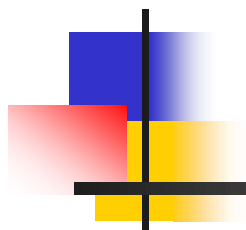


宽带通信网



信息与通信工程学院

靳 浩



宽带通信网络概论

- 关于宽带通信网
- 宽带通信网络的演进
- 宽带通信网络的未来
- 宽带通信网络技术的机理



关于宽带通信网

- 宽带通信网的概念
- 宽带通信网的关键技术
- 宽带通信网技术的外延
- 宽带通信网课程内容



宽带通信网的概念

- 宽带通信网是一种全数字化、高速、宽带、具有综合业务能力的智能化通信网络，宽带通信网的显著特点就是在信息数据传输上突破了速度、容量和时间空间的限制。
- 美国最早关于“宽带”的定义是超过200Kb/s；目前，对“宽带”的一般界定是速率在2Mbit/s以上；
- 目前的宽带网技术能达到的带宽
 - 接入（有线/无线）
 - 传输
 - 交换



宽带通信网的关键技术

- 宽带通信技术
- 微电子技术
- 数字信号处理技术



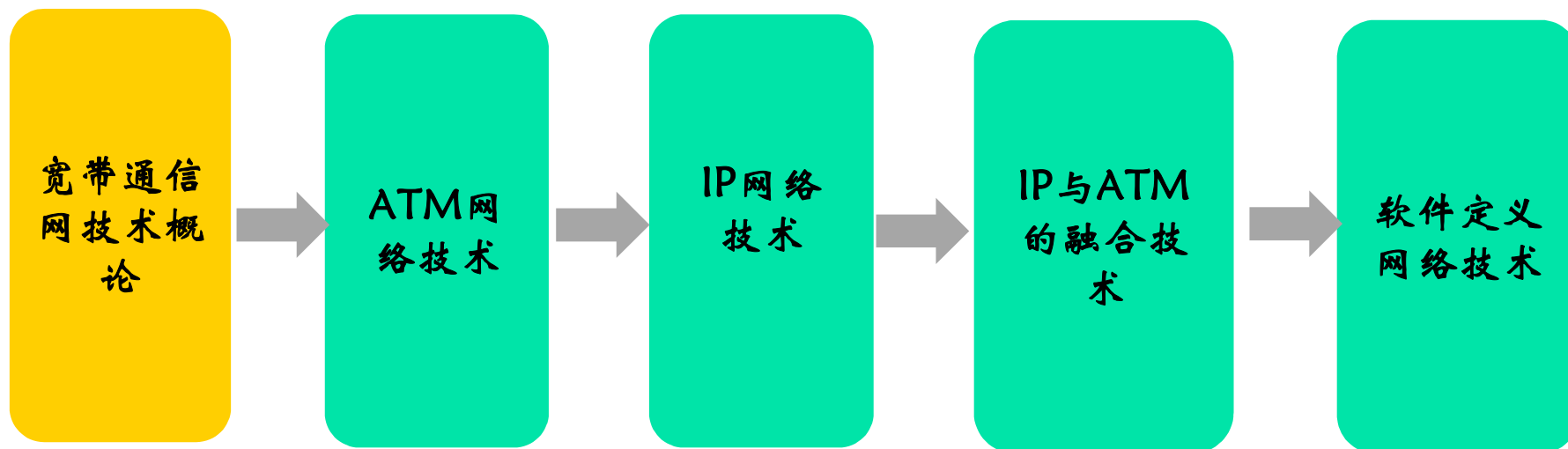
宽带通信网技术的外延

- 从基本功能角度对宽带通信网技术进行划分

- 宽带接入技术
- 宽带交换技术
- 宽带传输技术
- 宽带通信网络的业务
- 宽带通信网络的安全与管理技术
-

宽带通信网课程内容

- 以宽带网络技术中的核心网技术为中心，包括以下模块：

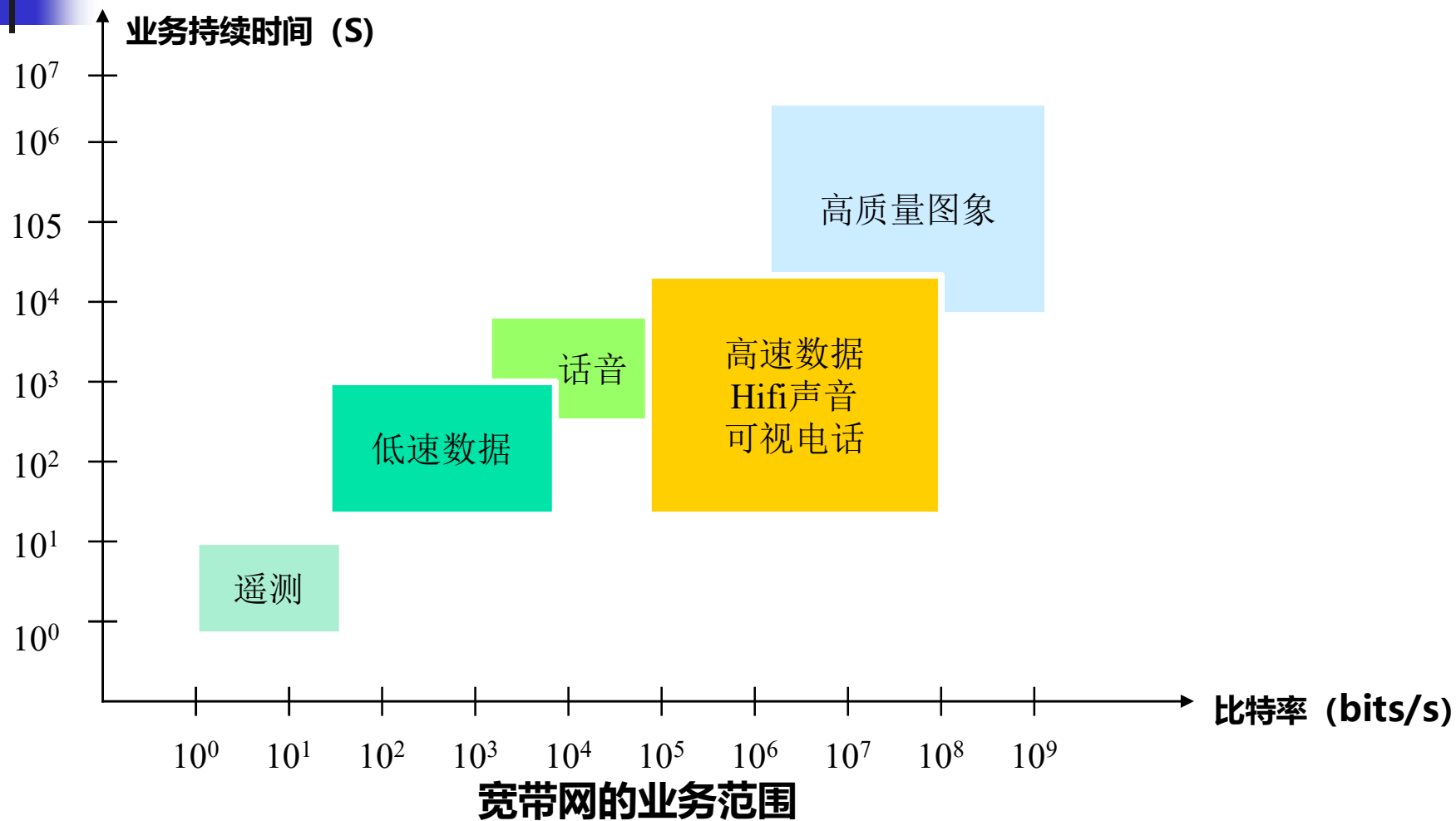




宽带通信网络的演进

- 宽带网络业务特点及其对网络的要求
- 通信网的网络技术-----传递方式的概念
- 通信网的各种传递方式
- 核心网络的演变

宽带网络业务特点及其对网络的要求





宽带网络业务特点及其对网络的要求

- 带宽差异大
- 业务突发性强
- 业务质量要求 (QoS) 不同:
 - 时延
 - 丢失率
- 业务的广播和组播要求

通信网需要一种网络技术, 以便支持包罗万象的业务类型。



通信网的网络技术--传递方式的概念

- 传递方式：是CCITT用来描述通信网使用的一项网络技术，它涉及传输、复用和交换的各方面。



通信网的各种传递方式

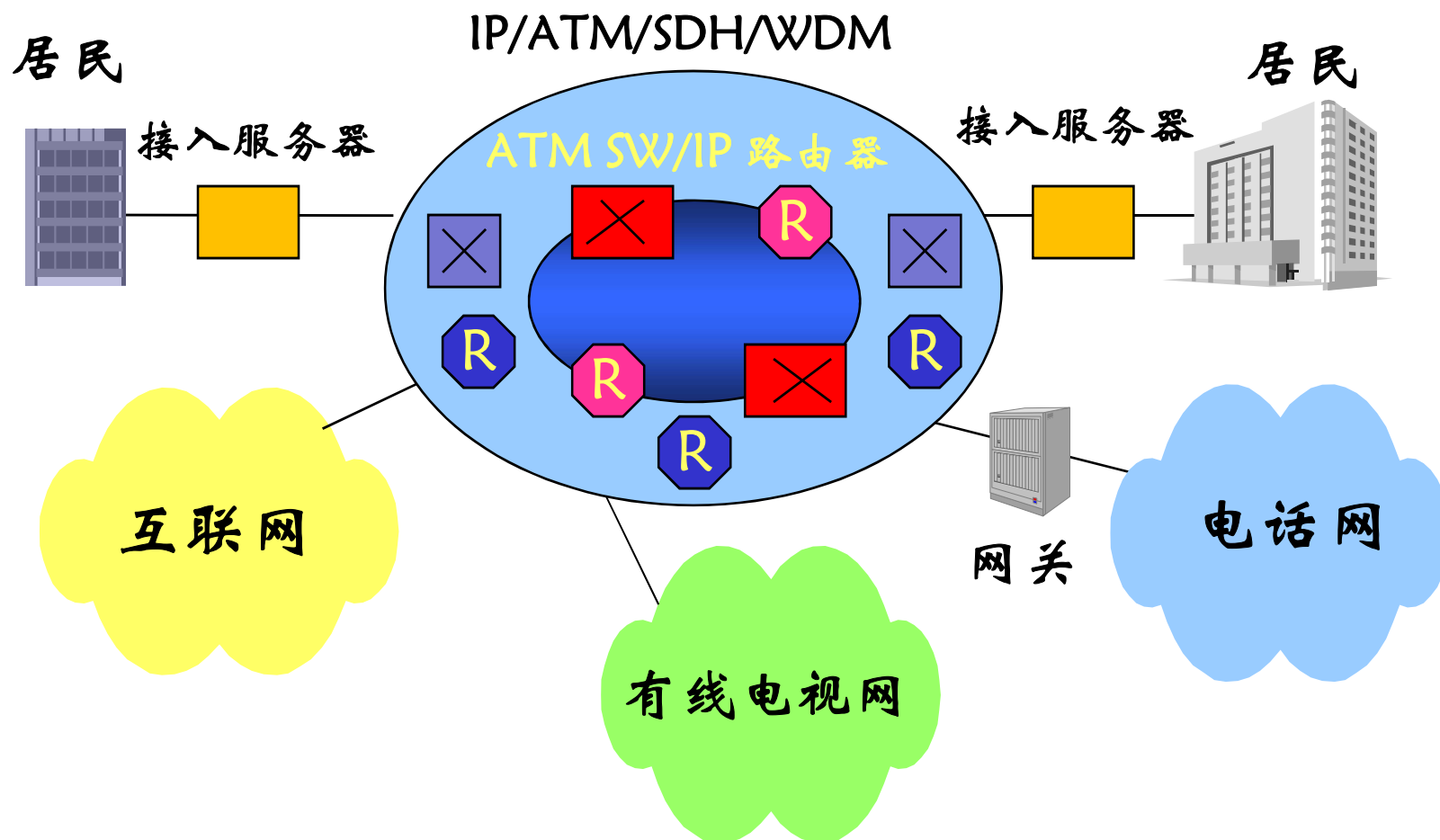
- 电路交换方式
- 多速率电路交换方式
- 快速电路交换方式
- 分组交换方式
- 快速分组交换方式
- ATM方式（异步传递方式）



核心网络的演变

- 基于数据业务的核心网络结构形成
- 电信核心网的演变趋势
- IP与ATM融合技术
- MPLS技术及其发展
- IP与ATM融合技术的主要贡献
- 网络融合

基于数据业务的核心网络结构形成



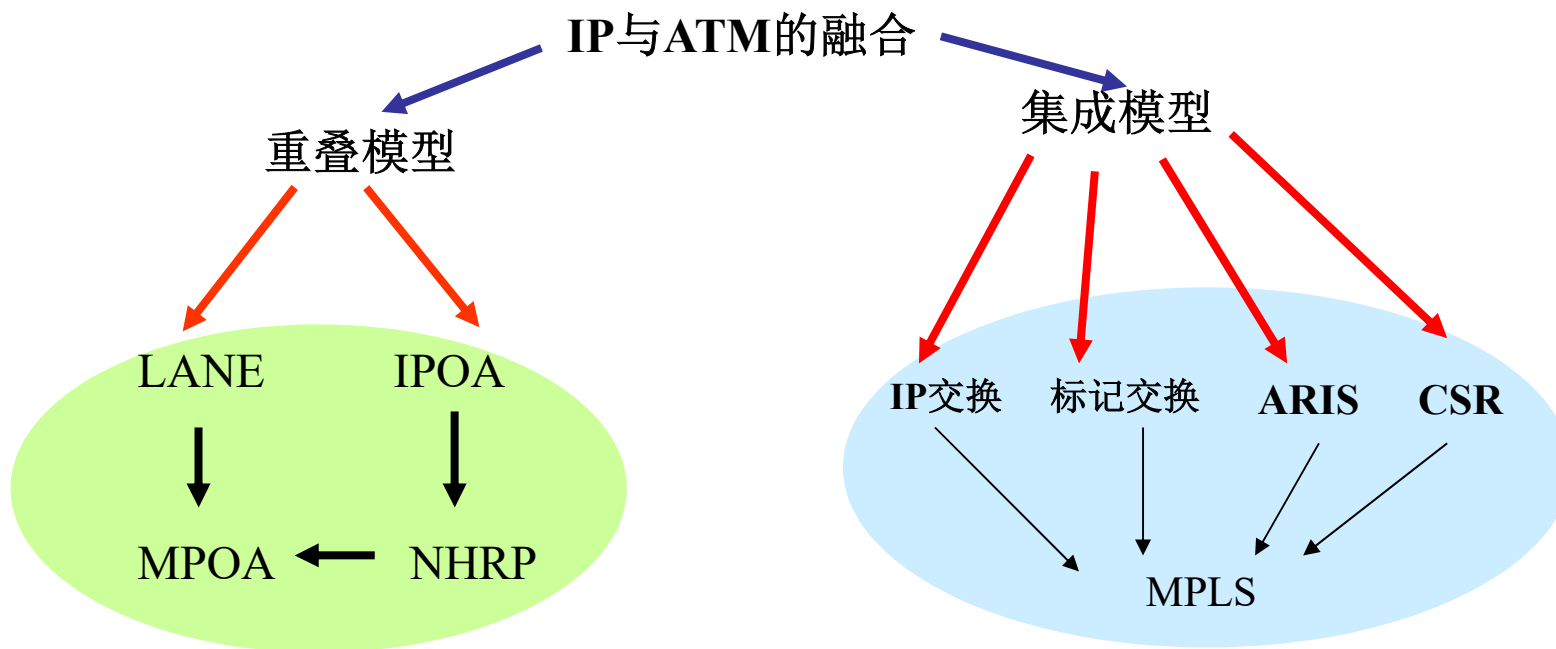


电信核心网的演变趋势

- 实现由ATM---->宽带IP网络的演变
 - ATM提供统一的网络平台（成本、速率、商用化）
 - 支持传统的电信业务网，如DDN、FR、ISDN、PSTN。
 - 支持非IP的数据业务和实时业务
 - 基于ATM网络结构支持IP业务
 - 支持宽带业务的QoS
 - IETF提出IP一统天下
 - IP提供统一的应用平台（Everything over IP）。
 - 所有的应用基于IP
 - IP提供统一的网络承载平台（IP over Everything）。
 - 数据骨干传送网是IP。
 - 利用路由器组建统一的网络平台（即现有的因特网模式）。

IP与ATM融合技术

■ 重叠模型与集成模型



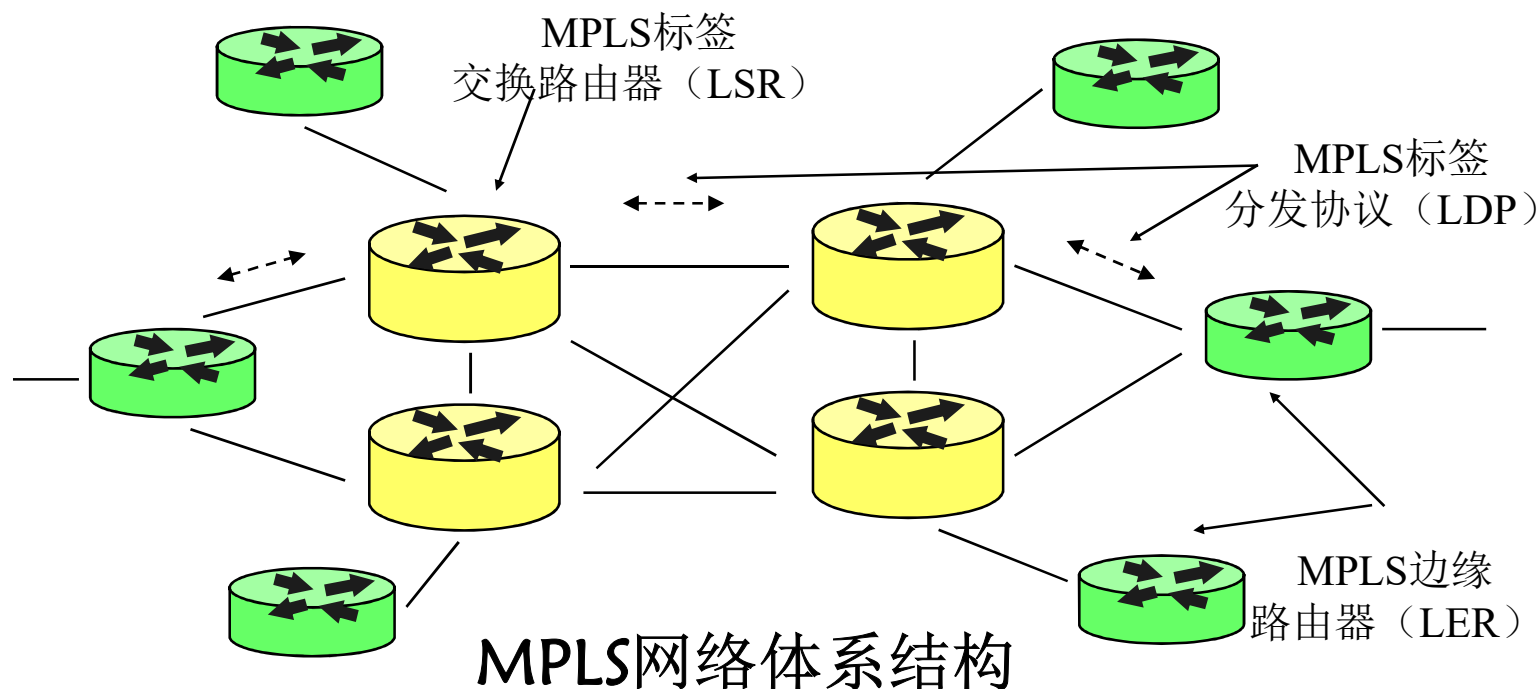


MPLS技术及其发展

- 网络的发展向宽带化、智能化和一体化演进
 - IP网络技术的简单性和业务质量
 - ATM技术的复杂性和对业务的QOS保证
- 各种IP与ATM的融合技术的局限性。
- 实现多种传输承载链路技术的网络互通需要迫切解决（ATM、FR、SDH/SONET、PPP等）。
- VPN提供新一代电信业务的潜力和现有网络技术在实现VPN时的扩展性、安全性和管理性的不足。

MPLS技术及其发展

- MPLS于1997年由IETF提出，是结合第二层和第三层的交换技术，MPLS的实质是将路由器移到网络的边缘，使快速、简单的路由器位于网络中心，一个连接采用实现一次路由和多次交换，从而提高了网络的性能。





MPLS技术及其发展

- MPLS技术的主要贡献
- MPLS技术的发展
 - MPLS向光网络的扩展
 - 多协议波长标签交换 (MPLmS) 技术
 - 通用多协议标签交换 (GMPLS) 技术
 - 传送MPLS (T-MPLS/MPLS-TP) 技术
 - 分组传送网 (PTN)



IP与ATM融合技术的主要贡献

- IP与ATM融合技术在核心网络不断整合了通信网络的组网和数据传输技术，使得核心网络的数据转发和路由技术优化起来，但是上层基于网络提供全业务的解决方案仍然任重道远；
- 各种网络提供的业务趋于一致，导致了几类网络的融合。



网络融合

- 通信网、计算机网和有线电视网形成的三网融合
- 三网融合的演进
- 因特网与电信网的融合
- 下一代网络



网络融合

- 通信网、计算机网和有线电视网形成的三网融合
 - 不同的网络结构往往适于传送不同的业务信号，而不同的业务信号也往往要求不同的网络结构来支持；
 - IP业务及其协议成为电信网的主导通信协议；
 - 在这种大形势下，网络向着对IP业务最佳的分组化网（特别是IP网）演进和融合。



网络融合

■ 三网融合的演进

- 电信级语音网关（统一协议，大容量，多信令）的出现和普及；
- ATM/IP在核心网代替长途交换机，然后在网络边缘逐渐代替新装端局交换机和移动交换机，核心网设备全部IP化……
- 移动核心网也逐渐基于全IP的架构，通过GPRS，3G、4G、5G不断演进……
- 接入网将长期分离、竞争、共存、发展。
- 全业务网络的产生。

