



تمرین کامپیوتری ۴ (فاز ۵ تمرین ۳)



عنوان: پیاده سازی شبیه ساز شبکه برای بررسی انتقال داده در TCP

درس: شبکه های کامپیوتری
استاد: دکتر ناصر یزدانی

دستیاران آموزشی: آراین فیروزی، عرفان احمدی، گلبو رشیدی، سهراب مرادی

نیمسال اول سال تحصیلی ۱۴۰۳-۰۴

هدف پروژه:

در این فاز پروژه، شما الگوریتم های TCP و ازدحام^۱ را در شبکه ی طراحی شده در فاز های قبلی پروژه پیاده سازی خواهید کرد. به منظور ایجاد یک سناریو منطبق بر واقعیت، با روشی شبیه به پروتکل Torrent ترافیکی با حجم بالا در شبکه ایجاد خواهد شد تا الگوریتم های پیاده سازی شده در شرایطی با بار زیاد بر روی شبکه صحت سنجی شوند.

توضیح تمرین:

۱. الگوریتم تورنت:

پروتکل بیت تورنت یک پروتکلی ارتباطی است که برای اشتراک گذاری فایل هایی با حجم بزرگ به صورت P2P و بدون نیاز به سرور مرکزی^۲ استفاده می شود، که نسخه ساده شده آن را در این تمرین پیاده سازی خواهید کرد. نحوه کارکرد این الگوریتم را شرح دهید.

۲. پیاده سازی ترافیک شبکه:

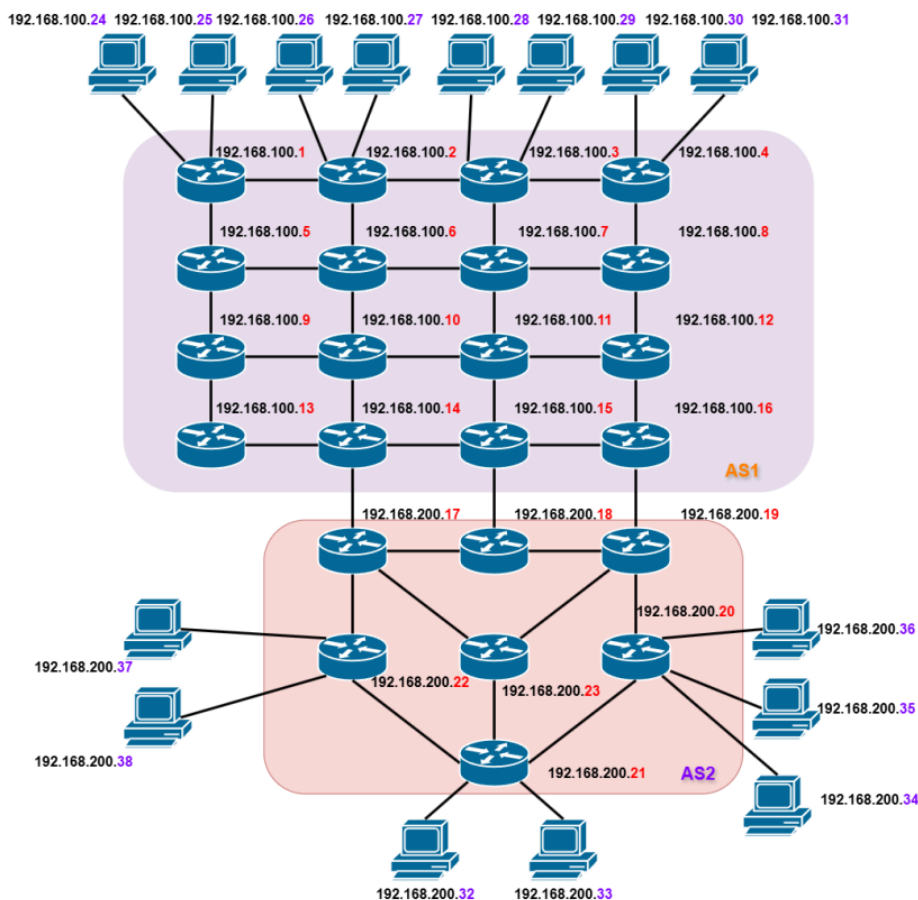
الگوریتمی که شما پیاده سازی خواهید کرد، ساده سازی شده و به شرح زیر است:

