

## 9.网络设计思想

1



### 文献阅读与参考文献



- 文献阅读及作业

- ✓ J. H. Saltzer, D. P. Reed, and D. D. Clark. 1984. End-to-end arguments in system design. ACM Trans. Comput. Syst. 2, 4 (Nov. 1984), 277–288.

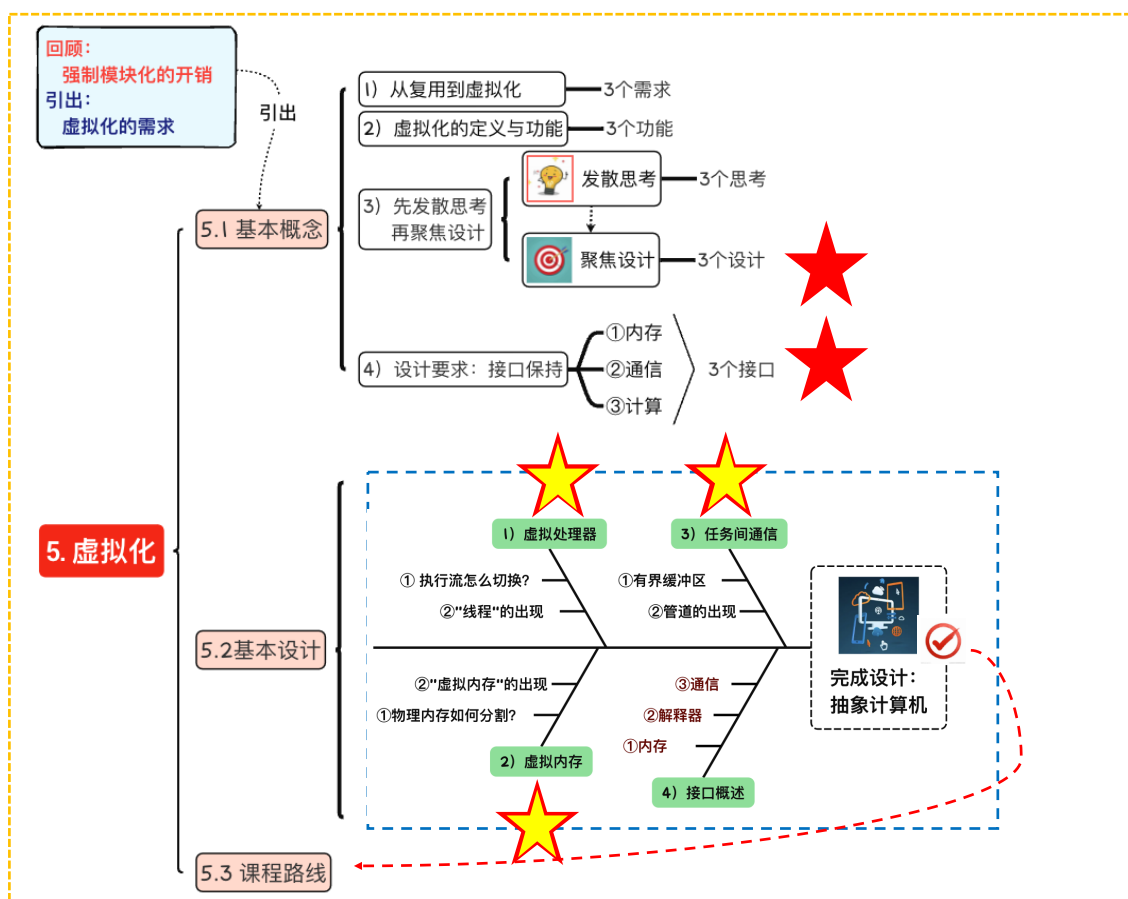
- 具体要求见Bb平台的作业发布

- 参考文献

- ✓ Clark, D. Annotated Version of The Design Philosophy of the DARPA Internet Protocols, March 2013.
  - ✓ Greenberg, A., Hjalmtysson, G., Maltz, D. A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., and Zhang, H. A Clean Slate 4D Approach to Network Control and Management. SIGCOMM Comput. Commun. Rev. 35, 5 (Oct. 2005), 41–54.