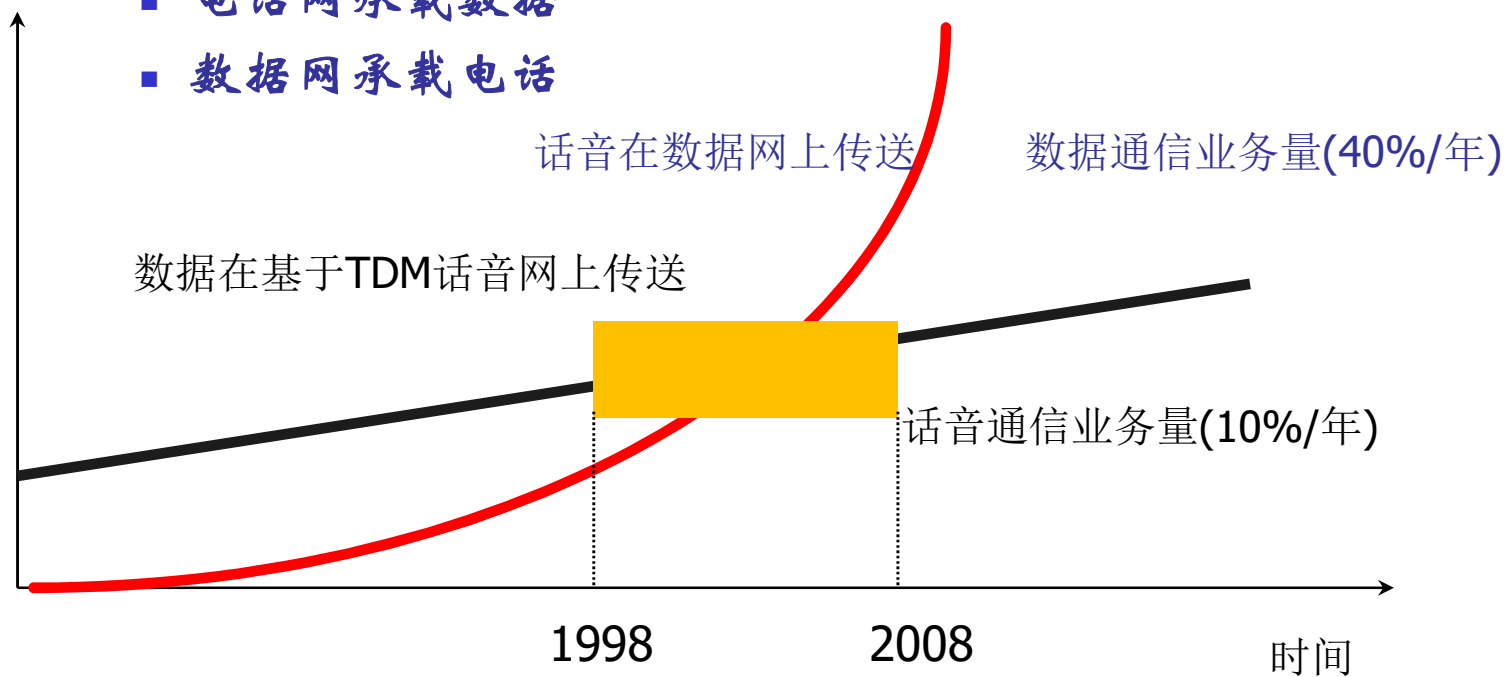
网络融合

■ 因特网与电信网的融合

■ 语音与数据

- 电话网承载数据

- 数据网承载电话





网络融合

■ 因特网与电信网技术的结合

- 单一基础设施
- 分组交换网络
- 智能分布/合作
- 分等级的可靠性、安全性与受控的QOS
- 新型的商业模式
- 高附加值的业务捆绑
- 及时的业务交付
- 智能的终端
- 宽带业务能力与业务综合



下一代网络 (NGN)

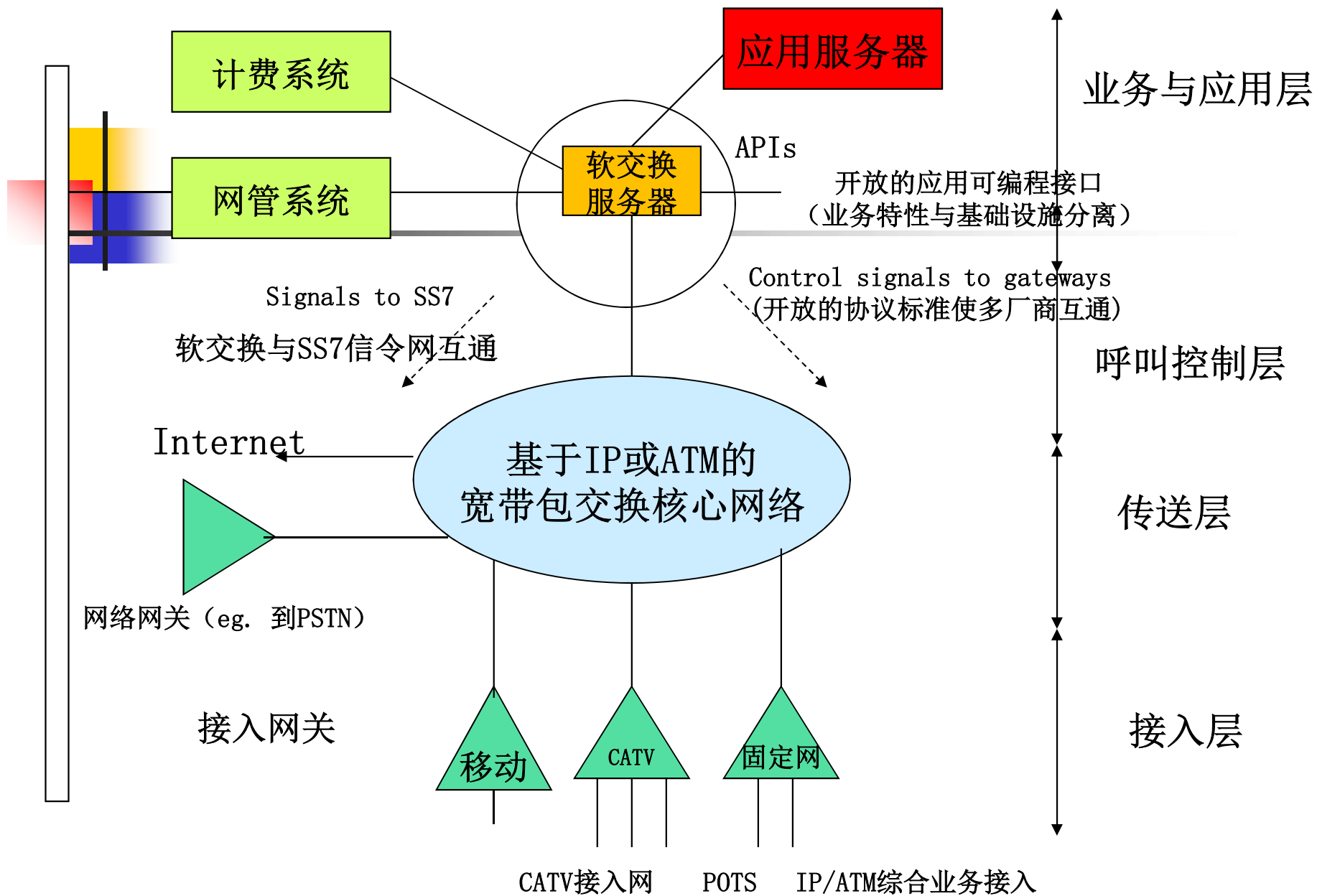
- NGN的概念
- 下一代网络 (NGN) 的体系结构
- NGN的优势
- 基于NGN的移动核心网络架构
- 固网移动融合 (FMC)
- 软件定义网络 (SDN)



NGN的概念

- 欧洲电信标准化组织（ETSI）对NGN给出的定义：

- NGN是一种规范和部署网络的概念，通过使用分层、分面和开放接口的方式，给业务提供者和运营者提供一个平台，借助这一平台逐步演进以生成、部署和管理新的业务。



下一代网 (NGN) 的体系结构



NGN的优势

- NGN是可以提供包括话音、数据和多媒体等各种业务的综合开放的网络架构，其优越性表现在：
 - 采用开放的网络架构体系
 - 交换机分离实现，各自独立发展；
 - 部件之间的协议接口标准化，易于异构网络互通；
 - 业务驱动的网络
 - 业务与控制相分离；
 - 控制与承载相分离；
 - 基于统一协议的分组网络
 - 基于IMS；
 - 固网与移动网络的融合。



固网移动融合 (FMC)

- FMC作为概念最早出现在ITU-T特别工作组 (SSG) 的第七课题组, 并第一次在Q.1761建议中给出了对FMC的明确解释:

FMC是一种机制, 它可以使IMT2000的用户根据用户的签约选项和接入能力, 通过固定网接入继续使用语音业务。SSG最早提出这个概念的想法主要是考虑到当IMT-2000的用户从归属网络漫游走以后, 能否利用非IMT2000的网络, 甚至是固定网络, 继续保证这些用户在不更换用户标志的前提下, 继续使用基本的服务。



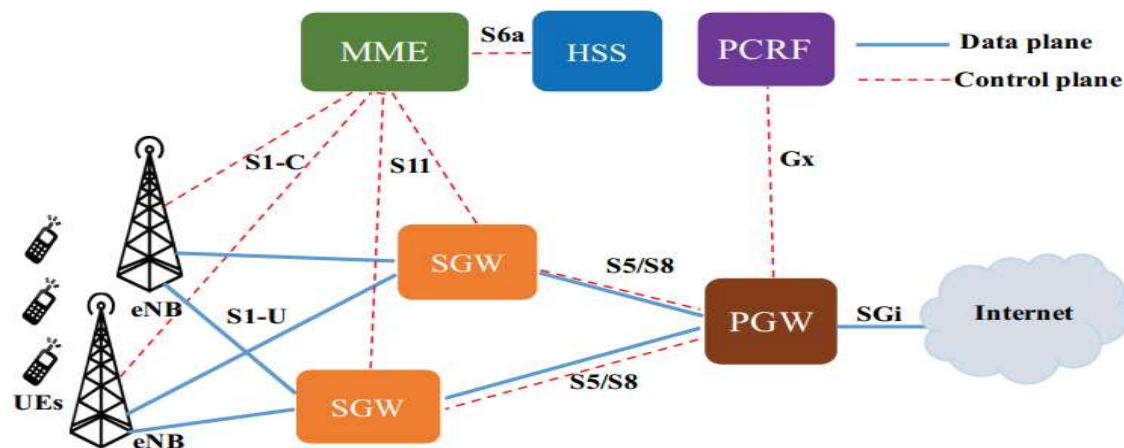
固网移动融合 (FMC)

- ITU-TSG13和SG19的联合研究中，在Rec.FMC.Req 建议草案中提出了相对完善的FMC定义：
 - FMC是在一个给定的网络环境中运营者可以使用的一种机制，可以向终端用户提供业务和应用，而不用考虑其接入技术和位置，体现了向终端用户提供业务时不受接入技术限制的需求。

基于NGN的移动核心网络架构

■ 当前的移动核心网络EPS(Evolved Packet System) 及其五个主要功能实体

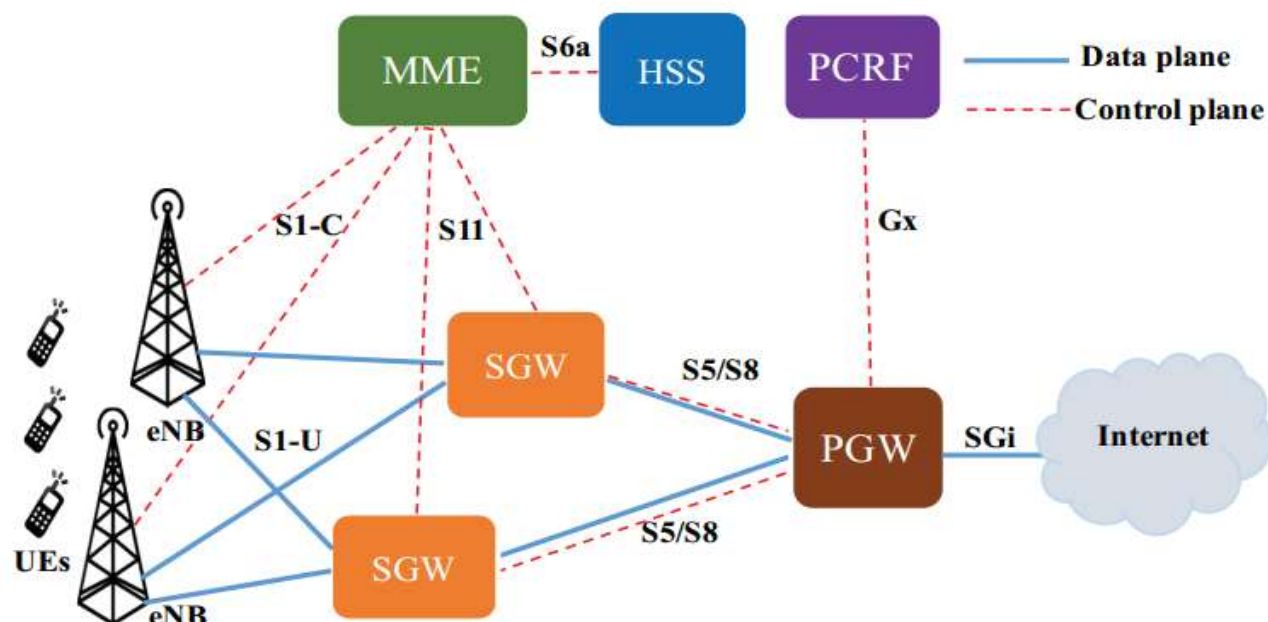
- MME: 负责移动性管理, 寻呼, 以及管理承载建立, 用户信息等;
- SGW: 负责在eNodeB和PGW之间转发和路由用户数据包, 并且用于eNodeB间切换的本地移动锚点;
- PGW: 转发移动网络到外部网络间的数据包, 其主要功能包括设备IP地址分配、策略执行、包过滤和计费等;
- HSS: 归属签约用户服务器, 用于存储用户签约信息;
- PCRF: 策略与计费规则功能单元, 向PGW提供QoS配置文件和计费规则。



基于NGN的移动核心网络架构

■ 现有的EPS存在的问题

- 由于基于硬件实施，因此部署和扩展的成本很高；
- 控制平面和数据平面之间未完全分离，因此，可扩展性和灵活性较差；
- 需要手动和静态网络配置，会导致资源供给和分配不足；
- 基于分层结构设计，因此，数据包转发和路由不是最优。





软件定义网络 (SDN)

作为近年来兴起的一个网络技术研究热点，SDN的概念最早起源于对分组网络提供用户可以控制的网络节点转发管理功能。

- 1996年Ipsilon尝试提出的通用交换管理协议 (GSMP)；1998年提出的一个安全、保证资源的可编程网络Tempest；IETF于2000年提出的转发和控制单元分离；2004年提出的路径计算单元；
- 2007年提出和实现的Ethane提供了一个安全管理架构，结合了简单的基于流的交换机和一个中心控制器协助管理和路由数据流；2008年，美国斯坦福大学clean slate研究组提出的OpenFlow，能使得交换机/路由器流表中的各项被外挂的服务器定义，从而灵活地控制了数据流的转发和路由规则。
- 目前，SDN已经不再局限于各类软件实现平台，而是逐渐形成了一种通用的平台和概念。



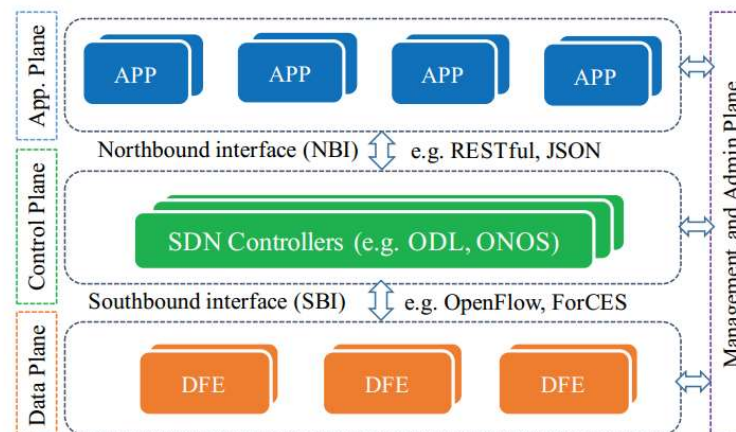
软件定义网络 (SDN)

- SDN 的概念

- SDN (Software Defined Network, 软件定义网络) 是一种新兴的控制与转发分离并直接可编程的网络架构。
- 传统网络设备紧耦合的网络架构被分拆成应用、控制、转发三层分离的架构，其中，控制功能被转移到了服务器，上层应用、底层转发设施被抽象成多个逻辑实体。

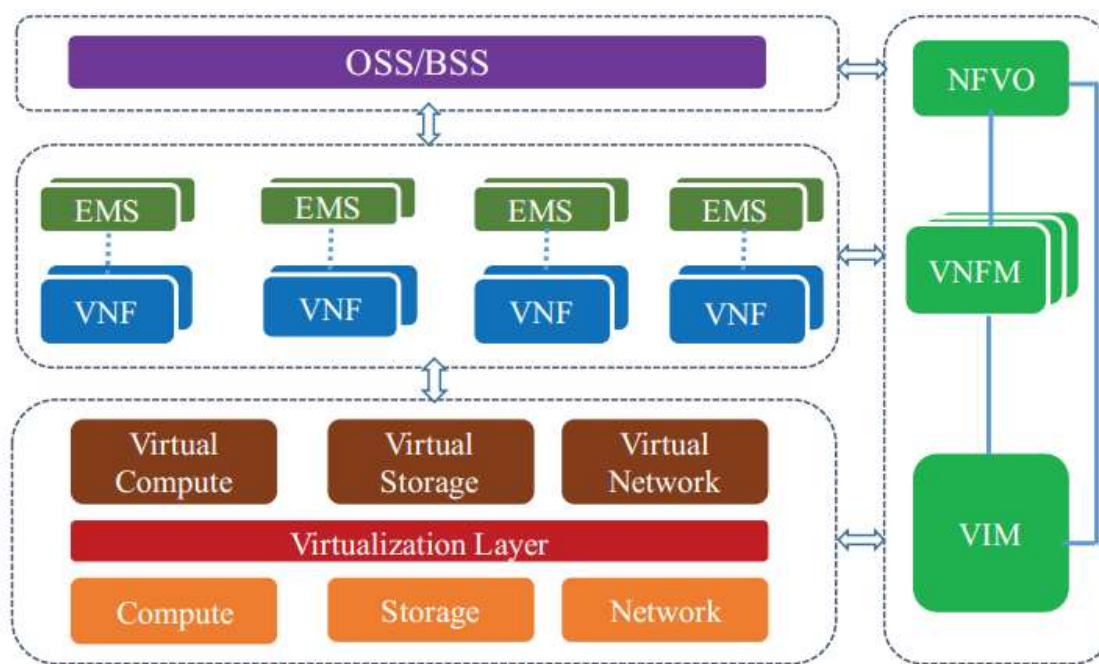
基于软件定义的组网 (Software Defined Networking)

- SDN本质上是一种集中式网络范型，其中控制平面在逻辑上集中在一个或一组控制实体（即SDN控制器）上，通过SDN控制器请求的应用和网络服务抽象为数据转发平面。
- 由应用程序（如路由和负载均衡）组成的应用平面通过北向接口与控制平面中的SDN控制器进行通信（例如，REST和JSON）
- 控制平面逻辑上保持全局和动态网络视图，提供控制任务，以根据来自应用的请求，通过南向接口（例如，OpenFlow，ForCES）来管理数据平面中的网络设备。控制器之间使用东西向接口相互通信。
- 数据平面由虚拟/物理交换机和路由器等数据转发单元（DFE）组成，根据SDN控制器安装的规则对数据包进行转发和路由。
- 管理平面负责与网络的配置和监控相关的所有活动。



网络功能虚拟化 (NFV)

- 网络功能虚拟化 (NFV) 本质上是将网络功能 (NF) 从专用硬件迁移到在云环境或通用服务器上运行的软件中。
- 网络功能被虚拟化成不同的虚拟网络功能 (VNF)，这些 VNF 在通用的基础设施 (NFVI) 上部署和执行。
- ETSI 标准化的 MANO 架构针对基于网络功能虚拟化的架构进行管理。





ETSI的MANO架构

- NFVI由虚拟资源组成，虚拟资源通过虚拟化层，对底层的物理硬件资源（计算、存储和网络资源）进行抽象和映射。
- NFV管理编排器（NFV MANO）负责协调和管理虚拟网络功能和虚拟网络基础设施，NFV MANO与外部运营和业务支撑系统（OSS/BSS）交互，完成来自OSS/BSS的业务请求。
 - NFV编排器（NFVO）负责管理网络服务（NS），包括网络服务的生命周期管理及其更新；
 - 虚拟网络功能管理器（VNFM）管理虚拟网络功能；
 - 虚拟化基础设施管理器（VIM）管理虚拟化的基础设施资源到物理资源的映射。

基于SDN与NFV的未来核心网络架构

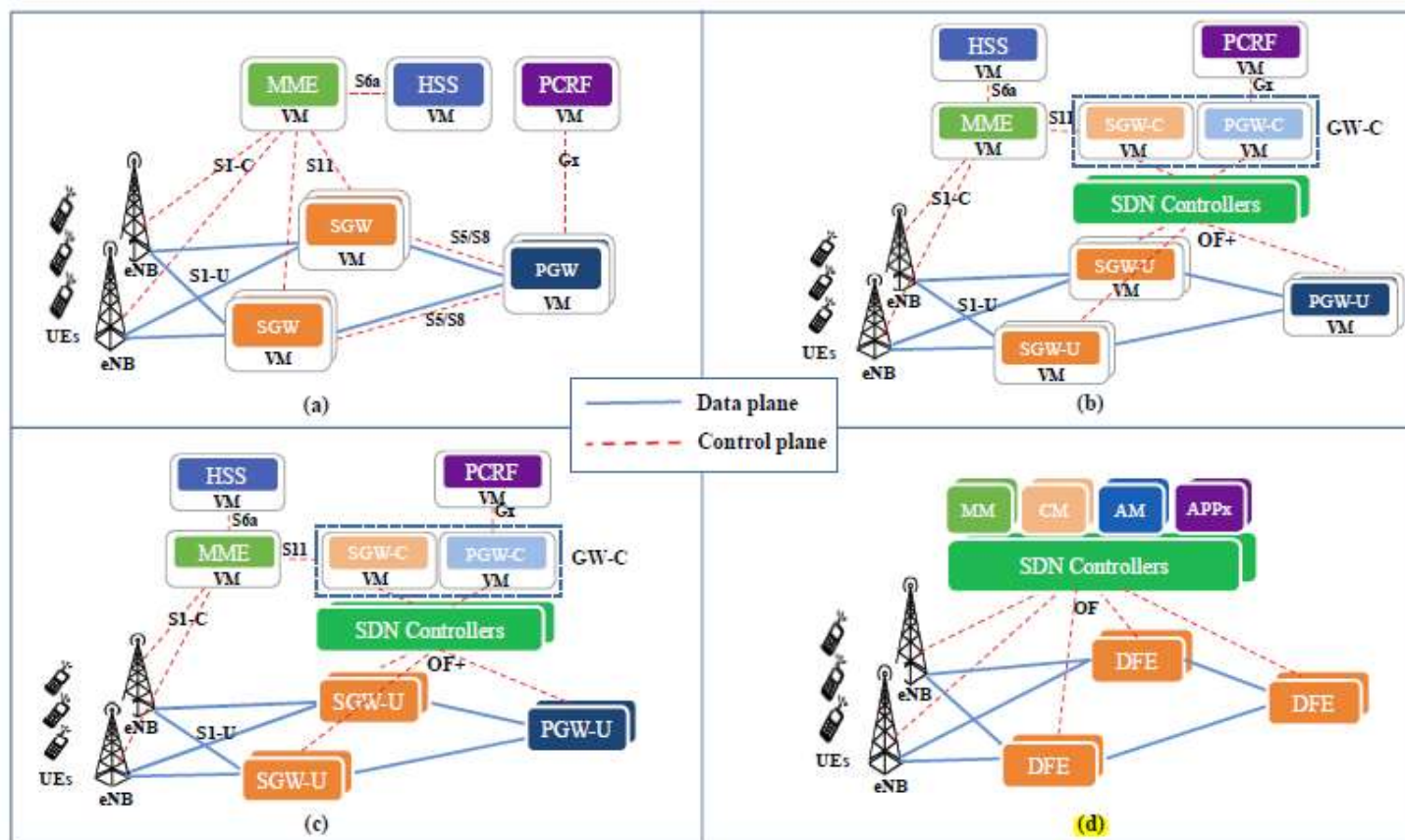


Fig. 6. Typical ways of re-architecting the MPC architecture with SDN and NFV: a) Fully NFV-based EPC architecture, b) SDN/NFV-based EPC architecture with virtualized data/user plane, c) SDN/NFV-based EPC architecture with non-virtualized data/user plane, d) Fully SDN-based MPC architecture



基于SDN与NFV的未来核心网络架构

- 演进性方案：演进性方案中，整个实体或一些现有MPC架构，标准接口和协议的一部分仍然存在或保持现有的核心网架构。
- 革命性方案：是指所有常规MPC实体以及它们之间使用的标准接口和协议的完全被替换掉的方案。

