

message router design

weizhou¹

2018 年 11 月 15 日

¹awokezhou@gmail.com

目录

第一章 引言	2
第二章 message route 消息处理流程	4
2.1 router init	4
2.2 router running time	5

第一章 引言

在 Linux 应用层开发过程中，通常以进程的方式完成某项特定功能。系统中会同时运行多个进程，而根据业务的需要，进程间需要相互通信几乎是必须的。Linux 为进程间通信提供了几种原始的方法：管道和命名管道、信号、信号量和 **unix socket**。多进程间传递消息经常被使用的是 **unix socket**，其他几种方式从设计理念和目的上不适合用来进行多进程间大量消息的传输。

要使用 **unix socket**，每个进程必须创建一个 **socket** 绑定一个文件作为本地通信的地址，如图 1.1 所示。

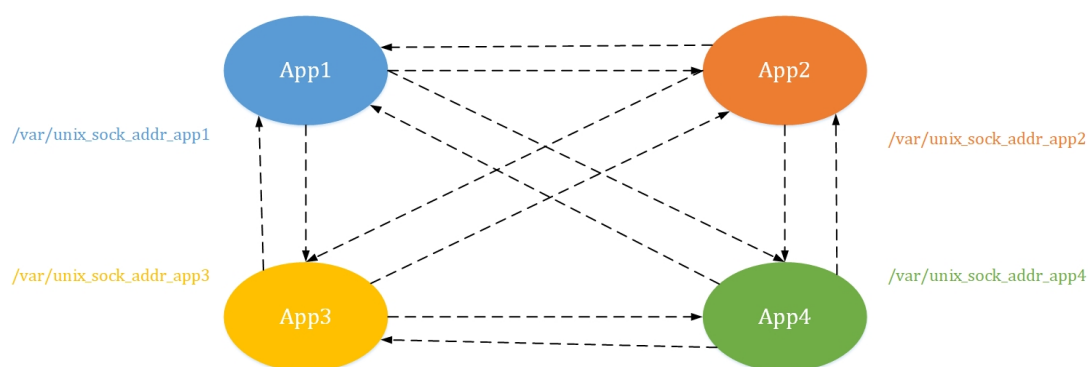


图 1.1: 多进程使用 **unix socket** 通信

如果多个应用要进行复杂的交互，那么需要为此创建多个地址文件，这虽然对于 Linux 系统来说不是什么难题，但是这样的通信模型对于应用软件的设计、开发和维护来说，都太过复杂和不必要。另外，如果某个应用想要想其他多个应用发送类似组播或者广播消息，这种通信结构就更加难以维护。

基于以上的问题和需求，提出消息路由 (**message route**) 的概念，即创建一个消息中转站，类似通信中的路由器，能够对消息做转发，所有进程间的消息通信，都统一发送到中转站，再由中转站根据消息的目的地址发往目的进程。如图 1.2 所示。

消息路由的设计仅限于本地通信的各个进程之间，并不包括远程服务和代理，其设计属于集中式结构，而不是分布式结构，需要一个专门用于提供中转服务的进程 **router** 来做消息转发，定义一个公共库 **librtmsg** 来规范各个进程的消息格式、发送和接收处理。而各个进程的消息通信 ID 必须预先在 **router** 中固定编写，不支持进程上线后临时上报，即未在 **router** 中记录的进程无法使用 **message route** 来转发消息。

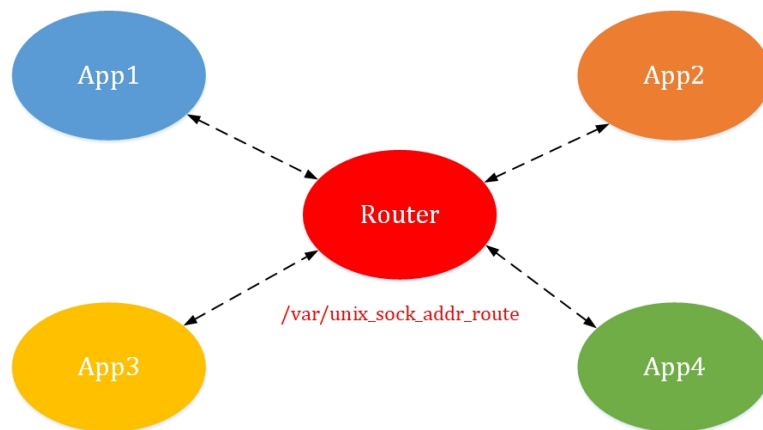


图 1.2: message route

本文章节安排如下：

1. 第二章对整个 message route 的处理流程进行描述，包括 router 如何初始化、进程与 router 的交互、router 的消息转发。
2. 第三章对消息格式定义以及 librtmsg 需要提供的公共 API 进行说明。

第二章 message route 消息处理流程

2.1 router init

首先有一个非常关键的点，router 的启动必须先于需要使用 message route 机制转发消息的其他进程，因为消息转发的核心在于 unix socket 的建立，如果 unix socket 还未建立之前其他进程就已经开始尝试转发消息，这是不好的设计，也是不安全的 (导致进程阻塞)。当然你也可以将其他进程 connection 动作设置为非阻塞的，但是这样的设计从根本上是劣质的。创建 unix socket 的任务理所应当放在了 router 中完成。

要进行消息转发，需要在中转站上存放一个中转表，就像交换机上有 MAC Table，路由器上有 Route Table 一样，router 中也需要一个转发表 (forward table, fdt)。想象一个最简单的模型，当一个进程 App1 要向进程 App2 发送消息时，App1 的消息中必定包含了发往进程的标识 dst id，此消息发送到 router 后，router 将根据这个 dst id 在转发表 fdt 中查找，获取对应进程的消息句柄 msg handle，以该句柄向 App2 发送消息。如图 2.1 所示。

