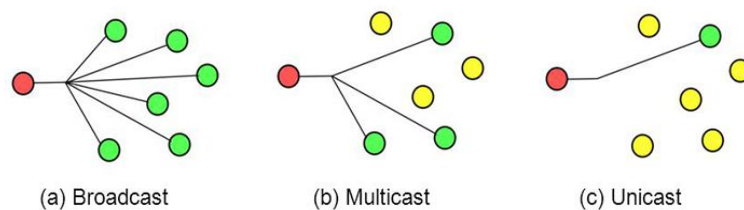


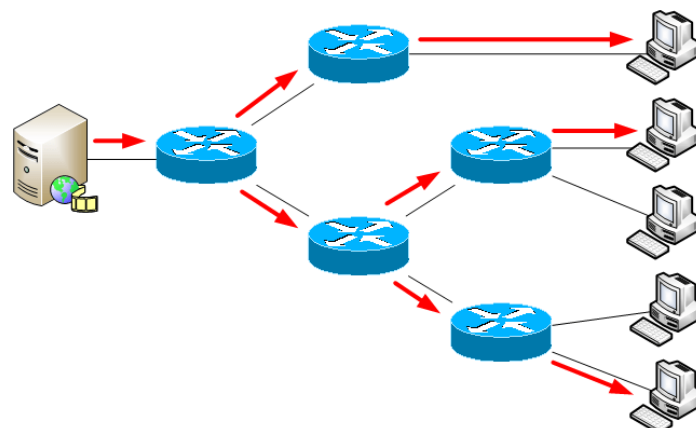
## طرح پروژه پنج‌دری

### توضیح عمومی پروژه

Multicast یکی از روش‌های متداول برقراری ارتباط در شبکه است که در مقابل Unicast و Broadcast قرار دارد. در Multicast ارسال کننده پیام را می‌فرستد، در حلقه‌های ارتباطی پیام هر جا که لازم باشد تکثیر می‌شود و در نهایت به گره‌هایی که علاقمند به دریافت آن پیام بوده‌اند می‌رسد. در این نوع ارتباط بر خلاف Unicast مقصد تنها یک گره نیست و می‌تواند گروهی از گره‌ها باشد، همچنین بر خلاف Broadcast پیام به همه ارسال نمی‌شود، بلکه تنها به گره‌هایی که علاقمند بوده‌اند ارسال می‌شود.



یکی از مهم‌ترین پیاده‌سازی‌های مفهوم Multicast، پیاده‌سازی آن در لایه‌ی شبکه و مبتنی بر پروتکل IPv4 است که با نام IP Multicast شناخته می‌شود. در این پیاده‌سازی یک گروه Multicast با یک IP خاص مشخص می‌شود، هر دریافت کننده برای اعلام علاقمندی به عضویت در گروه (معادلا دریافت پیام در آن گروه) پیام خاصی با نشانی آن گروه ارسال می‌کند. ارسال کننده‌ها می‌توانند خود عضو گروه نباشند. برای ارسال کافیست مقصد پیام نشانی گروه مورد نظر باشد. گره‌های میانی اعم از انواع تجهیزات شبکه وظیفه دارند پیام را با کوتاه‌ترین مسیر و تکثیر بین مسیرهای مختلف به دست دریافت کننده‌ها برسانند.



Multicast به صورت گسترده در برنامه‌های زیرساختی و تجاری استفاده می‌شود، کاربردهایی مانند کنفرانس صوتی یا تصویری، سرویس‌های IPTV و Streaming، بازی‌های کامپیوتری MMO، اشتراک فایل، به‌روزرسانی نرم‌افزار و سیستم عامل بخشی از

کاربردهای این نوع ارتباط است.

در کنار مزایای مهم Multicast چالش‌هایی نیز در استفاده از این نوع ارتباط وجود دارد:

**مقیاس پذیری:** پیاده‌سازی باید متناسب با کاربرد مورد استفاده، تحمل قابل قبولی در زمینه‌ی افزایش تعداد ارسال‌کننده‌ها، دریافت‌کننده‌ها، نرخ ارسال پیام، تعداد گره‌های میانی و مانند آن داشته باشد.

**امنیت:** مطابق با نیازمندی، قابلیت کنترل دریافت‌کننده‌ها، جلوگیری از تغییر داده‌ها و عدم افشای داده برای گره‌های میانی می‌تواند از دغدغه‌های استفاده از این نوع ارتباط باشد

**قابلیت اطمینان:** در کاربردی مانند ارسال Video ممکن است از دست رفتن بسته‌ها تا حد خوبی بدون مشکل و قابل صرف نظر باشد اما در بعضی کاربردهای دیگر مانند همگام‌سازی پایگاه‌داده‌ها یا انتقال اسناد، هدر رفته بسته به هیچ وجه قابل اغماض نیست. در چنین کاربردهایی باید روش‌های اضافه کردن قابلیت اطمینان به این نوع ارتباط مد نظر قرار بگیرد.