- سیستم کنترل داده که در فاز های قبلی پیاده سازی شده، تعداد داده ها را با یک توزیع آماری مشخص بین PC ها مشخص میکرد. در این بخش شما یک سیستم بسیار ساده شده از تورنت را رو به رو هستید که توسط ما پیاده سازی شده است. در واقع سیستم کنترل داده یک فایل با حجم نسبتاً زیاد را به صورت برابر بین PC های فرستنده توزیع خواهد کرد و به جای اینکه فایل از یک PC به PC دیگر فرستاده شود، چندین دستگاه تکه های مختلف یک فایل را به یک دستگاه گیرنده خواهند فرستاد که این بسته های مختلف این فایل با توجه به شماره هایی که بر مبنای ترتیب آن درج شده در مقصد سرهم خواهد شد (این شماره گذاری صرفاً برای اطلاع از ترتیب بسته ها در فایل اصلی است و ربطی به Sequence Number که مربوط به ترتیب بسته های ارسالی از یک دستگاه است ندارد). **(توسط دستیاران آموزشی پیاده سازی شده است)**
- هر دستگاه در شروع فایل را از روی دیسک میخواند و QByteArray متناظر با آن را به بسته هایی با payload با حجم حداکثر یک کیلوبایت تقسیم میکند. سپس شماره پکت هایی را که خواهد فرستاد را از سیستم کنترل داده دریافت می کند و طبق الگوریتم پیاده شده که در بخش بعدی شرح داده میشود، در هر تیک تعدادی پکت را میفرستد. **(توسط دستیاران آموزشی پیاده سازی شده است)**
- بین هر فرستنده و گیرنده، یک پروتکل TCP پیاده سازی کنید که با استفاده از three-way handshake ارتباط برقرار کنند. این شامل مراحل SYN، SYN-ACK و ACK میشود که در نهایت منجر به برقراری ارتباط خواهد شد. پکت های مختلف باید تایپ های متناسب را در Header پیاده کنند. **(امتیازی)**
- اگر بخش قبل را پیاده سازی نکردید، بخش بعدی بدون handshake و مستقیم (هارد کد) انجام خواهد شد.

^۱ Congestion

^۲ البته رابطه بین کاربران توسط یک سرور مرکزی به نام Tracker کنترل میشوند اما این سرور تنها برای ایجاد هماهنگی بین کاربران است و داده های فرستاده شده را نمیبیند

- دستگاه‌های موجود در شبکه RingStar به عنوان فرستنده فایل را ارسال خواهند کرد و یک دستگاه دلخواه از شبکه Mesh به عنوان گیرنده بسته‌ها را دریافت خواهد کرد. (ما PC شماره ۲۴ را به عنوان مقصد معرفی نموده ایم) داده‌ها باید به صورت بسته‌های مشخص ارسال شوند و گیرنده باید در پاسخ ACK ارسال کند. بررسی کنید که تمامی بسته‌ها به درستی دریافت و تأیید شده‌اند.



شکل ۱ - ساختار شبکه‌ی مورد بحث تمرین

- فایل‌های یک فایل صوتی است که در ایلرن درس قرار داده شده است. جهت صحت سنجی فرایند ارسال، لازم است گیرنده آن را با فرمت اولیه فایل بر روی دیسک ذخیره کند.
- اندازه صف^۳ روترها را به نحوی تنظیم کنید که شبکه دچار حالت ازدحام شود. (ما اندازه صف را برابر با ۶ بسته قرار داده ایم)

۳. پیاده‌سازی پروتکل کنترل ازدحام:

کنترل ازدحام TCP را با استفاده از الگوریتم TCP NewReno پیاده‌سازی کنید. همچنین در گزارش خود سه الگوریتم TCP Veno، TCP Vegas، TCP NewReno را با یکدیگر مقایسه کنید.

³ Queue

مشاهده نتایج:

برای مشاهده عملکرد الگوریتم های کنترل ازدحام، نمودار میزان بسته های موجود در بافر روتر مربوط به گیرنده (در مثال ما PC شماره ۲۴) و یکی از روتر هایی که مستقیماً به یکی از فرستنده ها وصل شدند (روتر شماره ۲۰) را با استفاده از کد قرار داده شده بر روی ایلرن رسم کنید و درستی کارکرد الگوریتم را نشان دهید. کد موجود در سایت آرایه ای از اعداد را با فرمت QVector دریافت میکند و نمودار متناظر با آن را رسم میکند. به منظور استفاده از آن میتوانید لاگی از تعداد پکت های موجود در صف روتر نگهداری کنید و بعد از اتمام فرایند آن را رسم کنید.

