IP协议详解

目录

回到顶部

Internet地址结构

表示IP地址

目前的IP版本有4和6。

目前最流行的就是IPv4，有十进制和二进制两种表示方法。分别是：

点分四组十进制。每一组范围是[0~255]，如：255.255.255.255

二进制。如：11111111 11111111 11111111 11111111

IPv6地址长度是128位，

由8块（或8个字段）组成，每一块都包含四个16进制数，每块由冒号分隔。

有以下特点：

1、一个块中前导的0不必书写。

2、全0的块可以省略，并用符号::代替。

3、IPv6可以兼容IPv4地址，即可以用IPv6格式表示IPv4地址。

表示方式为：IPv6块值为ffff，其后面紧跟“点分四组”的格式。如：

::ffff:10.0.0.1

可以代表IPv4：10.0.0.1

4、IPv6的低32位通常采用点分四组（就是上面那样）的表示法。

基本的IP地址结构

分类寻址

IPV4被分为五大类：ABCDE

A类为：点分四组中的第一组地址范围为0~127的IP地址。已二进制来看就是“首位为0”

B类：128~191.二进制首位为10

C类：192~223.二进制首位为110

随笔档案

D类：224~239.二进制首位为1110

E类：240~255.二进制首位为1111

子网寻址

如上可以看出，IP地址值存在两个子结构：网络号和主机ID，但这样就出现了一个问题。就是分配ip地址很麻烦。（即网络信息中心要负责每一台主机的ip地址分配，这样就太繁杂了。）。

于是就将ip地址从两级分成了三级：网络号、子网ID、主机ID

划分子网ID的方法是从“原有的主机号中借用若干位”作为子网号。（当然，主机号就减少了）

网络信息中心集中分配每一个网络号。

然后各个站点的管理人员再分配他们网络号下的子网ID和对应的主机ID。然后管理员在安排每一个子网下面的主机数。

子网掩码

在ip地址传播的时候，可以根据二进制首位的格式判断其属于第几类网，也就爱能判断其网络号有多少位。

但ip地址本身并没有包含任何关于子网划分的信息，所以光凭ip地址无法知道“其是如何划分子网的”。

于是就出现了子网掩码，和ip地址配套着出现，用来说明“该ip地址的子网ID是那几位数”。

图示如下：

2016年3月(8)
2016年2月(1)
2016年1月(2)
2015年12月(9)

最新评论

1. Re:springMVC操作cookie和session
spring cookie 讲的很好
--民工也Coding
2. Re:决策树
您好，邀您写书，有意向的话请加微信：yzbook01，如有打扰，请见谅。
--xiaoxiaoningmeng
3. Re:springCloud配置本地配中心SpringCloudConfig
老哥 读不到配置
--背对我煮面
4. Re:二进制正负数的原码、反码、补码之间的转化
@ 你高兴的太早了谢指出，已修改...
--Red_Code
5. Re:二进制正负数的原码、反码、补码之间的转化
你这负数原码、反码、补码错的有点离谱。。。
--你高兴的太早了

