# 一、名字系统及TCP/IP名字注册和名字解析

## DNS设计目标和目的

1.创建一个全球性的、可扩展的、一致性的名字空间

2.实现本地资源本地控制

3.分布式设计以避免瓶颈

4.应用的广泛性

5.支持多个底层协议

6.硬件普适性

## 域：单个对象或基于某种共性收集齐的一个对象集合

## DNS服务器支持功能

主要功能：存储和提供（通过相应来自DNS解析器的请求）名字数据

1. 与其他服务器进行交互
2. 地区管理和传输
3. 性能增强功能
4. 管理

### DNS名字服务器负载均衡

并非只能为一个DNS域名创建一条地址RR，而是可以创建多条这样的RR。这样就可以把多个IP地址与一个名字相关联，从而可以把针对一个域名的大量请求分散到多台物理IP设备上，这种做法使得DNS能够为繁忙的因特网服务器实现负载均衡

## 名字解析服务

1.标准名字解析：接受一个DNS名字作为输入并确定其对应的IP地址

2.反向名字解析：接受一个IP地址并确定与其关联的名字

3．电子邮件解析：根据报文中使用的电子邮件地址来确定应该把电子邮件报文发送到 何处。

### 名字解析器执行的功能

1．提供用户接口

2.生成并发送查询

3.处理相应

## DNS通用主文件记录格式

1. 域名
2. TTL
3. 类
4. 类型
5. 数据

### IPv6 DNS 扩展

1. 新的RR类型AAAA(IPv6地址)
2. 新的反向解析层次结构
3. 查询类型和解析过程的改动

# 二、网络文件和资源共享协议

网络文件系统（NFS）

文件传送协议（FTP）

超文本传送协议（HTTP）

## 网络文件和资源共享以及TCP/IP NFS

### 文件和资源共享协议的组件

1. 文件系统模型和体系结构
2. 资源访问方法
3. 操作集
4. 报文传送协议
5. 管理工具

### NFS专门为实现消除本地和远程文件之间差别的目标而设计

1. 研制网络文件系统（NFS）的目的是使客户机主机能够访问本地文件那样访问远程服务器上的文件。它的主要设计目标是性能、简单和跨供应商的兼容性。
2. NFS的主要组件



1. OSI模型：开放式系统互联通信参考模型
2. 通用数据交换方法(XDR):当从设备A向设备B传输关于如何访问一个文件的信息时，设备A首先将其从设备A的内部表示转换成这些数据类型的XDR表示，信息使用XDR编码跨越网络进行传输，然后设备B将其从XDR转换成自己的内部表示。
3. 使用XDR就可以在设备之间交换数据而不需要考虑每台设备使用了什么样的内部文件系统。这令NFS可以在客户机和服务器之间交换文件数据，这些客户机和服务器可能是使用完全不同的硬件和软件平台实现的。

#### 使用远程过程调用（RPC）的NFS客户机/服务器操作

##### RPC操作与运输协议用法

NFS 没有像其他大多数协议 那样采用专用报文格式。相反，客户和服务器使用RPC协议来交换文件操作请求和数据。

##### NFS中客户机和服务器的职责

NFS服务器被设计成无状态的，简言之，这意味着NFP服务器不会从一个请求到另一个请求地跟踪记录使用自己的客户机的状态。

##### 客户机和服务器高速缓存

NFS被设计成一种无状态协议，使用智能客户机和相对较哑的服务器，服务器只对请求做出相应，不维护有关那些文件正在使用的状态信息。为提高效率起见，NFS最初设计成使用UDP作为运输协议。这要求由NFS客户机负责检测丢失请求并进行重传。为能够利用TCP的可靠性和其他特色，NFSv4使用了TCP

#### NFS服务器过程和操作

#### NFS文件系统模型与装配协议

##### NFS文件系统模型

一种层次结构式的目录组织形式，目录中又包含文件和子目录。层次结构的顶端是根，根下面可以包含任意数量的文件和第一级目录，每一个目录都可以包含更多文件或其他目录，从而可以创建出任意的树状结构。

##### 装配协议

NFS版本2和版本3中没有包含用来打开或关闭某台远程服务器上的资源的过程。在这两个版本里，当NFS任务能够成功完成之前，首先必须利用专门的装配协议来装配一个文件系统，并创建一个文件句柄以便在句柄上访问中国文件，当文件系统不需要时装配协议也被用来装配一个文件系统，装配协议的实现方法与NFS自己的实现方法相似。定义了一系列使用RPC的XDR的过程。在NFSv4中装配协议已经不再需要，因为它所执行的任务已经作为NFSv4操作实现了。