در این پروژه، هدف اضافه کردن قابلیت اطمینان به یک کتابخانه‌ی موجود برای ارسال و دریافت به روش IP Multicast است.

## صورت مساله

هدف اصلی پروژه توسعه‌ی یک کلاس جدید با استفاده از این کلاس است که بتواند هر نوع خرابی بسته‌های دریافتی را تشخیص دهد و بر طرف کند. منظور از خرابی، از دست رفتن بسته و یا تکرار یک بسته است و یا تشخیص دریافت نامرتب آن‌ها. کلاس جدید باید با اضافه کردن شماره‌ی پیام و شماره‌ی ارسال‌کننده در متن پیام، این امکان را به دریافت‌کننده بدهد که مشکل را تشخیص دهد، همچنین باید سازوکاری برای ارتباط‌گیری دریافت‌کننده با ارسال‌کننده تهیه شود که با آن بتواند به ارسال‌کننده پیام داده و با دریافت جواب، مشکل خود را رفع کند. روش کلی اضافه کردن قابلیت اطمینان در این پروژه Nack است که توضیح داده می‌شود.

پروژه به چهار بخش تقسیم شده است که در ذیل به آن اشاره خواهد شد:

### بخش اول: پیاده‌سازی ابزار آزمایش پیام‌رسان گروهی

در ابتدا شما باید ابزاری برای استفاده از کلاس ارائه شده توسعه دهید و به کمک آن نرخ ارسال و دریافت را محاسبه کنید. در این بخش محیط توسعه را آماده خواهید کرد، با cmake آشنا می‌شوید و اولین برنامه‌ی پروژه را می‌سازید.

## بخش دوم: پیاده‌سازی کلاس Sequenced Messenger

در این بخش باید کلاسی با نام مشخص شده و با ارث‌بری از کلاس داده شده، پیاده‌سازی گردد که در سمت ارسال شماره‌ای افزایشی (Sequence Number) با طول ثابت به ابتدای پیام اضافه کند. در سمت دریافت کننده آن عدد را جدا نموده و از طریق آن تعداد بسته‌های گم شده، تعداد بسته‌های نامرتب شده و تعداد بسته‌های تکراری را تشخیص دهد.

در این قسمت آزمون‌های واحدی توسعه داده شده‌اند که کلاس شما باید از پس آن‌ها برآید. دستاورد این بخش اجرای صحیح این آزمون‌های واحد خواهد بود.

## بخش سوم: پیاده‌سازی ارتباط قابل اتکا

هدف این قسمت پیاده‌سازی یک کانال ارتباطی قابل اتکا بر پایه TCP است. نوع ارتباط به شکل Client-Server است که هر دو طرف امکان ارسال و دریافت دارند. از این بخش برای قابل اتکا کردن پیام‌رسان گروهی استفاده خواهد شد. پیاده‌سازی ارائه شده باید آزمون‌های واحد را با موفقیت پشت‌سر بگذارد.

## بخش چهارم: پیاده‌سازی پیام‌رسان گروهی قابل اتکا

در این باید کلاس Reliable Messenger پیاده‌سازی شود که علاوه بر عملکرد بخش دو، در صورت تشخیص مشکل، با فرستنده ارتباط برقرار می‌کند و تلاش می‌کند مشکل را رفع کند.

این بخش نیز آزمون‌های واحدی دارد که کلاس شما باید بتواند در سناریوهای آن‌ها عملکرد صحیحی داشته باشد.

## فضا، ابزار توسعه، ارزیابی

توسعه باید در محیط Ubuntu ۲۰,۰۴ بر روی ماشین‌های مجازی تعیین شده صورت بگیرد. کدها باید با کامپایلرهای g++ و clang سازگار باشند. کتابخانه‌های مورد استفاده باید به تایید مسئولین پروژه برسد. کدها باید cmake داشته باشند. نگاه‌داشت‌پذیری، استاندارد‌نویسی، توجه به اصول طراحی شی‌گرا و قابلیت‌های زبان از مسایلی است که باید در توسعه لحاظ گردد. کدها باید عاری از خطاها و نشتهای حافظه‌ای باشند. کارایی از مهم‌ترین فاکتورهای ارزیابی نهایی است.