فرآیند اجرا و پیاده سازی تمرین در شبیه ساز ارائه شده:

فایل موسیقی مورد نظر ما شامل حدود ۲۲ هزار بسته ی ۱ کیلوبایتی میشود و همانطور که در خروجی زیر مشاهده میکنید، به علت پرشدن بافر برخی از روتر ها در مسیر بسته، حدود ۱۹ هزار بسته به مقصد میرسد و باقی بسته های drop میشوند.

```
C:\QNTools\QICreator\bin\qpc X + -
PC 24 received 19512 th packet.
Router 14 : Buffer full. Dropping packet with dest IP "192.168.100.24"
PC 24 received 19513 th packet.
PC 24 received 19514 th packet.
PC 24 received 19515 th packet.
PC 24 received 19516 th packet.
PC 24 received 19517 th packet.
PC 24 received 19518 th packet.
PC 34 sent the last packet.
PC 35 sent the last packet.
Router 14 : Buffer full. Dropping packet with dest IP "192.168.100.24"
PC 36 sent the last packet.
PC 32 sent the last packet.
PC 33 sent the last packet.
PC 37 sent the last packet.
PC 38 sent the last packet.
PC 24 received 19519 th packet.
PC 24 received 19520 th packet.
PC 24 received 19521 th packet.
PC 24 received 19522 th packet.
PC 24 received 19523 th packet.
PC 24 received 19524 th packet.
Router 14 : Buffer full. Dropping packet with dest IP "192.168.100.24"
PC 24 received 19525 th packet.
PC 24 received 19526 th packet.
PC 24 received 19527 th packet.
PC 24 received 19528 th packet.
PC 24 received 19529 th packet.
PC 24 received 19530 th packet.
Router 14 : Buffer full. Dropping packet with dest IP "192.168.100.24"
PC 24 received 19531 th packet.
PC 24 received 19532 th packet.
PC 24 received 19533 th packet.
PC 24 received 19534 th packet.
PC 24 received 19535 th packet.
PC 24 received 19536 th packet.
PC 24 received 19537 th packet.
PC 24 received 19538 th packet.
PC 24 received 19539 th packet.
PC 24 received 19540 th packet.
PC 24 received 19541 th packet.
PC 24 received 19542 th packet.
PC 24 received 19543 th packet.
PC 24 received 19544 th packet.
PC 24 received 19545 th packet.
PC 24 received 19546 th packet.
PC 24 received 19547 th packet.
PC 24 received 19548 th packet.
PC 24 received 19549 th packet.
PC 24 received 19550 th packet.
PC 24 received 19551 th packet.
PC 24 received 19552 th packet.
PC 24 received 19553 th packet.
PC 24 received 19554 th packet.
PC 24 received 19555 th packet.
PC 24 received 19556 th packet.
PC 24 received 19557 th packet.
PC 24 received 19558 th packet.
PC 24 received 19559 th packet.
PC 24 received 19560 th packet.
PC 24 received 19561 th packet.
PC 24 received 19562 th packet.
PC 24 received 19563 th packet.
PC 24 received 19564 th packet.
PC 24 received 19565 th packet.
PC 24 received 19566 th packet.
PC 24 received 19567 th packet.
PC 24 received 19568 th packet.
PC 24 received 19569 th packet.
PC 24 received 19570 th packet.
```

ما اندازه ی بافر روتر ها و نحوه ی ارسال بسته ها از روتر ها را به گونه ای تغییر دادیم تا نسبت تعداد بسته های drop شده به کل بسته ها عدد معقولی باشد. شما با پیاده سازی الگوریتم مورد بحث، میتوانید تضمین کنید که تمام ۲۲ هزار بسته به سلامت به مقصد میرسند.

ما صحت و سلامت اجرای شبیه ساز پیاده سازی شده توسط دوستانتان را بر روی ویندوز با سه کامپایلر MSVC MinGW, MinGW-LLVM ۲۰۲۲، تست نموده ایم و اجرای شبیه ساز با MinGW دارای مشکل است، **پیشنهاد** میکنیم تا از MSVC یا LLVM-MinGW استفاده کنید.