第5-8章  
完成  
内核设计

第9-11章  
设计  
系统互联



## 回顾：3个基本抽象（第2章）

### 解释器抽象

- 虚拟化为（线程）

### 存储器抽象

- 虚拟化为（虚拟内存）

### 通信链路抽象

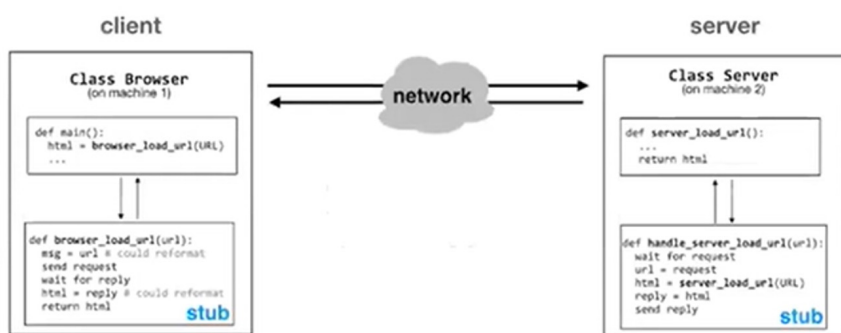
- 两个原语：send & receive @虚拟机和物理机



哪3个基本抽象？

系统能不能跨越空间阻隔，联合起来？

# 需求：模块化与C/S的通信设施（第4章）



假设已有通信链路，  
能否建成对上层**透明**的通信基础设施？  
这个通信基础设施应该是什么样的呢？

## 通信基础设施的需求

1. **更远**的连接、**任意**点对点的连接

链路之间可以接续

通过共享协议

2. 更**廉价**的通信

共享现有的通信链路以降低成本

通过共享链路

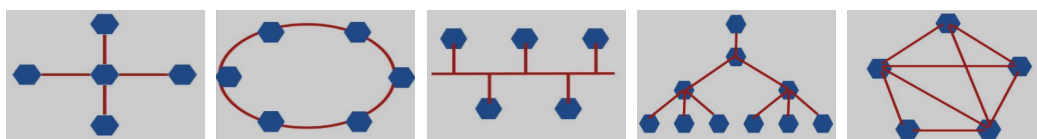
3. 更**健壮**的通信

两点间有多条路径

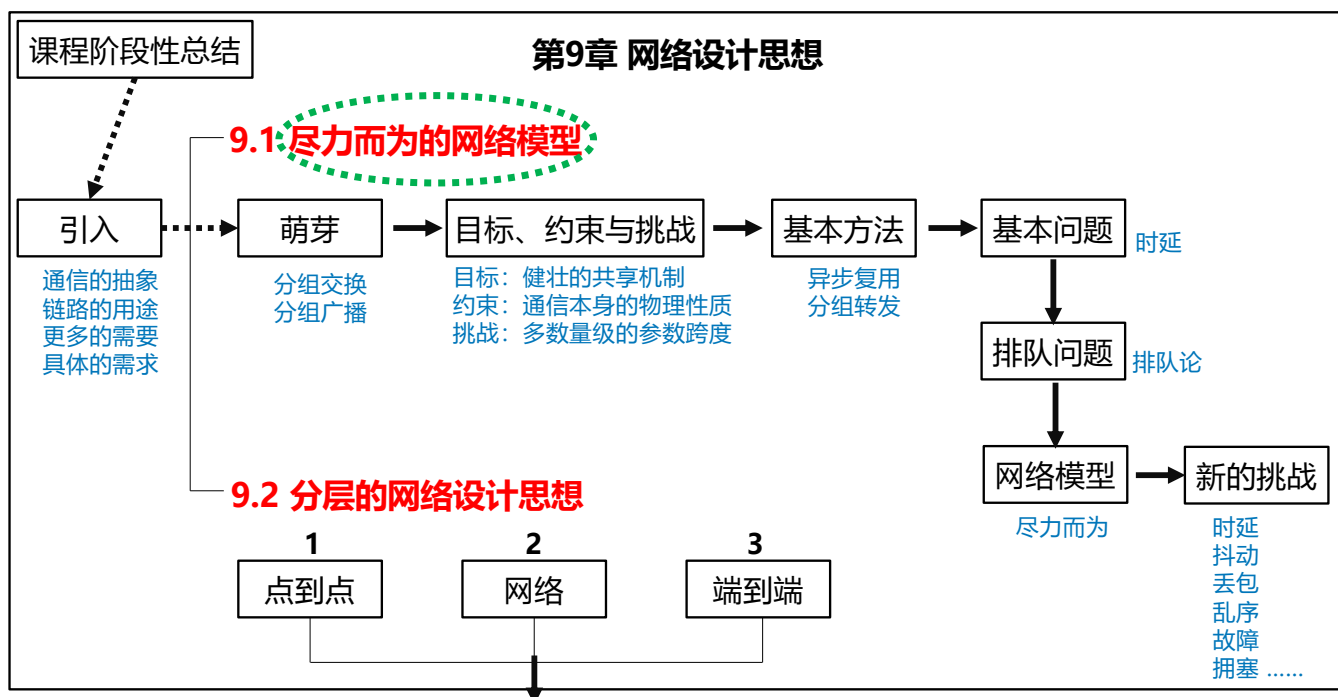
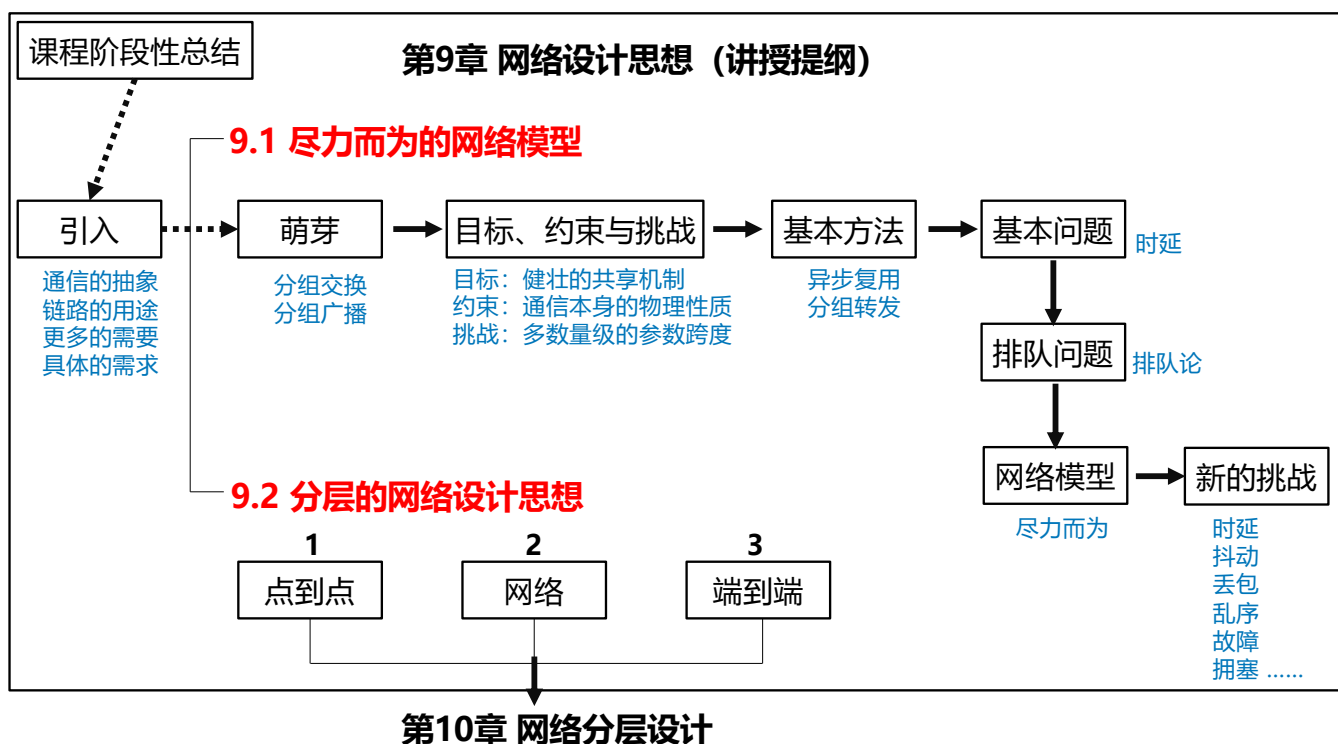
通过共享路径

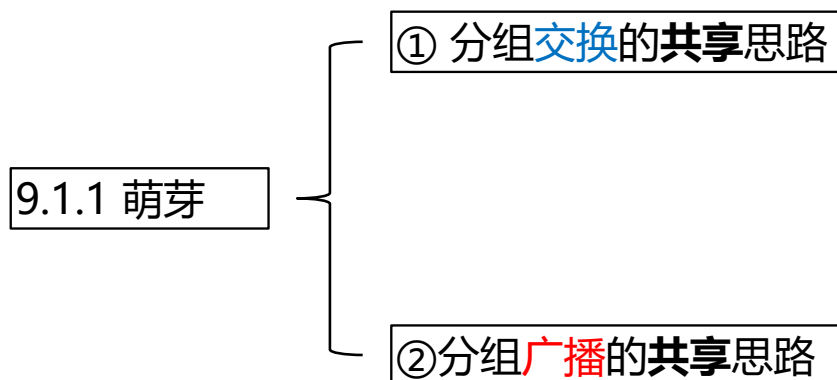
● 这些**共享**的需求产生了**网络**的思想和设计探索

共享！



我们用三次课重温前人的这些思想与探索





共享节约成本， **more with less**，是工程师孜孜不倦的追求！

## ①从电路交换到分组交换网

### ● 电路交换

- 电话交换机 1877，自动交换机 1888

适用于语音

### ● 包交换



- Paul Baran@RAND Co. in US, 1960-1964:

适用于数据

背景：Cold War, Nuclear War

灵感：Rand主管Frank Collbohm的模拟链路应急通信思想

架构：Distributed relay node architecture

发明：Store-and-forward packet switching, routing

- Donald Davies@National Physical Laboratory in UK, 1965-1968:

背景：计算机通信的"bursty"

