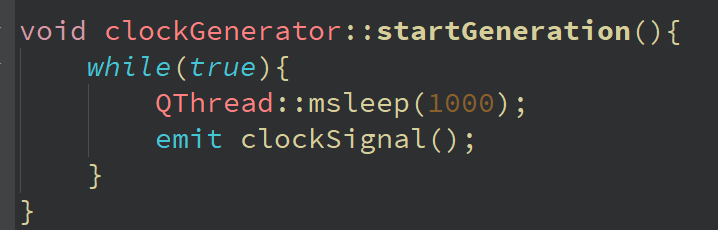
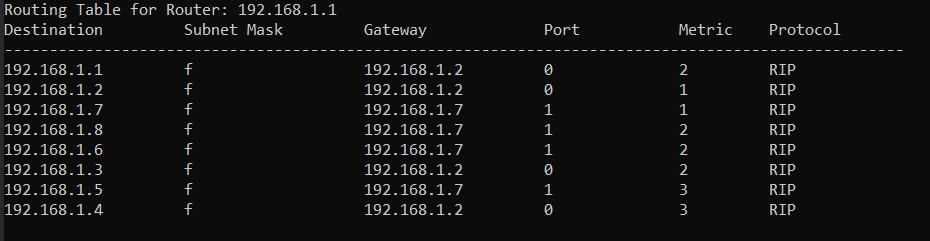
* Clock Generator

هر n ثانیه یکبار کلاس clock generator که در یک thread جدا قرار دارد سیگنالی را به تمام روتر ها میفرستد و هر روتر شروع به پردازش packet های داخل buffer خودش می کند.



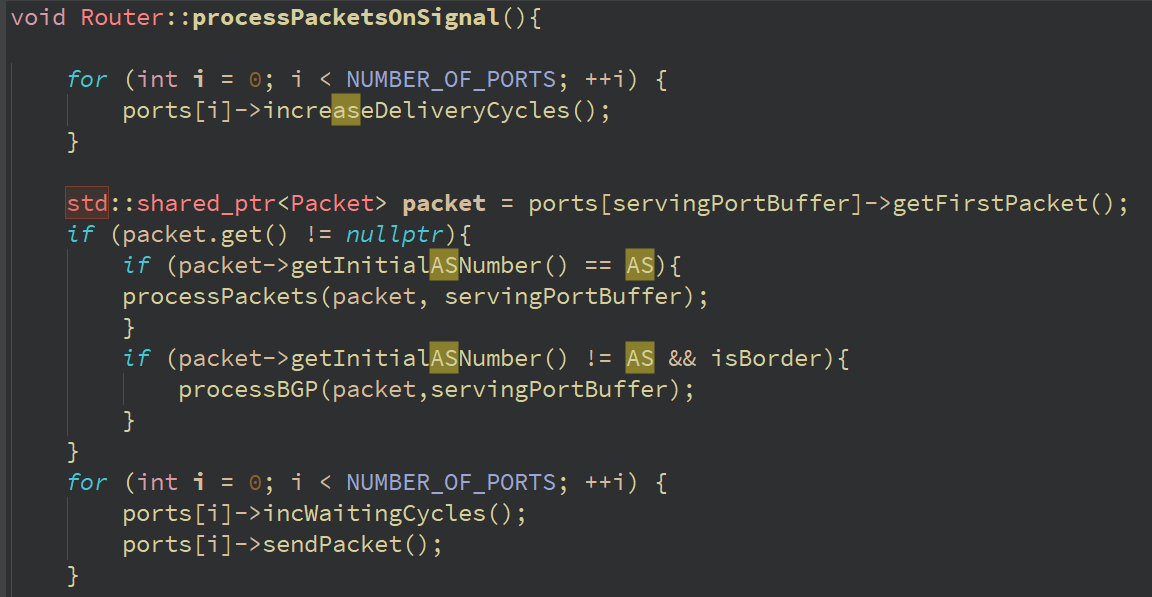
* Routing Table

Routing table ها بسته به نوع الگوریتم مسیریابی، به پیدا کردن کوتاه ترین مسیر کمک میکنند. روتینگ تیبل ها موارد زیر را نگه میدارند:



* Router

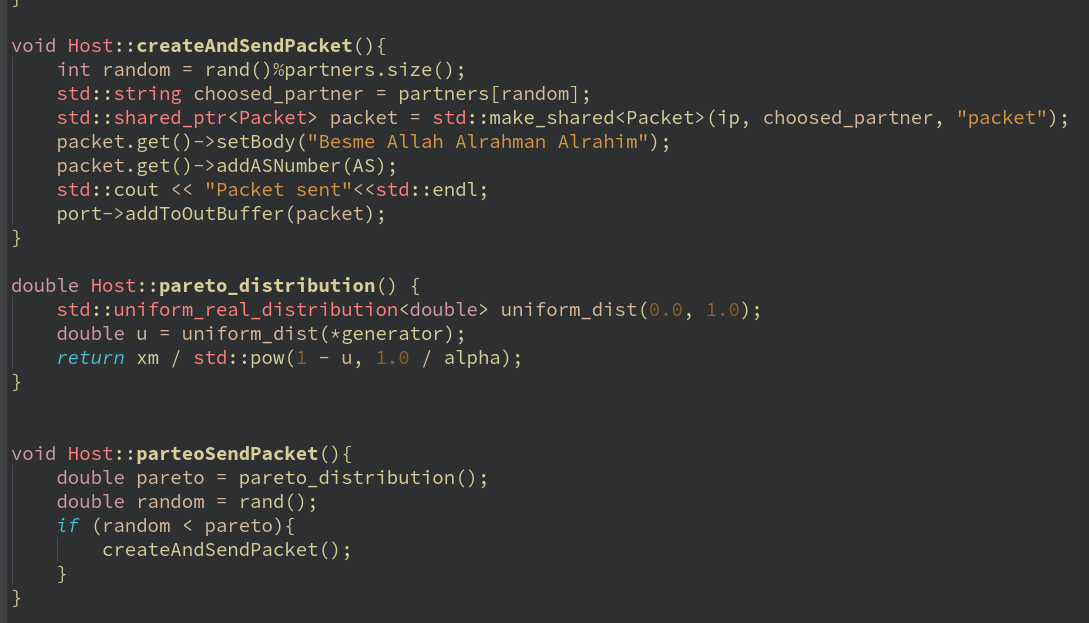
روتر ها در thread های متفاوتی قرار دارند و در هر کلاک یک packet از buffer خود به صورت round robin خوانده و شروع به پردازش آن می کنند:



هر روتر تعدادی Port دارد که از کلاس Buffer هستند و هر port یک buffer برای Input و یک بافر برای output دارد که به شکل زیر می باشد و پس از پردازش packet در روتر، packet به بافر خروجی فرستاده می شود تا در کلاک بعدی فرستاده شود:

  
CommandReader : این فایل پیاده سازی CLI برنامه است.  
  
Host : این فایل پیاده سازی هاست های برنامه است. هر هاست یک پورت دارد که به وسیله آن اطلاعات را به Router مربوطه ارسال می کند. از طرفی هر هاست لیستی از شرکایش در آن طرف شبکه دارد که طبق توزیع پارتو برای آن ها پکت می فرستد.

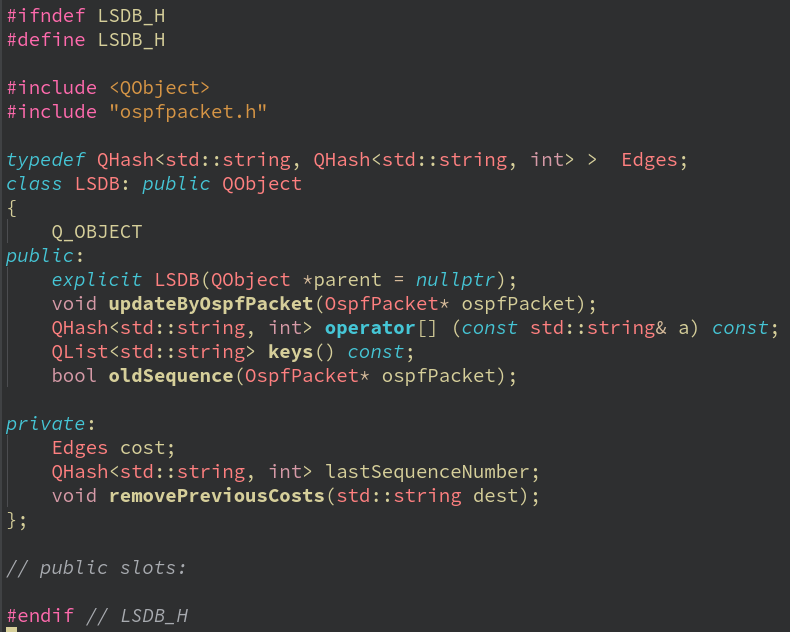
نحوه ارسال پکت با توزیع پارتو نیز به این صورت است که clockGenerator که پیشتر توضیح داده شد به دو اسلات نشان داده شده وصل اند و هر دفعه نمونه ای از توزیع پارتو گرفته می شود و بر اساس آن احتمال پکتی تولید و به شریک رندومی در طرف دیگر شبکه ارسال می شود.

کد موارد توضیح داده شده در بالا موجود است.

TODO : تابع handlePackets هم دو وظیفه دارد :

۱- خالی کردن بافر هاست و ارسال پکت ها به شبکه با کلاک

۲- هندل کردن پکت های دریافتی و ارجاع آن ها به کلاس گزارش که آن ها را برای تحلیل های لازم ذخیره کند.

  
LSDB : این فایل که برای پرتوکل OSPF به کار می رود در واقع گراف شبکه را در خود نگه می دارد.

طی تناوب ثابتی RoutingTable اطلاعات را از آن می گیرد و با الگوریتم Dijkstra کوتاه ترین مسیر ممکن برای ارسال پکت را می یابد.