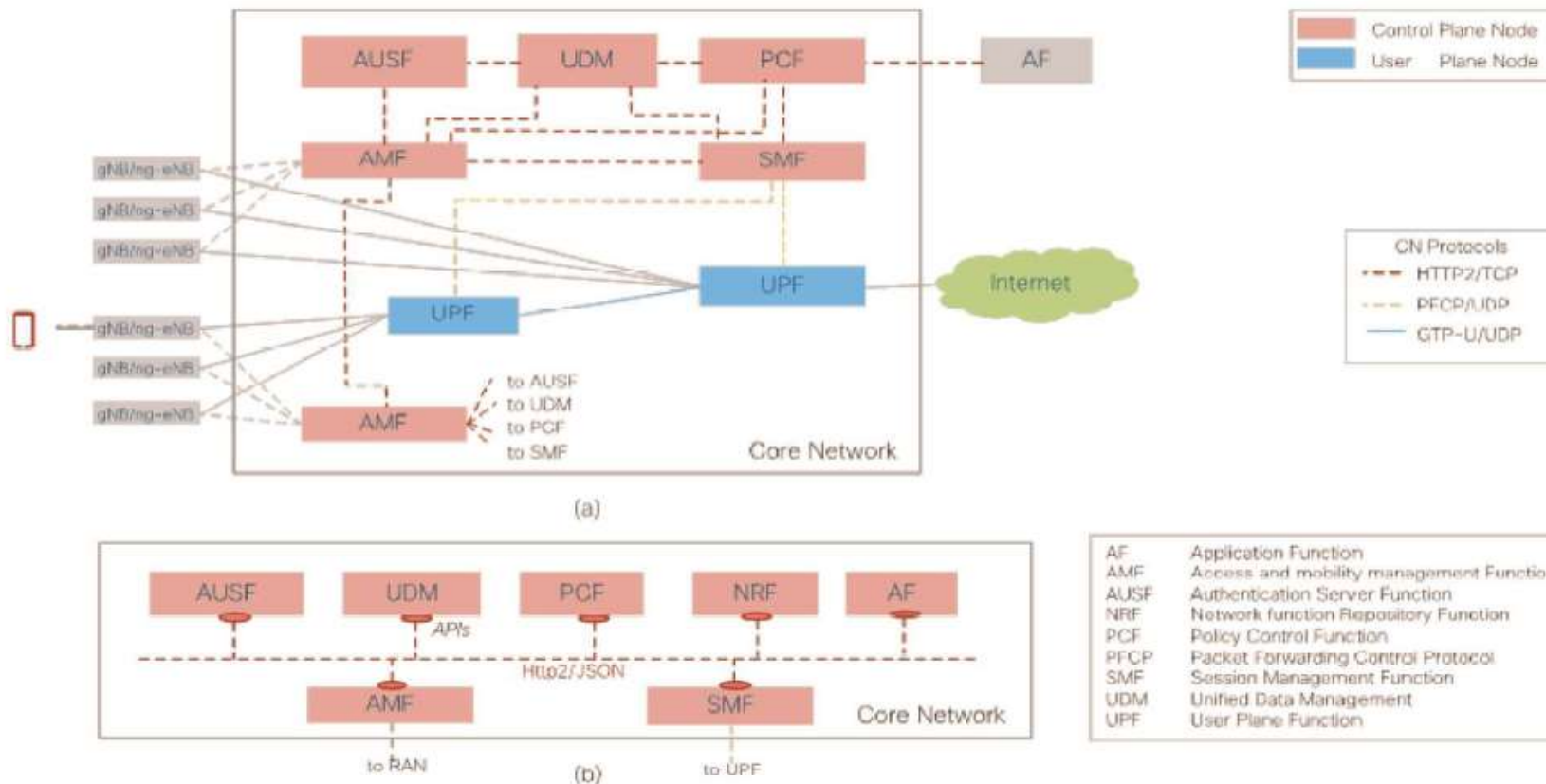
基于SDN与NFV的5G核心网络架构

5G应用场景需求



网络	流量密度	连接数密度	时延	移动性	能效	用户体验速率	频谱效率	峰值速率
4G	0.1Mbps/m ²	10万/km ²	空口10ms	350Km/h	1倍	10Mbps	1倍	1Gbps
5G	10Mbps/m ²	100万/km ²	空口1ms	500Km/h	100倍提升	100M-1Gbps	3-5倍提升	20Gbps

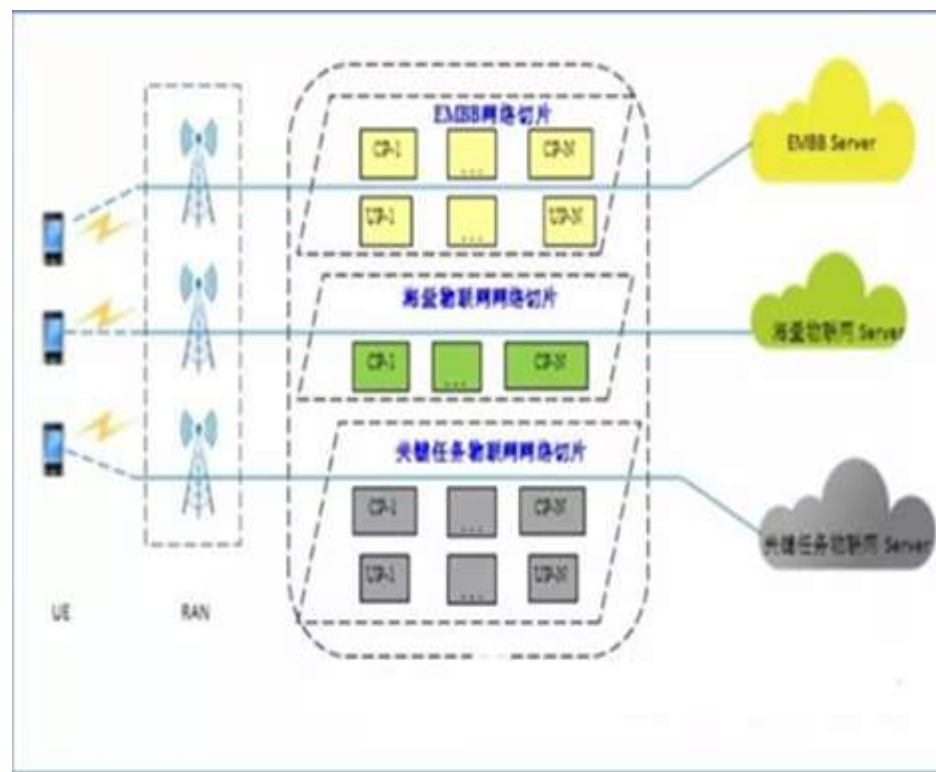
5G的核心网架构



5G core network. (a) Interface representation, and (b) API level representation.

5G核心网架构

■ 5G的网络切片技术





宽带通信网络的未来

- 宽带化
- 智能化 (SON)
- 泛在化 (异构)
- 融合化 (异构、云)
- 业务多样化、综合化 (人-人、人-机、机-机)
- 可编程化
-



宽带通信网络技术的重要机理

- 通信网络中不同交换技术的机理
- 通信系统和网络的描述方法
- 宽带通信网网络性能的主要评价方法

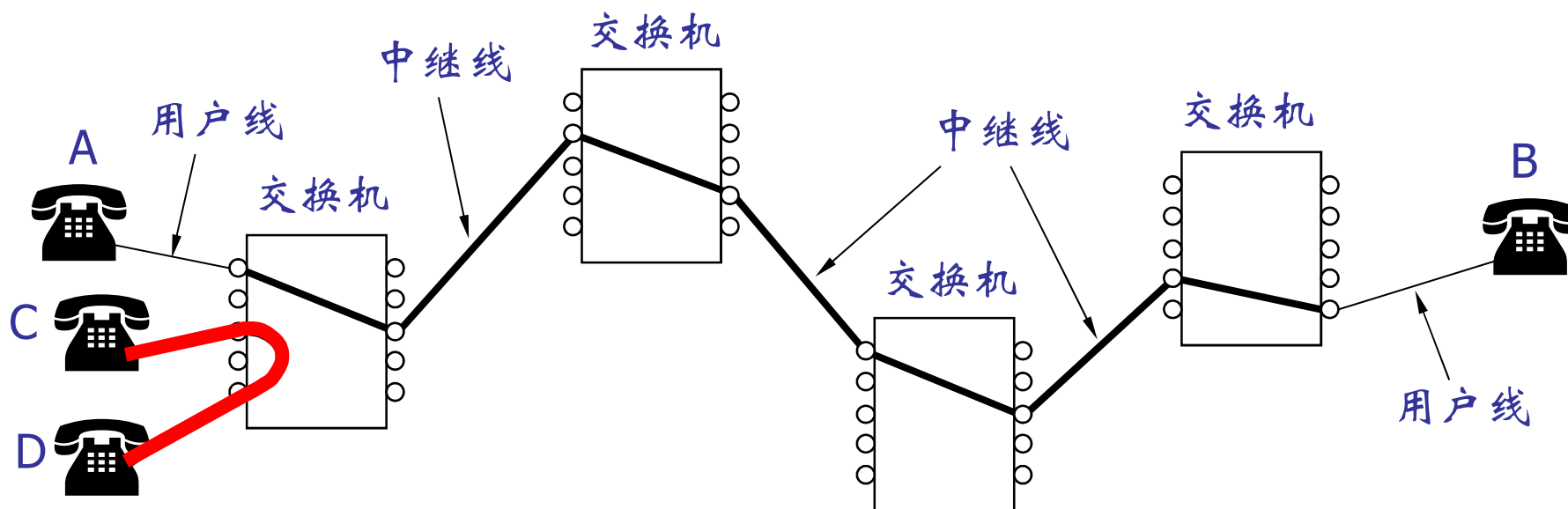


通信网络中不同交换技术的机理

- 电路交换技术
- 包交换技术

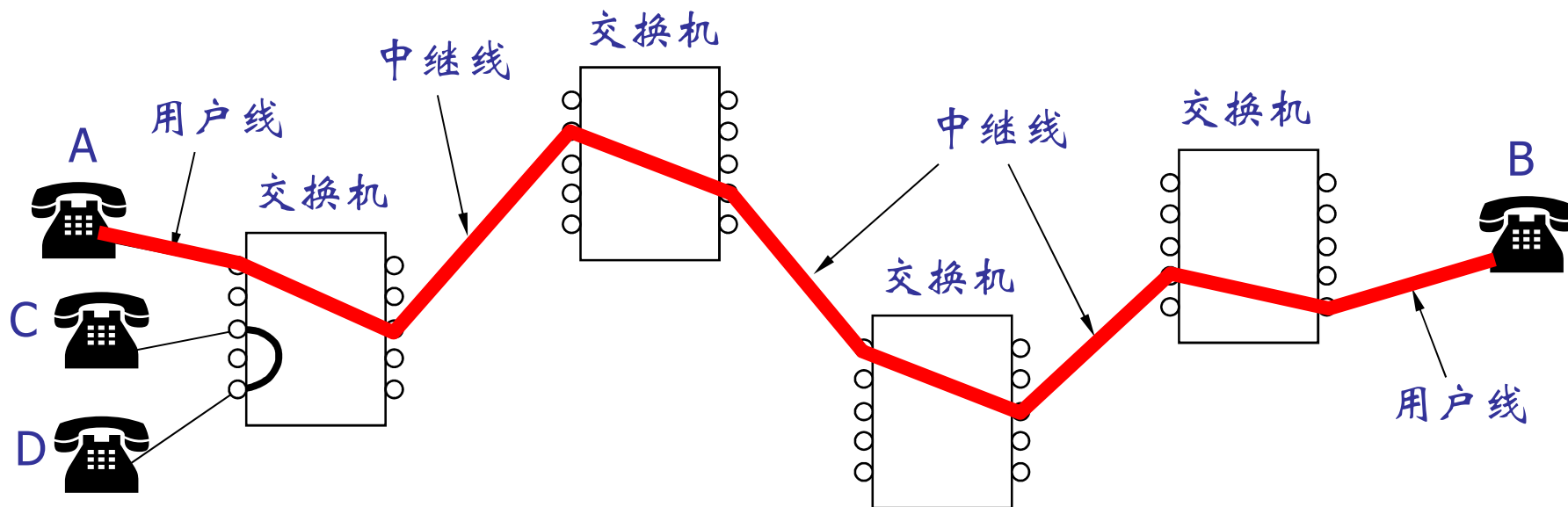
电路交换技术

- C 和 D 通话只经过一个本地交换机
- 通话在 C 到 D 的连接上进行



电路交换技术

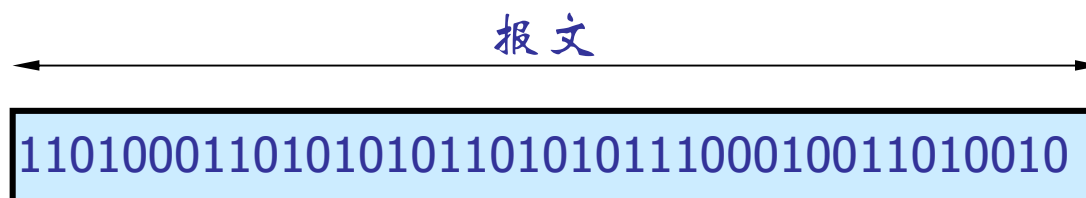
- A 和 B 通话经过四个交换机
- 通话在 A 到 B 的连接上进行



包交换技术

■ 分组的形式

- 在发送端，先把较长的报文划分成较短的、固定/可变长度的数据段。

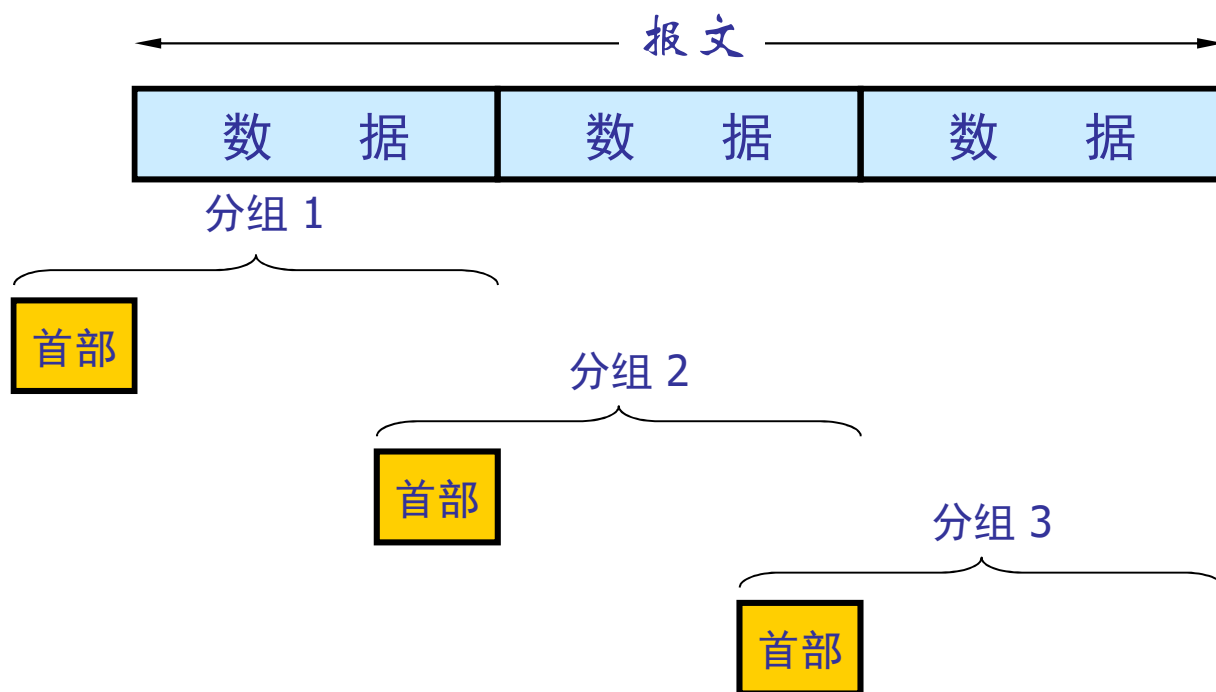


↑
当这个报文较长
不便于传输时

包交换技术

■ 分组的形成

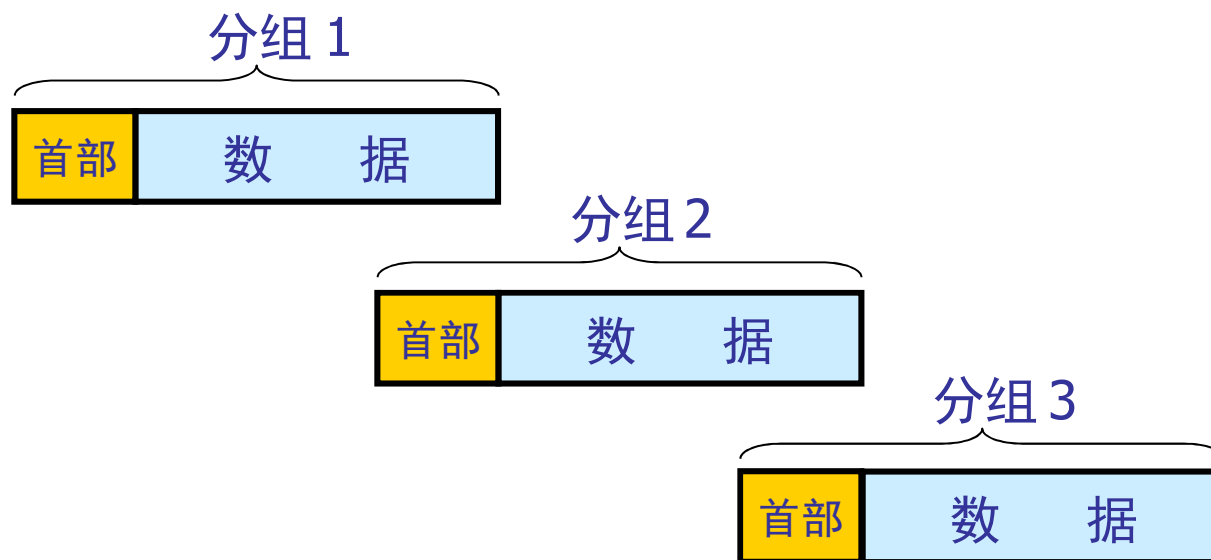
- 每一个数据段前面添加上首部构成分组。



包交换技术

■ 交换期间的传输单元

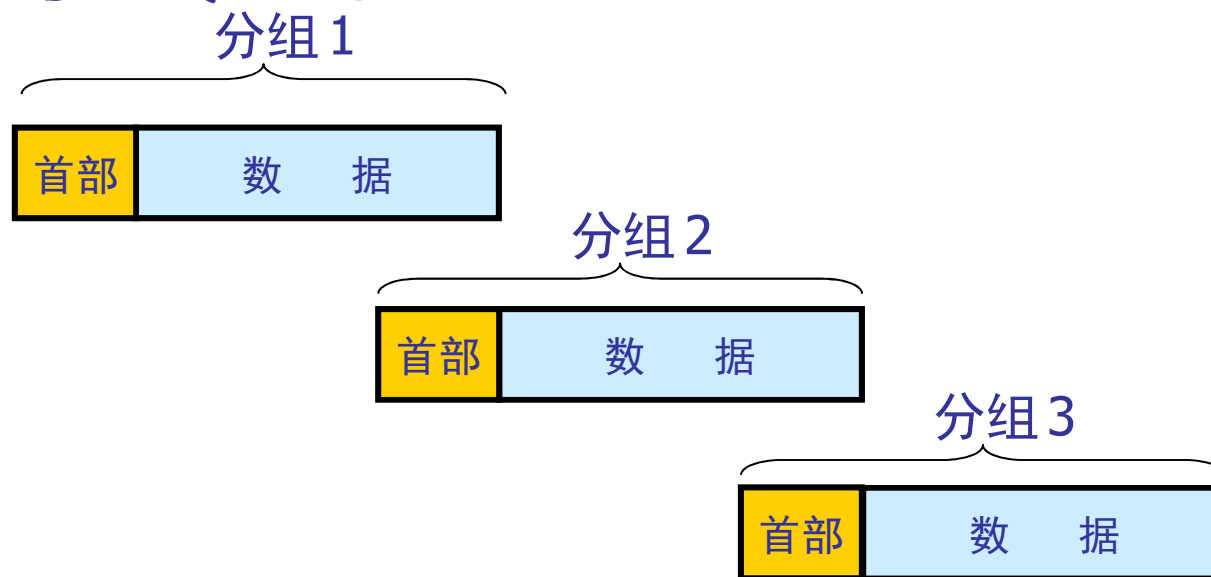
- 分组交换网以“**分组**”作为数据传输单元。
- **依次**把各分组发送到接收端（假定接收端在左边）。
- 分组交换网中的节点交换机根据收到的分组首部中的**地址信息**，存储，再把分组**转发**到下一个节点交换机。
- **各节点通过存储转发方式**，最后将分组送达**最终目的地**。



包交换技术

■ 接收端收到分组后的处理

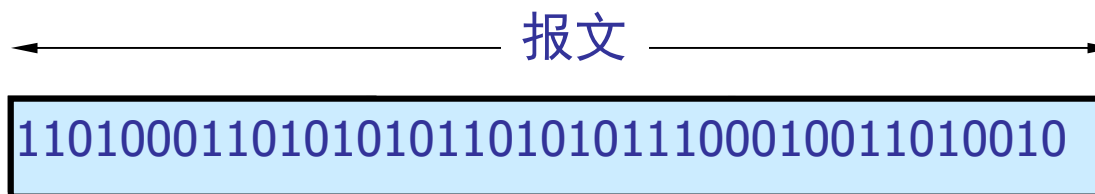
- 接收端收到分组后去掉其封装的首部，将数据还原成报文。



包交换技术

- 还原成原来的报文

- 最后，在接收端把收到的数据恢复成为原来的报文。



- 这里我们假定分组在传输过程中没有出现差错，在转发时也没有被丢弃。



通信网络中不同交换技术的机理

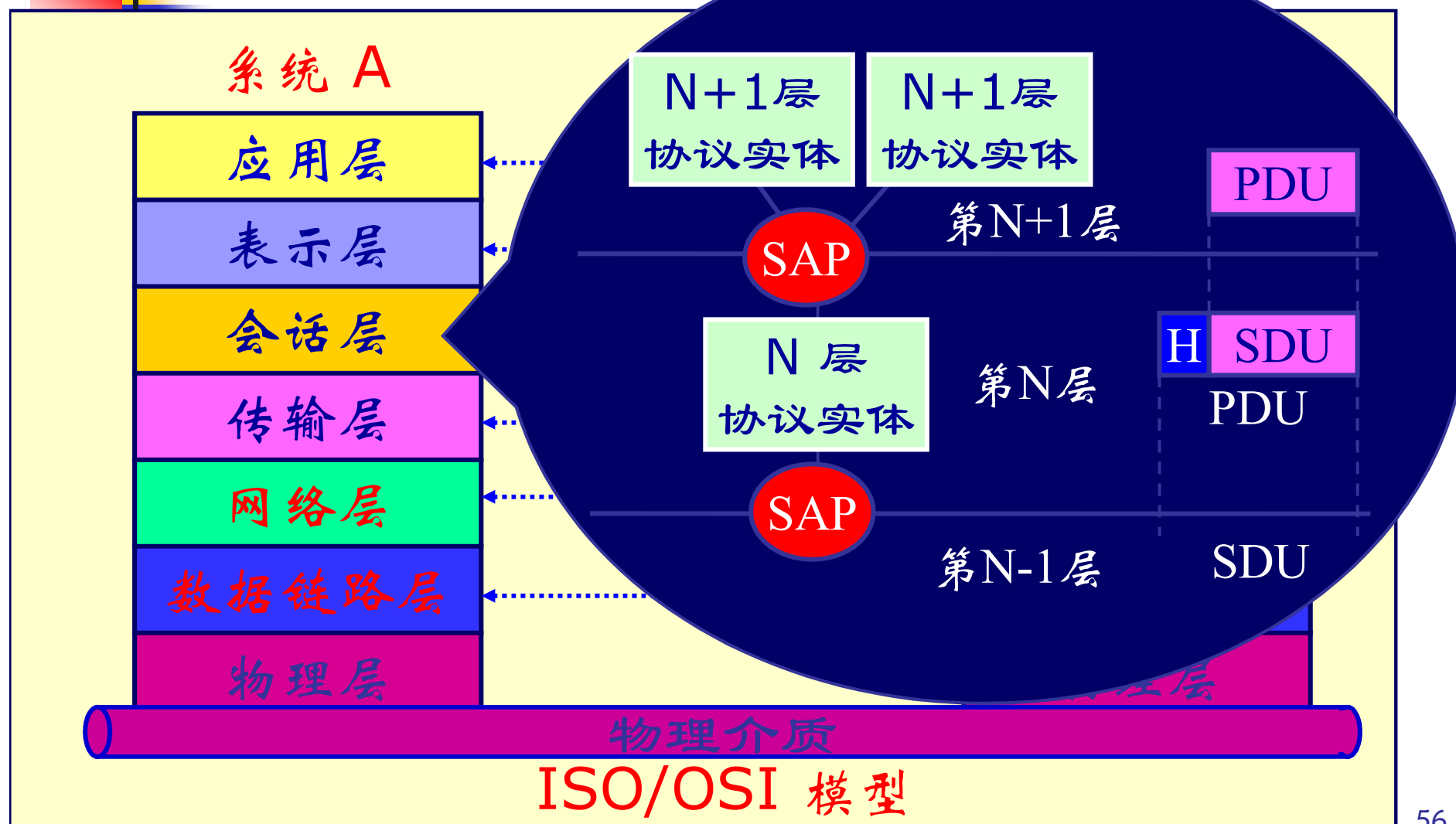
- 分组首部字段设置与网络功能
 - 每一个分组的首部都含有地址、控制等字段，每个字段根据实现的功能进行编码设计；
 - 分组报头的字段设置反映了该种网络技术支持的网络功能；
 - 分组包头、净荷与分组的额外开销；
- 网络中业务数据与控制数据相比，控制数据的开销比例；
- 面向连接与无连接（业务、网络技术）



