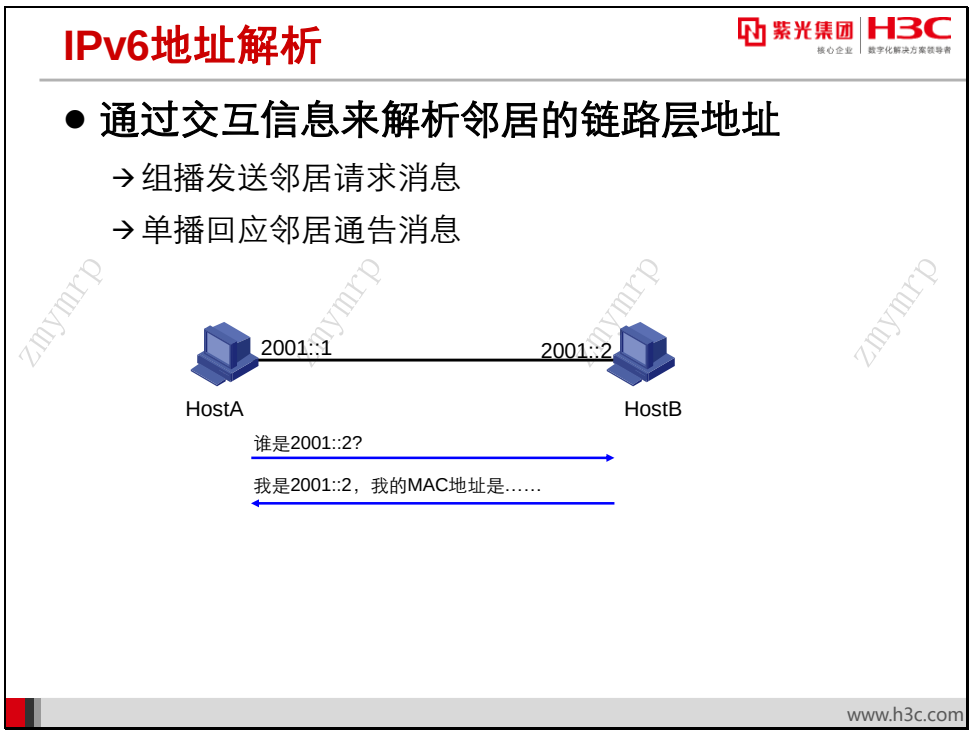
IPv4 网络中，当一个节点想和另外一个节点通信时，它需要知道另外一个节点的链路层地址。比如，以太网共享网段上的两台主机通信时，主机需要通过 ARP 协议解析出另一台主机的 MAC 地址，从而知道如何封装报文。在 IPv6 网络中也有解析链路层地址的需要，就是由邻居发现协议来完成的。

而路由器发现/前缀发现、地址自动配置功能则是 IPv4 协议中所不具备的，是 IPv6 协议为了简化主机配置而对 IPV4 协议的改进。

路由器发现/前缀发现是指主机能够获得路由器及所在网络的前缀，以及其他配置参数。如果在共享网段上有若干台 IPv6 主机和一台 IPv6 路由器，通过路由器发现/前缀发现功能，IPv6 主机会自动发现 IPv6 路由器上所配置的前缀及链路 MTU 等信息。

地址自动配置功能是指主机根据路由器发现/前缀发现所获取的信息，自动配置 IPv6 地址。在主机发现了路由器上所配置的前缀及链路 MTU 等信息后，主机会用这些信息来自动生成 IPv6 地址，然后用此地址来与其他主机进行通信。

