**1.概述**

协议三要素：语法、语义、同步

网络中互相通信的对等实体间进行数据交换而建立的规则、标准或约定

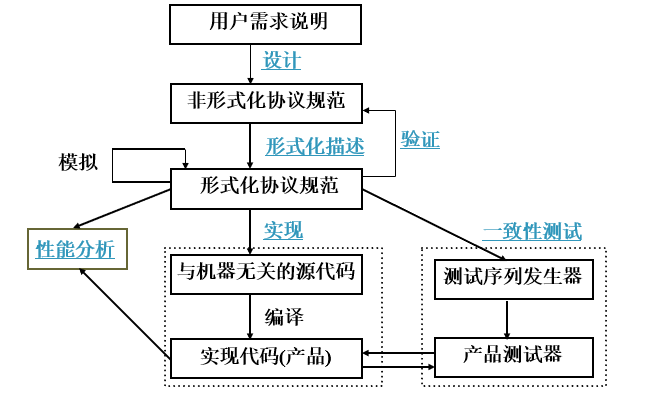
语法体现为数据报文中的控制信息(通常在报文的首部)和各种控制报文的结构、格式

语义可以理解为协议数据报文中的控制信息和控制报文所约定的含义

同步是指事件发生顺序的详细说明

协议工程过程是集成化、形式化的协议开发过程

协议工程过程中协议的表现形式：



根据协议的需求说明构造协议的非形式描述文本(经过非形式的验证分析) 称为协议设计；

而将“协议的设计、形式描述、验证、实现、测试、运行”这一全过程称为“协议开发”，即协议开发是指协议工程活动的全过程。

形式描述模型：有限状态机FSM、Petri网、时态逻辑TL等

验证方法：模型检查（重要问题：状态空间爆炸）、证明、模拟

**2.协议设计**

分层的好处：（1-48），分层的原则（2-92）

服务涉及层与层之间的接口，可看做面向对象语言中的抽象数据类型

协议涉及不同机器上对等实体之间发送的数据包

协议是“水平的”，即协议是控制对等实体之间通信的规则。

服务是“垂直的”，即服务是由下层向上层通过层间接口提供的

协议模型：51（PDU、SAP等）

4种服务原语

在进行交互时所要交换的一些必须信息(或命令)称为服务原语

同一系统相邻两层的实体进行交互的地方，通常称为服务访问点SAP

对等实体之间所交换的信息单元称为协议数据单元PDU (Protocol Data Unit)，PDU通常由2部分构成：用户数据和协议控制信息PCI，分为两种类型，数据PDU和控制PDU（不携带用户数据）

N层中任何两个协议实体通过(n-1)SAP所形成的逻辑数据通路称为(n-1)层通道，分类：空通道、非缓冲通道、缓冲通道，通道的工作方式：单工、半双工、全双工、同步和异步，通道的通信方式：点对点、点对多点

协议提供的服务：面向连接的服务、无连接的服务。面向连接的服务的三个阶段：连接建立、数据传输和连接释放。无连接的服务三种类型：数据报、证实交付、请求应答

协议功能：连接管理、数据交换

差错控制：序号、确认、计时器、重传（差错的四种类型：失真、丢失、重复、 失序）

拥塞控制：

差错控制和拥塞控制的区别和联系：47

协议组成：服务原语（原语名字、原语类型和原语参数）和服务原语时序、协议数据单元PDU 和PDU交换时序 、协议状态、协议事件、协议变量、协议操作和谓词

协议的主体是它的状态-事件转换机制

协议事件：输入事件（三类）、输出事件（两类）

通信事件、内部事件

性质：成对性、原子性、时序性

协议运行的三种方式：协议交替（交替活跃）、协议并发（同时活跃，并发运行）、协议并行（同时活跃，同时运行）

协议文本中最重要、最主要部分是协议元素的描述。必须对每一项协议元素作出准确、清晰、无二义性的定义。

一个好的结构化协议设计应具备的特点是简单和模块化

第3章

形式描述模型(FDM) ：

状态变迁模型

有限状态机FSM (Finite State Machine)