阅读排行榜

1. KindEditor用法介绍(27623)
2. spring集成jedis简单实例(25389)

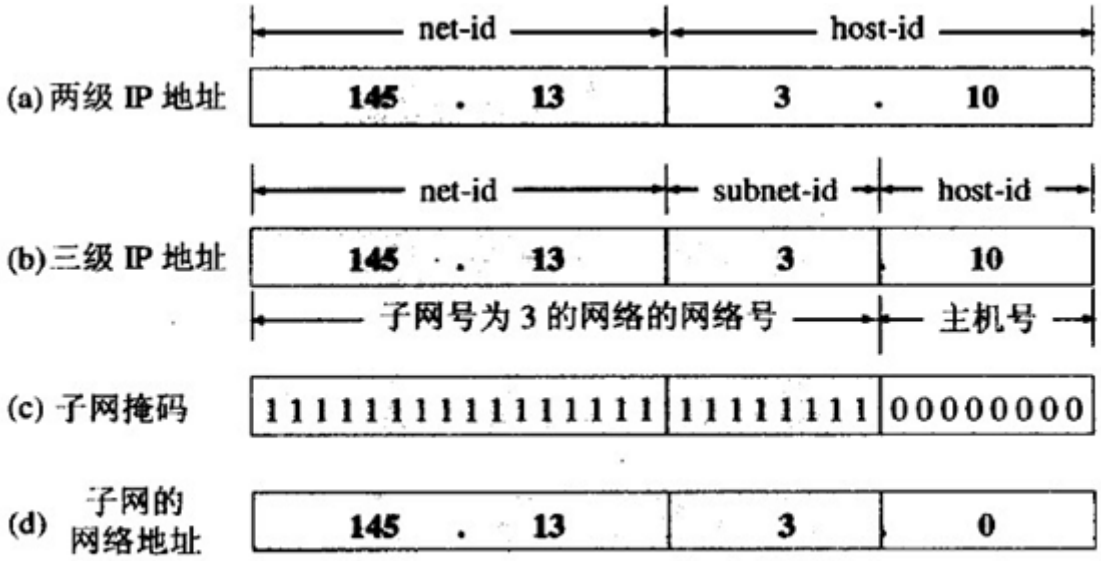


图 4-20 IP 地址的各字段和子网掩码（以 145.13.3.30 为例）

以二进制来看子网掩码，只有“连续的1”或“连续的0”。IP子网掩码的长度和他所对应的IP地址长度一样（即32位和128位）。

其中的“1”对应的是ip中的“网络号”和“子网号”，“0”对应的是ip中的“主机号”。

将ip地址与子网掩码进行“按位与”运算，就能得出用于路由的“子网标识符”，如：

ip地址： 128.32.1.14 对应二进制：10000000 00100000 00000001 00001110

掩码地址： 255.255.255.0 对应二进制：11111111 11111111 11111111 00000000

子网标识符 128.32.1.0 对应二进制：10000000 00100000 00000001 00000000

通过上面的运算我们可以看出：128.32.1.14这个地址属于子网“128.32.1.0”。

3. android获取本地图片并显示图片(24985)
4. springMVC操作cookie和session(24286)
5. IP协议详解(19509)

评论排行榜

1. 二进制正负数的原码、反码、补码之间的转化(3)
2. KindEditor用法介绍(1)
3. 决策树(1)
4. springCloud配置本地配中心SpringCloudConfig(1)
5. tomcat原理详解(1)

推荐排行榜

1. IP协议详解(5)
2. KindEditor用法介绍(4)
3. tomcat原理详解(3)
4. spring集成jedis简单实例(3)
5. springMVC操作cookie和session(3)

以上面的各个ip地址为例，整个流程可以如下所示：

1、某个站点申请到了一个B类网的网络号：128.32.x.x

2、然后该站点的管理员决定使用“255.255.255.0”作为该站点的子网掩码。这样就该站点有多少个子网划分好了（一旦决定使用该子网掩码，通过计算就能知道，将该站点划分了256个子网，每个子网里有254台主机（因为每个子网的第一个和最后一个地址无效））

3、然后为每一个子网中的主机都安排好ip地址。

4、假设现在有一个访问，请求访问ip地址128.32.1.14

5、先根据该ip地址的二进制前几位，发现该地址是一个B类网，所以网络号有16位，也就是说128.32是它的网络号。于是根据网络号128.32找到该站点。

6、该站点边界路由器将该ip与子网掩码进行“按位与”运算，发现子网标识符为128.32.1.0（即128.32.1，后面的.0不存在，因为每个子网的第一个和最后一个地址无效）