IPv6 中的地址自动配置具有与 IPv4 中的 DHCP 类似的功能。所以在 IPv6 中，DHCP 已不再是实现地址自动配置所必不可少的了。



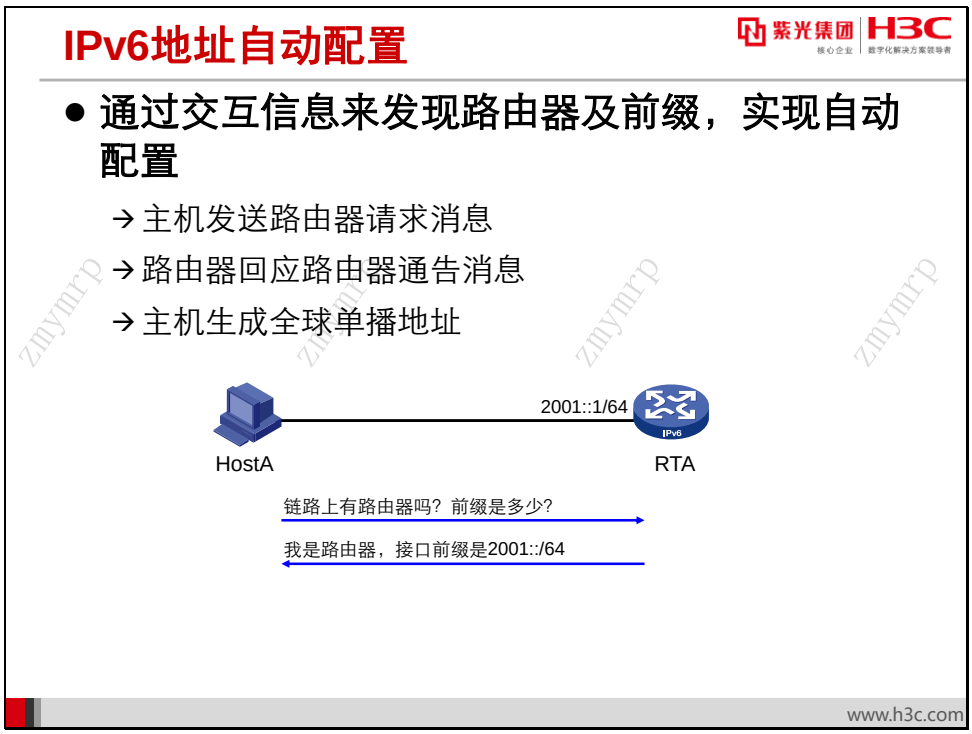
邻居发现协议能够通过地址解析功能来获取同一链路上邻居节点的链路层地址。所谓“同一链路”是指节点之间处于同一链路层上，中间没有网络层设备隔离。通过以太网介质相连的两台主机、通过运行 PPP 协议的串口链路连接的两台路由器，都是属于同一链路上的邻居节点。

地址解析通过节点交互邻居请求消息和邻居通告消息来实现。如图所示：

主机 A 想要与主机 B 通信，但不知道主机 B 的链路层地址，则会以组播方式发送邻居请求消息。邻居请求消息的目的地址是主机 B 的被请求节点组播地址。这样这个邻居请求消息就能够只被主机 B 所接收，其他主机会忽略这个消息。消息内容中包含了主机 A 的链路层地址。

主机 B 收到邻居请求消息后，则会以单播方式返回邻居通告消息。以单播方式返回的目的是减少网络中的组播流量，节省带宽。邻居通告消息中包含了自己的链路层地址。

主机 A 从收到的邻居通告消息中就可获取到主机 B 的链路层地址。之后主机 A 用主机 B 的链路层地址来进行数据报文封装，双方即可通信了。



IPv6 地址自动配置包括了路由器发现/前缀发现和地址自动配置。路由器发现/前缀发现是指主机从收到的路由器请求消息中获取邻居路由器及所在网络的前缀，以及其他配置参数。地址自动配置是指主机根据路由器发现/前缀发现所获取的信息，自动配置 IPv6 地址。