扩展的有限状态机EFSM (Extended FSM)模型

通信有限状态机CFSM (Communicating FSM)模型

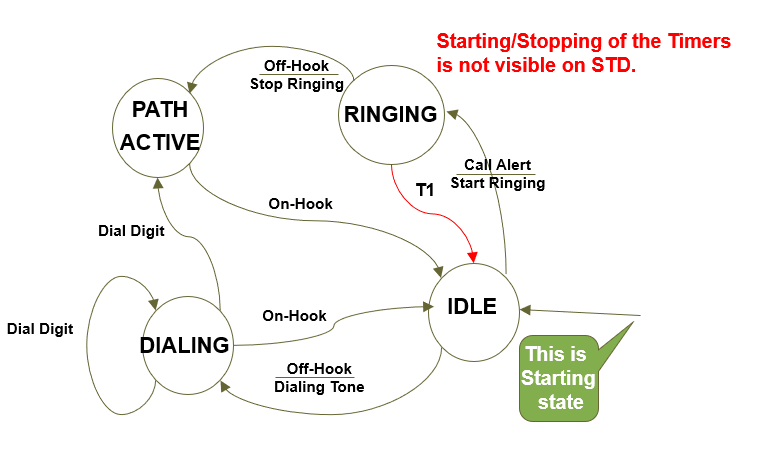
Carl Adam Petri的Petri网 (PetriNet)

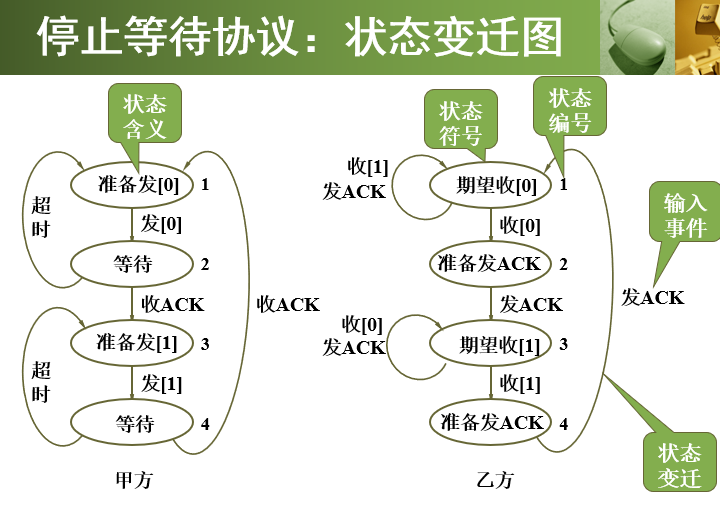
时态逻辑TL (Temporal Logic)

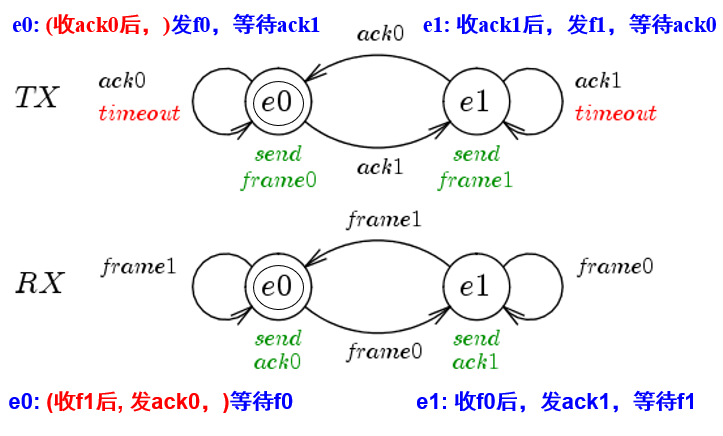
进程代数 (Algebra of Process)

