به نام خداوند بخشنده‌ی مهربان

تمرین برنامه‌نویسی دوم[1]
شبکه‌های کامپیوتری

مدرس: مهدی خرازی **پاییز ۹۰**

مقدمه

پروتکل TCP یکی از پروتکل‌های لایه‌ی انتقال در شبکه است که در برنامه‌های کاربردی متعددی در شبکه اینترنت استفاده می‌شود. این پروتکل با ایجاد ضمانتِ ارسال بسته‌ها، سرویس‌های متعددی را به لایه‌ی کاربرد می‌دهد. همچنین TCP با مشخص کردن هر میزبان نهایی[2] با شماره‌ی پورت و آدرس پروتکل اینترنت[3]، ایجاد کننده‌ی یک ارتباط کاملاً دوطرفه[4] بین گره‌ها است و با استفاده از یک پنجره‌ی متحرک همراه با مکانیزم‌های مختلف از قبیل Timeout، Retransmission و غیره به دنبال امن کردن کانال ارتباطی بین گره‌ها است. در تمرین صفرم با استفاده از کتابخانه‌ی «سوکت[5]» و امکانات این پروتکل مجموعه‌ای از دستورات را در کامپیوتر مقصد اجرا کردید. به منظور درک بهتر این پروتکل گره‌ی کارگزار در تمرین صفرم را به محیط «**پرتو**» منتقل می‌کنیم و در این محیط به پیاده سازی برخی از امکانات TCP Suit می‌پردازیم. همان‌طور که می‌دانید پروتکل TCP از لحاظ اجرا به ۳ فاز مختلف تقسیم می‌شود.

الف: فاز برقراری ارتباط[6]؛

ب: فاز انتقال داده[7]؛

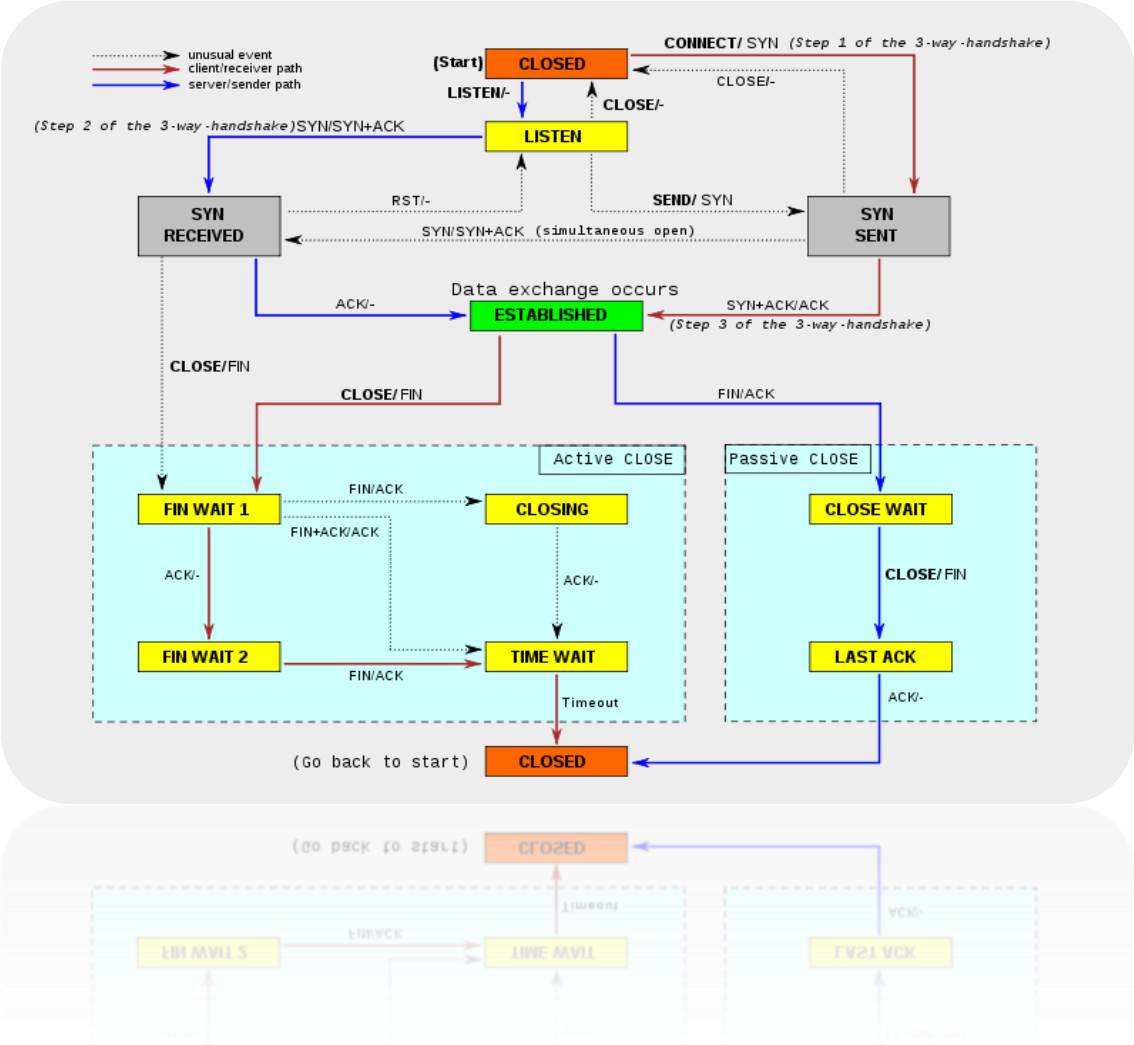
ج: فاز بستن ارتباط[8].