通信系统和网络的描述方法

- 开放系统互联互通的方式----OSI参考模型
- 通信系统和网络的形式化描述方法

开放系统互联互通的方式---OSI参考模型





开放系统互联互通的方式---OSI参考模型

- 开放系统互联互通的方式---OSI参考模型
 - 开放系统互连参考模型采用分层模型，将开放系统的功能抽象到了7个层次，定义各层次的功能和各层次之间的接口。
 - OSI模型规定了在同一系统中和不同系统中的通信规则。



通信系统和网络的形式化描述方法

- 规范描述语言
- 对宽带通信系统和网络的描述



规范描述语言 (SDL)

- SDL语言的概念
- SDL的发展
- SDL的特点
- SDL方法适用的系统
- SDL的应用
- SDL简介



SDL语言的概念

- SDL (Specification and Description Language) 语言是ITU-T用来描述系统模型的技术规范。
- SDL着重从全局视角来描述系统结构，表现系统是由哪些子系统构成的，这些子系统之间是如何相互作用的，并且能递归式地描述各子系统的功能和结构。
- SDL语言可以用来完成对系统的分析。



SDL的发展

- 1972年，国际电报电话咨询委员会（CCITT，现ITU-T）内的一个研究小组（分别来自贝尔、爱立信、Motorola等公司或组织）就开始研究协议的规范及其描述语言SDL(Specification and Description Language)；
- 1976年，提出了SDL的初步定义文本，建议书Z.101~Z.103（或称为Z.100系列建议书），开发了基本的图形描述语言；
- 1980年，修订为Z.101~Z.104；
- 1984年，又进行了扩充，加入了结构和数据，并使SDL的定义更加严格；为了便于协议的具体实现和验证，CCITT的第Ⅶ研究组又提出了面向PASCAL的SDL，X.250建议书。



SDL的发展

- 1988、1992年又进行了修订，在1992的修订版中，加入了面向对象的特征；
- 1996年，在1992年的版本基础上降低了许多规则的要求，使得SDL语言更容易理解和更灵活地使用，形成了1996版的SDL。该版本大大扩充了SDL的表示能力并简化了接口，由于改动较小，最终只是作为附录加入1992年的版本。
- 版本SDL-2000是1999年更新的，其中对面向对象的特征进行了更多的扩展，引入了其它语言，如ASN.1、ITU-T ODL、UML等，以便可以更好地支持对象建模和代码生成。
- 2010年，又推出了SDL-2010版本。
- 为了区分不同年份制定的SDL标准，通常用“SDL-yy”的形式来表示在yy年制定的SDL标准。例如，SDL-92表示1992年发表的SDL标准。



SDL的特点

- SDL语言是一种主要针对电信系统的需要，基于面向对象的形式描述语言，适于描述复杂的实时应用。
- SDL针对系统的结构、行为和数据有较强的描述能力，例如电信系统中的呼叫处理、维护和故障处理、系统控制、数据通信协议等。
- SDL是一种基于扩展有限状态机和抽象数据类型的混合技术。同其它的形式描述语言一样，SDL既可描述比较粗的系统的描述，亦可描述系统的详细设计。



SDL 的特点

- 可用于需求分析到具体实现的整个开发过程；
- 适应于实时系统；
- 基于扩展的有限状态机；
- 图形化表示方式，可视性强；
- 具有面向对象的特征。



SDL方法适用的系统

■ SDL方法适用的系统

- 主要与其他系统通信（嵌入式系统）；
- 对精确准时行为的需求不高；
- 与其他系统主要是基于数字通信，而不是采用模拟信号进行通信；
- 无须执行复杂计算；
- 需要并发运行，异步通信（并行系统）；
- 极少需要复杂数据关系模型。



SDL 的应用

- SDL 主要应用于电信领域，适用于描述 **活性离散系统**；
- 既可以用来说明（specification）一个系统应具有的行为（behaviour）；又可以用来描述（description）一个系统实际具有的行为（behaviour）。
- SDL 可用来制作一些 **文档**。



SDL 的应用

- 一般不将它作为实现语言来使用，可以基于SDL描述的规范将其自动转换成某种编程语言；
- 可以将SDL同其它的语言结合起来使用，例如，与 MSC(Message Sequence Chart, Z.120) ，ASN.1(Abstract Syntax Notation One, Z.105), TTCN(Tree and Tabular Combined Notation, ISO/IEC 9646-3)等。



SDL简介

- SDL语言的表示方法
- SDL的三种语法形式
- SDL的静态结构
- 基于SDL对系统的描述规范
- SDL对系统动态行为的描述
- SDL中的数据
- 基于SDL对协议的描述假设
- 基于SDL的系统协议结构



SDL语言的表示方法

■ SDL语言的表示方法

- 纯文本表示(SDL PR): 纯文本表示更容易被计算机处理;
- 图形表示(SDL GR): 图形表示更直观, 利于进行可视化建模;

图形表示也往往结合一定的文本表示, 图形表示在进行SDL模型检查前, 往往先转换为纯文本表示。



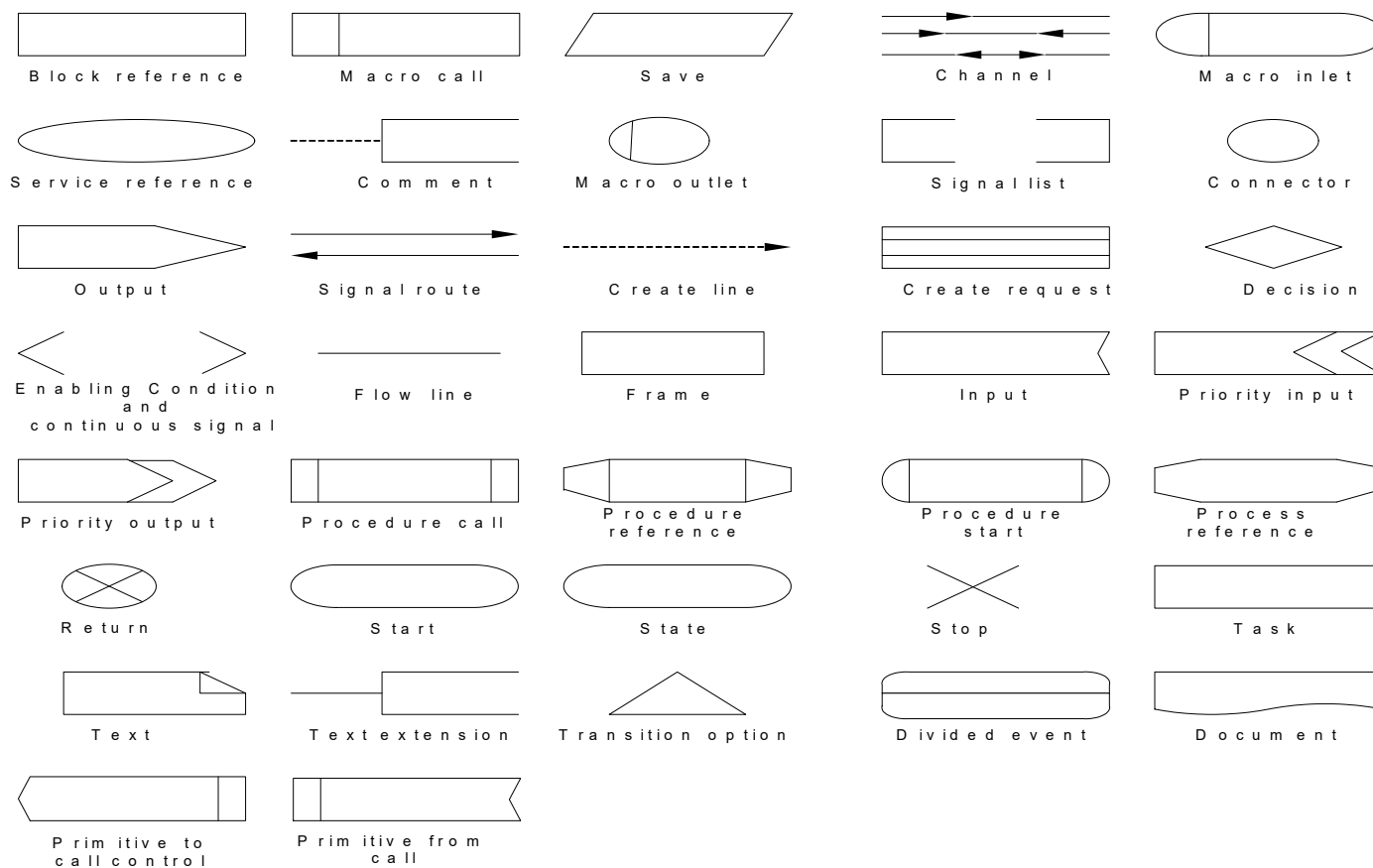
SDL的三种语法形式

■ SDL的三种语法形式

- 图形表示的SDL/GR(Graphical Representation): 比较适合于描述系统的结构和控制流;
 - 用文字短语表示的SDL/PR(Textual Phrase Representation): 比较适合于机器处理;
 - 程序语言形式的SDL, X.250。
- 为了保证等价性, 以便相互转换, SDL的语义严格地和三种具体语法定义分离来。
- SDL的语义是利用公共语言模型的抽象语法来定义的;
 - SDL提供了(但不是建议书的内容)公共语言模型的数字定义。

SDL的三种语法形式

■ SDL/PR使用的主要符号





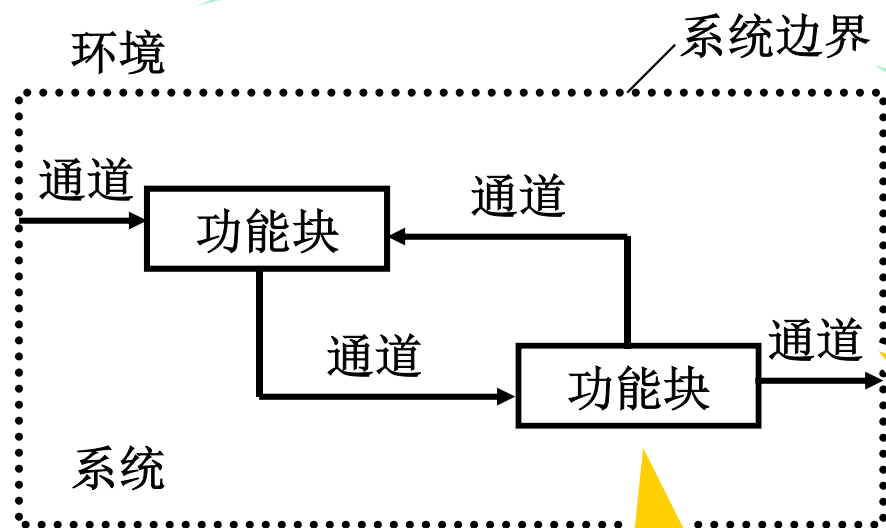
SDL的静态结构

- 在SDL中，把整个开放系统说明为由许多互连的**抽象机**（扩展的有限状态机FSM）所组成。
- 每个抽象机的动态行为用它与其他机器或环境的交互和对交互数据的操作来描述。
- 一个SDL的静态结构采用**系统**(System)、**功能块**(Block)**进程**(Process)、**过程**(Procedure)和**通道**(Channel)来描述：
 - 在SDL中，将系统、功能块、进程等称为**代理**(Agent)；
 - 各个代理用块图(Block Diagram)来表示，一个代理可以对应多个块图；
 - 各个块图以边框(Frame)为分界，主要包括四个部分，即**块图的类型和标记**、**页号**、**文本记号**以及**块图的主要内容**。

SDL的静态结构

将其它所有不属于系统的内容称为环境(ENV)，从环境中可以观察和影响系统的行为。一般假定环境的行为模式与SDL规定的行为模式相似，并且遵守系统规范中强加的限制。

- 根据系统与其环境之间有无交互，可将系统分为开放的(open)系统和封闭的(closed)系统。
- 如果系统是开放的，则系统与环境之间有交互：环境给系统发送信号，并从系统接收信号形式的响应；
- 如果系统是封闭的，则系统与环境之间没有交互。



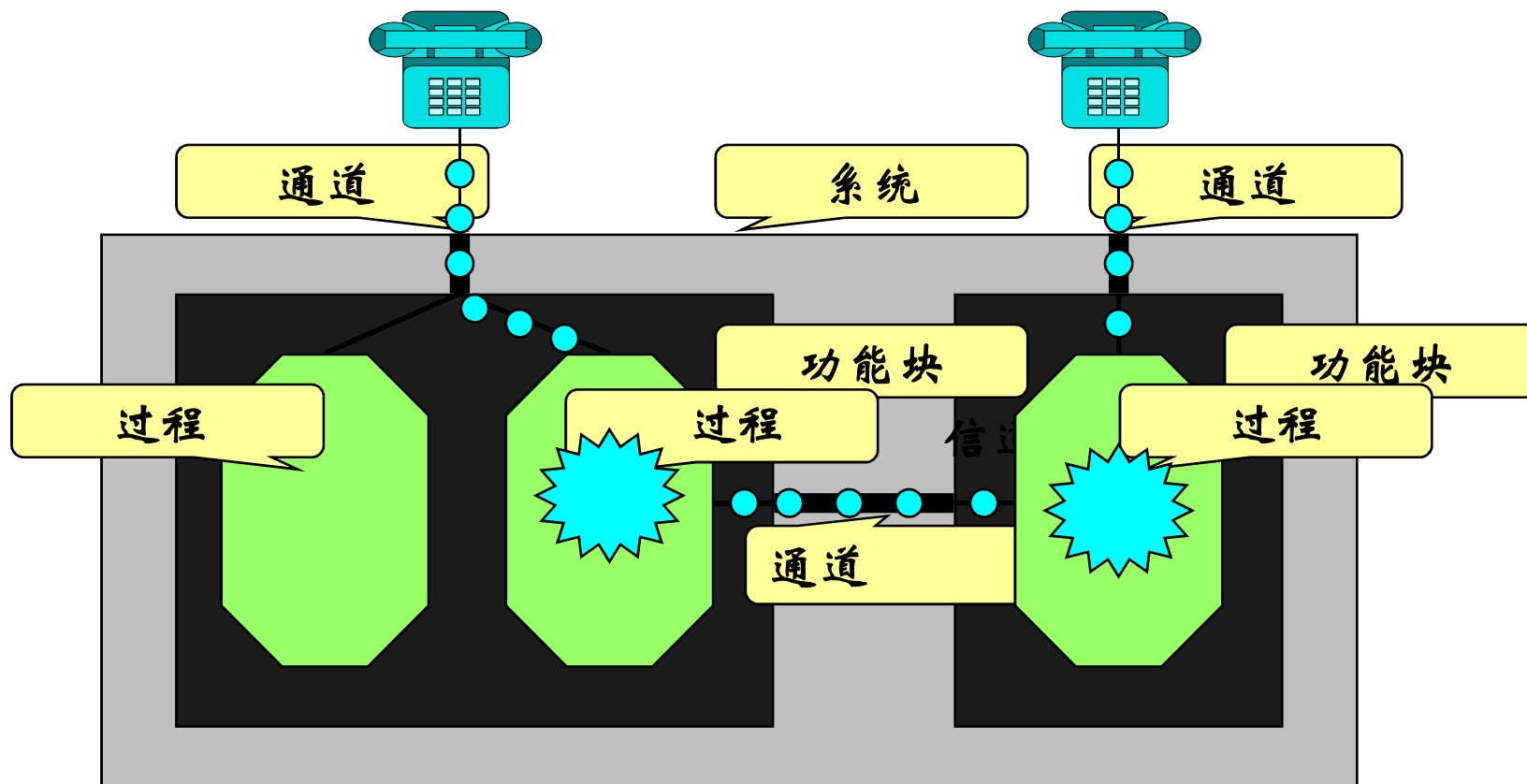
系统与其环境用系统边界分开。

通道在功能块之间以及在功能块与系统边界之间充当传输信号的媒介，既可是单方向的，也可以是双向的。

规范(Specification)所描述的对象称为一个系统。

功能块含有进程，并用来构造系统。

SDL 的结构和通信通道





基于SDL对系统的描述规范

- 在一个SDL应用中，一个系统的描述规范由以下几部分组成：
 - 一个或多个**功能块**：功能块通过通道互相连接，并通过**通道**连接到系统边界。
 - 在描述功能块时，**进程、服务类型、过程、数据类型和信号**均可作为功能块描述规范的一部分。
 - SDL进程可保留和处理数据。数据受进程的局部变量的约束。基于信号，数据值可在进程之间传送，也可发送到环境或从环境接收。进程的局部变量由该进程给出定义。
 - **所有通道、信号(Signal)、信号表(Signal List)、预定义的数据类型与/或宏(macro)的定义。**



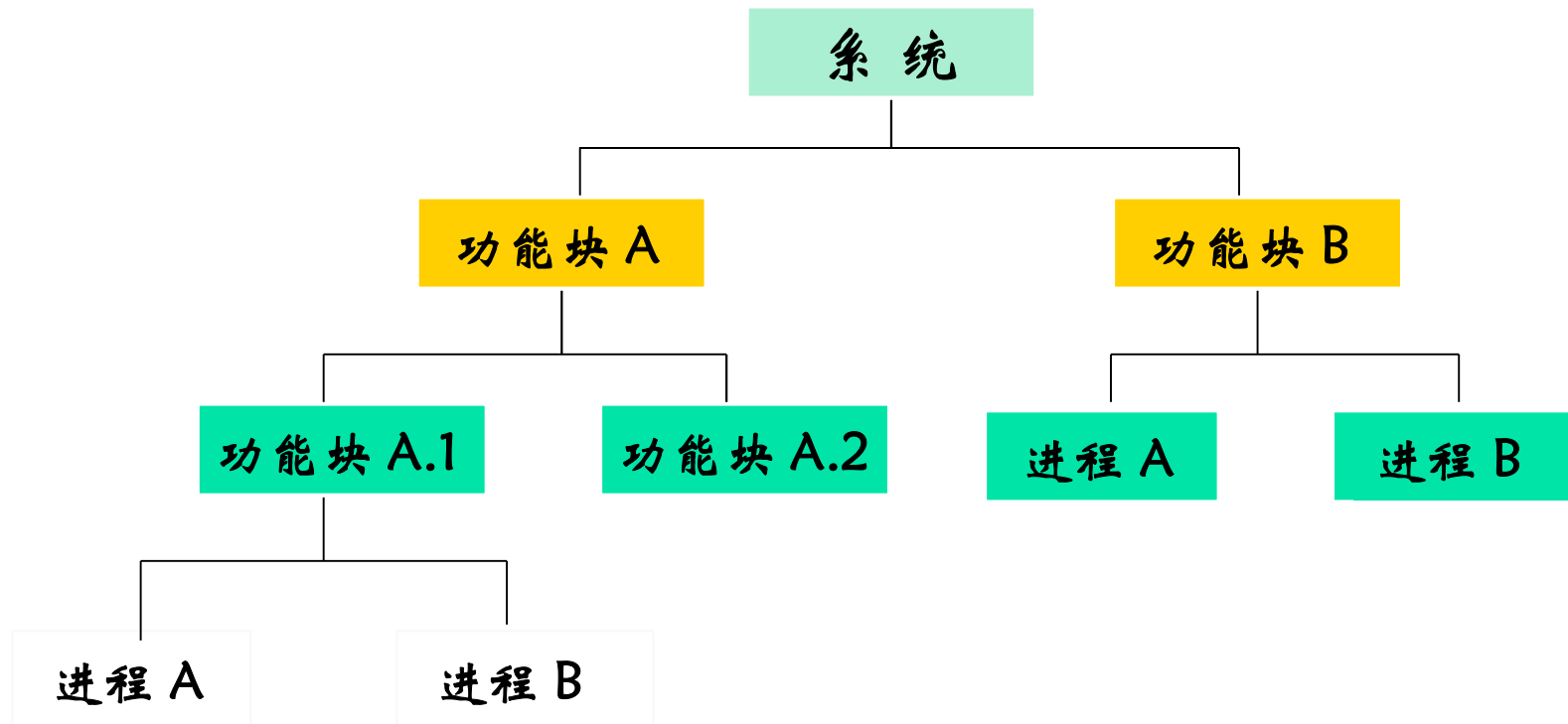
基于SDL对系统的描述规范

■ SDL的结构化概念

- 系统中的每一个功能块由一组进程或一组子功能块组成，进程间通过信号路由(Signal routes)或功能块的子结构(Substructure)相连接；
- 子功能块又可由一组进程或一组子功能块组成。这样，可以得到一个系统的分级多层描述。



模块的树形结构 (Tree of Blocks)





