# 分布式系统的特征

分布式系统：组件分布在联网的计算机上，组件之间通过消息来进行通信与协调：就是很多个体组成一个整体，协调工作；任何一个系统都是分布式系统，只不过组件之间的通讯方式不同；组件的并发性（独立）、缺乏全局时钟（独立）、组件故障的独立性（单点依赖）。

分布式系统发展的关键趋势：现代网络的泛在（无所不在）特性、移动与无处不在的计算、分布式多媒体系统。

分布式系统的挑战：组件的异构性、开放性、安全性、可伸缩性、故障处理、组件的并发性、透明性与提供服务的质量的问题。

## 简介

书主要是解释影响系统设计与实现的联网的计算机的特征，给出可帮助完成设计与实现的主要概念与技术（解决方法：概念与技术）

分布式系统的特征：并发（共享资源）、缺乏全局时钟（消息协作的时序问题）、故障的独立性（隔离感知）。构建与使用分布式系统的主要动力是对共享资源的期望 （变化的与不变的）。

## 分布式系统的例子

所有基于网络的服务都是分布式系统。

* 金融与商业：电商与支付；
* 信息社会：互联网，web搜索与wiki等知识网站；
* 创意产业与娱乐：游戏、视频；
* 医疗与保健：医疗信息化与远程医疗；
* 教育：电子教育；
* 交通与物流：地图等；
* 科学；
* 环境管理：结合传感器技术。

### Web搜索

对万维网的内容建立索引，google使用分布式系统完成这个任务。

### 大型多人在线游戏

快速响应时间与事件通知机制，对共享世界的一致的视图；c/s架构。

### 金融交易

## 分布式系统的趋势

### 泛在联网和现代互联网

现代互联网是各种网络的集合，也是一个大型的分布式系统，设计好的互联网通信协议连接了各个网络，这些网络在生活中无处不在

### 1.3.2无处不在的计算

设备小型化，每个设备都是一个分布式的资源，连接到分布式系统；移动计算与物联网（无处不在的计算）。

### 分布式多媒体系统

### 把分布式计算作为一个公共设施

PaaS/IaaS/云计算等；

## 关注共享资源

共享资源包括硬件、服务、数据等；共享的方式包含2种：一种是各个使用方不相关，一种是使用方相关的（协作）。服务用于管理共享资源与提供访问；服务定义了资源上的操作；服务是通过管理资源的程序提供的，这个程序也叫做服务器；服务器接受客户端请求，这个方案是客户-服务器计算；当客户发送请求时，就称为调用服务气上的操作，一次完整的请求与响应的过程称为远程调用。

## 挑战

* 异构性：网络、计算机硬件、操作系统、编程语言；中间件是指一个软件层，提供了编程抽象，屏蔽了底层网络、硬件、操作系统与编程语言的异构性；中间件为服务器与分布式应用提供了一致的计算模型；移动代码：applet或者javascript；
* 开放性：系统可以被扩展，也可以被重新实现；开放的分布式系统允许增加硬件或者软件比如（分布式协议与规约等）而进行扩展；
* 安全性：机密性（防止泄漏给未授权的个人）、完整性（防止被改变或破坏）、可用性（访问资源的手段进行干扰），拒绝服务攻击（大量无意义的请求）；
* 可伸缩性：资源数量能满足不断增长的用户数量就是可伸缩的，需要控制物理资源的开销，控制性能损失，防止软件资源用尽，避免性能瓶颈；
* 故障处理：分布式系统的故障是独立的，涉及的难点有检测故障、掩盖故障（重传、使用副本）、容错、故障恢复、冗余（实现容错比如副本机制，多种选择）；
* 并发性：分布式系统的共享资源服务要并法的处理请求，需要进行并法控制；
* 透明性：透明性指对用户或者应用程序员屏蔽分布式系统的组件的分离性，使系统被认为是一个整体；而不是独立组件的集合，透明性包含访问透明性、位置透明性、并发透明性、复制透明性、故障透明性、移动透明性、性能透明性、伸缩透明性；
* 服务质量：可靠性、安全性与性能、适应性；