در نتیجه می‌توان این پروتکل را به صورت ماشین حالت[9] پیاده سازی کرد. به صورت معمول برای پیاده سازی این پروتکل به صورت کامل، یک ماشین حالت با ۱۲ حالت مختلف را در نظر می‌گیرند که این ۱۲ حالت در تصویر زیر نشان داده شده است.

[1] با تشکر از بهنام مؤمنی، امیر شیخ‌ها، مهدی احمدی نژاد، کامیار الله وردی، سجاد فولادی و مهرداد مرادی

[2] End Host
[3] Internet Protocol Address(IP Address)
[4] Full Duplex
[5] Socket
[6] Connection Establishment
[7] Data Transfer
[8] Connection Termination
[9] FSM

اما در این تمرین شما تنها بخش‌هایی از این ماشین حالت را برای گره ی کارگزار در محیط «**پرتو**» پیاده سازی می‌کنید. همچنین اطلاعات بیشتری را می‌توانید درمورد این پروتکل در http://en.wikipedia.org/wiki/Transmission_Control_Protocol و http://www.ietf.org/rfc/rfc793.txt جست و جو کنید.



برنامه‌ی کارگزار-گره مجازی در محیط «پرتو»

برنامه‌ی کارگزار از لحاظ کارایی مانند تمرین صفرم می‌باشد. اما با توجه به نیازمندی پیاده سازی این پروتکل در لایه‌ی چهارم یا لایه‌ی انتقال، از محیط پرتو برای پیاده سازی استفاده می‌کنیم. در این تمرین شما برای سادگی تنها مکانیزم‌های قابلیت اطمینان[10] و بخش‌هایی از مکانیزم‌های کنترل ازدحام[11] را پیاده سازی می‌کنید. موارد زیر از ضروریات این تمرین هستند:

[10] Reliability Mechanisms
[11] Congestion Control Mechanisms

Simple TCP Suite
Three-Way Handshaking
Acknowledgment and Sequence Numbers
TCP Checksum
Connection Termination
Fast Retransmission(Triple Multiple Acknowledgments)
Retransmission

نکته‌ی اول :

در پروتکل TCP فرستنده و گیرنده در سرایند[۱۲] بسته‌های خود اندازه‌ی پنجره را تنظیم می‌کنند. فرستنده برای اینکه بتواند از Overflow کردن بافر گیرنده جلوگیری کند و همچنین Congestion را تا حدودی برطرف کند؛ حداکثر به اندازه Effective Window sizeبسته داده‌ای برای گیرنده ارسال می‌کند و این مقدار در زیر تعریف شده است. دقت کنید که و مقدار Congestion Windowرا باید کنید.

$$Maximum\ Window\ Size = Min\{Advetrised\ Window\ size, Congestion\ Window\}$$

$$Effective\ Window\ Size = Max\ Window\ Size - (LastByteSent - LastByteAcked)$$

نکته‌ی دوم:

در این تمرین لزومی ندارد Timeout را به صورت پویا تغییر دهید و در نظر گرفتن زمان ۵ ثانیه برای Retransmission کافی است.

نکته‌ی سوم:

در این تمرین برای سادگی در پیاده سازی فرض می‌کنیم که $Congestion\ Window\ Size$:

- همواره از $Advertised\ Window\ Size$ کمتر است.
- مقدار ثابتی است که به عنوان ورودی از خط فرمان دریافت می‌شود.