灵感：time-sharing computer systems

发明：packets, protocol

- ARPANET 1969 ARPA of DoD

不同的需求  
产生相同设计

## ②从模拟广播到分组广播网

模拟广播 → 数据广播

### 1. ALOHAnet (无线/数据包)

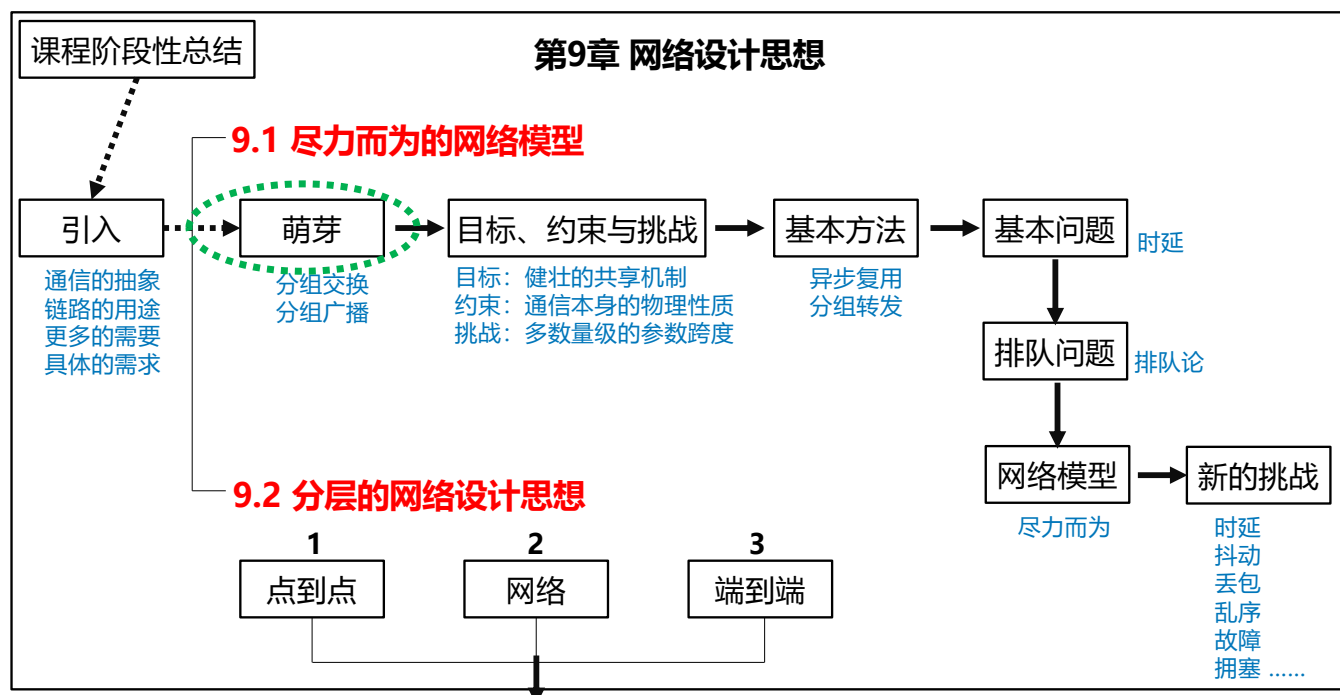
- 1971
- 夏威夷大学

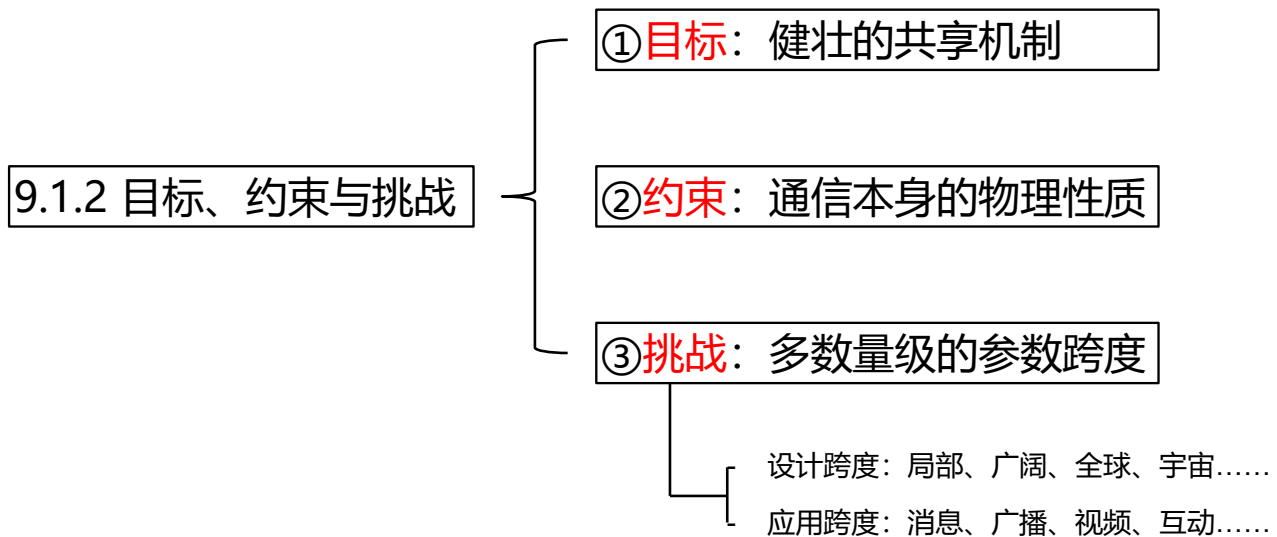


分组交换和分组广播  
哪个成本更低?  
哪个适用于远程传输?

### 2. 以太网 (Ethernet/Frame)

- 1974
  - Xerox PARC
- 分组也可以运输分组





## 设计目标

1. 共享的经济目标：
  - 减少链路数量，降低通信代价
2. 共享的实施对象：
  - 以链路为单位进行共享
  - 以带宽为单位进行共享
3. 共享的设计目标：
  - 复用、容错、高效、可靠：哪个更重要、哪个不能保证？
  - 网络设计目标：尽力而为（Best-effort）还是使命必达？

# 设计约束

## (1) 速度有上限

- 青岛-乌鲁木齐: 3000 km。 传播时延 (propagation delay) 10 ms
- 青岛-同步卫星-乌鲁木齐:  $36000 \times 2 = 72000$  km。 传播时延 240 ms
- 临近计算机: 3-10米。 传播时延 10 ns