## 实例研究：万维网

Web是扩展与开放的，基于下面3个主要的技术组件：

* HTML 超文本描述语言，指定显示形式；
* 统一资源定位器URL；
* 客户-服务器系统体系结构；

HTML制定内容的布局方式与组织方式；URL定位资源，具有统一的格式；HTTP协议定义了浏览器与服务器的交互方式。

## 练习

* 1. 列出能被共享的五种类型的硬件资源和五种类型的数据与软件资源，并举出他们在实际的分布式系统中发生共享的例子？

答：打印机、磁盘、传感器、文件、web搜索、支付、数据库、服务;共享的例子忽略。

* 1. 不参考外部时间源的情况下，通过本地网络连接的2台计算机的时钟如何同步？什么因素限制了你描述过程的准确性，有互联网大量连接的计算机的时钟是如何同步的？讨论该过程的准确性？

答：通过消息同步的，不确定消息的格式；外部基准时间源同步。

* 1. 考虑1.2.2节讨论的大型多人在线游戏的实现策略，采用单一服务器方法表示多个游戏玩家的游戏状态的好处是什么？这存在什么问题以及如何解决？

答：能够获得一致性的地图；单机负载过高，必须使用集群结构做分解。这种结构需要设计集群之间的通信机制。

* 1. 一个用户随身携带可以无线联网的PDA，来到一个从没有到过的火车站，请给出建议，在用户不输入火车站的名称与属性的前提下，如何得到关于本地服务和火车站环境的情况？要解决哪些技术问题？

答：

# 系统模型

分布式系统3种模型：

* 物理模型：硬件设备与连通方式；
* 体系结构模型：基于角色来思考分布式系统（客户-服务器或者对等模型）；
* 基础模型：抽象的观点描述分布式系统的问题。

## 2.1 简介

## 2.2 物理模型

物理模型是从计算机和所用网络技术的特定细节中抽象出来的分布式系统地层硬件元素的表示。

基线物理模型：计算机设别结点

* + 早期的分布式系统：局域网；
  + 互联网规模的分布式系统：网络的网络，异构性增加，中间件出现；
  + 当代的分布式网络：移动互联网与物联网与云计算。

系统的分布式系统也就是系统的系统。

## 2.3 体系结构模型

系统的体系结构指的是系统的组件以及组件之间的关系；目标是确保结构能满足将来的需求，主要关心的是系统的可靠性、可管理性、适应性、与性价比。

### 2.3.1 体系结构的元素

* 通信实体：系统角度是进程，也可以是计算物理结点，也可以是线程；从编程角度看是对象（分布式对象是面向对象的思维）、组件（面向问题的实体）、web服务（基于互联网协议的行为封装）；
* 通信泛型：进程间通信，就是简单的2台机器的基于互联网协议的进程间通信；远程调用，双向交换的技术，基于的是请求-应答的通信模式，是客户-服务器计算的结构的主要通信模式，分为远程过程调用与远程方法调用，远程过程调用（RPC），是从组件的角度来说的，从对象的角度来书就是远程方法调用（RMI），分布式对象之间的方法调用。上述的通信模型都是同步的，双方必须同时存在，通过第三方实体，允许发送者与接受者解耦合，达到空间解耦合（不知道发给谁）与时间解耦合（不需要同时存在），这个也叫做间接通信，间接通信的关键技术是：组通信，一对多通信，通过组标识符发送给组，组管理组的成员。发布-订阅系统（分布式基于事件的系统）一对多。消息队列，点对点服务。元组空间，结构化数据放入持久空间，消费者处理特定模式的元组。分布式共享内存（DSM）。
* 角色与责任：基于角色，分布式系统体系结构风格有客户-服务器风格与对等风格；客户-服务器是最重要也是最简单有效的体系结构；对等结构是所有进程都是相同的，扮演同样的角色，不区分客户或者服务器；相比于客户-服务器（访问的资源与服务受限于服务器的能力），对等体系解决了这个问题，每个拥有资源与服务的机器都参与计算；对等应用的形式有分解（每个进程有一部分数据）与冗余（增加副本，分散负载，容灾配置）2种；服务与数据如何放置到真实的物理结点涉及到分布式系统的性能、可靠性与安全性；放置策略：1）将服务映射到多个服务器，可以集中式（一台服务器），也可以是分布式的（服务分解），集群（副本）2）缓存，用于存储最近使用的对象；3)移动代码；4）移动代理。