SDL对系统动态行为的描述

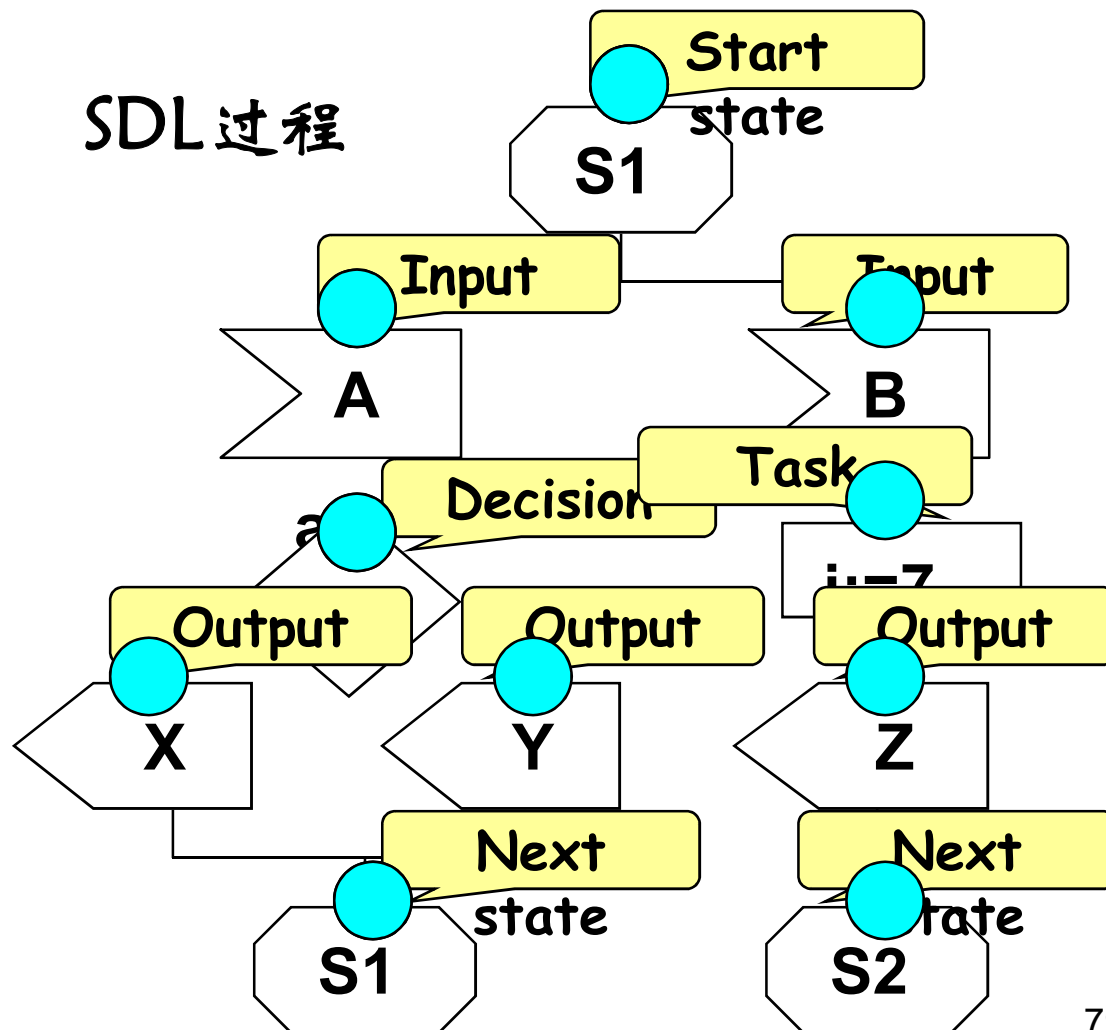
- 各功能块包含若干个进程，各进程的实例(Instance，即进程的一次活动，用Pid来表示)的并发操作产生了SDL系统的动态行为。
 - 进程的状态与动作是用扩展的FSM来模拟的；
 - 进程状态的变化受外来信号激励，进程也可向环境发回响应信号；
 - 一个进程可同时收到多个信号。SDL设置了队列以暂时存放这些信号(一般采用先进先出的方式)。但进程也可自行处理保留信号，保留信号是指在某些状态下需等待其他信号到达并处理之后方能处理的信号。

SDL的行为 (例子)

本SDL描述了一个系统在S1时能够收到信号A或B的过程：

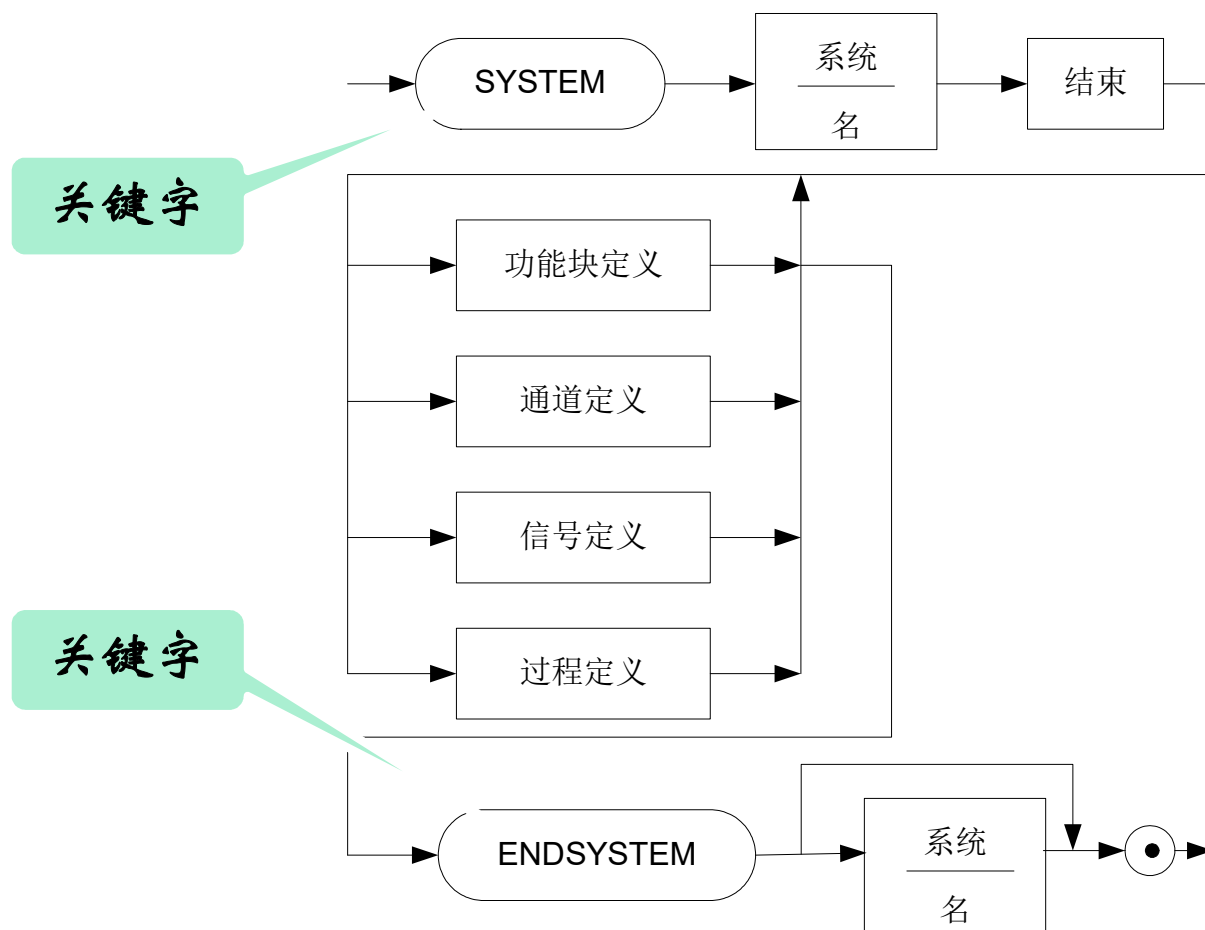
- 如果系统收到信号A，则系统可以输出X或者Y，并且回到S1（等待更多的A或B输入）；
- 如果系统收到信号B，系统将输出Z，并且转移到状态S2，等待新信号的到来。

SDL过程



SDL/PR 对系统的定义语法图

SDL/PR的语法采用Pascal的语法图形式定义





SDL中的数据

- 在SDL中，数据的定义一般基于抽象数据类型 (Abstract data type)
 - 抽象数据类型：是指其定义并不描述(数据)类型是如何实现的，而仅仅描述当将操作符 (Operators) 应用到该类型的数据值上时产生何种结果。
 - 例如，逻辑值“真”或“假”在具体的程序设计语言中可实现为{true; false}或{T, F}或{0, 1}。在数学上，一种抽象数据类型就是一种代数。



SDL预定义数据类型

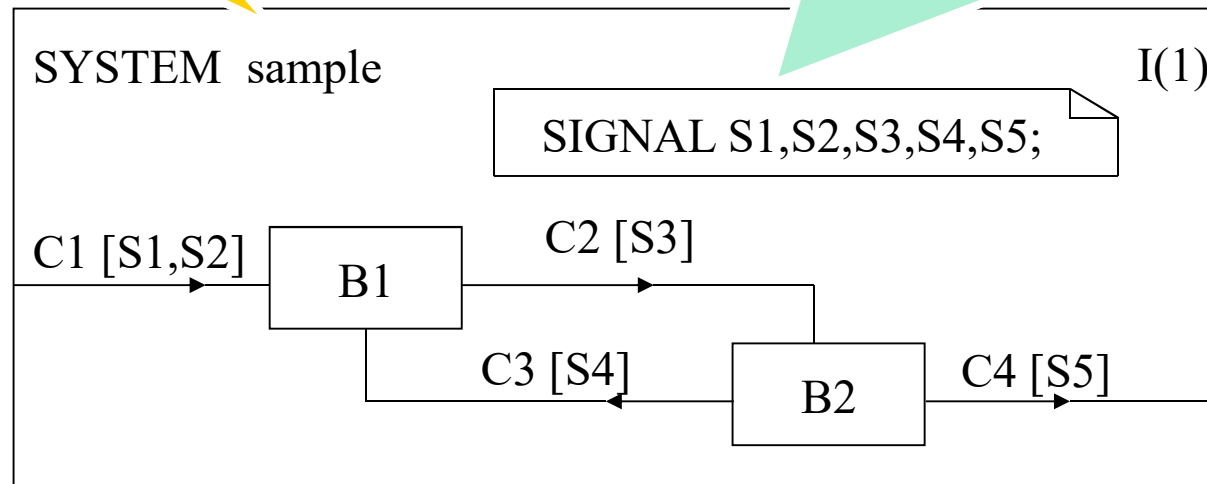
■ SDL中的数据

- SDL定义了一些预定义的数据类型，如布尔(Boolean)、整数(Integer)、自然数(Natural)、实数(Real)、字符(Character)、字符串(Charstring)、进程标识符(PId)、时刻(Time)、持续时间(Duration)。
- 此外，SDL还预定义了一些产生器(Generator)，如数组(Array)、串(String)、计时器(Timer)和幂集(Powerset)等。
- 这些预定义的数据类可在任何级别上SDL描述中使用。用户还可用上述数据类型自行定义其他的数据类型。

系统的描述

■ 用SDL/GR表示的系统例子

系统名

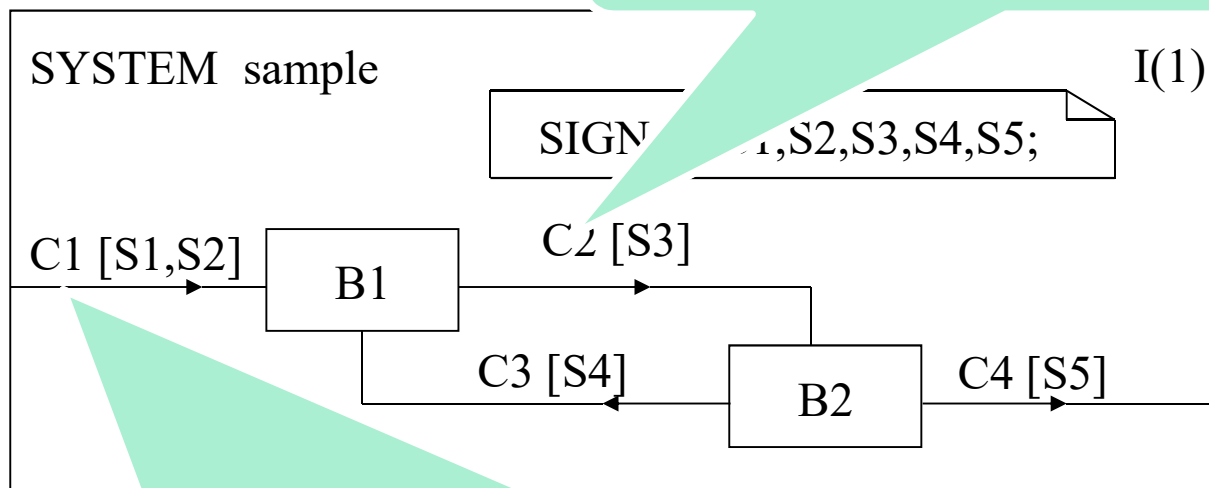


信号：描述系统中所有功能块之间或功能块与环境之间交互的信号，也可能还包括信号携带的值的类型的说明。系统级定义的信号既可用于本级系统描述，还可用于更低层次上的描述，如描述功能块中的信号。

信号表描述(Signal list specification)：信号表通常是指一组相关信号，并可以嵌套定义。使用信号表的目的是节省空间，并使得描述更简洁。在SDL/PR中，信号表用关键字 SIGNALLIST 来定义，而在SDL/PR中，用信号表符号(一个中括号)来表示。

系统的描述

- 通道名必须尽可能地靠近通道符号线;
- 通道的信号表在中括号中给出, 且靠近通道符号线上的箭头, 指示信号的方向。



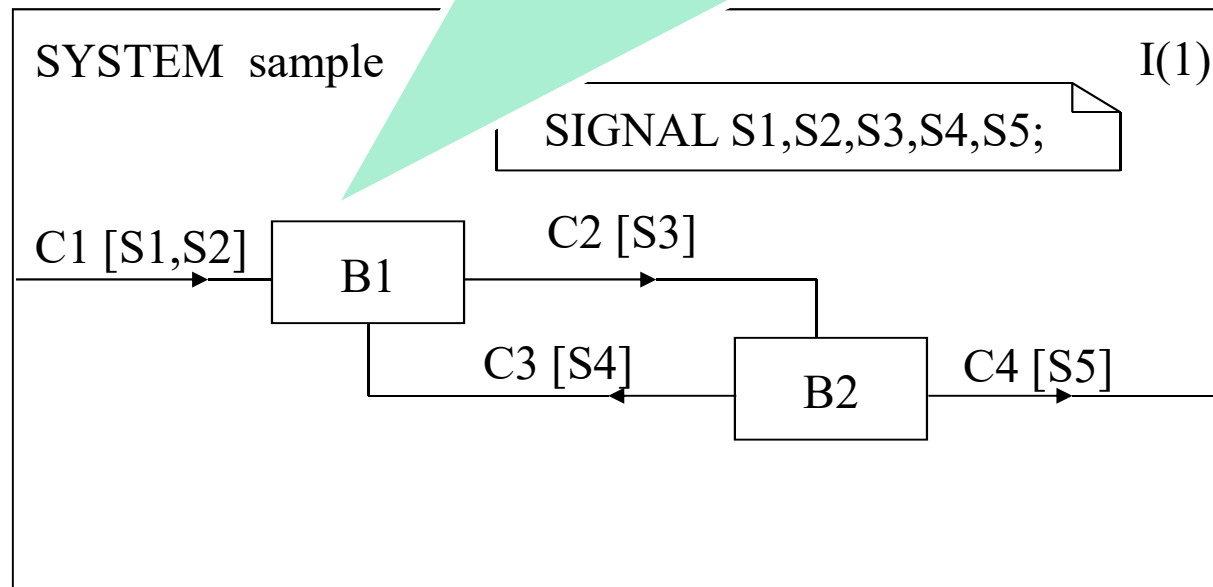
通道描述(Channel specification):

- 一个通道定义包括: 通道名, 通道可以传输的信号列表以及通道端点标志。通道用通道符号 (一根带箭头的线条) 来表示。
- 通道端点中至少有一个必须是功能块。在SDL/PR中, 通道的定义包括在关键字CHANNEL和ENDCHANNEL之间, 关键字FROM和TO表示通信路径, 而关键字WITH表示信号列表。

系统的描述（用SDL/GR表示的系统例子）

功能块描述(Block specification):

- 一个系统中必须至少包含一个功能块;
- 在系统图(diagram)中, 对功能块的描述有两种形式: 一种是功能块引用(block reference), 这种方式只需要指出功能块的名字; 另一种是详细描述功能块内的进程以及进程间的交互的功能块图(block diagram)。



- 数据描述(Data specification): 这一部分描述用户定义的对所有功能块可见的数据类(NEWTYPE, SYNTYPE, GENERATOR)。



系统的描述（用SDL/PR表示的系统例子）

```
SYSTEM sample;  
  SIGNAL S1, S2, S3, S4, S5;  
  CHANNEL C1 FROM ENV TO B1 WITH S1, S2;  
  ENDCHANNEL C1;  
  CHANNEL C2 FROM B1 TO B2 WITH S3;  
  ENDCHANNEL C2;  
  CHANNEL C3 FROM B2 TO B1 WITH S4;  
  ENDCHANNEL C3;  
  CHANNEL C4 FROM B2 TO ENV WITH S5;  
  ENDCHANNEL C4;  
  BLOCK B1 REFERENCED;  
  BLOCK B2 REFERENCED;  
ENDSYSTEM sample;
```



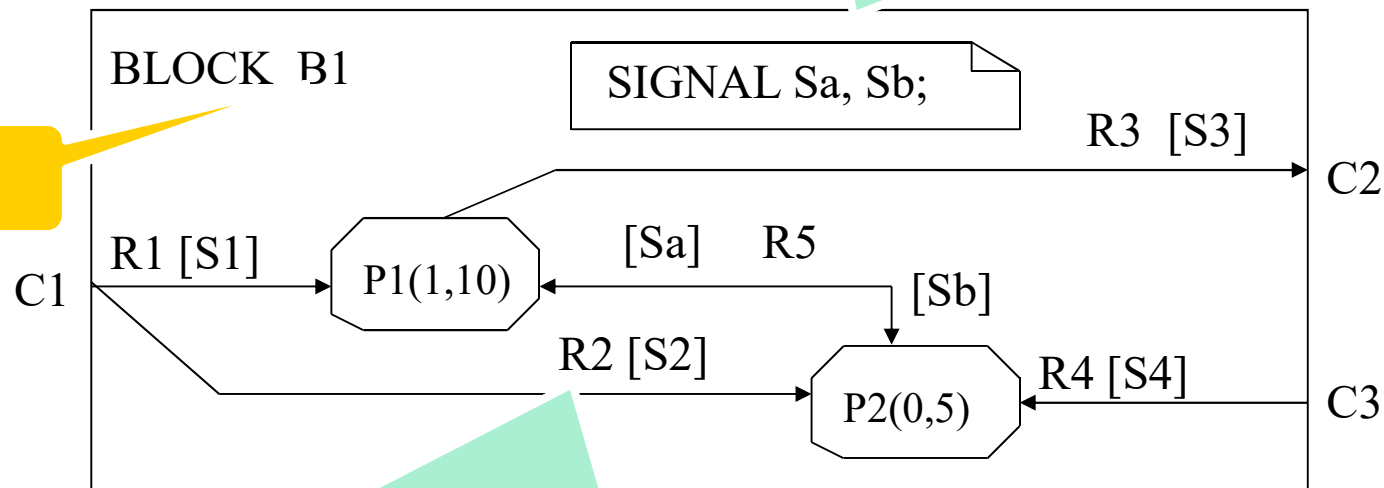
功能块的描述

- 功能块的主要目的是描述系统结构。
- 利用功能块，可以将一个大的系统分层描述，不同层次对应着系统的不同抽象级别。
- 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，通信手段主要包括共享变量或通过信号路由(signal routes)交换信号。

功能块的描述

信号或信号表描述

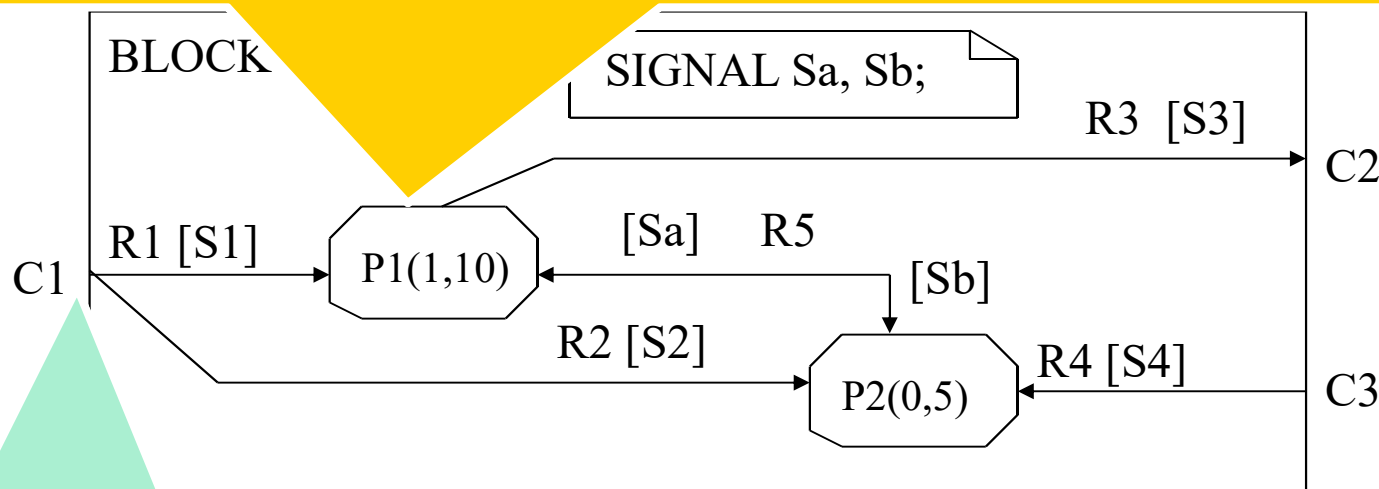
块名



信号路由描述(Signal route specification): 信号路由是指连接功能块内的不同进程以及将功能块内的进程与环境连接起来的通信路径。与通道一样, 信号路由也是用来表示通信路径的, 既可是单向的, 也有双向信号路由。信号路由中的箭头在线条的两端。

功能块的描述 (SDL/GR表示的功能块例子)

进程描述(Process specification): 描述功能块内的进程。如果功能块内没有子功能块, 则必须至少有一个进程在此处描述。进程可以有两种表现形式, 一种是仅仅包含进程名的进程引用(Process reference), 另一种是与进程的有限状态机相一致的详细描述(进程图)。



通道与信号路由间的连接(Channel-to-route connection): 通道与信号路由间的连接是指功能块外面的通道与功能块内部的信号路由间的连接。

数据描述(Data specification): 这一部分描述用户定义的对所有功能块可见的数据类(NEWTYPE, SYNTYPE, GENERATOR)。



功能块的描述

■ 用SDL/PR表示的功能块例子

```
BLOCK B1;  
  SIGNAL Sa, Sb;  
  SIGNALROUTE R1 FROM ENV TO P1 WITH S1;  
  SIGNALROUTE R2 FROM ENV TO P2 WITH S2;  
  SIGNALROUTE R3 FROM P1 TO ENV WITH S3;  
  SIGNALROUTE R4 FROM ENV TO P2 WITH S4;  
  SIGNALROUTE R5 FROM P1 TO P2 WITH Sa;  
                                FROM P2 TO P1 WITH Sb;  
  
  CONNECT C1 AND R1, R2;  
  CONNECT C2 AND R3;  
  CONNECT C3 AND R4;  
  PROCESS P1 REFERENCED;  
  PROCESS P2 REFERENCED;  
ENDBLOCK B1;
```



进程的描述

■ 系统的动态行为通过进程来体现

- 进程包含很多状态，在不同状态下对收到的信号作出不同的处理；
- 可以将状态看作是进程运行过程中的某一点，在该点进程不执行任何动作，但却在监视输入队列中是否有信号到达；
- 进程在某一状态等待一个信号，当收到一个信号时，进程执行与该信号相关的指定操作；执行完所有规定的操作后，进程进入下一个状态，等待新的输入信号。