## (2) 环境较恶劣

- 无线、电缆、光纤、海底光缆
- 噪声、损坏

## (3) 带宽有上限

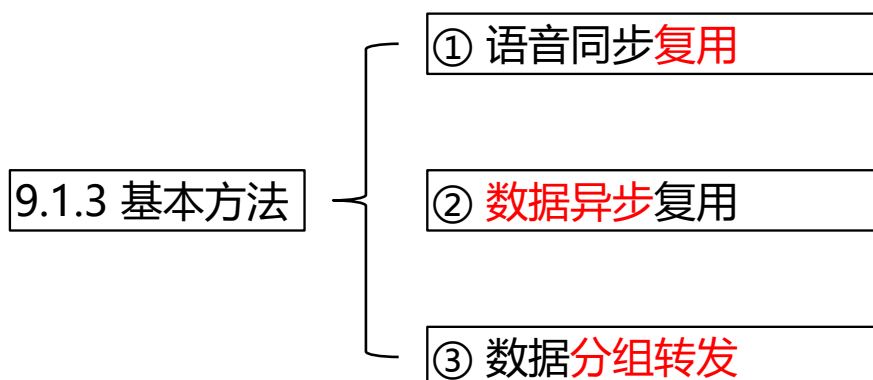
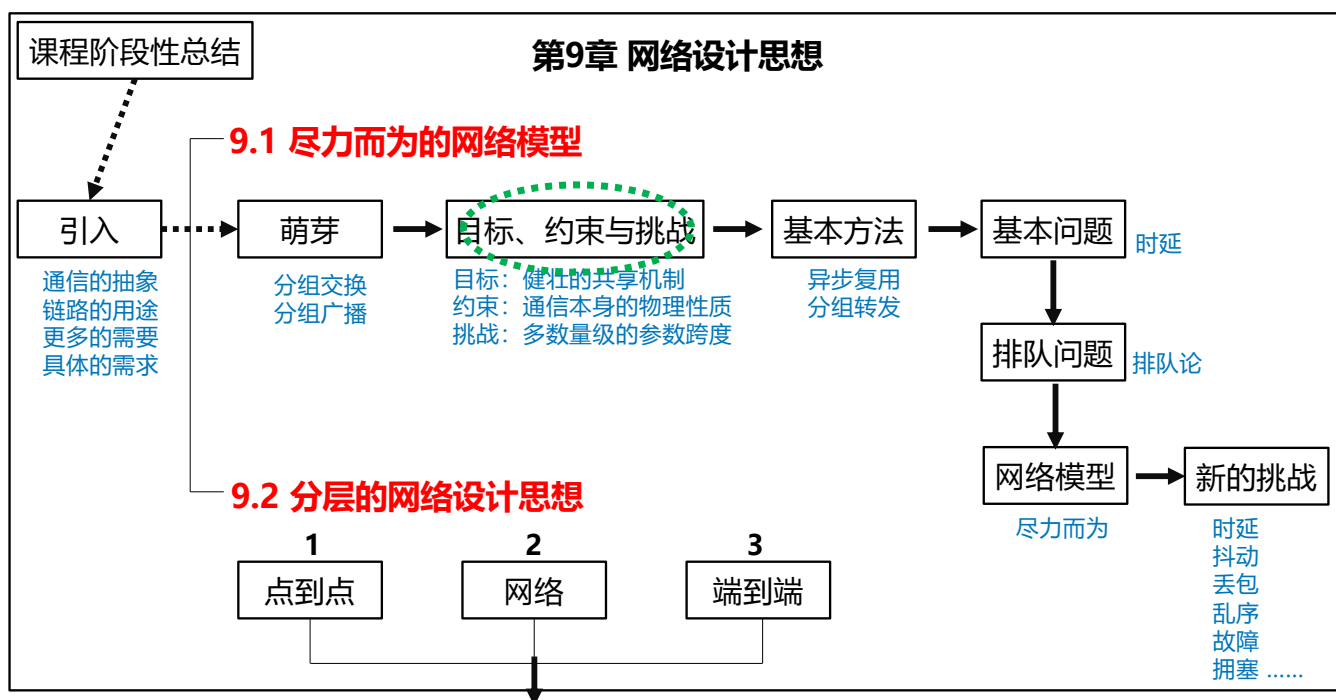
- 信号率受距离、衰减特性约束, 信号率、功率、噪声决定了数据率上限
- 香农信道容量公式 (Shannon's capacity theorem)

# 设计挑战

## 物理性质和网络规模: 参数跨越7个以上的数量级

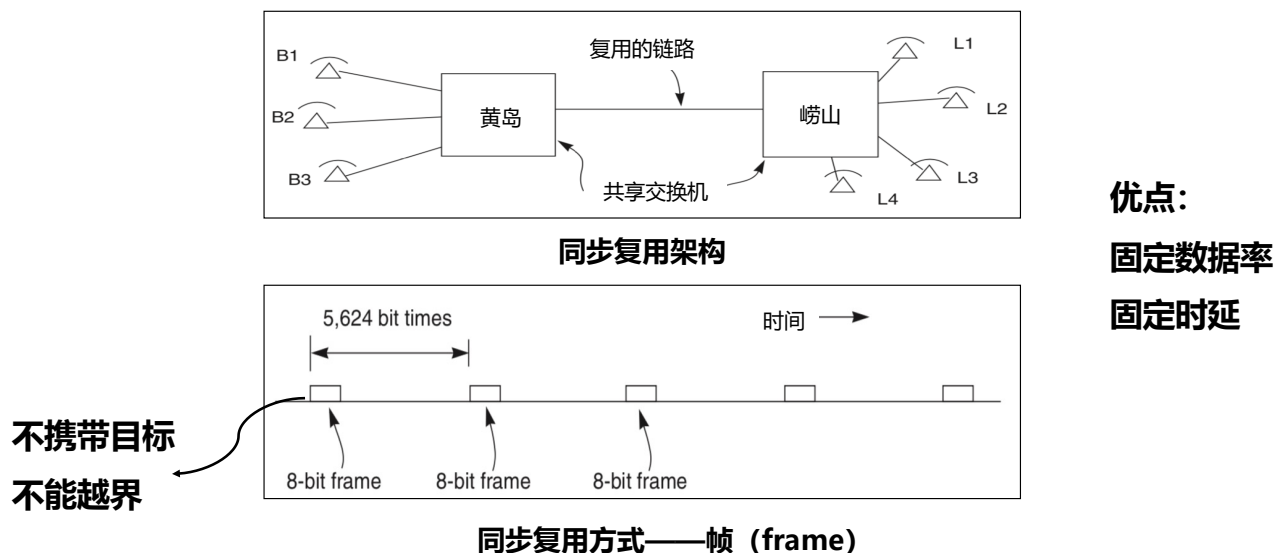
- 传播时间
- 数据率
- 计算机数量
- 网络负载
- .....



