

به نام خدا



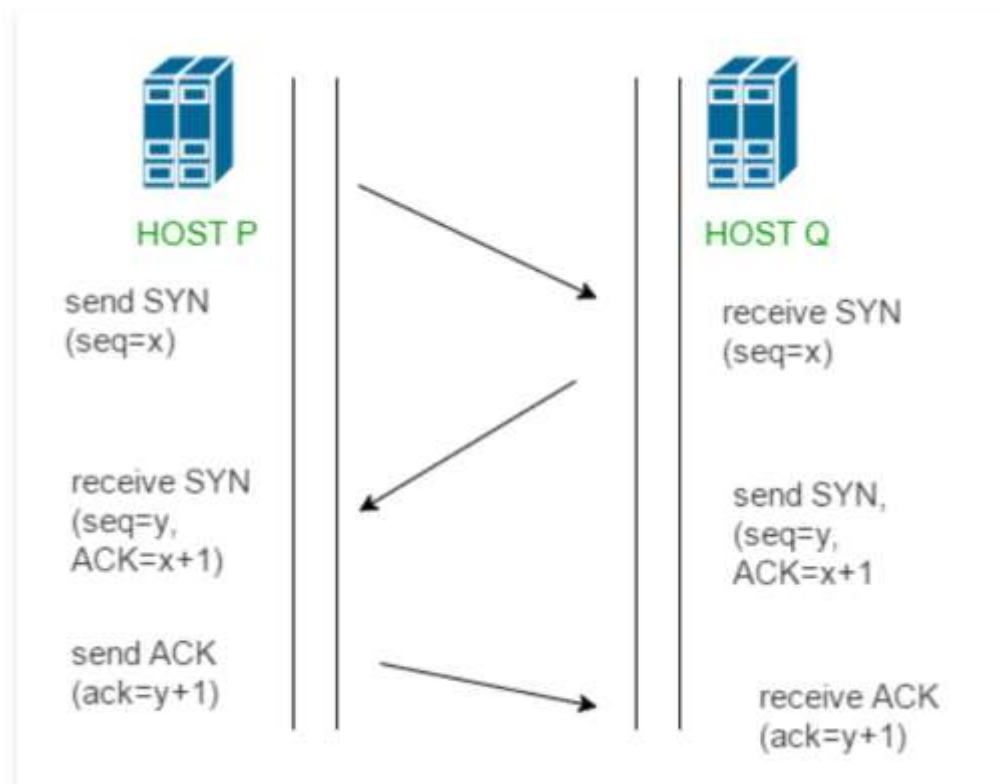
## شبکه های کامپیوتری - پاییز 98

پروژه سوم



توضیحات هر مرحله:

### 1. پیاده سازی three way handshake برای شروع اتصال:



### 2. ارسال داده ها توسط پروتکل Go Back N:

ارسال داده ها باید براساس قوانین موجود در این پروتکل پیاده سازی شوند. برای فهم بهتر این پروتکل پیشنهاد می شود این [لینک](#) را ببینید.

### 3. Congestion Control

در این پروژه، قصد، کنترل ازدحام به وسیله tcp vegas است که تفاوت هایی در قوانین کنترل ازدحام با tcp reno دارد؛ برای فهم بهتر این پروتکل و تفاوت های بین این دو که برای پیاده سازی این پروژه نیاز است، این [لینک](#) را ببینید.

برخلاف reno و new reno که زمانی که packet loss اتفاق افتاد، میتوانند congestion را تشخیص دهند. این پروتکل براساس RTT یک بسته، congestion را تشخیص می دهد. الگوریتم این پروتکل وابسته به مقدار RTT است. اگر این مقدار خیلی کم باشد، throughput از bandwidth کمتر خواهد شد و زمانی که زیاد باشد، Overrun connection می شود.

1. RTT در این پروتکل به فاصله زمانی بین ارسال یک بسته تا دریافت آن گفته می شود و به همین دلیل به واقعیت نزدیک تر است نسبت به RTT که در tcp reno گرفته می شود.

2. بعد از dup ack و دریافت اولین و دومین ACK باید با توجه time out بسته ای که ack آن دریافت می شود باید بسته را retransmit کرد.

3. در vegas وقتی که یک بسته loss می شود CWND آن کم می شود ولی برای تمام بسته هایی که ارسال آنها مربوط به قبل از دریافت dupack اولین بسته است CWND تغییری نمی کند. به طور مثال وقتی بسته ۲ loss می شود و dupack آن دریافت می شود اگر بسته ۳ قبل از dupack2 فرستاده شده باشد و loss شده باشد در این صورت با دریافت dupack3 مقدار CWND تغییری نمی کند.

4. Congestion از طریق مقایسه actual throughput و expected throughput مشخص می شود. (نحوه محاسبه این مقادیر را می توانید در لینک ببینید). مقادیر آلفا و بتا را به طور اختیاری (ولی منطقی!) انتخاب کنید. تغییر مقدار CWND هم با استفاده از MSS انجام شود.

5. Slow Start هم با مقایسه دو مقدار actual و expected throughput انجام می شود. اگر این مقدار از threshold که به طور اختیاری انتخاب می شود، بزرگتر باشد، فرض بر این است که congestion رخ داده و در غیر اینصورت CWND به طور نمایی افزایش میابد.

سایر قسمت هایی از congestion control که بیان نشده است مشابه tcp reno می باشد.

#### 4. TCP Connection Termination

برای قطع ارتباط بین دو طرف، لازم است از این پروتکل استفاده شود. (برای این قسمت می توانید از این [لینک](#) کمک بگیرید.)

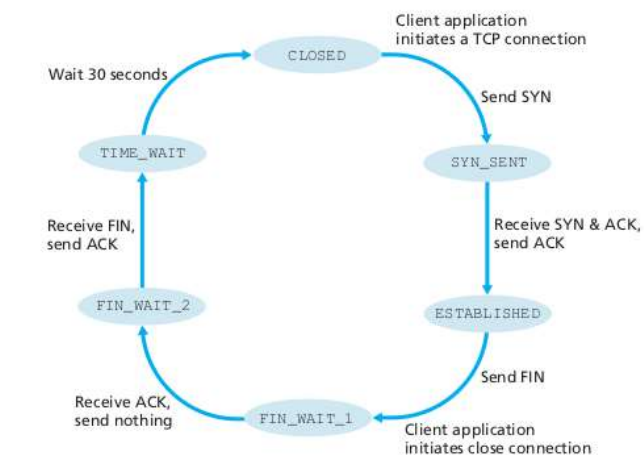


Figure 3.41 • A typical sequence of TCP states visited by a client TCP

نکات:

1. کد پروژه از این [لینک](#) قابل دسترسی است.
2. برای ارسال داده ها تنها از کلاس EnhancedDatagramSocket استفاده کنید.
3. Payload بسته های ارسالی نباید بیشتر از ۱۴۸۰ بایت باشد.
4. این پروژه به صورت گروهی است.
5. لطفا در صورتی که هر ابهامی درباره پروژه دارید قبل از سررسید پروژه بپرسید!!