در حین اجرای الگوریتم، تنظیمات ابتدایی را به شکل زیر وارد کنید:

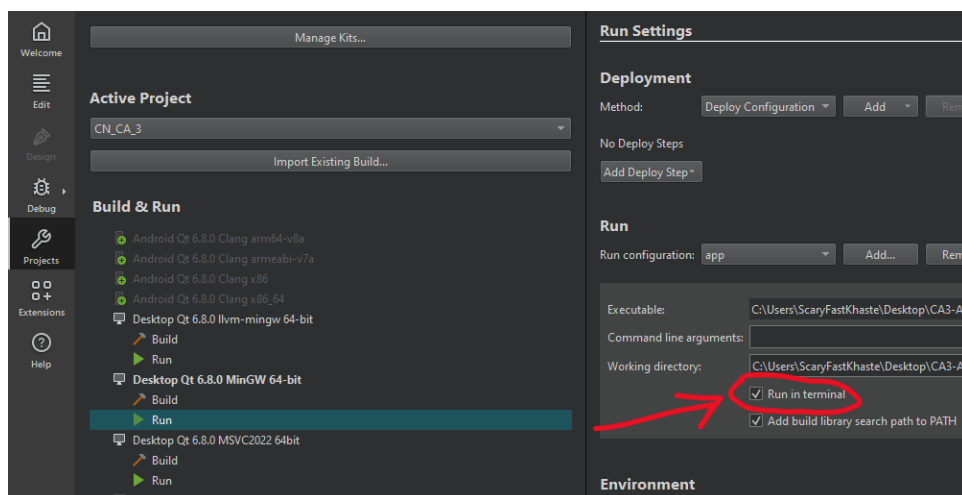
```

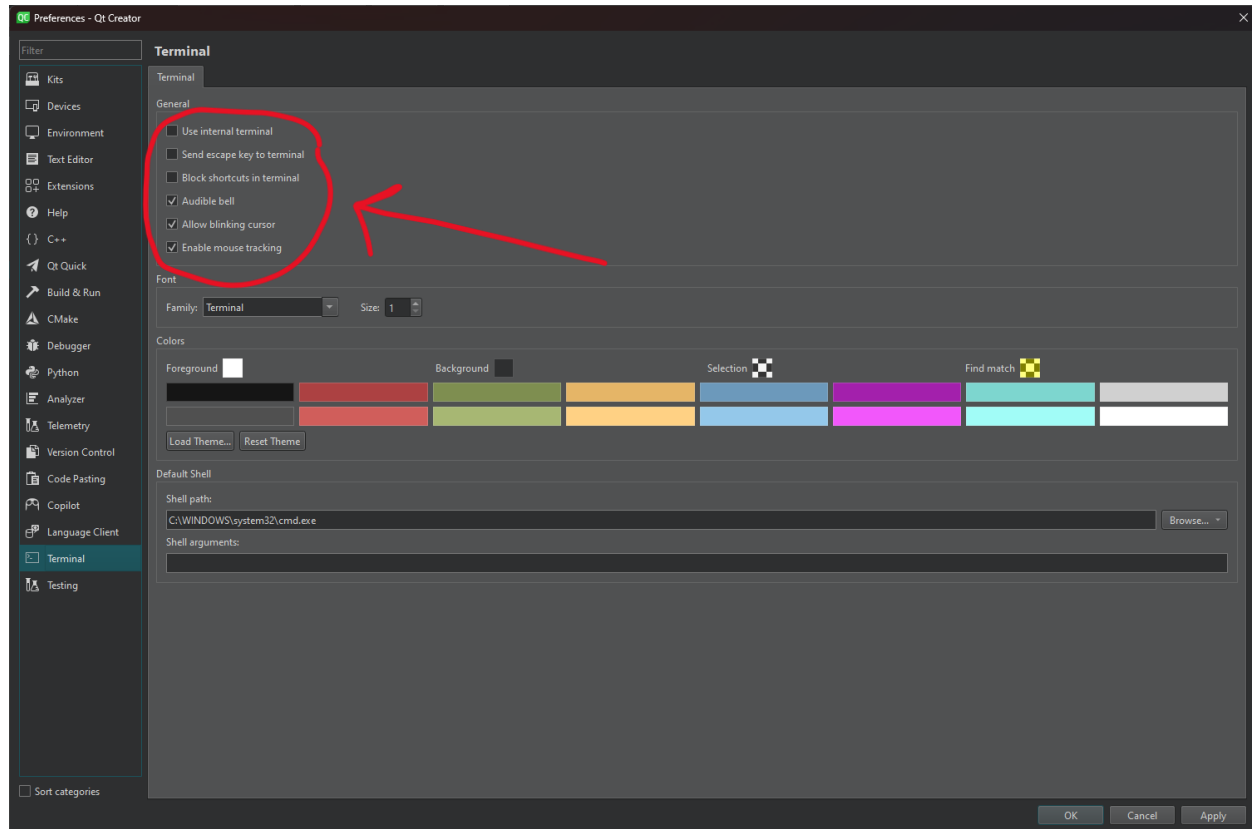
C:\Qt\Tools\QtCreator\bin\qtc
ApplicationContext initialized.
Simulator instance set.
ID pre-assignment done.
Welcome To Our UT Network Simulator !!!
Do you want BGP? (yes/no): no
Enter the main routing algorithm you want (1 for RIP, 2 for OSPF): 2
Do you want to add a torus topology? (yes/no): no