### 2.3.2 体系结构模式

1.分层体系结构

分层与抽象紧密相关，上层依赖下层，每层都是一个软件抽象，更高的层对细节透明。分布式系统的分层相当于把服务垂直组织成服务层；分层体系结构的常规视图：



中间件是为了屏蔽异构性，给应用程序员提供方便的编程模型，中间件也是计算机上的进程或者对象，可以是软件的基础构造块，用来抽象底层操作，比如远程方法调用、通信、事件通知等。

1. 层次化的体系结构

层次化与分层其实差不多，不过这个层次化指的是对某个分层进行层次化，就是再次分层，比如说应用层的MVC模型。

1. 瘦客户

将复杂性从最终用户设备移向互联网服务；

1. 代理

主要用于支持远程过程调用与远程方法调用透明性。

1. 业务代理（注册中心）
2. 反射。

### 2.3.3 相关的中间件解决方案

中间间用来提供一个抽象层，主要是为了屏蔽底层基础设施的异构性；提升互操作性与移植性；中间件的类别有：

|  |  |  |
| --- | --- | --- |
| 主类 | 子类 | 系统例子 |
| 分布式对象 | 标准 | RM-ODP |
| 平台 | CPRBA,RMI |
| 分布式组件 | 轻量级组件 | Fractal、OpenCOM |
| 应用服务器 | SUN EJB、JBoss |
| 发布-订阅系统 |  | JMS、Scribe,CORBA事件服务 |
| 消息队列 |  | WebsphereMQ、JMS |
| Web服务 | Web服务 | Apache Axis |
| 网格服务 | Globus Toolkit |
| 对等 | 路有覆盖网 | Pastry、Tapestry |
| 应用年特定的 | Squirrel、Ivy、Gnutella |

中间件在分布式系统中完成通信与数据共享，但是服务的可靠性还是需要应用层面作处理；实际就是中间件与上层服务的功能划分，比较困难，因为服务的正确性是依赖于上层服务的应用逻辑。

## 2.4 基础模型（这里没有理解）

分布式系统的设计的要求：进程与网络的高性能、高可靠性以及系统中资源的安全。

### 2.4.1 交互模型

分布式系统的进程要交互，相比于单个进程，分布式进程多了消息传递，不能很好的控制每个机器上的操作的执行时序，单机可以控制自己的进程的执行时序；影响进程交互的2个因素：

* 通信性能是一个限制特性；
* 不可能维护一个全局时间概念；

通信会有延迟（数据发送返回、网络堵塞、进程处理）、带宽（可以使用的共享带宽）、抖动（消息传递时间不稳定）。

通信涉及到的本地时钟因为时钟漂移的原因，可能影响时序。

同步分布式系统：进程执行时间有限制、消息传递时间有限制、本地时钟的偏移在给定的范围内；

异步分布式系统：对上述的延迟时间没有限制，时钟偏移也没有限制；典型的就是互联网。

分布式系统的分布式事件排序，比如2个分布式进程的事件发生顺序。

### 2.4.2 故障模型

* + 遗漏故障： 进程与通信信道故障；
  + 随机故障：随时可能出现的错误；
  + 时序故障：里面的所有的处理时间等故障；
  + 故障屏蔽：

### 2.4.3 安全模型

保证进程、通信信道、进程里面的对象数据的安全就是分布式系统的安全；

# 网络与网际互连

## 3.1 简介

分布式系统的网络需要：

* 传输介质；
* 硬件设备；
* 软件组件。

未分布式系统提供通信功能的所有组件叫做通信子系统，使用通信子系统通信的叫做主机，主机+交换机=结点。

分布式系统的联网问题：

* + 性能；
  + 可伸缩性；
  + 可靠性；
  + 安全性；
  + 移动性；
  + 服务质量；
  + 组播。

## 3.2 网络类型

* 个域网：
* 局域网；
* 广域网；
* 城域网；
* 无线局域网；
* 无线城域网；
* 无线广域网；
* 互联网络；
* 网络错误。

## 3.3 网络原理

包交换技术，不同于电路交换技术。

1. 数据包的传输：信息分割为小的数据包，是为了满足网络中计算机的缓冲空间要求，避免数据太大需要等待通信信道的时间过长；
2. 数据流协议，因为数据流的特点，对带宽与延迟有较高的要求，需要预先分配足够的资源，RSVP协议与RTP协议；
3. 交换模式：广播（以太网，蜂窝网络）、电路交换（传统的电话交换网络）、包交换（存储转发网络）、帧中继（针对包交换需要存储然后转发造成延迟高的特点，包更小（帧），不存储帧，只根据头转发，形成了一种流）式的结构；
4. 协议：进程间通信需要用到的规则与格式，协议包含2个部分：消息的顺序与消息的格式，协议层与协议栈，因为网络是分层处理，每层有很多的协议，OSI 7层协议栈：

