Net流处理NetSvc模块

此模块主要提供了封装好的网络服务供各个模块使用。此模块通过其类实例以及发布的接口和其余模块之间进行交互。同时因为我们通过字节流进行传输，每个模块还需要定义其自己传输数据的字节流解析器从而对接收到的数据进行解码提取。

对于解析器，每个模块都应定义其自己的解析器类MsgParser, 并在自己的模块CB中对其实例化。该抽象类提供push()，get()，decode()三个纯虚接口，分别对应将网络层接收到的字节流放入对应的缓存区，将填满的缓存区中的数据提取出，以及将收到的消息进行解码三个功能。每个模块需自己实现其功能，如需要也可添加其他成员函数。

对于通讯服务模块本身来说其包含一个管理组件类NetSvcManager和一个通讯服务提供类NetSvcItem。管理组件类主要负责管理实际服务提供对象，例如在其他模块需要通讯服务时创建并提供服务提供对象。在该服务不再被需要时负责释放通讯服务提供对象。其主要对外接口包含

createNetSvcItem —> 更具体的说，这个接口的实现方式会根据传入的NetMsgHandler句柄是否为空进行区分。对于传入为句柄的调用，最终所有的数据交换操作会通过句柄进行。对于传入句柄为空的调用，所有的数据会直接被推送到该EDU的message queue中。同时传入参数还应包括已经生成并初始化完成的MsgPaser对象，MsgRouter对象。该函数最终会返回一个NetSvcItem对象指针。

freeNetSvcItem (NetSvcItem) —> 清空数据并释放对象

通讯服务提供类NetSvcItem实例化的对象是实际上为各个模块提供通讯服务接口的抽象层。
其对外提供的接口主要包括通讯功能:

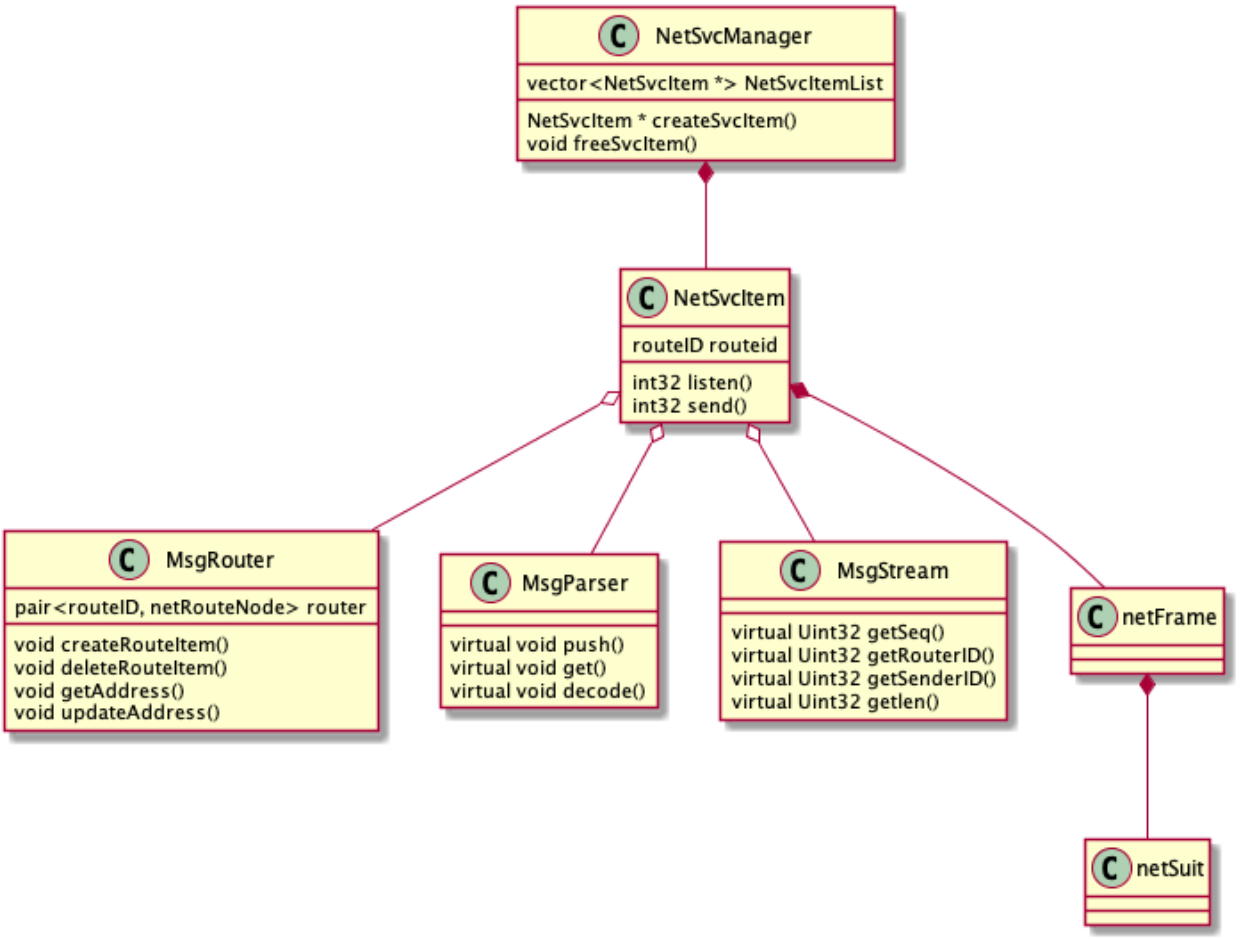
listen（）, 当一个模块需要监听一个固定端口并根据传入的连接请求进行连接。
send(MsgStream * pMsg)，同步发送