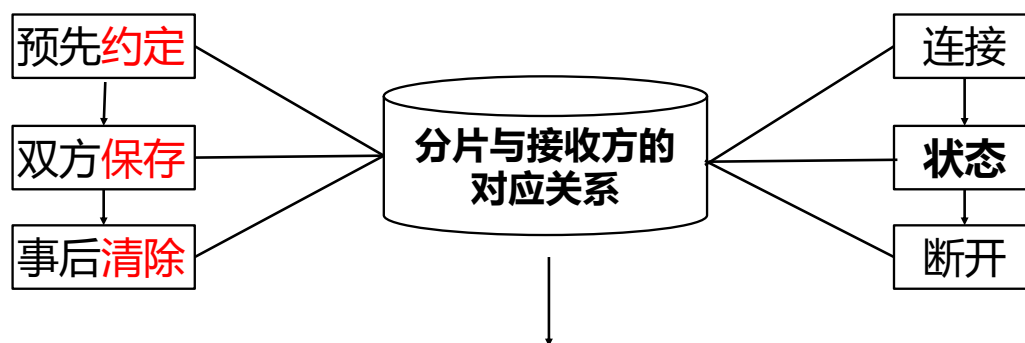


巧妙的协调，工程师的艺术！

# ① 语音数字通信的同步复用



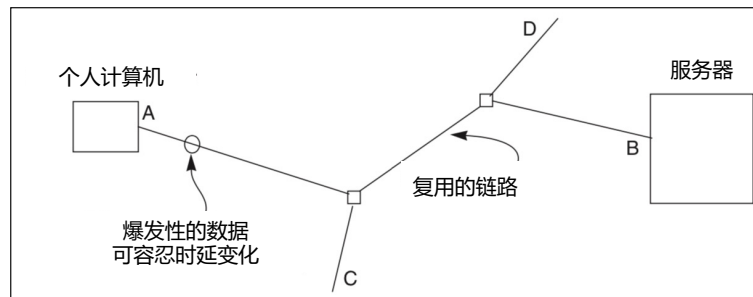
## 同步复用的开销与缺点



1. 提前约定状态，不灵活
2. 资源固定占用，没有伸缩性

# 数据通信的特点

1. 时间无规律性、数据大小不定长
2. 不要求、也无法要求固定数据率和时延



数据通信的特点

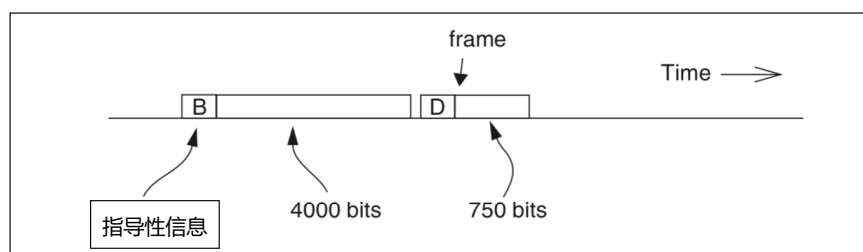
## ② 数据通信的异步复用



### 思路

1. 不提前建立、不存储状态、不需要终止 → 无连接

2. 为数据帧增加目标信息
  3. 为数据帧增加定界信息
- 异步数据分组



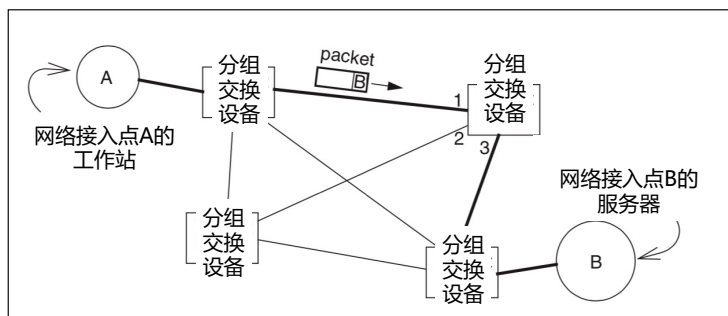
数据的异步复用方式——带指导信息的frame

### ③ 数据通信的异步转发



- 数据异步通信架构：分组转发网络


- 转发设备：分组交换设备（交换机switcher/路由器router）
- 接入设备：网络接入点（network attachment point）
- 需求：确定路径（routing）、转发（forwarding）



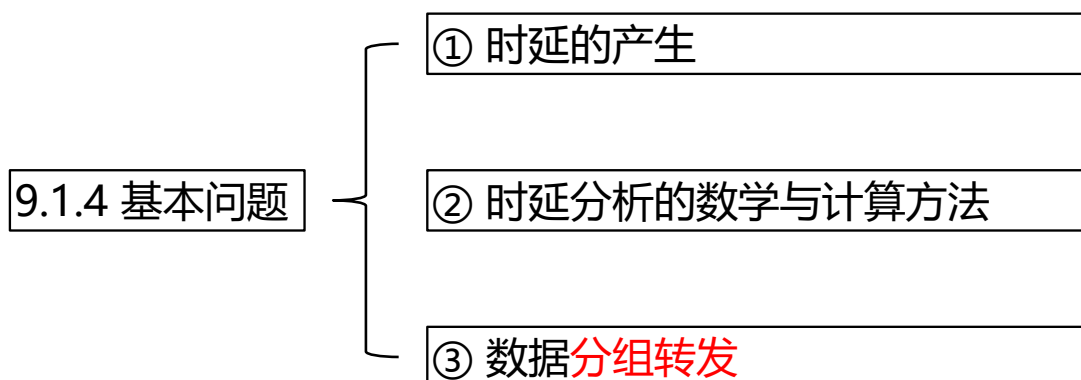
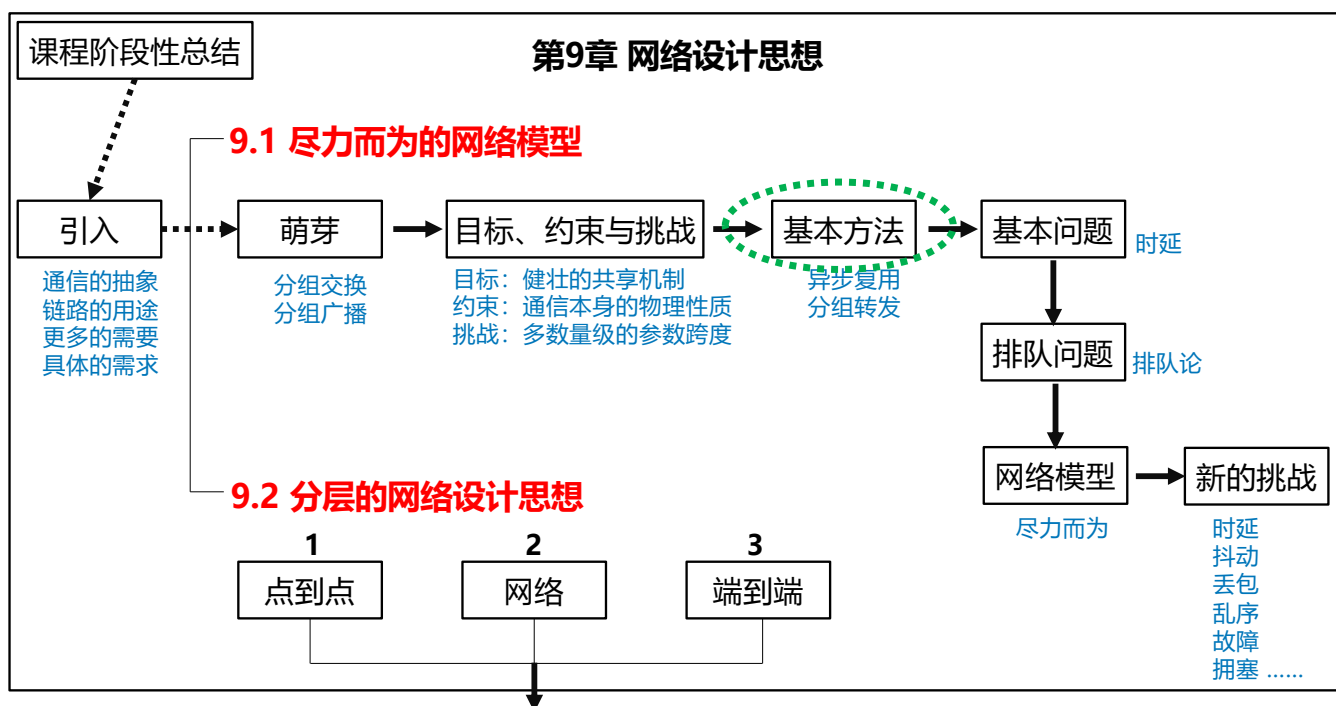
异步通信架构

## 休息一下

- 我们现在打国际长途电话，使用的是：

- 同步复用？
- 异步复用？
- 为什么？
  - ▶ 伸缩性、成本





将数理知识用于工程设计，是工程师的必备的能力！

# ①基本问题：时延的产生

复用 → 异步通信链路 → 异步通信结构 → 转发 → 时延

转发形成了流水线结构，带来**排队**问题

- 排队带来性能的波动
- 排队可导致服务失效

# ②时延分析的数学与计算方法

复用 → 异步通信链路 → 异步通信结构 → 转发 → 时延

用数学工具研究排队：运筹学 - 排队论

- 为管理决策提供科学依据的应用数学
- 以 概率论、随机过程、博弈论等为基础

排队论恰恰起源于对电话排队研究：

Erlang A K. The theory of probabilities and telephone conversations[J]. Nyt. Tidsskr. Mat. Ser. B, 1909, 20: 33-39.