```

به جهت اجرای شبیه ساز در ترمینال سیستم عامل (و نه ترمینال داخلی کیوتی، برای اینکه بتوانید ورودی های مربوطه را وارد کنید) تنظیمات زیر را در Qt Creator انجام دهید:

- ۱- در تب Projects بخش run مربوط به کامپایلر انتخابی، گزینه ی Run in terminal را فعال کنید
- ۲- از سربرگ Tools -> External -> Configure وارد بخش ترمینال شوید و گزینه ی Use Internal Terminal را غیر فعال کنید.





بسته های ACK را به عنوان بسته های داده با محتوای TCP_ACK ارسال کنید و به همین شکل در PC مقصد مدیریت کنید:

```
if(packet->getType() == PacketType::Data)
{
    if(m_ipAddress->getIp() == "192.168.100.24")
    {
        qDebug() << Q_FUNC_INFO << "PC" << m_id << "ip:" << m_ipAddress->getIp()
            << "received packet with payload size:" << packet->getPayload().size();
    }

    QString destinationIP = packet->destinationIP()->getIp();
    QString sourceIP = packet->sourceIP()->getIp();
    QString payload = packet->getPayload();

    packet->addToPathTaken(destinationIP);

    if(payload == "TCP_ACK")
    {
        /**
         * =====
         * =====
         * =====
         * @attention Handle the received ACK packet.
         * =====
         * =====
         */
    }
}
```

نحوه ی ایجاد یک بسته برای ارسال:

```
static const QString destination = "192.168.100.24";  
auto destinationIP = QSharedPointer<IP>::create(destination);  
  
auto packet = m_packetStorage.takeFirst();  
  
packet->addToPath(m_ipAddress->getIp());  
packet->addToPathTaken(m_ipAddress->getIp());  
packet->addToPath(destination);  
packet->setDestinationIP(destinationIP);  
packet->setSourceIP(m_ipAddress);
```

به یاد داشته باشید که IP مقصد برای سادگی بیشتر hard code شده است و نیازی به تغییر آن نیست.

نکات پیاده سازی:

لطفا به نکات زیر به دقت توجه داشته باشید:

