# توضیحات مربوط به کلاس‌ها

## host

در این کلاس port و IP مربوط به هر گره قرار می‌گیرد.

## NeighborsInformation

در این کلاس اطلاعات مربوط به یک همسایه یک node اعم از host همسایه، زمان آخرین hello message دریافت شده از این همسایه، تعداد hello message هایی که از این همسایه دریافت شده است، تعداد hello message هایی که به این همسایه فرستاده شده است، آخرین زمانی که این همسایه جزو همسایه‌های دوطرفه شده است و کل زمان‌هایی که این همسایه جزو همسایه‌های دوطرفه بوده است.

## UdpSocket

در این کلاس کارهای مربوط به socket و bind مربوط به یک node انجام می‌شود.

## HelloMessage

پیام‌هایی که بین node ها رد و بدل می‌شوند از جنس این کلاس هستند. هر HelloMessage حاوی اطلاعاتی شامل شماره،‌ port و IP فرستنده، نوع پیام، همسایه‌های یک طرفه و دو طرفه فرستنده، زمان آخرین پیام فرستاده شده از فرستنده به گیرنده و زمان آخرین پیام دریافت شده از طرف گیرنده می‌باشد.

## Node

برای هر گره اطلاعاتی نظیر همسایه‌های temp، یکطرفه و دوطرفه یک گره، آیا گره خاموش است. آخرین زمانی که گره خاموش شده است و host مربوط به گره را نگهداری می‌کنیم.

در تابع init این کلاس بعد از انجام تعریف اولیه هر node به وسیله یک ریسه handle می‌شود. تابع هدف این ریسه به شرح زیر است.

در این تابع ابتدا چک می‌شود که آیا گره خاموش است یا خیر. اگر خاموش بود و ۲۰ ثانیه از زمان خاموش شدنش می‌گذشت گره روشن می‌شود. در غیر این صورت اگر گره خاموش بود کاری انجام نمی‌دهد.

همچنین در این تابع هر ۱۰ ثانیه یک بار به صورت رندم یکی از گره‌ها انتخاب می‌شود که خاموش شود. چون انجام این کار بین تمام گره‌ها مشترک است و تنها لازم است هر ۱۰ ثانیه یکی از گره‌ها توسط یک ریسه خاموش شود. بنابراین هر وقت۱۰ ثانیه از آخرین باری که یک گره خاموش شده بود می‌گذشت، اولین ریسه‌ای که در حال اجرا بود یک گره را به صورت تصادفی خاموش می‌کند و متغیر مربوط به آخرین زمانی که یک گره خاموش شده است (که یک متغیر global است) را آپدیت می‌کند. چون تنها یکی از ریسه ها باید این قسمت از کد را اجرا کند، این قسمت به صورت atomic و داخل یک mutex luck انجام می‌شود.

سپس با استفاده از توابع select و recvfrom پیام‌هایی که برای این گره آمده است را دریافت می‌کنیم. سپس با توجه به اینکه پیام رسیده شده از چه پورتی است لیست همسایگان را تغییر می‌دهیم. نمودار فعالیت این تغییر لیست در صفحه بعد آمده است.

سپس چک می‌کنیم اگر در همسایگان یک طرفه و دو طرفه، همسایه‌ای وجود داشت که از زمان آخرین پیام ارسال شده از طرف او بیشتر از ۸ ثانیه می‌گذشت،‌ آن همسایه را از لیست حذف می‌کنیم.

سپس سراغ این می‌رویم که برای همسایه‌های این گره hello message ارسال کنیم.

برای این کار ابتدا چک می‌کنیم که آیا مجموع همسایگان موجود در لیست یک‌طرفه و دوطرفه کمتر از N = 3 است یا خیر. اگر کمتر بود از بین تمام گرهای موجود دیگر که در لیست همسایگان یک‌طرفه و دوطرفه نیستند یک گره را به عنوان همسایه temp انتخاب می‌کنیم تا به آن hello message بفرستیم. سپس به تمام همسایگان یک‌طرفه و دوطرفه‌ای هم که از قبل داشتیم hello message می‌فرستیم.

آیا گره ما در لیست همسایگان به تازگی شنیده شده فرستنده وجود ندارد؟

بله

آیا گره ما در لیست همسایگان به تازگی شنیده شده فرستنده وجود دارد؟

بله

خیر

بله

گره را در لیست همسایگان به یک‌طرفه قرار بده

گره را در لیست همسایگان به دوطرفه قرار بده

خیر

آیا فرستنده در لیست همسایگان دوطرفه و یا یک‌طرفه گره ما قرار دارد؟

آیا فرستنده در لیست همسایگان دوطرفه گره ما قرار دارد؟

فرستنده بسته را پیدا کن

بله

بله

بله

آیا فرستنده در لیست همسایگان یک‌طرفه گره ما قرار دارد؟

آیا گره ما در لیست همسایگان به تازگی شنیده شده فرستنده وجود دارد؟

بله

آیا بسته‌ای برای دریافت وجود داشت؟

همسایه را از لیست همسایگان یک‌طرفه به دوطرفه انتقال بده

همسایه را از لیست همسایگان دوطرفه به یک‌طرفه انتقال بده

یک بسته جدید دریافت کن