نحوه اجرای برنامه‌ی کارگزار

برنامه‌ی کارگزار به صورت زیر در خط فرمان اجرا می‌شود و از ورودی مقدار MSS[۱۳]، *Congestion window size* و اندازه‌ی بافر را دریافت می‌کند:

به عبارت دیگر برنامه‌ی کارگزار در خط فرمان به صورت مقابل اجرا می‌گردد:

$ ./server.sh <MSS><Congestion window size><Buffer size>

[12] Header

[۱۳] قابل توجه است که ممکن است برای تستِ برنامه شما این مقدار برابر با هر مقدار صحیحی از بایت(و درعین حال کوچکتر از MSS واقعی TCP)قرار گیرد.

همچنین، مقدار *MSS*، *Congestion window size* و اندازه‌ی بافر در محیط پرتو قبل از فراخوانی تابع initialize به عنوان آرگومان ورودی تابع parseArgument قابل دسترسی می‌باشد. همچنین برنامه‌ی کارگزار مجموعه اطلاعات لازم برای پیاده سازی‌های مذکور را از طریق Information Custom به دست می‌آورد که قالب آن در زیر آمده است :

توضیح قالب:

در خط اول مقدار اولیه‌ی Sequence number آمده است که کارگزار آن را باید در مرحله‌ی Handshaking به برنامه‌ی کارخواه انتقال دهد. در خط بعد شماره پورتی که باید کارگزار به Listen کردن بپردازد آورده شده است. با توجه به اینکه کارگزار برای دسترسی به گره‌های خارج از محیط پرتو باید آدرس فیزیکیِ (یا آدرس لایه دوم یا آدرس MAC) دروازه دسترسی به شبکه بیرون را بداند؛ این آدرس در خط بعد آمده است. همچنین در خط سوم تعداد مدخل[۱۴] های موجود در پایگاه داده و به دنبال آن در خط چهارم نام کاربری، رمز عبور و آدرس برنامه‌ی Shell را می‌توانید مشاهده کرد.

Server Custom Information
Initial Sequence Number of Server
Port Number(Listening Port)
Mac Address of Gateway to Client
Entry Size
Username/Password/Shell Program Address

در زیر یک نمونه از اطلاعات آورده شده است:

Server Custom Information

26764

5335

01:23:45:67:89:ab

2

Sadjad Allaahverdi

1234

/bash/sh

Mehrdad Moradi

76867

/bash/sh

نکته‌ی چهارم:در این تمرین کارگزار نباید دستور Connection established را برای کارخواه ارسال کند.

نکته‌ی پنجم: در این تمرین دستور EXIT از مجموعه دستورهایی که باید پیاده سازی شوند حذف شده است.

[14] Entry

برنامه‌ی کارخواه-گره در محیط «سوکت»

نکته ششم:برنامه‌ی کارخواه در این تمرین مانند تمرین‌های قبلی می‌باشد.

مثال: در زیر یک نمونه از نشست بین کارخواه و کارگزار آورده شده است.

```
HELO
Please authenticate.
USER justin
Invalid username.
USER sadjad
Username accepted, enter password.
ls
Not authenticated.
PASS rsdtjkff
Welcome, sadjad.
ls
.     ..    temp  files games
rm -rf home/
date
Wed Aug 5 11:15:30 IRDT 2011
cat /home/sadjad/temp/1.txt
There are four PAs:
1. PA0
2. PA1
TERMINATE
```

نکات ضروری

- برای ارسال این تمرین باید یک Makefile بنویسید. برای این کار می‌توانید از راهنمای نوشتن Makefile که در سایت درس قرار دارد استفاده کنید. برنامه‌ی شما باید با دستور make کامپایل شود و فایل‌های اجرایی client و server را تولید کند.

- فایل‌های client.cpp، پوشه‌ی user و Makefile را با فرمت ZIP فشرده کرده و بر روی سیستم داوری خودکار[15] بارگذاری کنید. از ارسال فایل‌های باینری خودداری کرده و حتماً پیش از ارسال make clean را اجرا نمایید.

- در این تمرین می‌توانید IP Address و Physical Address(MAC)مختص هر واسط[16] را به ترتیب از طریق تابع عمومی کلاس Interface با نام getIP و آرایه‌ی mac استخراج کنید.

موفق باشید.

[15] http://partov.sharif.edu/
[16] Interface