7、找到子网128.32.1后，再根据.14找到主机。

广播地址

之前我们讲到了，每个子网的第一个和最后一个地址无效。这个无效更多的是指“这两个地址不是指向的某个真实的主机”。

上面的子网掩码计算完后，得到的是“子网掩码.0”，此处的“.0”（子网的第一个地址），意味着该ip地址不指向某一个主机，而是表示“子网标识符”。

而“.255”（子网的最后一个地址），则也“不是指向”255主机，而是表示了“该子网下的所有主机”。从而引出了广播地址。

IP网络上发送信息都是要有准确的ip地址的，假设要给“统一网络下的”10台主机发信息，那么就得写10个精准的ip地址。

而利用广播地址，我们就能通过一条ip地址，将信息发送给某个子网下的所有主机。

即：只写出网络号和子网号，而将主机ID部分用.255代替。

想要计算一个站点的广播地址也要用到子网掩码，计算方法如下：

1、首先将站点的子网掩码“取反”（1变0,0变1）。

2、然后与站点ip地址进行“按位或”运算（1或0的1,0或0的0,1或1得1）。

3、这样就既能保留网络号和子网号的地址，然后又能将主机号部分替换为255。

前缀（CIDR）

因为现有IPV4地址已经耗尽，所以制定了一个前缀方案。

即：将ip地址的某些位固定住，剩余的为可设置为0和1的任意组合。（比如值固定住第一位，那么后面又能有好多种组合出来的ip）。

CIDR是没有分类只说的（也就是说带有前缀的ip地址，不属于ABCDE里面的任何一类）。

之前ABC的网络号被限制为8位、16位、24位

而CIDR的网络号由前缀来控制，如222.80.18.18/25，其中“/25”表示其前面地址中的前25位代表网络号，其余位数代表主机ID。

写法是在现有的点分四组后面加上"/前缀"