# 时延分析：最简单的M/M/1排队模型

- 请求抵达的分布： $\sim P(\lambda)$ 
  - 请求是独立、随机到达的 → 请求的间隔时间呈指数分布
- 服务时间分布： $\sim P(\mu)$ 
  - 指数分布、独立的服务时间
- 服务数量：单服务
- 服务策略：先来先服务 (FCFS)
- 排队和请求的最大容量均无上限

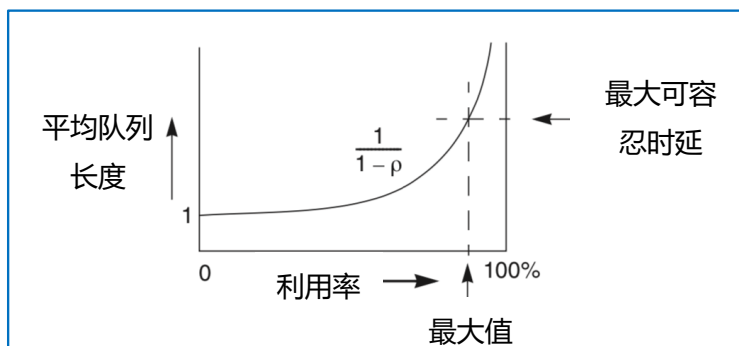


1.  $\lambda$ 比 $\mu$ 大会怎么样?
2. 单位时间到来的请求数量呈什么分布?
3. 随着利用率的提升, 队伍有什么变化?

## 最大利用率的计算

利用率与排队时延的关系：平均排队长度 =  $1/(1-\rho)$

- 应用：求**最大利用率**



- 这是**随机**假设下的平均结果 (**不随机会怎样?**)
- 能否改善?
  - 在高利用率的情况下, 通过设置排队上限, 把问题推给上一环节

# 排队缓冲区长度的计算

**问题：排队的缓冲区需要设置为多长，才能保证排队的请求不丢失？**

- 假设：队列长度的均值和方差都为 $1/(1-p)$ ，近似为正态分布

1. 传统方法：使用z变换 $z=(x-\mu)/\sigma$ 转换为标准正态分布，再[查表](#)

2. 使用计算工具 (scipy.py)：

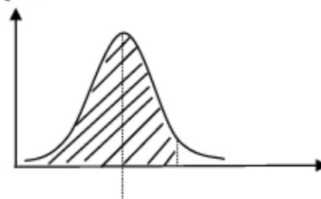
- 直接计算累积分布 (cdf)
- 或给定累积分布求分位点 (ppf)

- [\[试一试\]](#) 几个标准差的区间概率才感觉是安全的？

- $\sigma$ 、 $2\sigma$ 、 $4\sigma$ 、 $6\sigma$ 分别是多少？

标准正态分布函数表（形式1）

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$



$\Phi(x)$ x										
x	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0.5	0.504	0.508	0.512	0.516	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.591	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.648	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.67	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.695	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.719	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.758	0.7611	0.7642	0.7673	0.7703	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.791	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.834	0.8365	0.8389
1	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.877	0.879	0.881	0.883
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.898	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9278	0.9292	0.9306	0.9319



# 应对排队问题的不同服务策略

## 1. 服务可应对最坏情况

- 最坏情况难以预测 → 拥塞 (congestion)

## 2. 服务可应对常规情况，不能应对则向前反馈

- 不可行 → 丢弃

为什么反馈 (fight back) 不可行？

## 3. 服务可应对常规情况，不能应对则丢弃分组

- 简单粗暴 → 却是唯一可行的

# 实际中的网络排队模型

## 要考虑用户端系统的因素

- 端系统改进：速率自适应模型 (automatic rate adaptation)
  - 缓解网络拥塞
- 模型的变化：不再依照标准排队模型
  - 受到网络和网络应用程序的策略影响
  - 更适用于博弈论方法

## 👉设计者思考：保证交付还是尽力而为？



### 保证交付 (guaranteed- delivery) 网络模型设计

- 使用message而非packet
- 依靠非易失存储
- 跟踪交付情况
- 失败向上报告

### 尽力而为 (best-effort) 网络模型设计

- 复用→异步→时延→排队→丢弃→best effort

## 设计决策：尽力而为

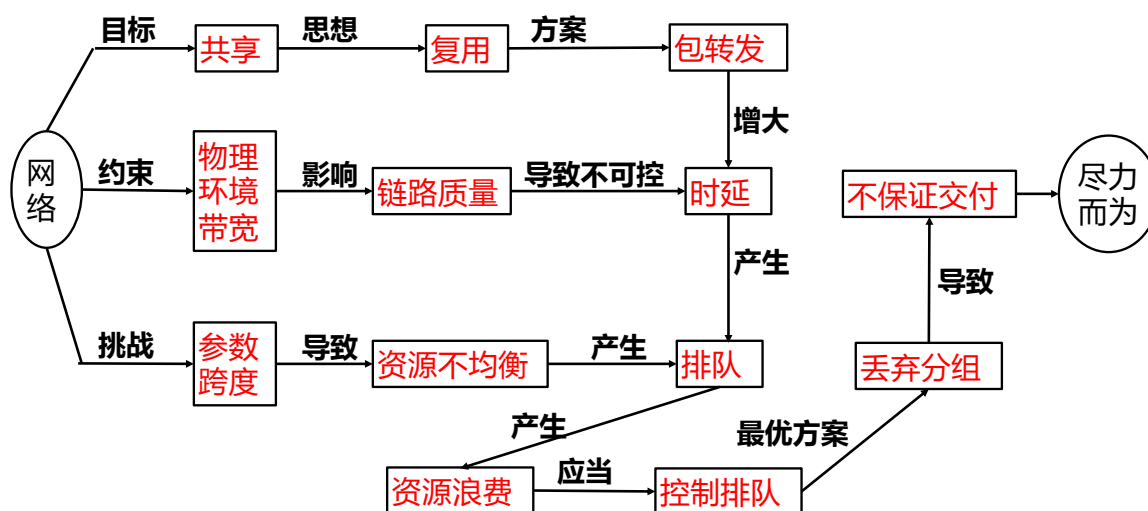


- 可靠性与成本的权衡取舍 (trade-off)
- End-to-End论点 (课后阅读)
  - Saltzer J H, Reed D P, Clark D D. End-to-end arguments in system design[J]. ACM Transactions on Computer Systems (TOCS), 1984, 2(4): 277-288.
    - google引用3632 (.2.20)
    - Saltzer J H: MIT,
    - Reed D P: UDP协议设计者,
    - Clark D D: 因特网首席协议架构师, IAB主席 (1981~1989)