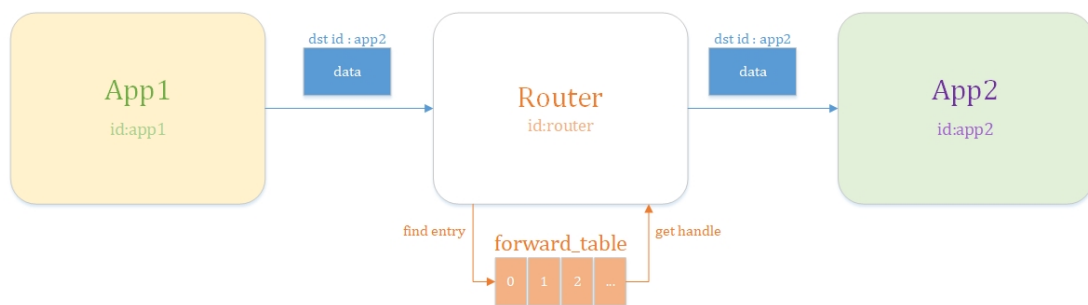


图 2.1: 两个进程消息通信

那么转发表如何生成呢？其中的每一个表项存放什么内容呢？目前转发表固定写死在 librtmsg 库中，以静态数组呈现，router 初始化时将其中的每一个表项添加到自己的转发表中。以上文的分析，每个表项至少应该存放进程标识 id，和消息句柄 msg handle。

因此，router 的初始化关键的两个步骤：

- 创建 unix socket
- 初始化转发表

2.2 router running time

当 router 以上述方式初始化完成后, 则在 unix socket 上监听消息, 但是此时如果直接进行消息转发, 则存在以下两个问题:

- 如果消息转发的目的 App 未启动, 则会转发失败
- 初始消息转发的目的 App 未和 router 建立连接, router 没有获取到 App 的消息句柄, 转发也会失败

以上问题总结来说, 就是只有消息通信 App 双方都已经启动, 并且与 router 建立了连接, 才可以进行消息转发, 那么如何解决呢? router 需要对进程的状态进行记录和管理, 转发表的表项中需要对 App 的状态有所记录, 当 App 与 router 建立连接后, 转发表中对应此 App 的表项将 App 状态变为可转发的, 然后将此 App 的消息句柄记录。当 App 的状态得到记录, 并且消息句柄存在时, 就可以进行正常的转发了。router 根据消息结构中的目的标识将此消息转发给目的 App。

router 运行时处理流程如图 2.2 所示。

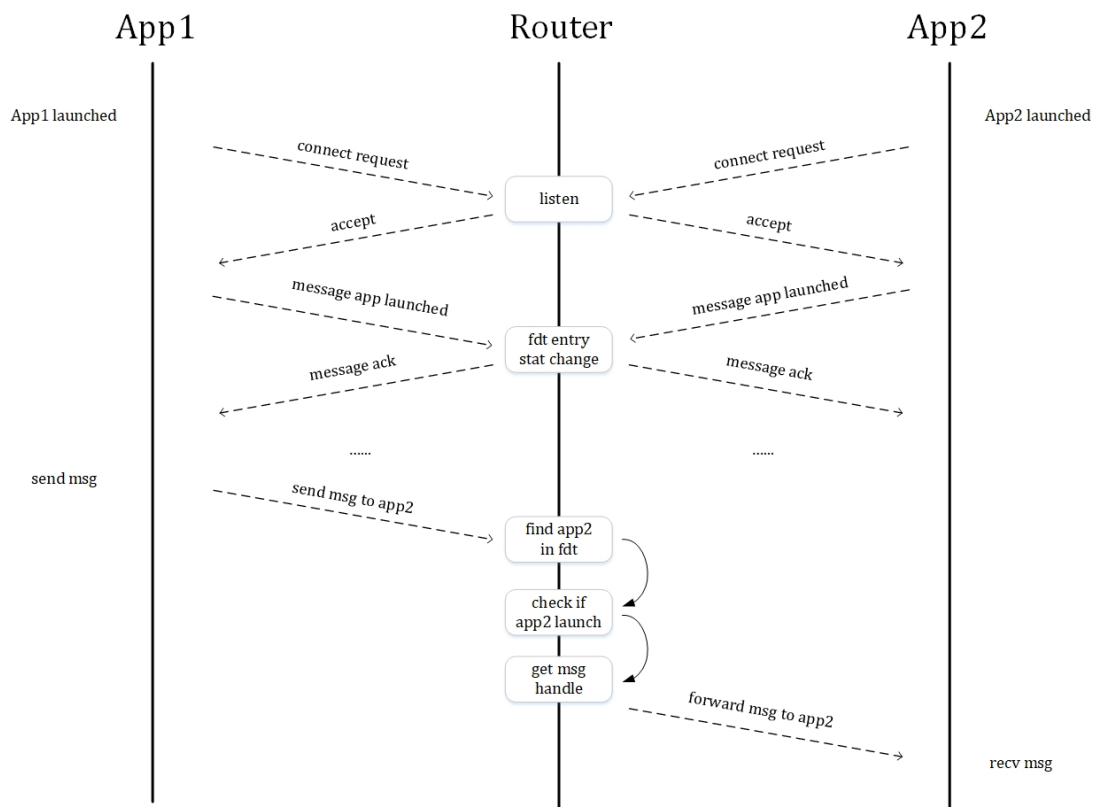


图 2.2: 两个进程消息通信