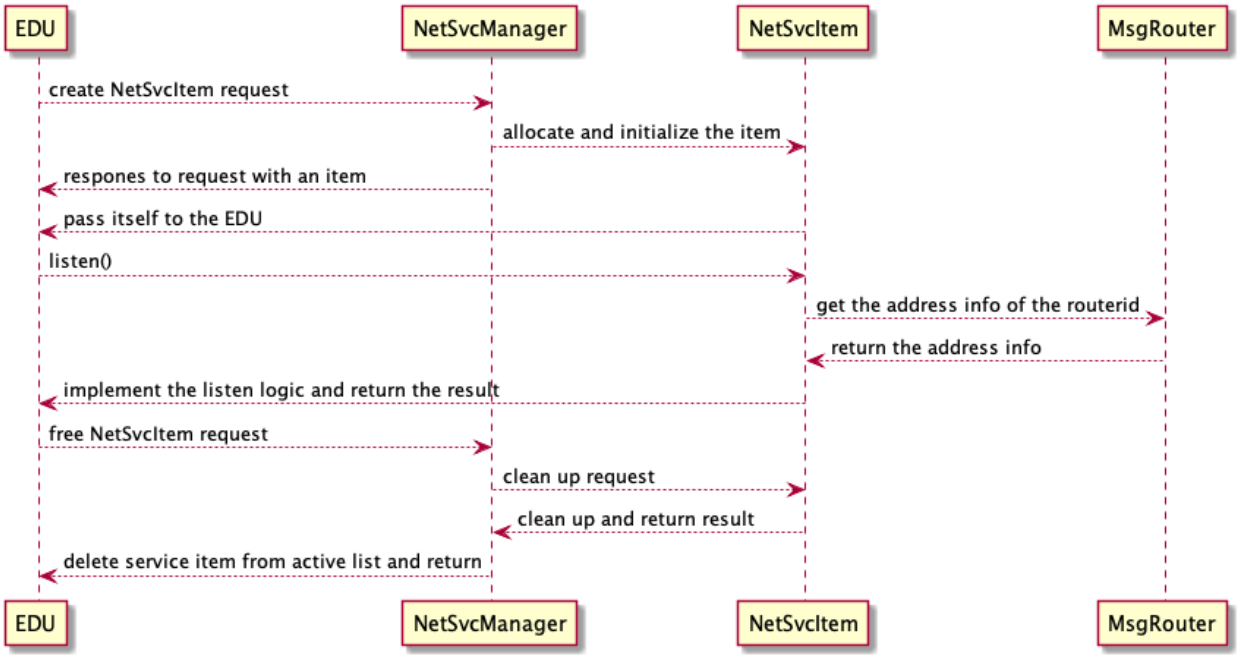
关于MsgStream这个结构，其是一个抽象的信息结构类，提供一组纯虚接口来获得相应的seq number，routerID，senderID，msg length等信息。除了实现这些接口，每个模块也可以根据其具体业务需求添加相应的成员函数。