# 三、主机配置与TCP/IP主机配置协议

## 主机配置概念、问题和动机

### 主机配置的目的

在TCP/IP网络中，必须为每台设备分配一个IP地址。除此之外，对一个TCP/IP网络来说，我们可能希望被告知每台主机下面的一些参数：

1. 本地网络上默认路由器的地址
2. 主机应该使用的网络掩码
3. 为主机提供特定服务的服务器的地址，例如邮件服务器或域名系统（DNS）名字服务器
4. 本地网络的最大传输单元（MTU）
5. 应该给IP数据报使用什么样的寿命（TTL，TIME TO LIVE）值

### 主机人工配置存在的问题

1. 远程配置
2. 移动设备配置
3. 哑主机配置
4. 地址共享

### 使配置过程自动化：主机配置协议

主机配置协议使管理员能够将主机设置为由主机自动确定自己的地址和其他的一些重要参数。主机配置协议之所以有用不仅因为他们节省了人工配置所需要的大量工作，而且因为它们支持远程设备、无存储设备或移动设备的自动设置。

### 主机配置协议在TCP/IP中起的作用

主机配置驻留在高层的有点：

1. 协议的操作不依赖于其上运行的硬件，从而令协议更为通用。
2. 能够在网络之间传送主机配置报文，而使用一个运行在本地网络上的低层协议则不可能做到这一点

## TCP/IP引导协议（BOOTP）

BOOTP是动态主机配置协议（DHCP）的前身。

### BOOTP概述、历史和标准

#### BOOTP：客服RARP的缺陷

反向地址解析协议（RARP）是第一个试图解决这种引导问题的尝试。

RARP能够向一台无盘设备提供其IP地址

RARP的困难在于它有太多的局限，它使用硬件广播方式运行在相当低的层次，因此他要求调整以适应不同的硬件类型。

BOOTP用来客服RARP的确定：

1. BOOTP仍基于客户机/服务器交换模式，只是作为一种高层软件协议实现，采用用户数据包协议（UDP）进行报文传输。它不像RARP那样依赖于特定的网络硬件。
2. 除了IP地址之外它还支持向客户机发送额外的配置信息，额外信息通畅可以放在一个报文中发送以提高效率。
3. 它能够处理客户机和服务器位于同一个互联网络的不同网络的情况。这使得对提供IP地址的服务器的管理能够更加集中，节省了费用同时也节省了管理时间和麻烦。

首个广泛应用于TCP/IP网络的主机配置协议是引导协议（BOOTP）。该协议专门设计用来客服PAPR的诸多缺点，支持主机配置。BOOTP计划在无存储器设备两阶段引导过程中的第一个阶段中使用，在利用BOOTP获得IP地址和其他配置参数以后，设备江勇诸如TFTP这样的协议来下载为在网络上工作所必须的软件。

#### BOOTP的修改与DHCP的研发

BOOTP的最重要的一个缺点是缺少对动态地址分配的支持。

20世纪90年代后期对动态地址分配的需求显著提高促成了动态主机配置协议（DHCP）的研发。

### BOOTP客户机/服务器报文传递和寻址

#### BOOTP报文传递和传输

BOOTP报文传递使用UDP作为自己的第四层传输协议，有以下两条原因：

1. UDP比另一个第四层运输协议即传输控制协议（TCP）要简单的多，特别适用于像BOOTP这样的简单请求/应答协议
2. 由于客户机显然不知道BOOTP服务器的地址，所以请求是在其本地网络上传播出去的。UDP支持广播，TCP则不然。

#### 丢失报文的重传

使用UDP传递BOOTP报文带来了简单性的好处，但缺点是UDP不可靠，这意味着BOOTP请求在到达服务器之前可能会丢失，或者服务器的相应可能无法返回客户机。

BOOTP使用不提供可靠特性的UDP作为传输协议，为此，BOOTP客户机必须检测自己的请求何时丢失并在必要时重传请求。

### BOOTP详细操作

#### BOOTP引导过程

1. 客户机创建请求
2. 客户机发送请求
3. 服务器收到请求对其进行处理
4. 服务器创建回答
5. 服务器发送回答
6. 客户机处理回答
7. 客户机完成引导过程

## DHCP概述与地址分配概念