IPv6 地址自动配置通过路由器请求消息和路由器通告消息来实现，过程如下：

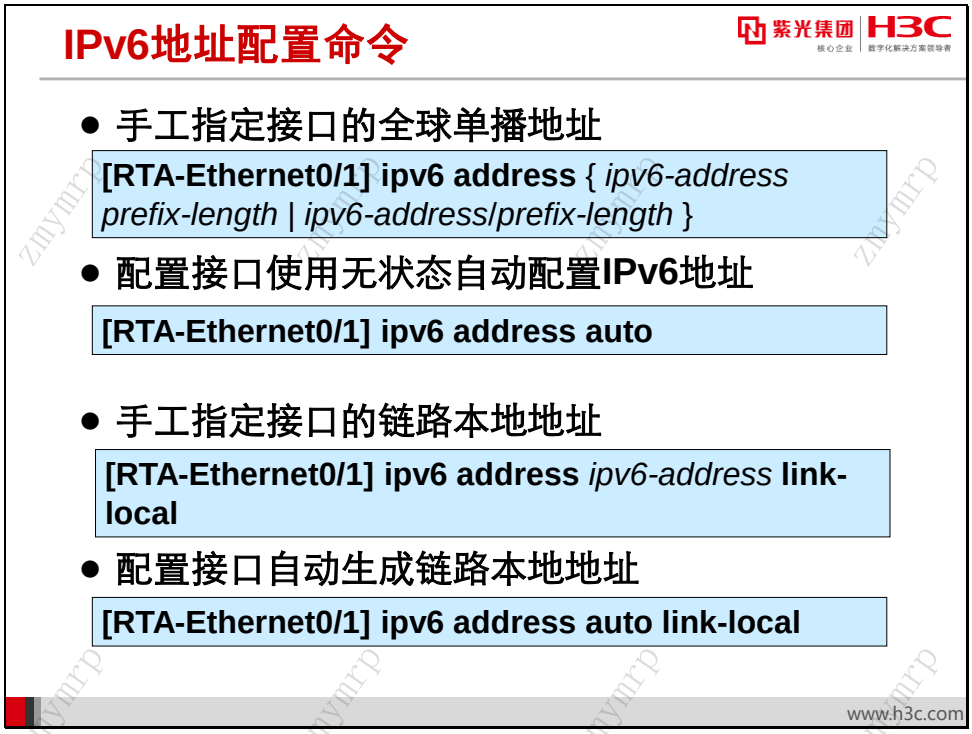
主机启动时，通过路由器请求消息向路由器发出请求，请求前缀和其他配置信息，以便用于主机的配置。路由器请求消息的目的地址是 FF02::2（链路本地范围所有路由器组播地址），这样所有路由器就会收到这个消息。

路由器收到路由器请求消息后，会返回路由器通告消息，其中包括前缀和其他配置参数信息（路由器也会周期性地发布路由器通告消息）。路由器通告消息的目的地址是 FF02::1（链路本地范围所有节点组播地址），以便所有节点都能收到这个消息。

主机利用路由器返回的路由器通告消息中的地址前缀及其他配置参数，自动配置接口的 IPv6 地址及其他信息，从而生成全球单播地址。

如图所示，主机在启动时发送路由器请求消息，路由器收到后，会把接口前缀 2001::/64 信息通过路由器通告消息通告给主机，然后主机以此前缀再加上 EUI-64 格式的接口标识符，生成一个全球单播地址。

20.5 IPv6地址配置



缺省情况下，路由器已经启用了 IPv6 功能。所以，根据需要，在路由器的接口上手工指定或者自动生成 IPv6 地址。

在接口视图下，手工指定接口的全球单播地址的命令为：

[RTA-Ethernet0/1] **ipv6 address** { *ipv6-address prefix-length* | *ipv6-address/prefix-length* }