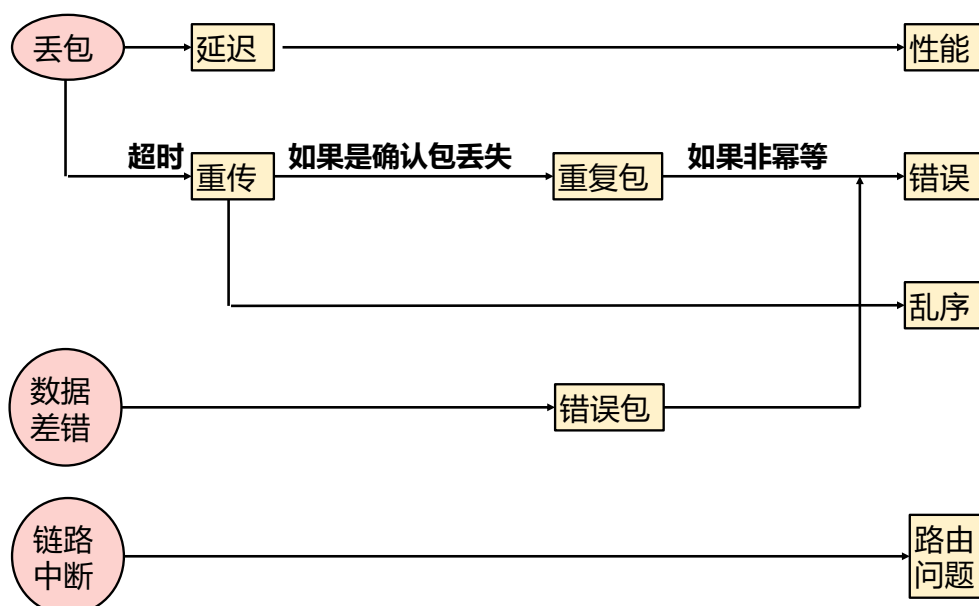
# 权衡：尽力而为是工程权衡的结果



- 尽力而为 (Best-effort) ~ 权衡取舍 (Trade-off)



## 新问题：尽力而为带来的问题

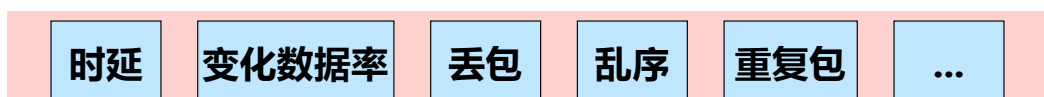


## 迎接挑战：环境与设计带来的若干属性

- 大范围变化的
  - 数据率
  - 传播、传输、排队、处理时延
- 恶劣环境导致的
  - 噪声破坏数据
  - 链路中断



尽力而为



# 分层设计

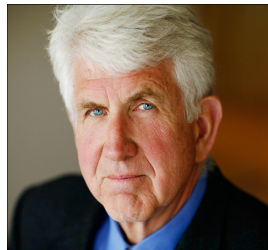
# 网络的设计——三个主要任务

## 设计任务

### 1. 分组点到点：复用物理链路

- 胜出：以太网/交换以太网

2022 ACM A.M. Turing Award  
Bob Metcalfe, US



For           Invention,  
Standardization,   and  
Commercialization of  
**Ethernet.**

# 网络的设计——三个主要任务

## 设计任务

### 2. 分组交换网络：共享虚拟链路

- 胜出：IP

### 3. 分组端到端：面向应用的传输

- 胜出：TCP、UDP、TLS

2004 ACM A.M. Turing Award  
VINTON GRAY CERF, US      ROBERT ELLIOT KAHN, US



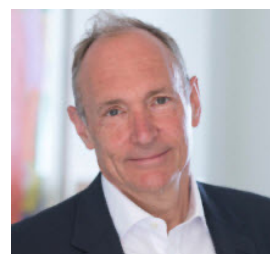
For pioneering work on internetworking, including the design and implementation of the Internet's basic communications protocols, **TCP/IP**, and for inspired leadership in networking.

# 网络的设计——三个主要任务

## 设计任务

- 基于网络的计算/应用：面向应用需求
  - 胜出：HTTP、IMAP、DNS、其他分布式计算协议

2016 ACM A.M. Turing Award  
SIR TIM BERNERS-LEE, UK



For inventing the **World Wide Web**, the first web browser, and the fundamental protocols and algorithms allowing the Web to scale.

# 网络的设计——三个主要任务

## 设计任务

- **分组点到点：复用物理链路**
  - 胜出：以太网/交换以太网
- **分组交换网络：共享虚拟链路**
  - 胜出：IP
- **分组端到端：面向应用的传输**
  - 胜出：TCP、UDP、TLS
- 基于网络的计算/应用：面向应用需求
  - 胜出：HTTP、IMAP、DNS、其他分布式计算协议