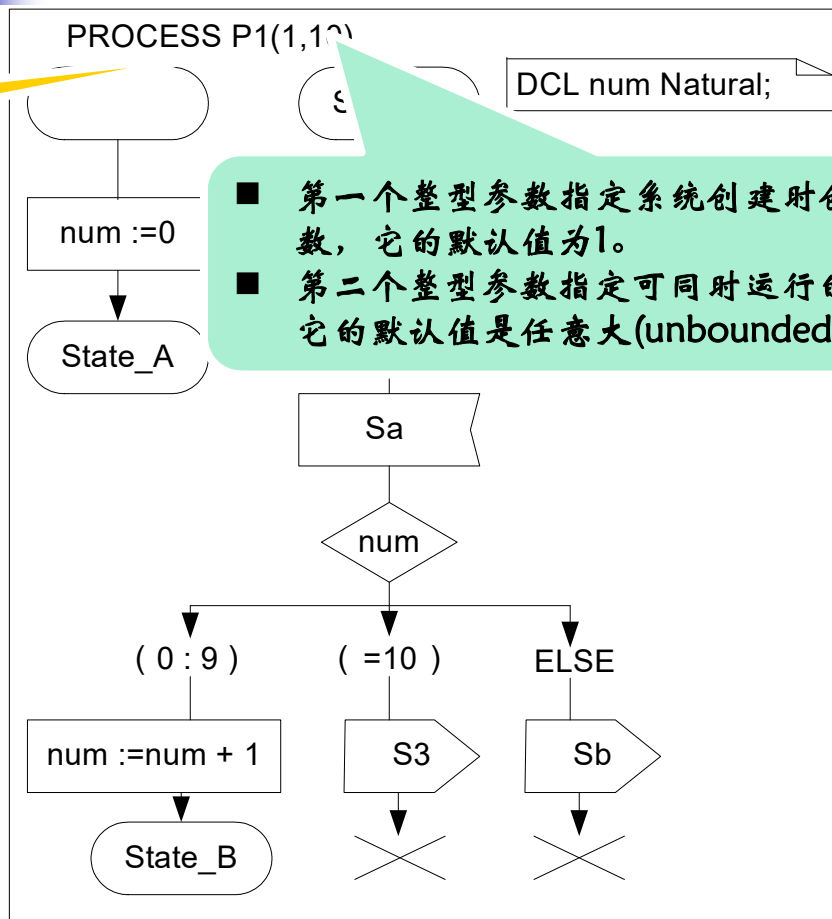
进程的描述

- 进程的描述规范实际上定义的是进程的类型。
同一类型的进程可以有多个实例，这些实例可
同时被创建和并行运行。
- 进程创建有两种方式（进程实例）：
 - 一种是在创建系统时产生；
 - 另一种是用CREATE调用来显示创建，这种方式一般
只被同一功能块中的另一个进程来使用；
 - 进程可以一直存活下去，或执行一个停止操作(Stop)
来终止运行。

进程的描述

变量定义: 可以将变量定义成与同一功能块中其它进程共享的变量(REVEALED变量), 或可被其它功能块中的进程访问的变量(EXPORTED变量)。

进程名



(a) 用SDL/GR表示的进程例子

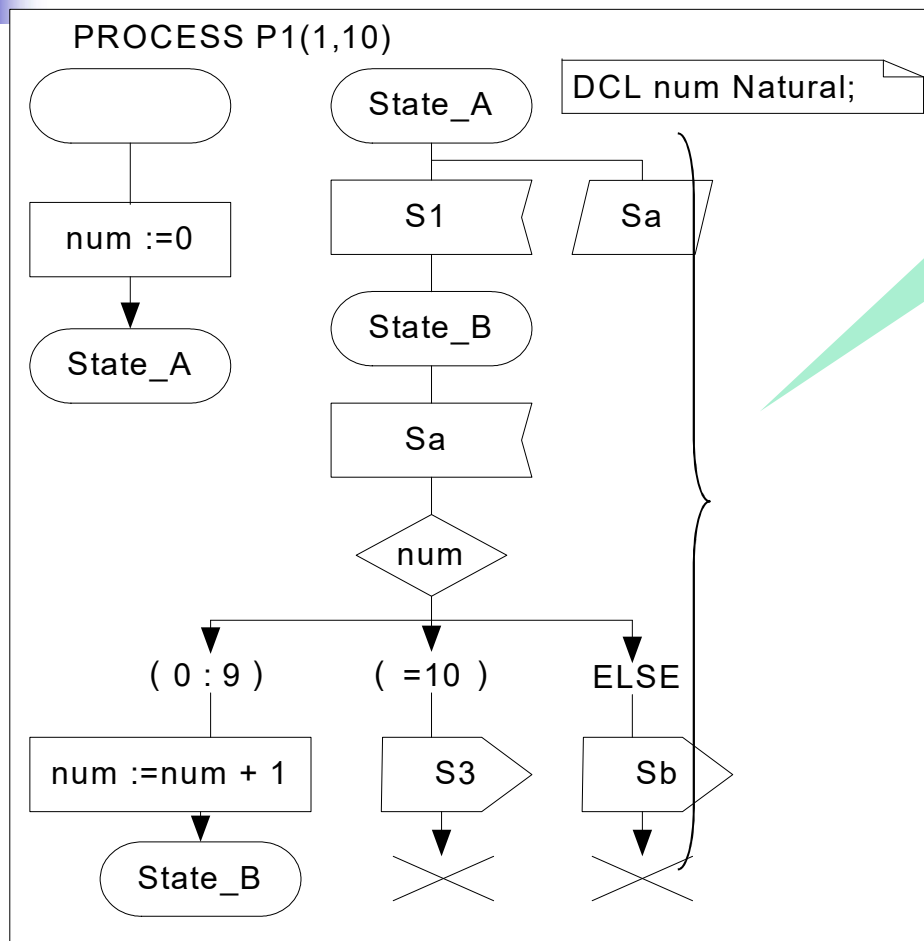
```
PROCESS P1(1,10);
DCL num Natural;

:=0;
'E State_A;
A;
ATE State_B;

ENDSTATE;
STATE State_B:
INPUT Sa;
DECISION num;
(0:9):
TASK num := num+1;
NEXTSTATE State_B;
(=10):
OUTPUT S3;
STOP;
ELSE
OUTPUT Sb;
STOP;
ENDDECISION;
ENDSTATE;
ENDPROCESS;
```

(b) 用SDL/PR表示的进程例子

进程的描述



(a) 用SDL/GR表示的进程例子

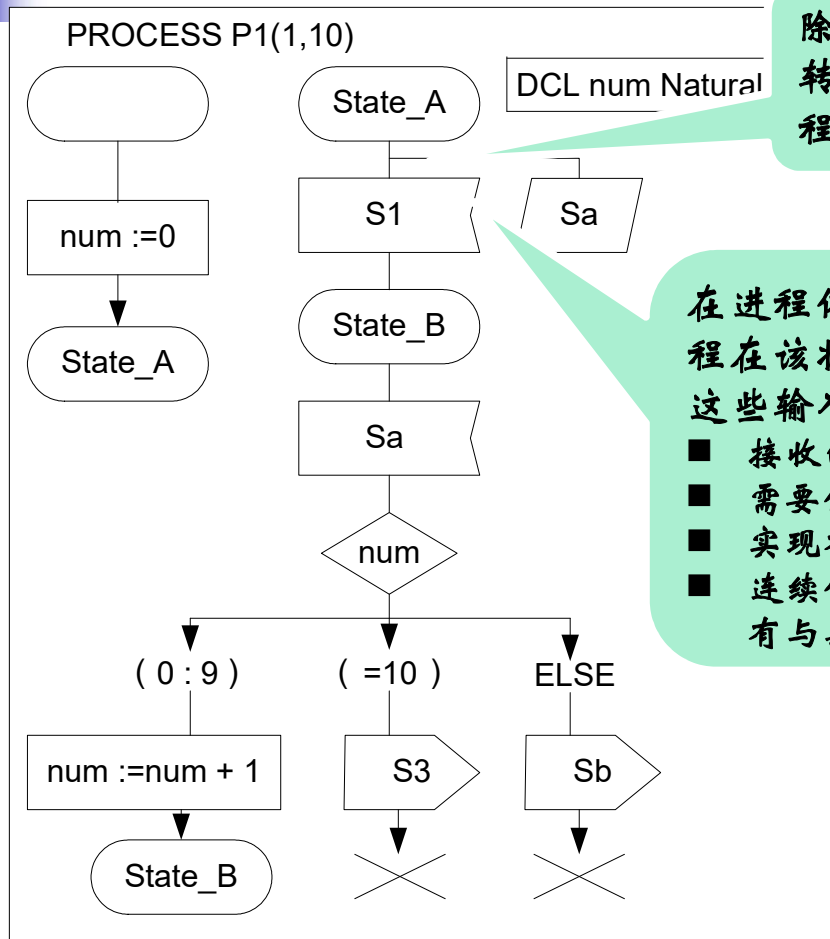
- 进程体描述进程的行为，由状态、输入、输出、任务等构成。
- 进程体的描述规范总是以START开始，其后是一组动作(转换或变迁)的描述。
- 当进程实例被创建时，进程开始解释执行。

```

PROCESS P1(1,10);
    DCL num Natural;
    START;
    TASK num := 0;
    NEXTSTATE State_A;
    STATE State_A;
    INPUT S1;
    NEXTSTATE State_B;
    SAVE Sa;
    ENDSTATE;
    STATE State_B;
    INPUT Sa;
    DECISION num;
    (0:9):
        TASK num := num + 1;
        NEXTSTATE State_B;
    (=10):
        OUTPUT S3;
        STOP;
    ELSE
        OUTPUT Sb;
        STOP;
    ENDDECISION;
    ENDSTATE;
ENDPROCESS;
    
```

(b) 用SDL/PR表示的进程例子

进程的描述



(a) 用SDL/GR表示的进程例子

除了Save以外，上述每一个输入必须有对应的转换(动作系列)定义，当这些输入出现时，进程将执行定义的转换。

在进程体中，每一个状态定义开始时都必须描述进程在该状态等待的所有可能的输入或激励(Stimuli)，这些输入包括：

- 接收的所有输入信号(Inputs)
- 需要保存下来在将来处理的信号(Saves)
- 实现有条件接收信号的使能条件(Enabling conditions)
- 连续信号(Continuous signals，与使能条件相似，但没有与具体的信号相关联)。

```

ELSE
  OUTPUT Sb;
  STOP;
ENDDECISION;
ENDSTATE;
ENDPROCESS;

```

(b) 用SDL/PR表示的进程例子



进程的描述

■ 其它的描述内容

- 合法的输入信号集(Valid input signal set): 定义进程可以接收的信号, 用信号标志符表来表示。
- 过程描述(Procedure specification): 指定进程可以调用的过程; 过程的描述也可只包含一个过程名, 表示一个可独立定义的本地过程, 这种方式称为过程引用(Procedure call)。
- 数据描述(Data specification): 定义进程使用的局部数据类。
- 视图变量定义(View definition): 声明可以保存其它进程实例拥有的变量值的变量, 称为视图变量; 同样, 对于每一个视图变量, 必须指定数据类型。

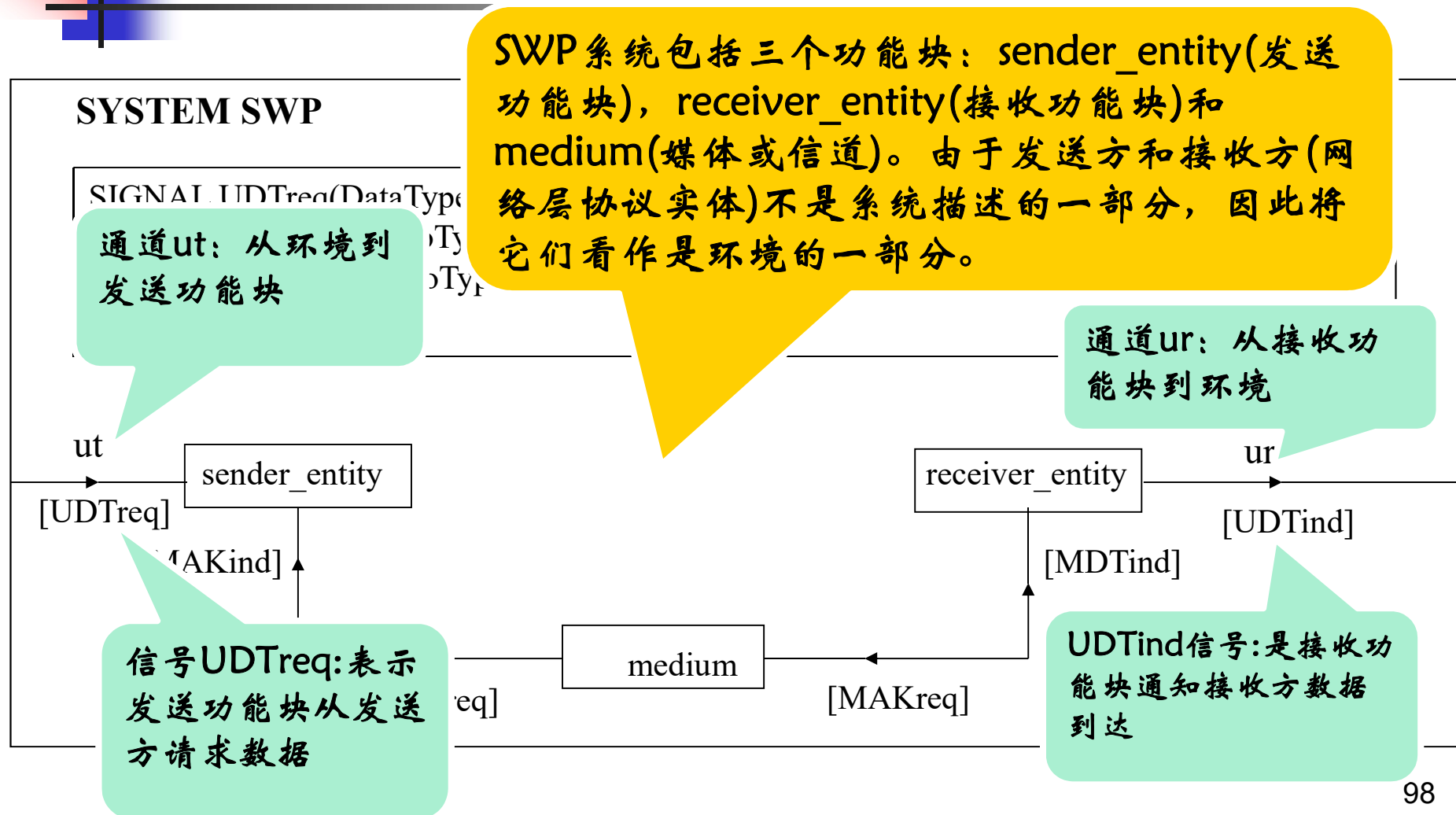


进程的描述

■ 其它的描述内容

- 进口变量定义(Import definition): 声明一个进程想要访问的其它进程中的变量, 称为进口变量; 同样, 对于每一个进口变量, 必须指定数据类型。
- 计时器定义(Timer definition): 声明一个进程体中使用的计时器。

基于SDL描述滑动窗口协议 (SWP: Sliding Window Protocol)



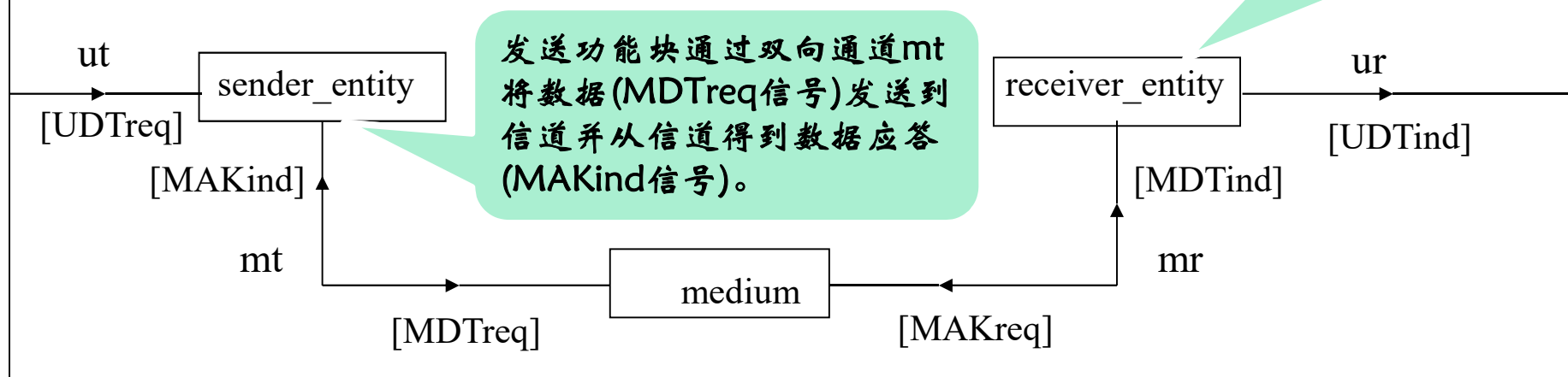
基于SDL描述滑动窗口协议 (SWP)

SYSTEM SWP

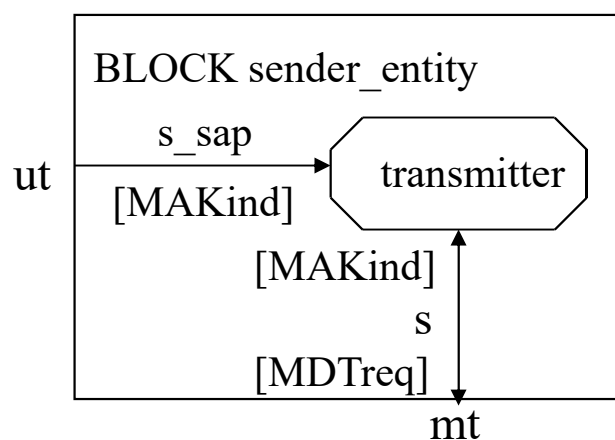
```
SIGNAL UDTreq(DataType), UDTind(DataType),  
      MDTreq(SeqnoType, DataType, datacrc), MDTind(SeqnoType,  
      MAKreq(SeqnoType, ackcrc), MAKind(SeqnoType, ackcrc);  
MACRO DataTypeDef;
```

接收功能块则通过双向通道mr从信道得到数据(MDTind信号)并发送数据应答(MAKreq信号)到信道上。

发送功能块通过双向通道mt将数据(MDTreq信号)发送到信道并从信道得到数据应答(MAKind信号)。

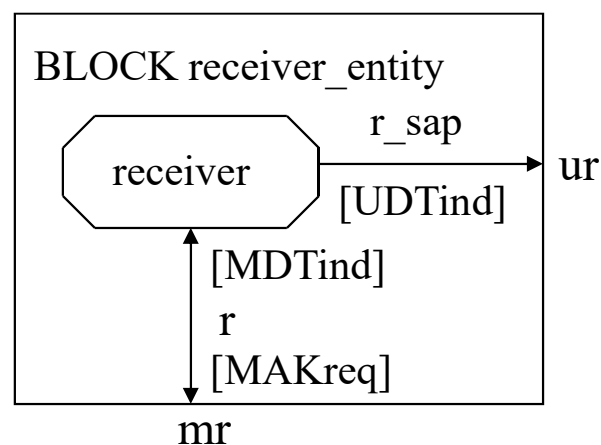


基于SDL描述滑动窗口协议 (SWP)



(a) 功能块 sender_entity

发送功能块只包含一个进程 transmitter，该进程在系统启动时被实例化一次，且只被实例化一次。



(b) 功能块 receiver_entity

接收功能块同样也只包含一个进程 receiver，且在系统启动时被实例化一次。

基于SDL描述滑

信道由两个队列管理进程(MsgManager和AckManager)和两个错误处理进程(MsgHazard和AckHazard)组成。

VP)

信号及信号表定义

BLOCK medium

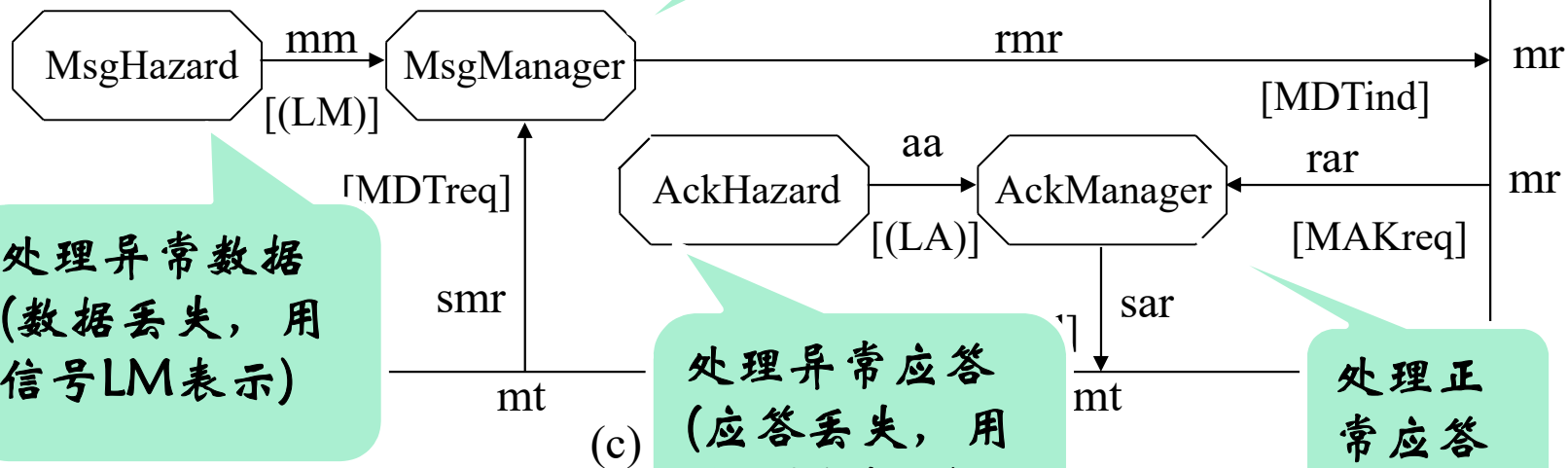
SIGNAL ANormal, ALost, ADup, ACorrupt;
MNormal, MLost, MDup, MReord, MCorrupt;
SIGNALLIST LA=ANormal, ALoose, ACorrupt;
SIGNALLIST LM=MNormal, MLoose, MReord, MCorrupt;

处理正常数据

处理异常数据
(数据丢失, 用信号LM表示)

处理异常应答
(应答丢失, 用信号LA表示)

处理正常应答



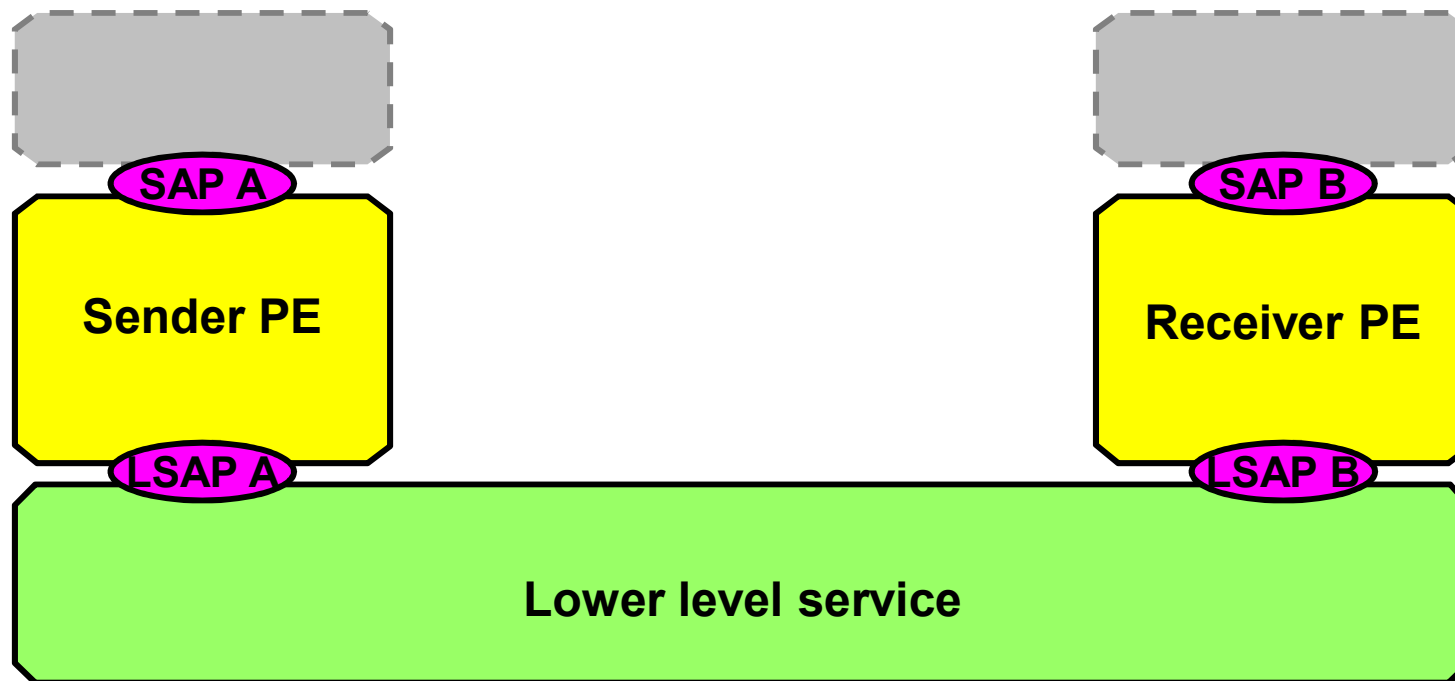


基于SDL描述滑动窗口协议 (SWP)