如：128.0.0.0/24

特殊用途地址

某些地址被用于特殊用途所以不会被分配。

比较为人们所熟知的是127.0.0.1，指向本机。实际上整个127段都是主机回送地址。

也就是说127.0.0.0/8是特殊地址（前缀8将127框住，后面可随意组合）。127.0.0.1~127.255.255.255之间的地址都会回送本机。

组播地址：

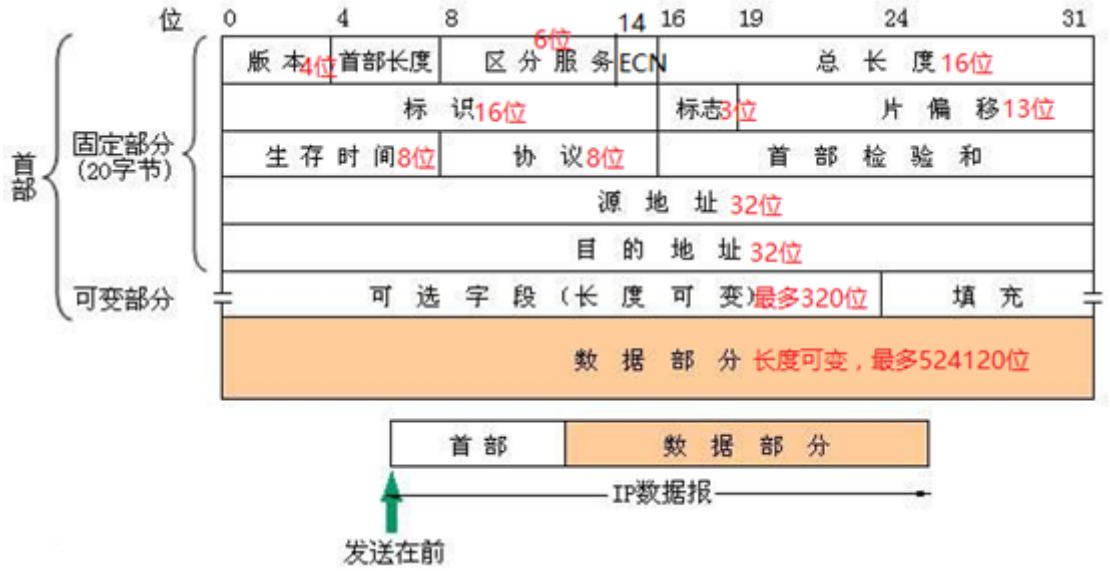
组播IP地址使用的是D类网。

使用组播IP地址作为目的地址，已加入“组”的所有主机都将接收发送到该组的任何数据报。

发送方甚至不知道有多少主机接收到数据报。

回到顶部

IP协议



IPv4首部

版本：

包含IP数据报的版本号：ipv4为4，ipv6为6

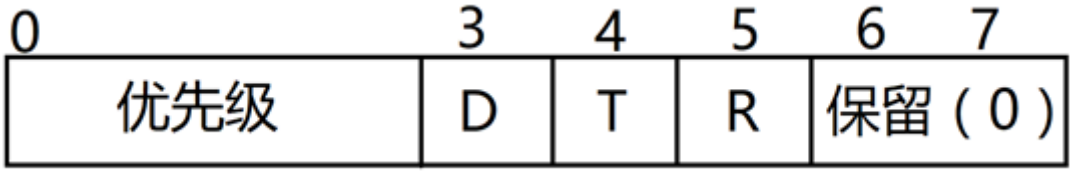
首部长度：

其中保存的是整个首部中的“32位字”的数量。

这个字段正常的值为：5（假设“可选字段长度为0”）

该字段最大值为：15（可选字段长度全满加上原有字段）

区分服务：



优先级（3位）和数据链路层的QoS机制有关，定义了8个服务级别。当Qos选择了某种服务模型后，优先级越高，字段越优先传输。

D、T、R分别表示延时、吞吐量、可靠性。当这些值都为1时，分别表示低延时、高吞吐量、高可靠性。

ECN：

用于为数据报标记“拥塞标识符”。

当一个带有ECN标记的分组发送后，如果接收端“持续拥塞”且“具有感知ECN的能力”（如TCP），那么接收端会通知发送端降低发送速度。

总长度：

该字段指的是IPv4数据报的总长度（以字节为单位）。

通过该字段和“首部长度”字段，我们可以推测出ip数据报中“数据部分”从哪开始以及长度。

标识、标志、分偏移：

该字段帮助标识由IPv4主机发送的数据报。

这个字段对实现分片很重要，大多数数据链路层不支持过长的ip数据报，所以要把ip数据报分片，每一片都是一个独立的ipv4数据报。

发送主机每次发送数据报都讲一个“内部计数器”加1，然后将数值复制“标识”字段中。

生存时间：

该字段用于设置一个“数据报可经过的路由器数量”的上限。

发送方在初始发送时设定某个值（建议为64,、128或255），每台路由器再转发时都将其减一，当字段达到0时，该数据报被丢弃，并使用一个ICMP消息通知发送方。

协议：

包含一个数字，该数字对应一个“有效载荷部分的数据类型”。比如17代表UDP，6代表TCP。

首部校验和：

该字段“仅计算”IPv4首部。也就是说只“校验”首部。并不检查数据报的“数据部分”。

首先将“首部校验和”设置为0,。

然后对首部（整个首部是一个16位字的“序列”）计算16位二进制反码和。该值被存储在首部校验和字段中。

当接收方接收到数据报后，也对其首部进行校验计算，如果结果与“首部校验和”的值不同，就丢弃收到的数据报。

可选字段：

IP支持很多可选选项。

如果选项存在的话，它在IPv4分组中紧跟在基本IPv4头部之后。

IP转发

主机和路由器都能转发ip，不同之处在于：主机不转发那些不是由它生成的数据报，但路由器会转发。

IP地址可以接收一个数据报。

当IP模块接收到一个数据报时，

首先检查此数据报的目的地址是否为自己的IP地址，

如果不是，且IP层配置为一台路由器，则根据“转发表”转发该数据报。

否则丢弃此数据报，然后返回给源节点一个信息，表明错误。

转发表

IP转发表通常需要包含以下信息。

目的地：

32位字段（用于IPv4），

当目的地为“默认路由”（当路由表中与包的目的地址之间没有匹配的表项时路由器能够做出的选择。如果没有默认路由，那么目的地址在路由表中没有匹配表项的包将被丢弃）时，目的地可设置为0。