* 应用层：满足特定应用的通信需求定义的，比如HTTP，  
  FTP，STMP；
* 表示层：数据转换，比如加密（TLS）；
* 会话层：可靠性与适应性，故障检测与自动恢复（SIP）；
* 传输层：处理消息的最低一层与端口相关联，比如TCP、UDP；
* 网络层：在网络路径中传输数据包比如IP、ATM虚电路；
* 数据链路层：有直接物理连接的2个结点间传输数据包，比如：MAC、PPP、ATM;
* 物理层：驱动网络的硬件与电路或者光纤或者无线网络电磁信号。

1. 数据包组装，因为网络中的存储要对包的大小的要求，这个就是MTU(Maximum Transfer Unit),端口是计算机中进程的目的点，0-1023 是公共端口，其他是个人端口；
2. 数据包传递：2种方法传递数据包：

* 数据报报包传递：发送后不管；
* 虚电路包传递：建立虚电路路由路径，包中保存虚电路号，在中间结点通过虚电路号转发。

1. 路由：自适应路由，路由信息会根据网络的情况定期的变更；路由算法包含2个部分：决定路由路径与根据网络情况或者故障动态变更信息；
2. 距离向量算法或者叫做Bellman-Ford算法，Router Infomation Protocol 路由信息协议，相邻的路由器互相交换路由信息，新增或者更新路由表信息，故障路由也会被检测到，最终替换故障路由的路径；
3. 网络的能力受通信链路性能与交换结点性能的限制，当链路与结点的负载接近极限时，发送的主机就会建立队列，当队里持续增长，超过缓冲区，就会发生数据包丢失，拥塞控制，通知路由上的结点，缓存数据包或者端到端的流量控制，通过额外的拥塞信息，降低发送的速率；
4. 路由器、网桥、集线器、交换机、隧道；

## 3.4 互联网协议

* 1. IP寻址：互联网中的每个主机都有一个IP地址，分为2个部分，子网编号与主机编号；分为A，B，C，D（组播），E（保留） 几种类别，具体可以参考网上的一些说明，3类可以使用的地址分别用于不同的用途，主机标示部分全为0代表本机，全为1代表广播消息，广播给子网内的全部主机；地址不够的3种方案：1.IPv6，2.无等级域间路由，3.NAT；
  2. IP协议：IP数据报包含头部与数据2部分，并且提供了头部的校验和，IP报文没有传输的保证，可能丢失，重复，错乱等，组装好后，放入数据链路层，数据链路层需要根据IP得到下一跳路由器的MAC地址（底层网络使用的地址），地址解析协议（ARP），IP伪冒，就是填入虚假的发送方地址；
  3. IP路由：主干（自治系统的概念），路由协议，链路-状态算法，解决路由器路由表过大的2种方案：1.IP地址人为设定地域，2.默认路由；网关路由器有个互联网地址，每个地址都对应一个网络；
  4. CIDR，分配连续的C类地址或者分割B类地址，增加掩码域，用来任意的分割IP地址中的子网地址与主机地址，相比分类地址更为灵活；
  5. NAT，不需要注册，直接访问网络，NAT算法的寻址过程：1.路由器接到内部计算机发送的TCP或者UDP，根据IP与端口号生成一个翻译表项，2.路由器使用一个虚拟端口号，指向翻译表项，并用它与路由器的IP地址替换包中的源端口与IP地址，3.当接到返回的包时，根据目的端口号查找翻译表项，然后替换包中的IP地址与端口号，然后转发到内部网；
  6. IPv6：IP地址使用128比特表示。
  7. 移动IP：移动计算机访问互联网，如果只是客户端，不需要保留固定的IP地址，移动时接入不同的子网就行，子网使用DHCP(动态主机配置协议)分配IP地址，但是如果是打电话这种，必须保持，手机的IP地址是固定的，才能接收到别人打过来的电话，这时需要使用移动IP的解决方案，在所有的子网中，有固定的IP地址，剩下的没看懂；
  8. TCP与UDP：端口号，16位整数，UDP不可靠传输，但是开销最小，TCP需要额外的信息保证是可靠传输，可靠传输的机制有：1.排序，TCP报文分割后有序号，2.流控制，比如确认机制、窗口控制、重传机制、缓冲与校验和。
  9. 域名：域名解析。什么是域？什么是域名。
  10. 防火墙：一组进程，用于控制外网与内部网的通信，用于服务控制、行为控制、用户控制。