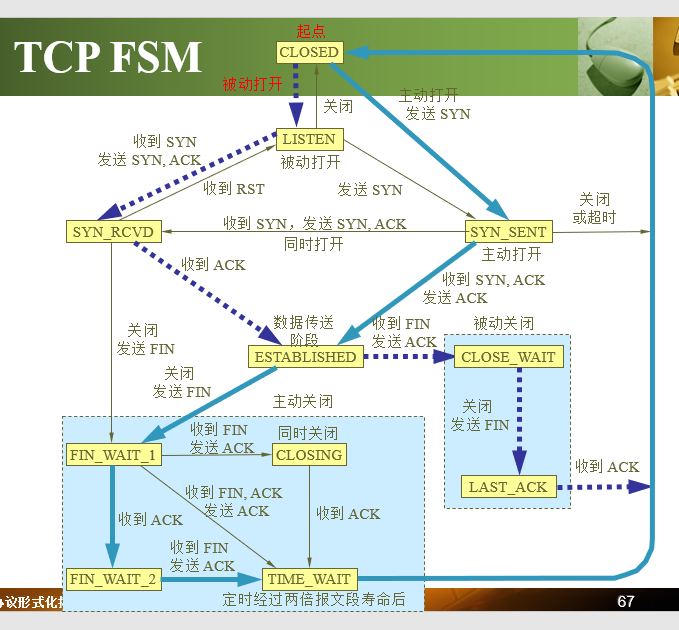
Moore机：输出与状态有关

Mealy机：输出与状态和输入有关









FSM的简化：

1. 状态层次化

2. 使用原子过程

3. 隐藏内部协同事件

4. 简化通道FSM

5. 使用协议变量

四种常见的FSM扩展模型：

EFSM（Extended FSM）

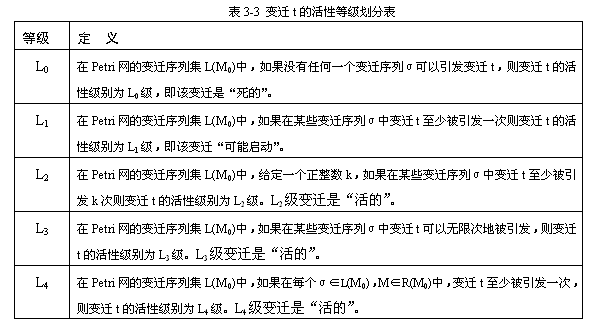
CFSM（Communicating FSM）

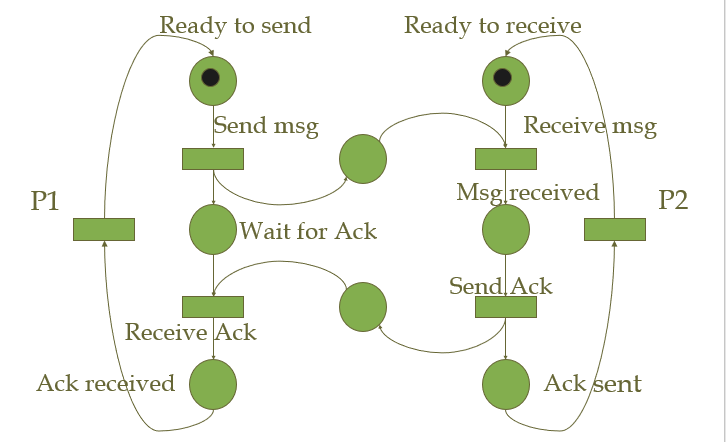
CEFSM（Comunicating EFSM）

PFSM（Probability FSM）

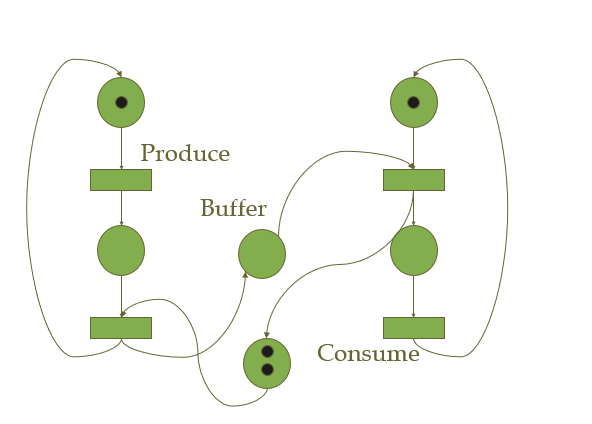
FSM的错误类型：死锁（进行没有可以接收的输入）、活锁（进程在几个状态之间陷入死循环）、无限分支（无法停止的程序分支）、不可达状态（即不可能执行到的状态）