对于仅描述一个目的地的主机路由，目的地可设为完整的ip地址。

掩码：

之前讲过了，掩码是和目的地ip组合使用的，将ip地址与子网掩码进行“按位与”运算，就能得出用于路由的“子网标识符”。

下一跳：

下一个IP实体（路由器或主机）的IP地址。

接口：

包含一个网络层使用的标识符，用来确定将数据报发送到下一跳的网络接口。

IP转发行动

当一台主机或路由器需要向下一跳转发数据报时，它首先检查数据报中的IP地址。在算法表中使用该IP地址来执行最长前缀匹配算法。

最长前缀匹配算法：

举个例子，现在路由器中有两个路由表（转发表）其目的地为：192.168.2.0/24和192.168.0.0/16

假设现在路由器接收到了一条数据报，其目标地址为192.168.2.3

那么实际上这两个路由表都匹配，但路由器会选择192.168.2.0/24作为该数据报的下一跳，

因为192.168.2.0/24这个ip地址“前缀更长”，匹配成功的部分更多。

IP分片

链路层通常对可传输的每一个帧的最大长度都有上限。为了使超过此上限的ip数据报能够正常传输，

IP引入了“分片”和“重组”。

当IP层接收到要发送的IP数据报时，通过查找“转发表”，会判断该数据报应该从那个本地“接口”发送以及MTU（最大传输单元，指一种通信协议的某一层上面所能通过的最大数据包大小（以字节为单位））是多少。

不同的网络类型，其MTU都不相同，如以太网中MTU为1518字节，FDDI为4500字节。

如果IP数据报超过MTU值，则进行分片。IPv4的分片可以在原始发送方主机和端到端路径上的任何中间路由器上进行，即：一个分片在到达接收主机的路径中，还可能被继续分片。（而IPv6只允许源主机进行分片）

当一个IP数据报被分片后，直到它到达最终目的地才会被重组。（因为同一个数据报的不同分片，可能经由不同的路径到达相同的最终目的地，所以中途有可能无法重组）

之前提到过，ip分组后，每一个分片都是一个完整的ip数据报。

其中首部里的“总长度”字段，是该分片的总长度。

分片技术与ip首部中以下三个字段有关：

标识符：

主机将数据报分片后，在发送前，会给每一个分片数据报一个ID值，放在16位的标识符字段中。

这个ID值可以用来识别哪些分片是属于同一个数据报的，方便重组。

标志：

标志字段在IP报头中占3位，

第1位作为保留，置0;

第2位，分段，有两个不同的取值：该位置0，表示可以分段;该位置1，表示不能分段;

第3位，更多分段，同样有两个取值：该位置0，表示这是数据流中的最后一个分段，该位置1，表示数据流未完，后续还有分段，当一个数据报没有分段时，则该位置0，表示这是唯一的一个分段。

当目的主机接收到一个IP数据报时，会首先查看该数据报的标识符，并且检查标志位的第3位是置0或置1，以确定是否还有更多的分段，如果还有后续报文，接收主机则将接收到的报文放在缓存直到接收完所有具有相同标识

符的数据报，然后再进行重组。

偏移量：

就像之前说过的，各个IP分片数据报在发送到目的主机时可能是无序的，所以就需要“偏移量”字段来指明“该分片在原数据报中的位置顺序”。

发送主机对第一个数据报的偏移量置为0，而后续的分片数据报的偏移量则以网络的MTU大小赋值。

如：

假设：网络接口MTU大小为1400字节，要传输的数据报为3800字节。

那么，将要传输的数据报分为3片即可（3800=1400+1400+1000）

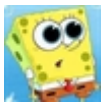
偏移量的计算方法为：已经“装载”好的分片字节数/8（偏移量就是某片在原分组的相对位置，所以8个字节作为偏移单位。）

分片1偏移量为：0（0=0/8，因为该偏移量之前没有装载过任何分片，自然也就是0除以8了）

分片2偏移量为：175（175=1400/8，由于分片1已经装载好了1400字节，所以此分片的位置就在1400字节之后，也就用1400除以8了）

分片3偏移量为：350（350=2800/8，目前已经装好了两个分片也就是2800字节已经装载完了，那么分片3自然就跟在2800字节之后了，也就是用2800除以8）

分类: Network



Red_Code
关注 - 0
粉丝 - 20
+加关注

5 0

« 上一篇： Http协议详解

» 下一篇： ARP协议详解

posted @ 2017-07-07 14:25 Red_Code 阅读(19510) 评论(0) 编辑 收藏

相关博文：

- TCP/IP协议、HTTP协议、SOCKET通讯详解
- TCP/IP协议详解---概述
- TCP/IP协议详解
- IP协议详解
- IP协议详解