## 3.5 实例研究：以太网、WiFi、蓝牙

### 3.5.1 以太网

802.3标准，类似一个有分支的总线连接，通过集线器或者中继器连接为一个以太网，通过交换机或者网桥连接多个以太网，以太网使用“具有冲突检测的载波监听多路访问”的机制传输数据，也是一种竞争总线的策略。

### 3.5.2 IEEE 802.11 无线LAN

也叫作WIFI，扩展了802.3的载波监听与多路复用协议；通过基站接入有线网络叫做基于基础设置的无线网络，还有没有基站的叫做自组织网络，802.11标准使用无线电传输信号，有多个版本，每个版本工作在不同的频率上，可以互相独立组成无线网，并且也是用MAC（媒体访问控制协议），由于无线点信号传输的特殊性，可能会造成子网内：1.工作站隐蔽（2个主机离的太远，检测不到对方），2.信号衰减，3.冲突屏蔽（因为本地的信号用于比远程传过来的好，这样一对比，主机就看不到冲突了）；CSMA/CD加入时隙保留机制变成CSMA/CA协议；主要的机制就是加锁-完成-通知的一种逻辑，这整个的过程中，时隙代表约定的加锁的时间；就是发送方要发送信息时，广播一个RTS帧，代表请求时隙，告诉别的主机，信道我要用，你们都别用，然后别的主机回复一个CTS帧，代表，你加锁成功，然后传输完成，基站通知所有的主机信道空闲了。

### 3.5.3 IEEE 802.15.1 蓝牙无线PAN

无线个域网，名字取自与一个海盗的名字，蓝牙比无线网带宽少，能耗小，用于低开销传输设备。

# 进程间通信

## 4.1 简介



UDP应用程序接口提供了消息传递抽象；TCP应用程序接口提供了进程对之间的双向流抽象。

## 4.2 互联网协议的API

### 4.2.1 进程间通信的特征

每个消息目的地都维持了一个队列，发送者就是生产者，接收者就是消费者，send/receive是2个通信原语，通信也可以是异步或者同步的：

* 同步：发送进程与接收进程在每个消息上同步（必须关联发生，在这期间不能干别的，必须都完成了，都才能向下走，这就是同步），send/receive都是阻塞操作，发送者在send后，一直阻塞直到接收端receive并正确处理，receive的消息处理完后，服务端也会一直阻塞等待新的消息；
* 异步：send操作是非阻塞的，消息被复制到缓冲区，发送者去干别的，生产与发送是同时进行的，异步的意思就是（不管你干没干完，我这边不用管，我继续干我的），receive操作可以是阻塞的，与同步的形式一样，也可以是非阻塞的，那就是接收进程receive后复制到缓冲区，继续receive，不用等处理完，接收这需要轮训或者定期的检查事件去读取消息。
* 每个消息都有消息目的地，里面含有端口；
* 可靠性；
* 排序。

### 4.2.2 套接字

TCP/UDP都使用套接字抽象，套接字是进程通信的端点的抽象，起源于UNIX，每个套接字绑定到<IP,PORT>对；java语言中，InetAddress抽象表示互联网地址。