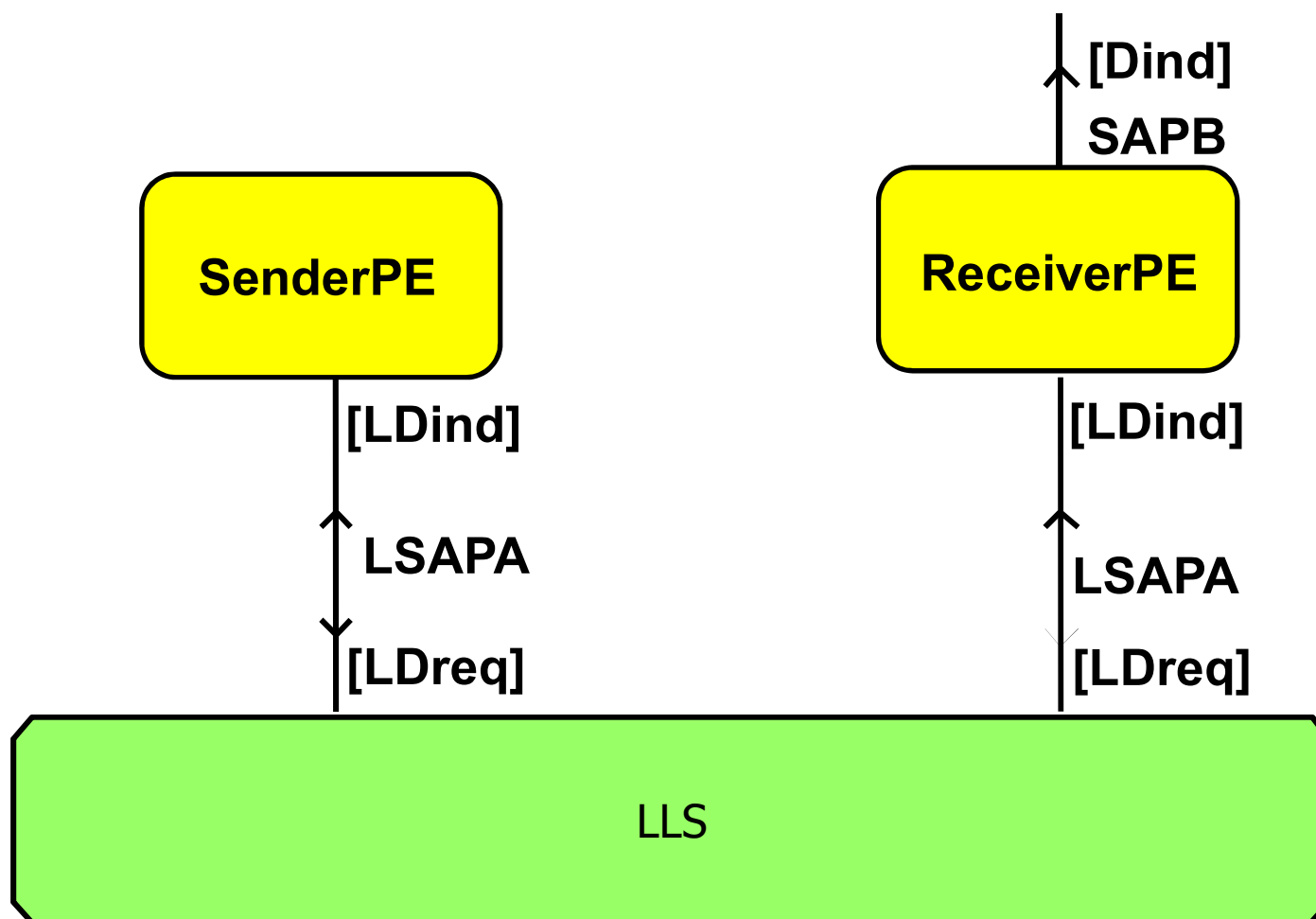
- 上面的讨论只是描述了滑动窗口协议中部分协议操作。对于协议中的一个很重要的协议操作，即计时器则没有描述。
 - 在滑动窗口协议中，协议需要为每一个发送的数据帧启动一个计时器。
 - 可以用SDL中的Timer类和时间信号(tim)来模型化计时器(tim的不同实例分别对应不同的发送报文的计时器)，用Duration类来描述超时值，并定义计时器操作原语SET和RESET。
- 有关细节可参考ISO相关技术报告。

基于SDL对协议的描述假设

- 协议的两个实体
 - 发送者端实体 (PE) 和接收者端实体 (RE)



基于SDL的系统协议结构





X-on/X-off protocol

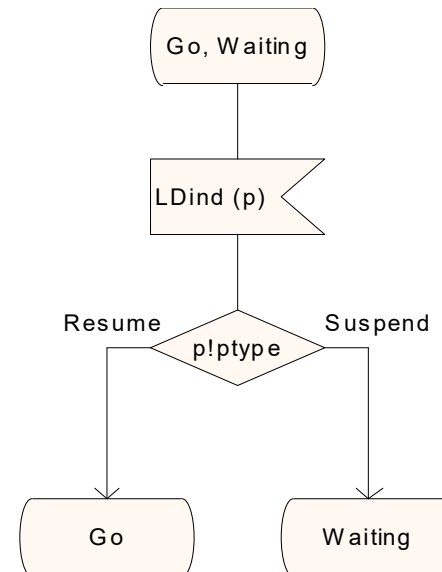
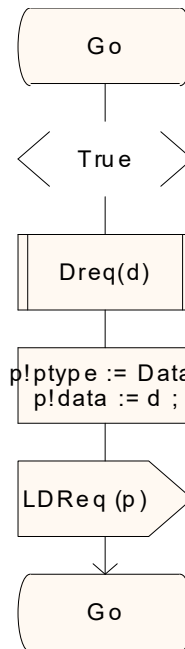
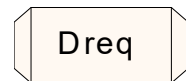
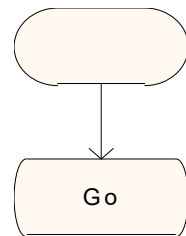
- Three PDU types
 - **Data PDU**: sent to the receiver PE by the sender PE; it contains user data;
 - **Suspend PDU**: sent to the sender PE by the receiver PE to signal that the sender should stop sending data PDUs;
 - **Resume PDU**: sent to the sender PE by the receiver PE to signal that the sender should resume sending data PDUs.

X-on/X-off protocol: SenderPE

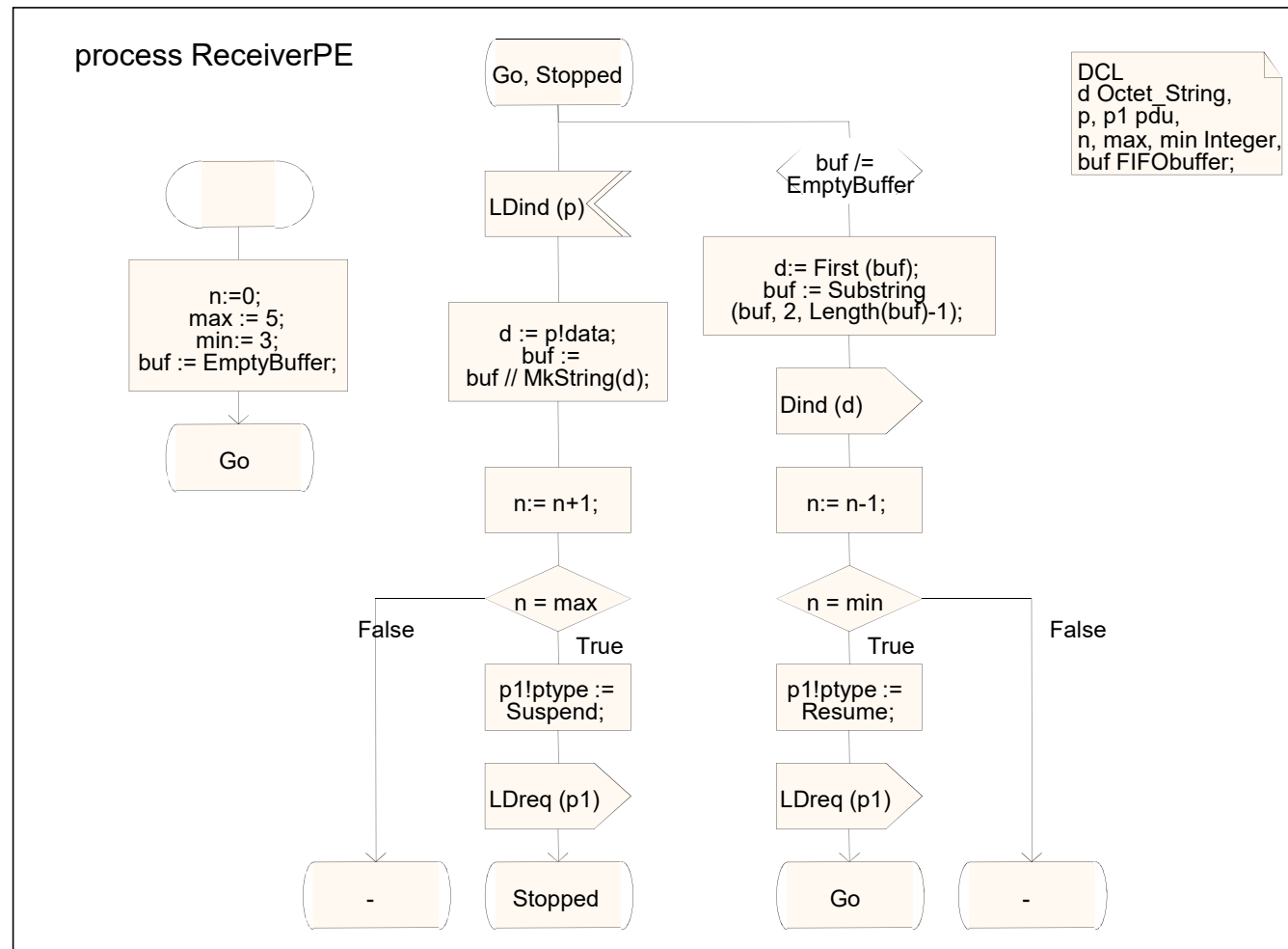
process SenderPE



DCL
d Octet_String,
p pdu;



X-on/X-off protocol: Receiver PE





对宽带通信系统和网络的描述

- 宽带通信网络的组成---节点的功能定义
- 宽带通信网络中节点的主要功能
- 宽带通信网络的节点/系统功能有效划分
- 通信协议



宽带通信网络的组成---节点的功能定义

- 不同网络的节点组成以及实现的功能不同
 - 电路交换网络（例如：用户终端、交换机、接入设备……）
 - 包交换网络（例如：交换机、接入设备、用户终端……）
 - 互联网络（例如：主机、路由器……）



宽带通信网络中节点的主要功能

■ 节点可能支持的功能

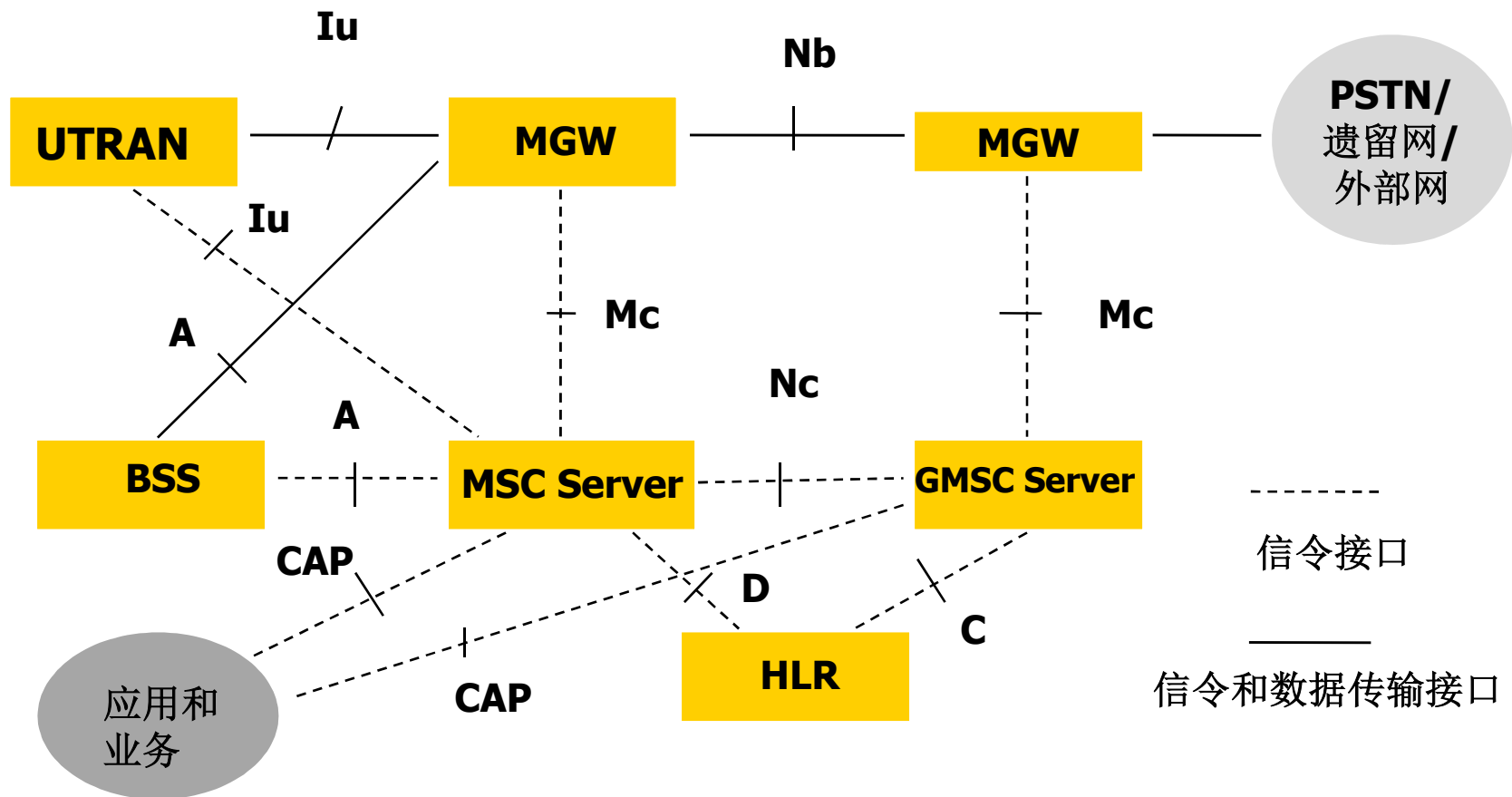
- 接入控制
- 路由选择与交换
- 信息转发
- 资源分配
- 服务质量控制
- 业务及流量管理、拥塞控制
- 运行管理、故障处理
-



宽带通信网络的节点/系统功能有效划分

- 网络由不同的系统互联而成；
- 网络和系统中传送的数据流按照其完成的功能，被划分成不同平面（控制平面/用户平面/管理平面）
- 系统功能有效划分
 - 节点/实际设备/端口；
 - 功能模块/逻辑实体/接口；
 - 接口与协议栈

宽带通信网络的节点/系统功能有效划分（移动核心网络R4版本的例子）





宽带通信网络中的通信协议

- 通信协议给出了对等实体之间进行通信的语义、语法和规则（what/how/when）。
- 通信协议的要素
 - 协议支持的能力；
 - 所在层、数据平面/控制平面；
 - 协议的实体；
 - 功能模块及其动态行为（状态、状态转移；过程、进程）；
 - 协议包含的原语/消息集、数据包的格式；
 - 各场景下的交互流程。
- 基于规范描述语言对协议的描述
 - 一个协议的实例（LDP）

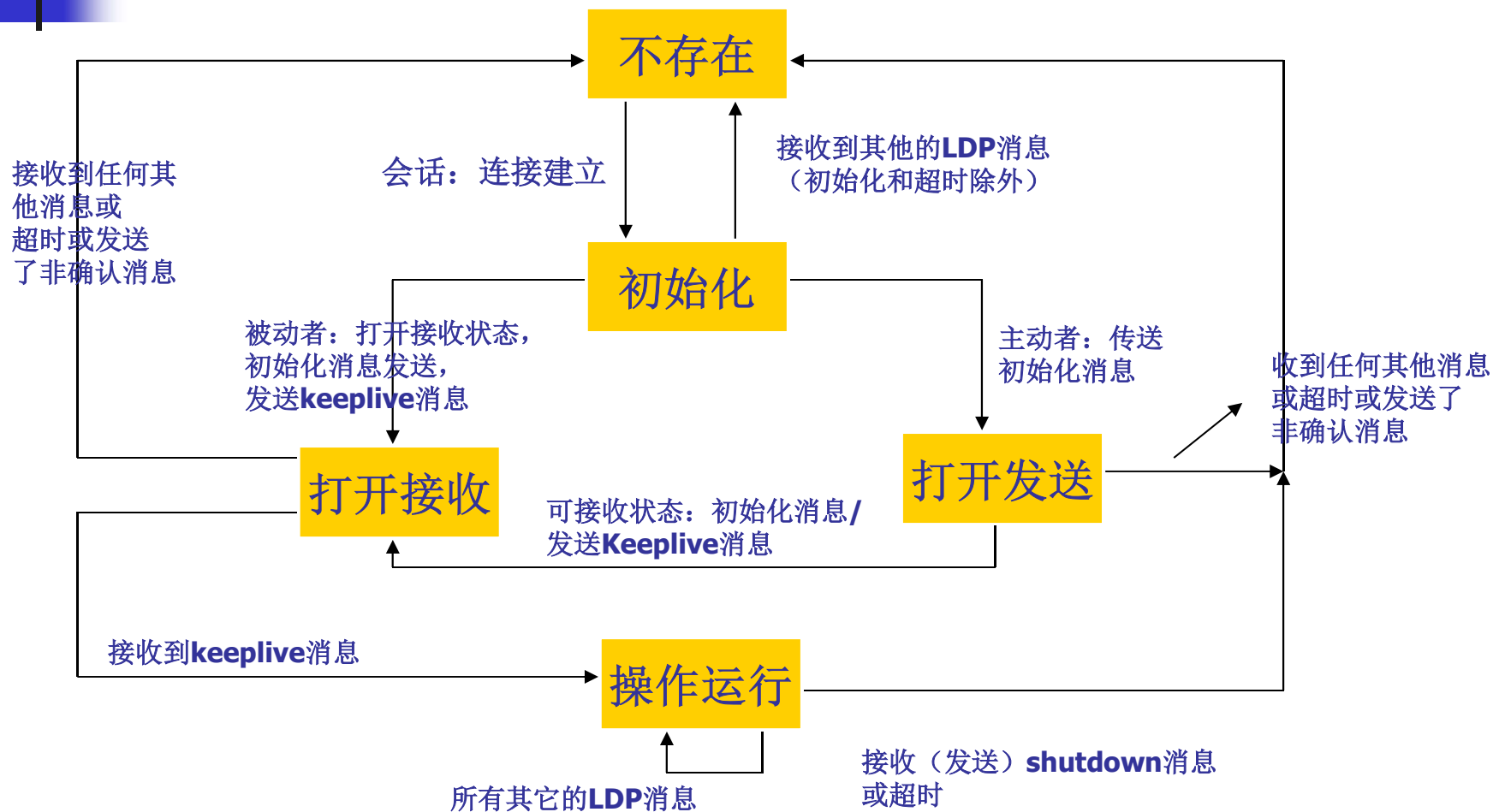


LDP协议支持的能力(实例)

■ LDP支持的能力

- 转发等价类到LSP的映射
- 标签空间、LDP标识、会话与传输
- LDP对等层侦测
- LDP会话建立与维护
- 标签分发与管理
- 循环探测与预防
- LDP错误处理

LDP协议的状态转移图（会话初始化状态）





LDP协议的功能

- LDP协议功能
 - LDP PDU消息格式
 - LDP的消息类型
 - LDP的消息格式
 - LDP的消息TLV编码格式
 - LDP的消息处理



LDP的消息集合

消息名	消息功能
地址取消	LSR取消先前公布给其对等层的接口地址
标签映射	LSR公布给对等层有关的FEC-标签绑定
标签请求	LSR用以向对等层申请一个FEC-标签绑定
标签申请退出	取消前面发出的标签申请
标签取消	解除FEC-标签绑定
标签释放	释放LSR不再需要的前面申请的标签

LDP的通用消息格式和各消息的具体格式

版本号 (2字节)	PDU长度 (2字节)
LDP标识 (6字节)	

LDP PDU消息格式

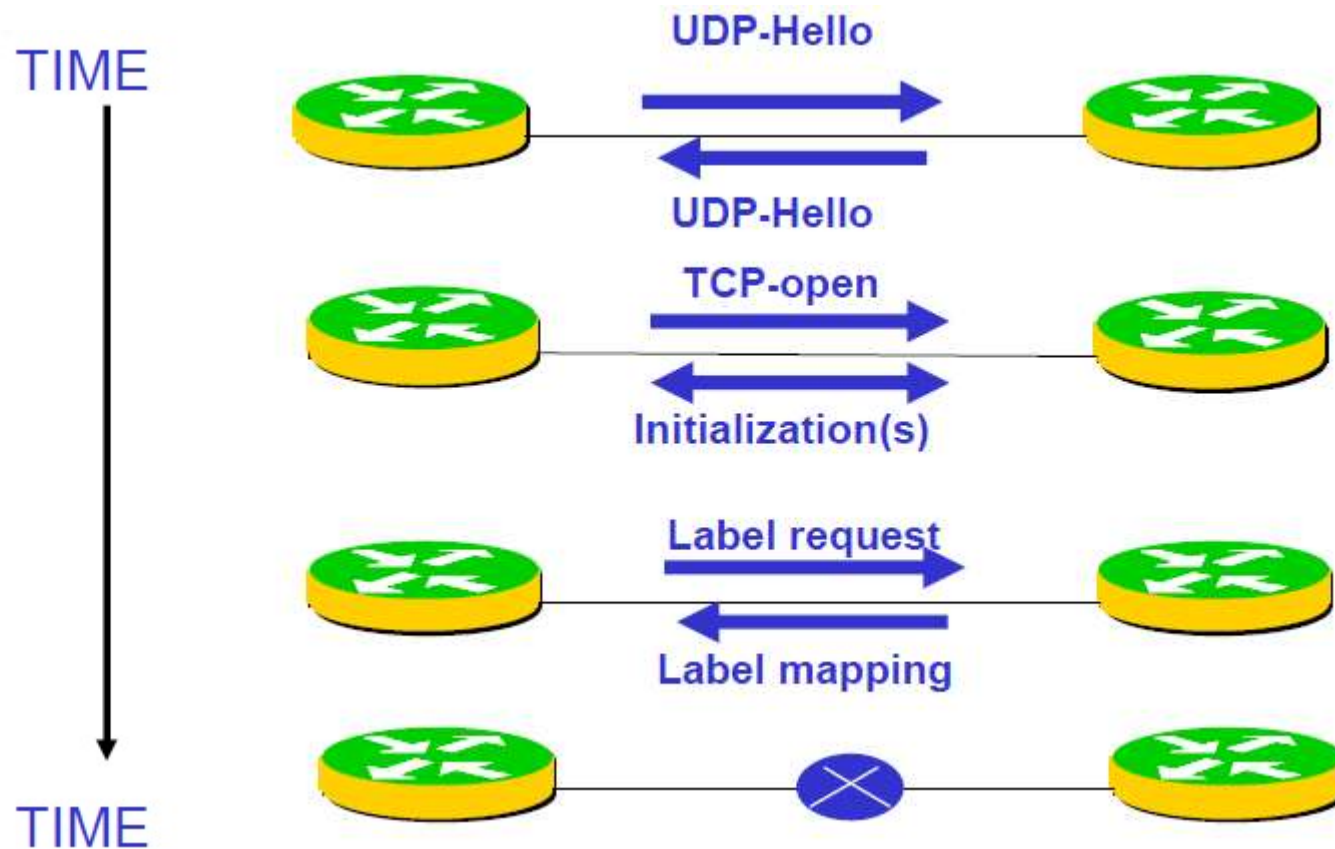
U比特 (1bit)	F比特 (1bit)
类型域 (14bit)	
长度 (2bytes)	
值域 (可变长度)	