PetriNet性质：可达性、有界性、安全性、活性、可逆性、可覆盖性、个持续性、公平性





限制缓冲区大小：



**协议形式化描述语言**

SDL是基于抽象数据类型和扩展有限状态机的混合技术

SDL三种语法：图形表示的SDL/GR、文字短语表示的SDL/PR、程序语言形式的SDL，X.250

一个SDL应用的静态结构由系统（System）、功能块（Block）、进程（Process）、过程（Procedure）和通道（Channel）构成

功能块描述规范包括：进程、过程(Procedure)、服务类型、数据类型和信号。进程的状态与动作是用EFSM (扩展有限状态机)来建模的

功能块描述部分主要描述进程与进程之间、进程与通道之间的通信，

功能块内通信手段主要包括共享变量或通过信号路由交换信号。

在SDL中，将系统、功能块、进程称为代理（Agent）

SDL中类的定义：值定义、操作符定义和等价式定义

SDL/GR和SDL/PR的相互转换

**协议验证技术**

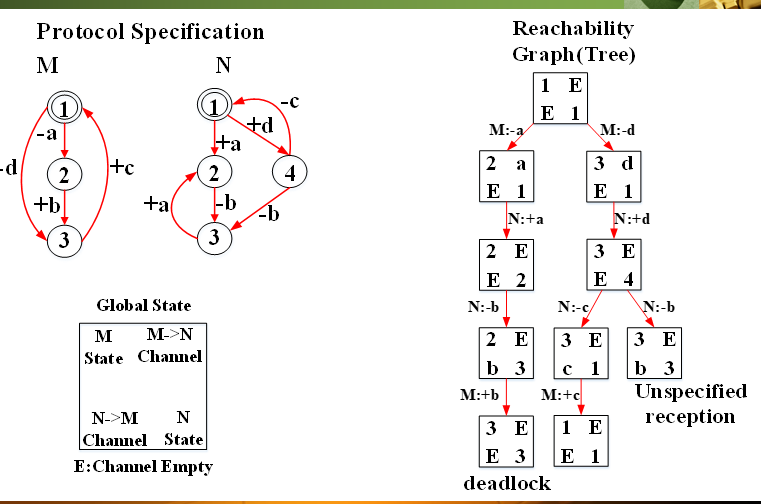
协议验证，可以获知协议设计是否满足正确性、完备性和一致性等要求。

方法：模型检测(Model Checking)：基于状态搜索(State Exploration)

演绎验证(Deductive Verification)：基于定理证明(Theorem Proving)

协议验证着重在于一般性质，而协议测试则着重于特殊性质。

可达性分析(RA: Reachability analysis):检查协议描述中是否有不可达，死锁、未定义、活锁等协议模型错误；包括状态穷举，状态随机枚举，状态概率枚举等方法



可达性分析算法

穷尽性可达性分析算法

受控部分搜索算法

随机模拟算法

三者优缺点比较：

穷尽性可达性分析算法的优点在于可以证明协议中没有错误，但问题在于其应用范围有限。对于复杂协议来说，属于低质量的部分搜索

受控部分搜索算法的目的是证明错误的存在，而不是证明没有错误。可以利用有限的资源来验证协议的最重要的部分，从而最大限度地发现错误，但必须能够预测出协议中的错误的可能位置（很困难），需要多次验证。

随机模拟算法与协议系统的大小和复杂性无关，但没有明确的终止，也无法确定是否发现了全部的错误。

解决状态空间爆炸问题（状态空间爆炸是因为不同实体执行的动作相互交织所造成的）的方法：

部分规范描述和验证：根据具体描述方法的特点，只选取协议的某些方面进行描述

分解或划分：将协议分解或划分成多个可控制的子模块，然后对每一个子模块进行独立地描述和验证

选择大的动作单元：可以通过合并一些状态或动作来减少状态及变迁数

按断言来分类状态：考虑状态类而不是单个状态，通过选择合适的谓词，可以大大减少状态数目

不变性分析两种途径：不变性证明系统、不变性监测系统

**协议一致性测试技术**

验证一项新的协议实现，一般需要考虑以下四个方面的测试：一致性测试、互操作测试、性能测试、鲁棒性测试

一致性测试方法：功能测试、结构测试

一致性测试级别（由低到高）：

基本互连测试

能力测试

行为测试（可以分为覆盖性测试和穷尽性测试）

一致性分解测试

一致性测试要求：静态一致性要求（规定了选择选项的限制条件，包括协议实现应该提供的最小能力，不同选项之间的组合和协调性）、动态一致性要求（对实现与环境通信的可观察行为的要求）

简述一致性测试的流程：

确定测试目的。一致性测试者根据协议规范、服务规范确定测试目的

测试集生成。根据协议规范、服务规范以及测试目的生成并描述测试集

测试集实现。将抽象测试集用测试执行系统能够认识的语言描述

测试执行。对被测试协议实现运行已具体化的测试例，对IUT外部响应进行观察和记录

结果评估。根据测试记录并参照PICS和PIXIT对IUT进行评估，并给出一致性测试报告

一致性测试方法：本地测试法、分布式测试法、协调测试法、远程测试法

简述本地测试法、分布式测试法、协调测试法、远程测试法的区别

本地测试法：在这种方法中，LT，UT，IUT同处于一台机器中，测试不需要底层通信系统的支持

分布式测试法：在这种方法中，IUT和UT处于一台机器中，而LT则分布在其他机器中。LT相当于（n-1）层服务的使用者

协同测试法：协同测试法和分布式测试法的根本区别在于协同测试法引入测试管理协议TMP。有了TMP，UT和LT就通过交换TM-PDU实现测试协同过程

远程测试法：这种测试法没有UT，因此也不存在UT和LT之间的协同问题

基于FSM的测试集生成方法：变迁游历法、特殊交互作用序列法