**每个任务建立在上一任务之上  
→ 设计模式：分层**

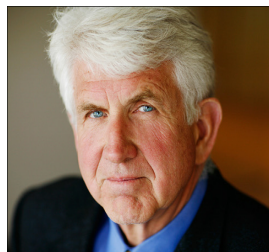
# 计算机网络的先驱，图灵奖得主

2004 ACM A.M. Turing Award  
VINTON GRAY CERF, US ROBERT ELLIOT KAHN, US

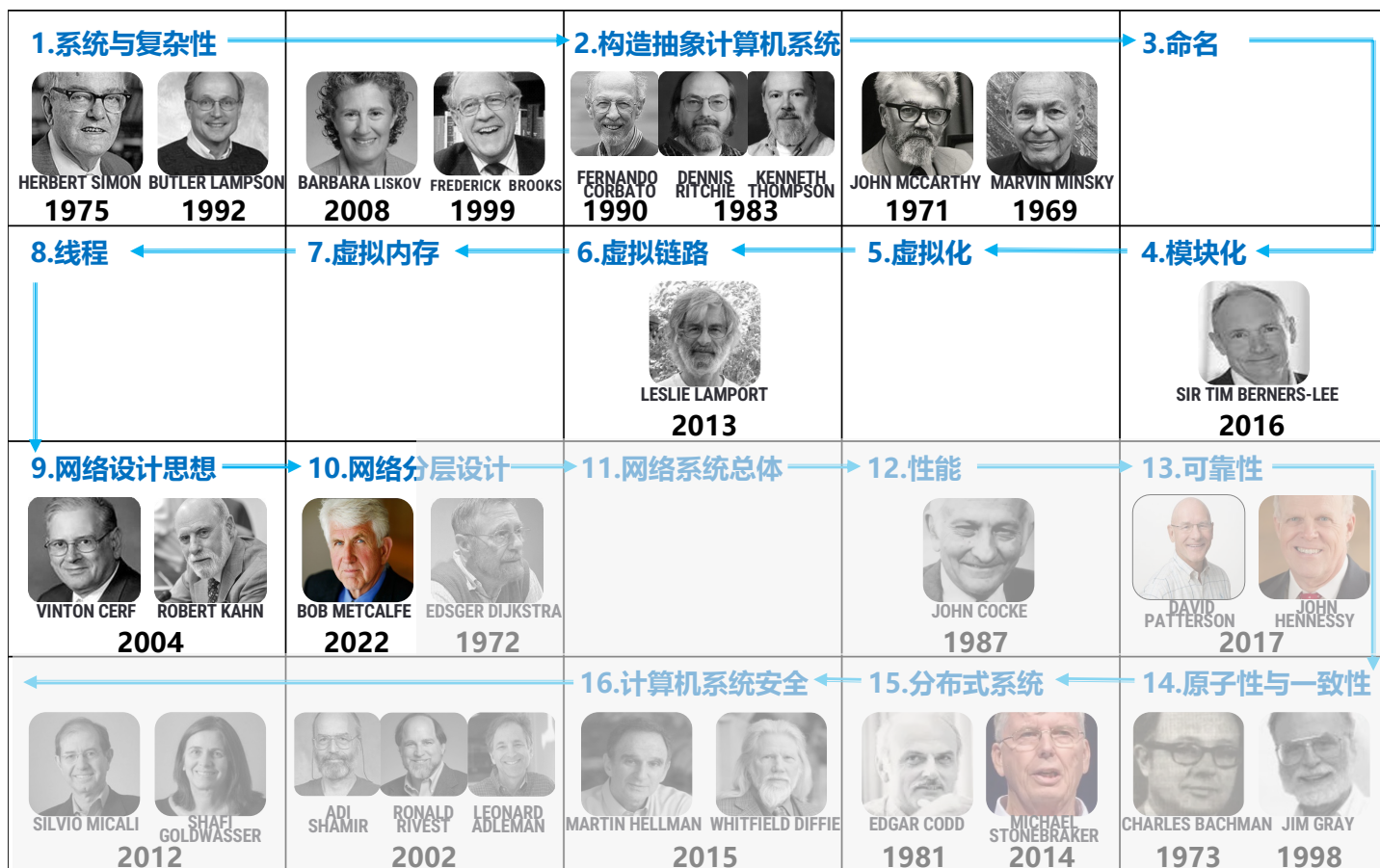


For pioneering work on internetworking, including the design and implementation of the Internet's basic communications protocols, **TCP/IP**, and for inspired leadership in networking.

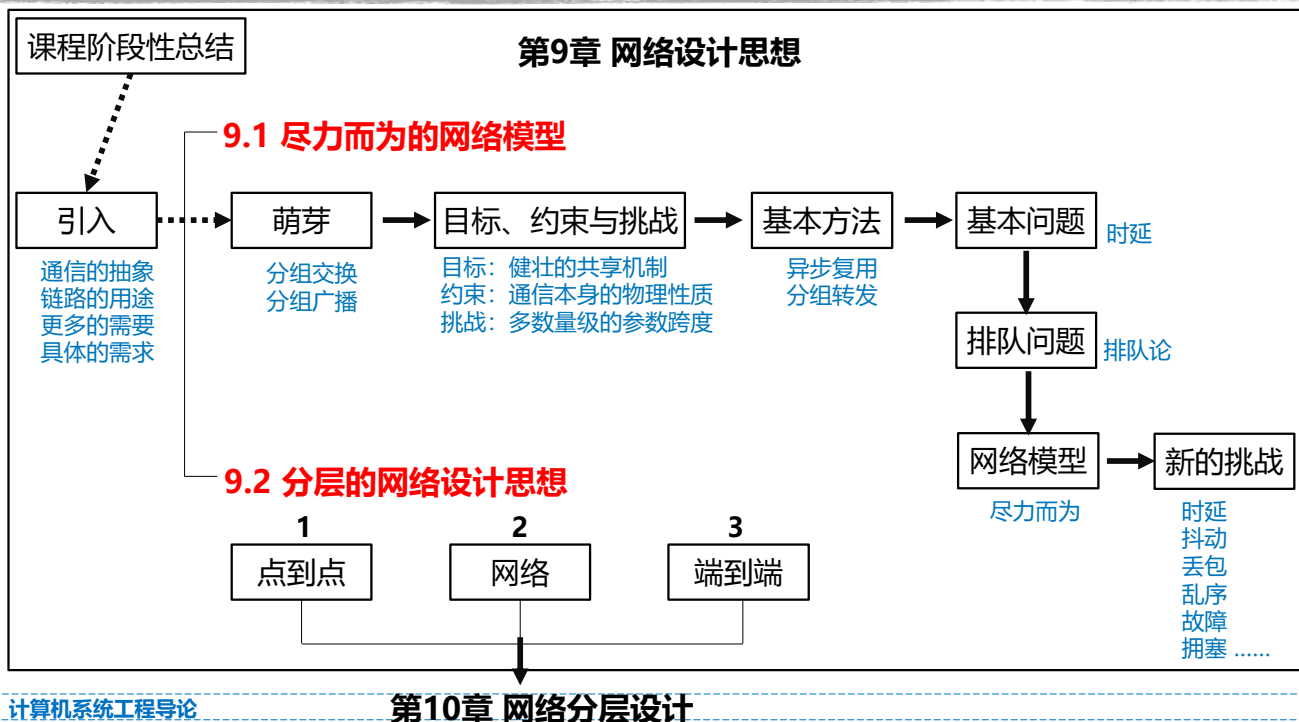
2016 ACM A.M. Turing Award  
SIR TIM BERNERS-LEE, UK 2022 ACM A.M. Turing Award  
Bob Metcalfe, US



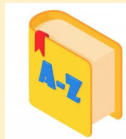
For inventing the **World Wide Web**, the first web browser, and the fundamental protocols and algorithms allowing the Web to scale. For Invention, Standardization, and Commercialization of **Ethernet**.



# 总结



## 重要术语中英文对照



速率自适应：automatic rate adaptation

保证交付：guaranteed- delivery

消息：message

帧：frame

包：packet

数据报：datagram

尽力而为：best-effort

拥塞：congestion

交换机：switcher

路由器：router

传播时延：propagation delay





# 文献阅读与参考文献



## ● 文献阅读及作业

- ✎ J. H. Saltzer, D. P. Reed, and D. D. Clark. 1984. End-to-end arguments in system design. *ACM Trans. Comput. Syst.* 2, 4 (Nov. 1984), 277–288.

## ● 参考文献

- ✓ Clark, D. Annotated Version of The Design Philosophy of the DARPA Internet Protocols, March 2013.
- ✓ Greenberg, A., Hjalmtysson, G., Maltz, D. A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., and Zhang, H. A Clean Slate 4D Approach to Network Control and Management. *SIGCOMM Comput. Commun. Rev.* 35, 5 (Oct. 2005), 41–54.



# 文献阅读与作业



## ● 论文 “End-to-End Arguments in System Design” 。

- 该论文发表在ACM TOCS上。
- 作者Saltzer是MIT计算机系统工程课程的创始人之一，是另两位作者的博士导师。作者Reed是两个主要的数据传输协议之一的UDP协议的发明人，作者David Clark曾任IAB（因特网架构委员会，ISOC的技术顾问组织，管理IETF、IATF和RFC制定等）首届主席以及因特网首席协议架构师。
- 论文针对系统的功能应当在何处实现这一设计决策问题提出了讨论。前两章给出了部分该问题的实例，后面的章节则进行了更为细化的讨论。



# 文献阅读与作业



## 阅读论文时，请思考以下问题

1. 是否用加密来保证安全通信应该在端到端过程实现，而非在系统通信底层实现？
2. 通信系统确保数据包的顺序和不重复，这对所有应用都是好的设计吗？
3. 对于迄今本课程已经涉及到的内容，End-to-End论点适用于哪些问题？

## 作业：请回答以下问题

1. 端到端论点是什么 (它阐述了什么)？
2. 这一论点如何在实际中运用？至少举出一个例子。
3. 你同意这一论点吗？为什么？

## 休息一下

frame、packet、datagram、segment、fragment、message  
有什么区别？

frame：可以定界的数据单元——帧

packet：下层承载的数据单元——分组(数据包)

--datagram：自我包含的数据包——数据报

--segment：有上下文的数据包——数据段

--fragment：数据包被分割成小数据包——分片

message：应用有完整语义的数据单元——消息



# 第9章 结束