LDP的消息TLV编码格式

0	0	地址列表(0x0101) 14bit
长度 (2字节)		
地址簇 (2字节)		
地址 (可变长度)		

地址列表TLV的格式

LDP协议的信息交互(实例)

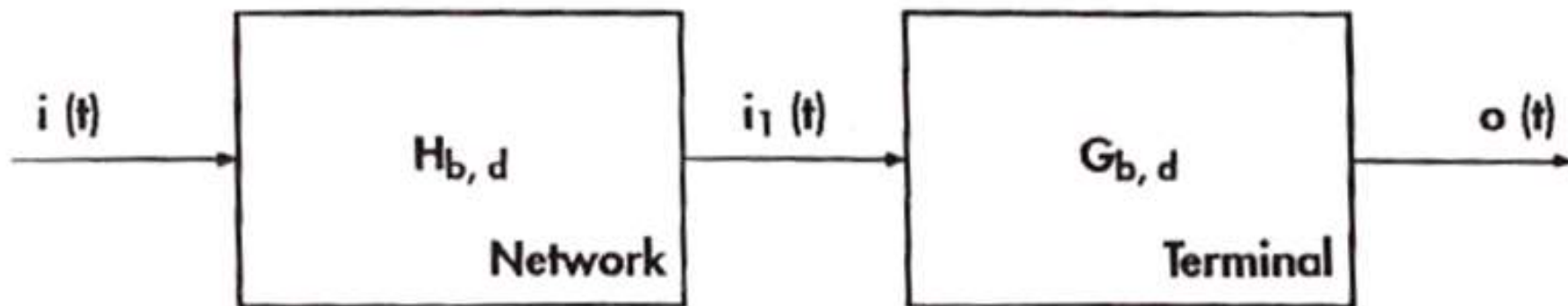
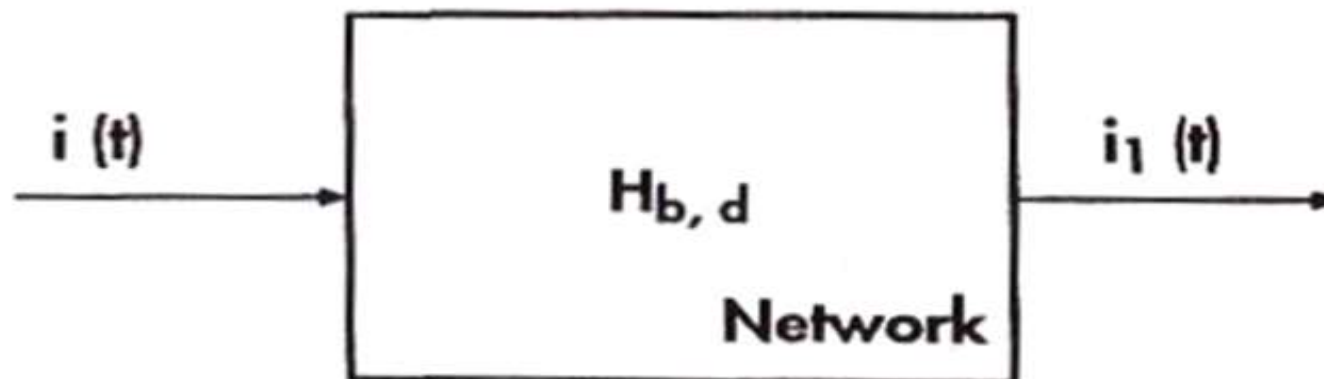




宽带通信网网络性能的主要评估方法

- 宽带通信网络的一般模型
- 宽带通信网的业务特征评估---业务特征参数
- 宽带通信网的网络性能评估---网络性能参数

宽带通信网络的一般模型



宽带通信网的业务特征评估---业务特征参数

■ 自然信息速率 (Natural Information Rate)

信息速率通常是一个变化的量，受使用人的行为、传输内容和信号处理方法等方面的影响，单位为bps：

■ 信息速率通常由以下参数描述：

■ 峰值速率 $S = \max s(t)$
 $0 \leq t \leq T$

■ 平均速率 $E[s(t)] = \frac{1}{T} \int_0^T s(t) dt$

宽带通信网的业务特征评估---业务特征参数

- 突发性: 业务流的突发性可由突发度来描述;

- 业务的突发度 (突发性) $B = \frac{S}{E[s(t)]}$

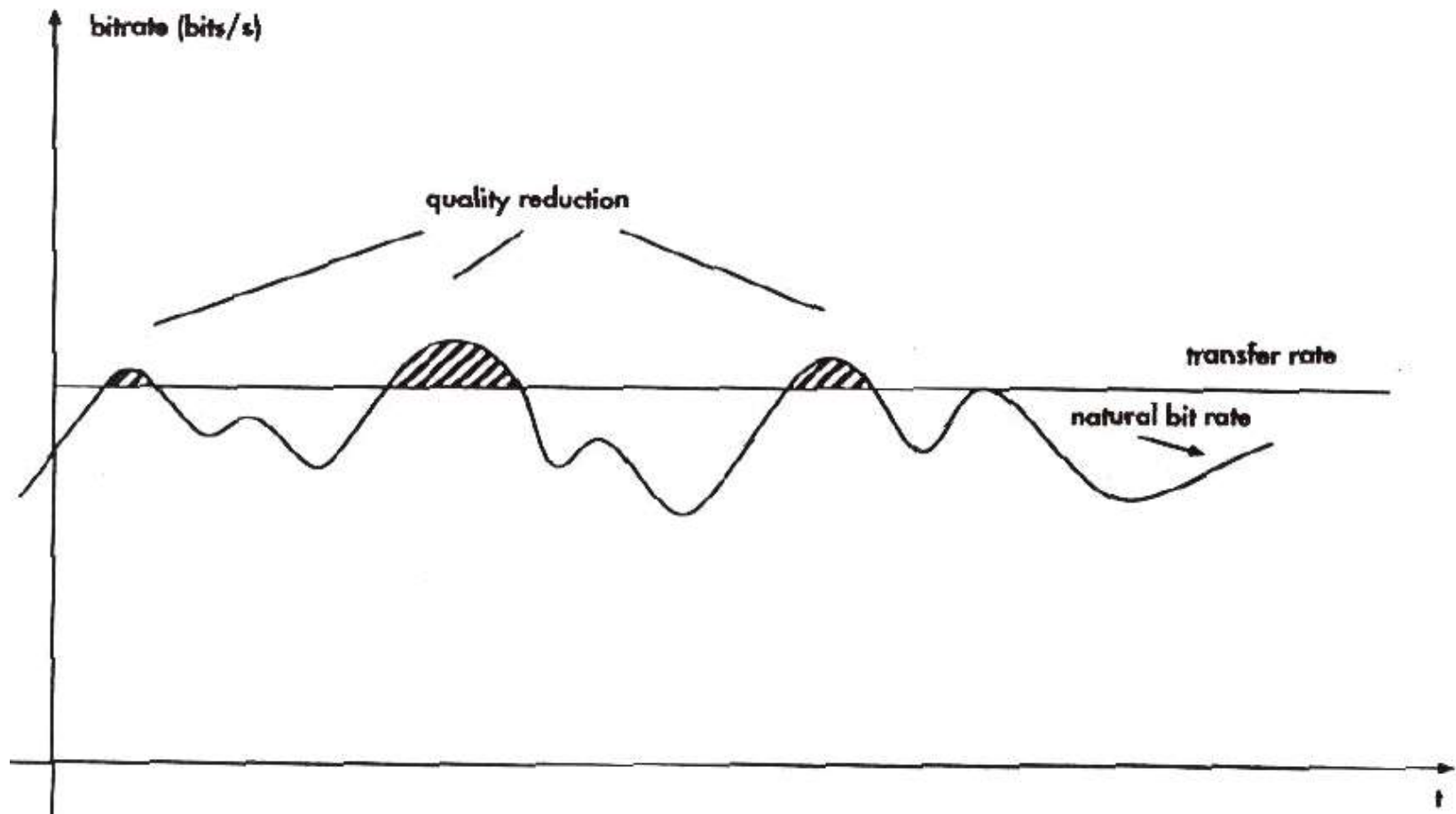
- 具有突发性的业务自然信息速率通常可以通过两个参数来描述和限定:

- 平均速率: $\bar{s} = E[s(t)]$

- 瞬间允许的最大突发量: F

- 通常要求用户传输速率应满足: $s(t) \leq \bar{s} \cdot t + F$

宽带通信网的业务特征评估---业务特征参数





宽带通信网的业务特征评估---业务特征参数

■ 传输速率分配

- 带宽：一般通过传输速率来衡量，带宽资源的动态优化分配是宽带网络的基本研究问题之一。
- 带宽分配方法
 - 按照峰值速率来分配带宽：延时最小但效率最低；
 - 按照平均速率来分配带宽：延时很大但效率最高；
 - 实际的带宽分配方法：取值介于平均速率与峰值速率之间。

宽带通信网的网络性能评估

---网络性能参数

■ 语义透明性

■ 时间透明性

宽带通信网的网络性能评估

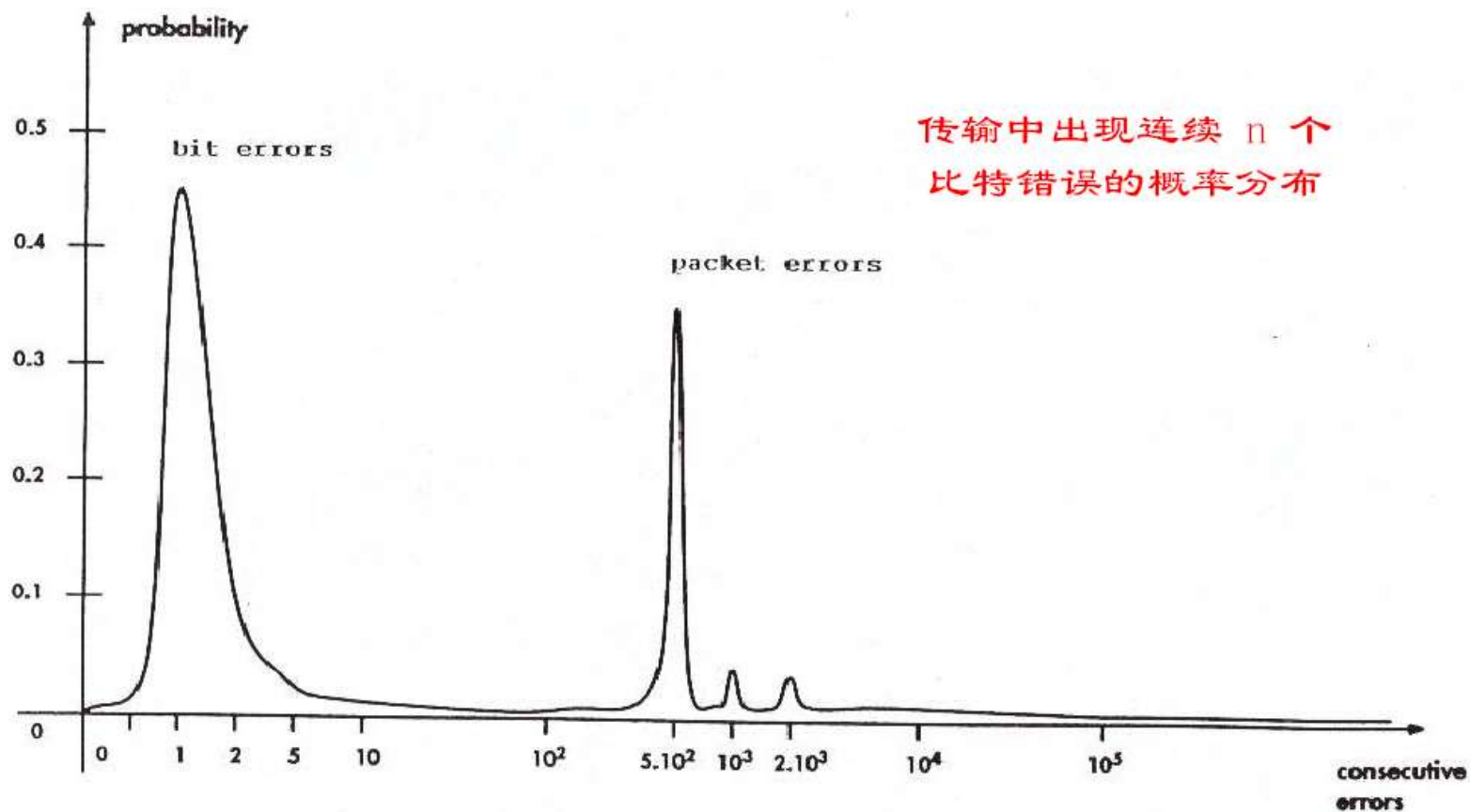
---网络性能参数

- 语义透明性：是指网络无差错传送信息的可能性；
 - 不同的场合、不同的业务类型对网络的语义透明性要求有所不同。
- 传送出错的原因
 - 信道畸变
 - 噪声和干扰
 - 交换/复用过程出错
 - 用户需求和网络可用资源不匹配

宽带通信网的网络性能评估

---网络性能参数（用例）

■ 语义透明性



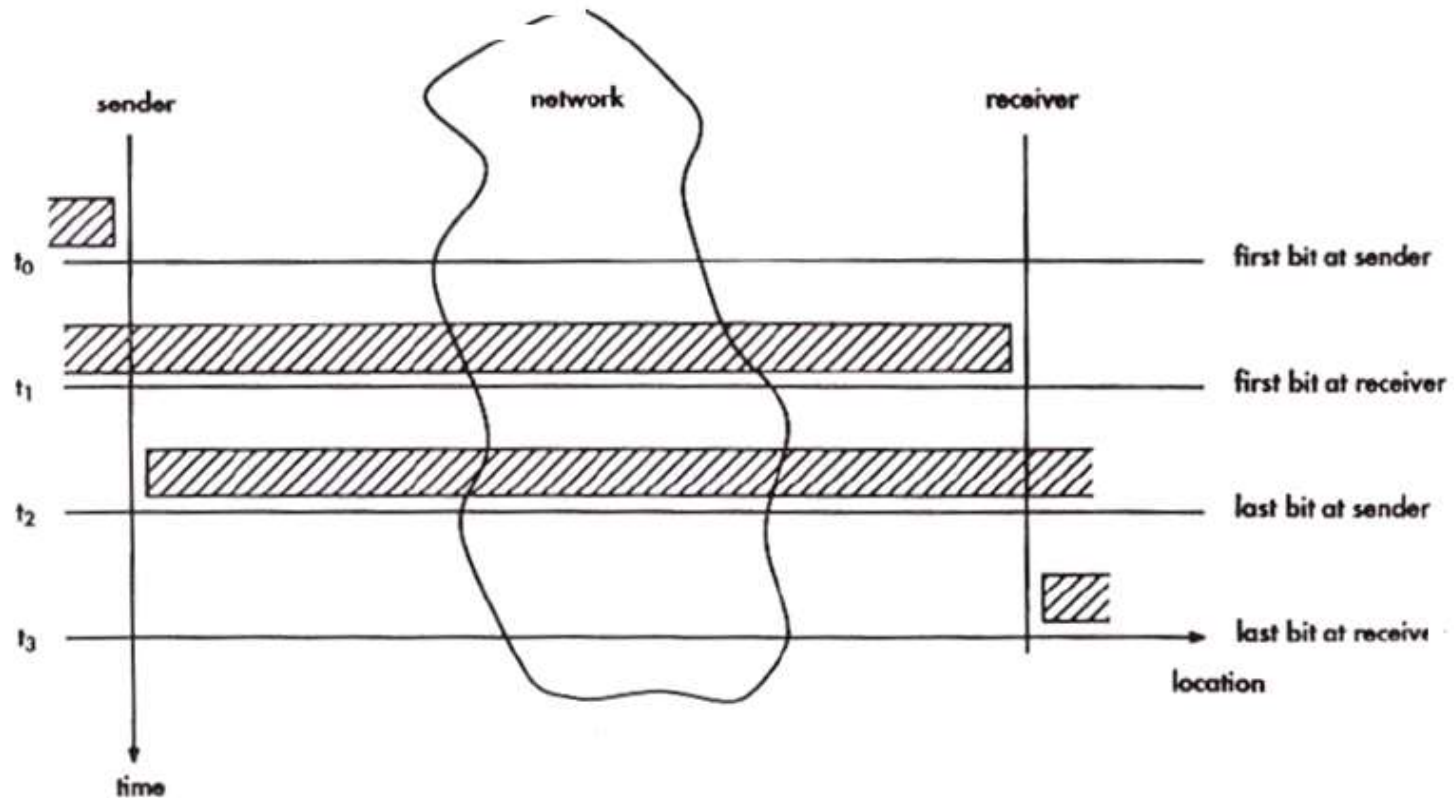
宽带通信网的网络性能评估

---网络性能参数（用例）

- 时间透明性：网络以最小时间将信息从源传送到目的地的能力。
- 有关时间透明性的主要参数（例子）
 - 最大时延；
 - 最小时延；
 - 平均时延；
 - 时延抖动：最大时延与最小时延之差。
- 分组网的时延也可用概率密度函数来表示。

宽带通信网的网络性能评估

---网络性能参数（用例）



$$D_f = t_1 - t_0, \quad D_l = t_3 - t_2, \quad \text{Delay jitter} = D_{\text{Max}} - D_{\text{min}}$$

通信网络中的时延

宽带通信网的网络性能评估

---网络性能参数

■ 语义透明性

■ 误码率

■ BER (误比特率)

■ PER (分组差错率)

■ PLR (分组丢失率)

■ PIR (分组误插率)

■

■ 时间透明性

■ 时延

■ 时延抖动

宽带通信网的网络性能评估---网络性能参数（例子）

■ 几种典型业务的传输差错率和时延

业务	BER	PLR	PIR	延迟
电话业务	10^{-7}	10^{-3}	10^{-3}	25ms/500ms
数据通信	10^{-7}	10^{-6}	10^{-6}	1000ms(50ms)
视频广播	10^{-6}	10^{-8}	10^{-8}	1000ms
高质量音频	10^{-5}	10^{-7}	10^{-7}	1000ms
远端过程控制	10^{-5}	10^{-3}	10^{-3}	1000ms



宽带通信网的网络性能评估---网络性能参数

改善网络性能的一些基本方法

■ 差错控制

- 检错重发：逐段链路进行操作(链路层),端到端进行操作(网络层);
- 前向纠错：在传输的信息比特流中插入的编码方式的冗余校验码。

■ 资源调度和流量控制

- 区分业务类别/等级;
- 资源预留;
- 接入控制/流量控制/拥塞控制。
- 数据流监管;

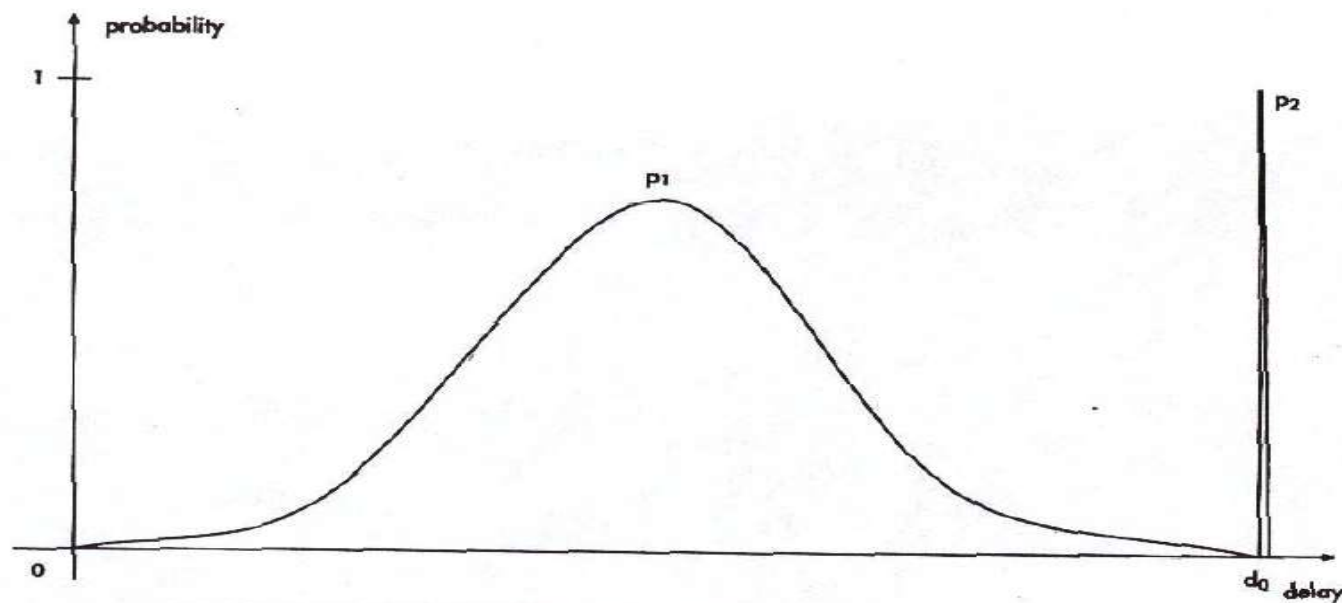
■ 流量整形

- 改变分组的大小和结构,例如采用固定长度分组(例如ATM信元)改善网络节点的功能。
- 改变分组流量的特性。

宽带通信网的网络性能评估---网络性能参数

改善网络性能的基本方法

- 提高时间透明性的措施：在接收端通过设置数据包存储来平滑延迟抖动。(VoIP)



P1: PROBABILITY DENSITY FUNCTION
OF DELAY OF THE NETWORK

P2: PROBABILITY DENSITY FUNCTION
AFTER CONDITIONING BY TERMINAL



思考题

- 简述宽带通信网络的核心网络技术的演变过程。
- 简述电路交换和包交换技术在工作机理上的主要差别？
- 分别简述语义透明性和时间透明性的定义。
- 评价宽带网络业务特征的主要业务参数有哪些？
- 评价宽带网络性能的主要标准有哪两类？分别列举其中包含的典型性能指标。