- ۱- میتوانید از پیاده سازی خود یا دوستانتان در فاز های قبلی برای پیاده سازی این فاز استفاده کنید.
- ۲- در پیاده سازی ای که ما در اختیارتان قرار میدهیم لازم است تا توابع generatePacket و processDataPacket از کلاس PC را به همراه موارد خواسته شده در بخش مشاهده نتایج پیاده سازی کنید.
- ۳- در این دو تابع، شما بسته ها را ارسال و دریافت میکنید و برای بسته های دریافتی ACK ارسال میکنید. همچنین الگوریتم خواسته در همین توابع باید پیاده سازی شود.
- ۴- پیاده سازی هرگونه تابع کمکی در شبیه ساز به شرط ذکر در گزارش مجاز است.
- ۵- آی پی کاربر شماره ۲۴ در تابع generatePacket هارد کد شده است.
- ۶- ما پیاده سازی خودمان (که نسخه ی دستکاری شده ی پیاده سازی آقایان کوروش سجادی و علی سامانیان است) را با کامپایلر MSVC بر روی ویندوز تست کرده ایم.
- ۷- برای سنکرون شدن، شبیه ساز باید با موارد زیر کانفیگ شود:
 - الگوریتم مسیریابی: OSPF.
 - استفاده از BGP: خیر.
 - استفاده از توپولوژی تورس: خیر.
 - اندازه بافر: ۶ بسته.
 - اندازه ی بافر PC ها: نامحدود.
 - تعداد بسته هایی که PC ها در هر سیکل میتوانند ارسال کنند: نامحدود متناسب با الگوریتم TCP
 - اگر برای انجام تمرین از پیاده سازی خودتان استفاده میکنید میتوانید بخش هایی از این تمرین که ما به صورت عمومی پیاده سازی کرده ایم را از ما تحویل گرفته و متناسب با پیاده سازی خودتان شخصی سازی کنید. دستیاران در این مورد کمک کرده و ایده های طراحی درست را به شما خواهند داد.
 - اگر در این پیاده سازی باگی دیدید حتما اطلاع دهید تا در سریع ترین زمان ممکن رفع شود. لطفا برای رپورت باگ به آقای مرادی پیام بدهید.
 - تاخیر این فاز روزی ۲ درصد محاسبه خواهد شد.

جمع بندی و نکات پایانی:

- مهلت تحویل: ۲۶ دی ماه ۱۴۰۳
- پروژه در گروههای ۲ نفره انجام میشود. (گروه بندی در سامانه ایلرن نیز انجام می شود و تحویل تمرین به صورت گروهی خواهد بود)
- هر ۲ نفر می بایست کار را تقسیم کنند. همچنین از Git برای ساختن branch و تقسیم issue ها استفاده نمایید. (با استفاده از commit ها و تعیین issue ها میزان مشارکت هر نفر مشخص می شود). بعد از انجام این کار کدها را در یک repository به نام ۵_۳_CN_CHomeworks در اکانتتهای GitLab/GitHub خود قرار دهید (به صورت private). همچنین در یک فایل md.README میتوانید report و داکيومنت خود را کامل کنید و در کنار repository قرار دهید. در نهایت لینک این repository را در محل پاسخ تمرین قرار دهید. (از فرستادن فایل به صورت زیپ جدا خودداری نمایید). اکانت تی ایهای این تمرین رو به Repo خودتون به پروژه اضافه کنید.
- Git Accounts: @ArianFiroozi - @erfun-ahmadi - @thegolboo
- برای پیاده سازی این تمرین از ++C استفاده کنید.
- دقت کنید گزارش نهایی شما می بایست همانند یک Document باشد و شامل توضیح کد و ساختار کد، همچنین نتیجه نهایی اجرای کد و اسکرین شاتهای دقیق از تمام مراحل باشد. (در فایل md.Readme کنار فایل های اصلی خود و در Repo مربوطه قرار دهید). **این نکته حائز اهمیت است که فایل PDF به هیچ عنوان مورد پذیرش قرار نخواهد گرفت.**
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره این پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- برای هر قسمت کد، گزارش دقیق و شفاف بنویسید. کدهای ضمیمه شده بدون گزارش مربوطه نمره ای نخواهند داشت.
- هدف این تمرین یادگیری شماسست. لطفا تمرین را خودتان انجام دهید. در صورت مشاهدهی مشابهت بین کدهای دو گروه، مطابقت سیاست درس با گروه متقلب و تقلب دهنده برخورد خواهد شد. همچنین توجه داشته باشید استفاده از ابزارهای AI توجیهی برای شباهت کدهای تحویل داده شده توسط گروه های مختلف نمی باشد.
- سوالات خود را تا حد ممکن در گروه درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن بهره مند شوند. در صورتی که قصد مطرح کردن سوال خاص تری دارید، از طریق ایمیل زیر ارتباط برقرار کنید. توجه داشته باشید که سایر شبکه های اجتماعی راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیطهای غیر رسمی نیستند.

- ایمیل دستیاران طراح:

arianfiroozim@gmail.com ○
erfahmadi۰۳@gmail.com ○
golgolrashidi@gmail.com ○

موفق باشید!