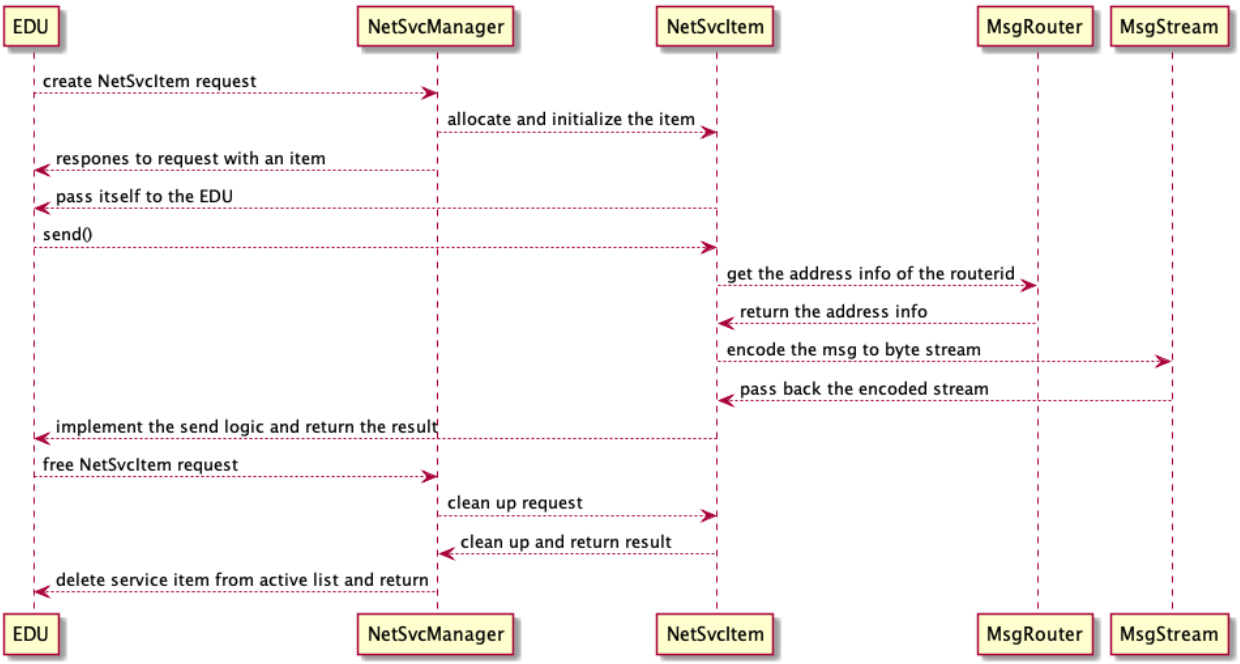
同时我们还需要一个msgRouter模块来对通讯地址进行转换与封装。业务逻辑只需要将通讯地址的host name和所需服务的service name传入，该模块便会对应生成一组routeID与其地址组成的pair。其提供createRouteItem，deleteRouteItem，getAddress() 和 updateAddress()功能接口。

当某个模块需要网络服务时，其首先应通过调用createNetSvcItem来创建一个实际服务提供对象。调用该接口时需要传入<句柄/ID, MsgPaser，MsgRouter>。当对象创建完成之后会返回一个

对象指针给调用者。接下来的业务逻辑可以直接通过调用对象所提供的接口进行网路通讯。
当需要监听的时候，应调用listen接口进行监听。当需要发送信息时需调用send接口并传入所需发送的内容MsgStream对象。






EDU
NetSvcManager
NetSvcItem
MsgRouter
MsgStream
create NetSvcItem request
allocate and initialize the item
respones to request with an item
pass itself to the EDU
send()
get the address info of the routerid
return the address info
encode the msg to byte stream
pass back the encoded stream
implement the send logic and return the result
free NetSvcItem request
clean up request
clean up and return result
delete service item from active list and return
EDU
NetSvcManager
NetSvcItem
MsgRouter
MsgStream