### 4.2.3 UDP数据报通信

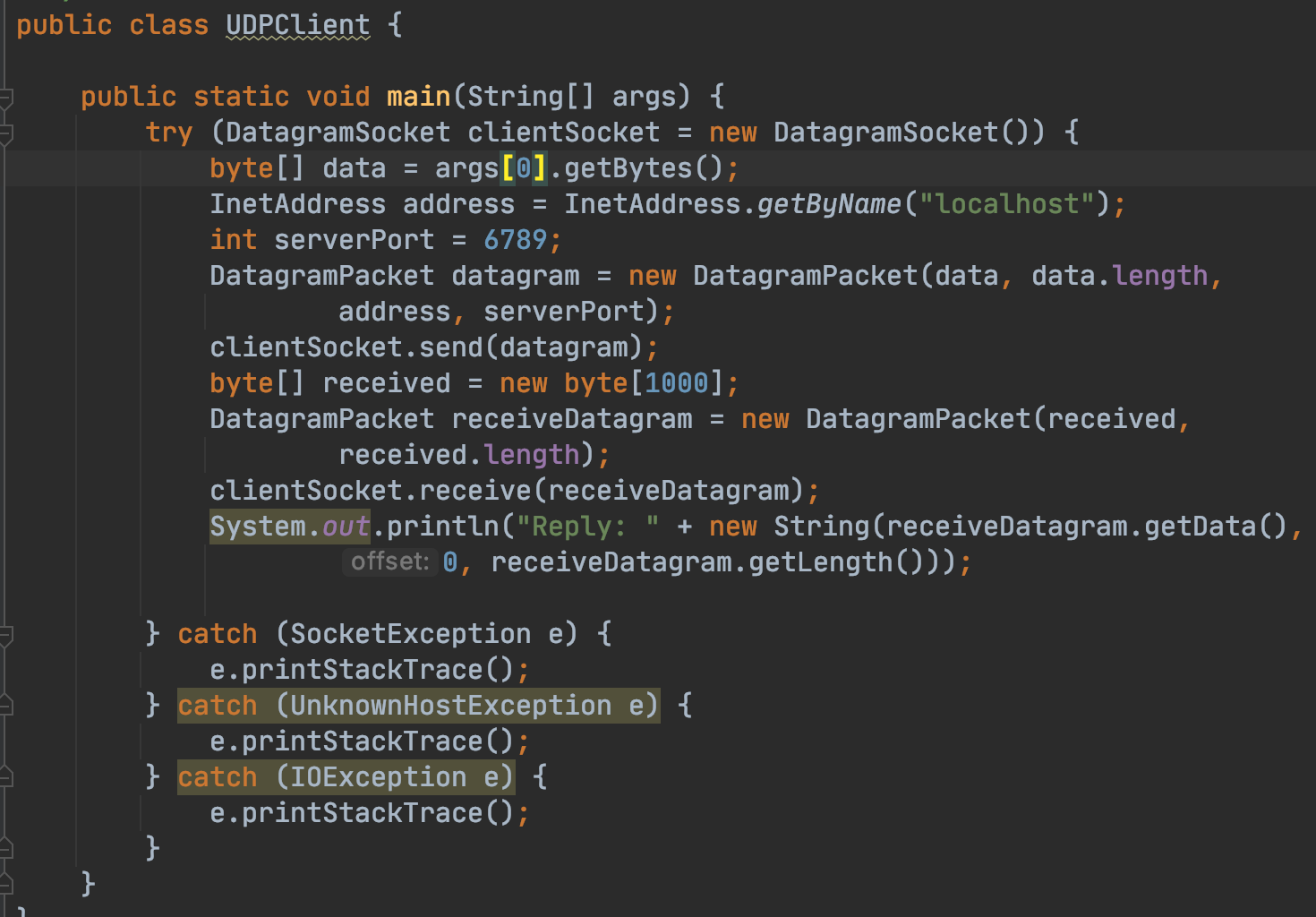
这种通信模式只是简单的发送IP数据报，但是通信也有一些注意点：

* + 消息大小：接收进程对接收消息的大小有限制，超出则会截断；
  + 阻塞：send是非阻塞，复制到缓冲区，不管发没发送，receive一般是阻塞，等到消息，到来后，送入单独的线程中处理；
  + 超时：设定receive的等待时间；
  + 任意接收：可以限制接收的来源根据IP与端口号。