在接口视图下，配置接口使用无状态自动配置 IPv6 地址的命令为：

[RTA-Ethernet0/1] **ipv6 address auto**

在接口视图下，手工指定接口的链路本地地址的命令为：

[RTA-Ethernet0/1] **ipv6 address** *ipv6-address* **link-local**

在接口视图下，配置接口自动生成链路本地地址的命令为：

[RTA-Ethernet0/1] **ipv6 address auto link-local**

缺省情况下，接口上没有链路本地地址。当接口配置了 IPv6 全球单播地址后，系统会为接口自动生成链路本地地址。

下面给出一个配置示例。如果想在路由器接口上同时手工配置一个全球单播地址 2000::1 以及一个链路本地地址 FE80::1，则命令如下：

```
[H3C]interface GigabitEthernet 0/0
```

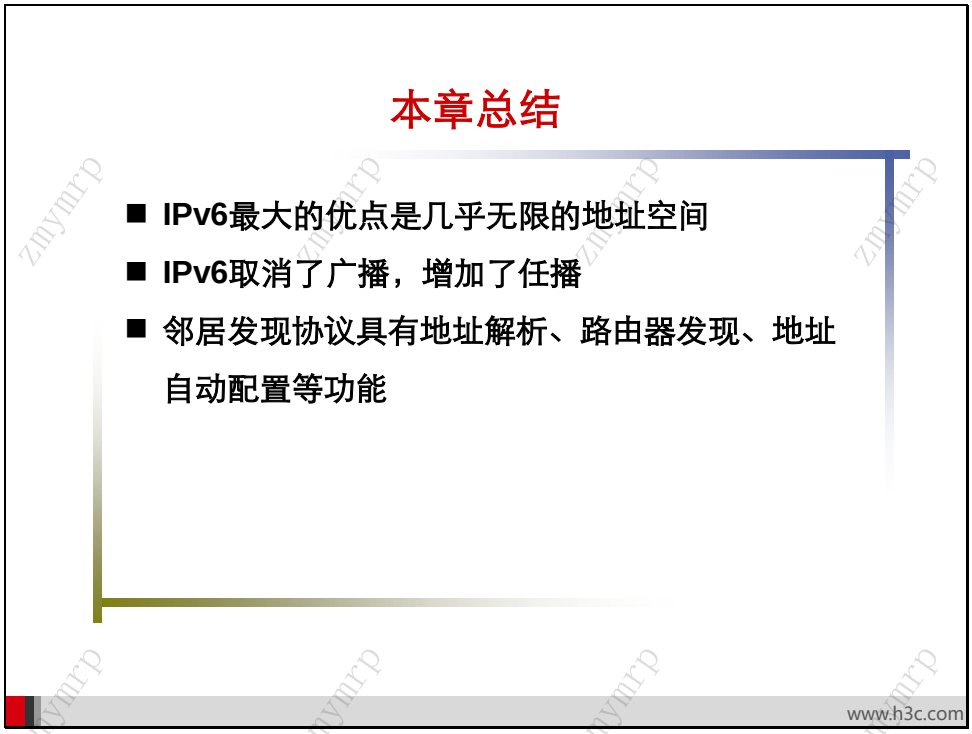
```
[H3C-GigabitEthernet0/0] ipv6 address FE80::1 link-local
[H3C-GigabitEthernet0/0] ipv6 address 2000::1/64
```

配置地址完成后，可以使用 **display ipv6 interface** [*interface-type* [*interface-number*]] [**brief**]来进行地址的查看。如下为上面示例中命令输出信息：

```
[RTA] display ipv6 interface GigabitEthernet 0/0
GigabitEthernet0/0 current state: UP
Line protocol current state: UP
IPv6 is enabled, link-local address is FE80::1
  Global unicast address(es):
    2000::1, subnet is 2000::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
  MTU is 1500 bytes
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
……
……
```

由以上的命令输出可以看到，此接口上配置了链路本地地址 FE80::1，全球单播地址 2001::1。

20.6 本章总结


本章总结
IPv6最大的优点是几乎无限的地址空间
IPv6取消了广播，增加了任播
邻居发现协议具有地址解析、路由器发现、地址自动配置等功能
www.h3c.com