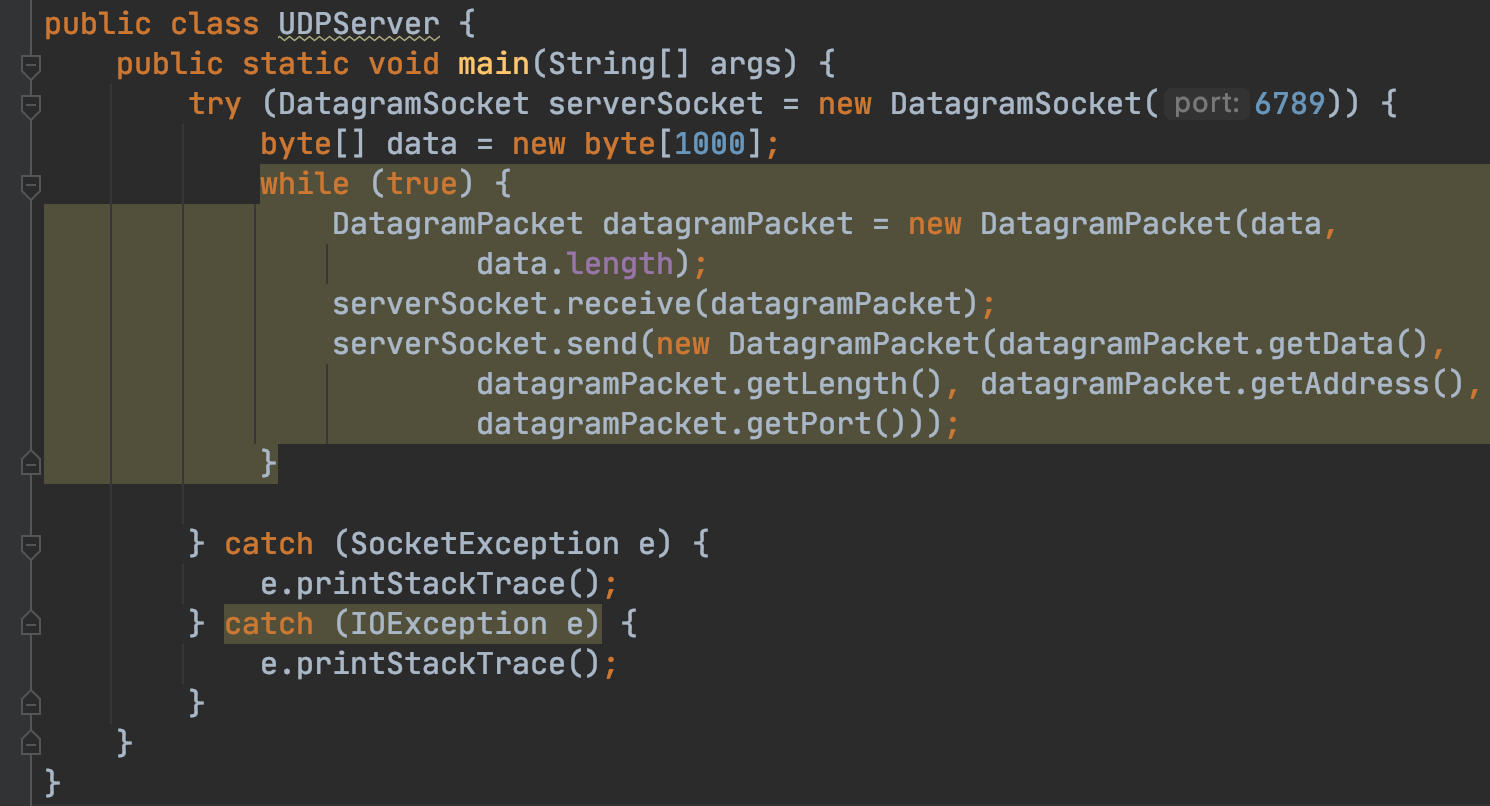
UDP数据报故障模型：

* 遗漏故障：也就是消息丢失；
* 乱序；

java中对UDP的抽象类是DatagramSocket与DatagramPacket。



服务端的设置如下：



### 4.2.4 TCP流通信

TCP提供了一种网络通信中的流抽象，流的特点是可靠双向传输，底层的保证机制有：

* + 消息大小可以通过分片的机制变成任意的，可以随时发送；
  + 消息重传机制，或者滑动窗口机制保证消息不会丢失；
  + 消息传送的速度控制；
  + 消息重复丢弃与消息编号保证顺序；
  + 只需要建立一次连接开销；

套接字有点消息队列的意思，发送队列与接收队列，好好想象；每个套接字都有一个输入流一个输出流，分别用于发送与接收；close套接字的时候，关闭输出流（记住没有关闭输入流），并在最后发送的数据末尾添加EOF，指明流已经结束。