

به نام خدا

محمد مهدی آقاجانی

تمرین چهارم شبکه های کامپیوتری

استاد : دکتر صادقیان

### سوال سه

مکمل یک آن می شود : ۱۱۰۱۰۰۱۱

همچنین از مکمل یک استفاده میکنند که در آخر اگر صفر نبود بدانند خطایی رخ داده است و تشخیص آن با

یک OR ساده امکان پذیر می باشد که سخت افزاری ست و بسیار سریع است.

این نحوه تشخیص فقط تشخیص میدهد که خطایی رخ داده است و نمیتواند محل خطا را مشخص کند از

طرفی یک burst خطا را میتواند تشخیص دهد و اگر خطا بزرگتر از یک burst باشد ممکن است تشخیص

ندهد.

### سوال چهارده

۸ میکرو ثانیه طول میکشد تا بسته ارسال گردد . برای اینکه ارسال کننده بیش از ۸۰ درصد مواقع مشغول باشد باید :

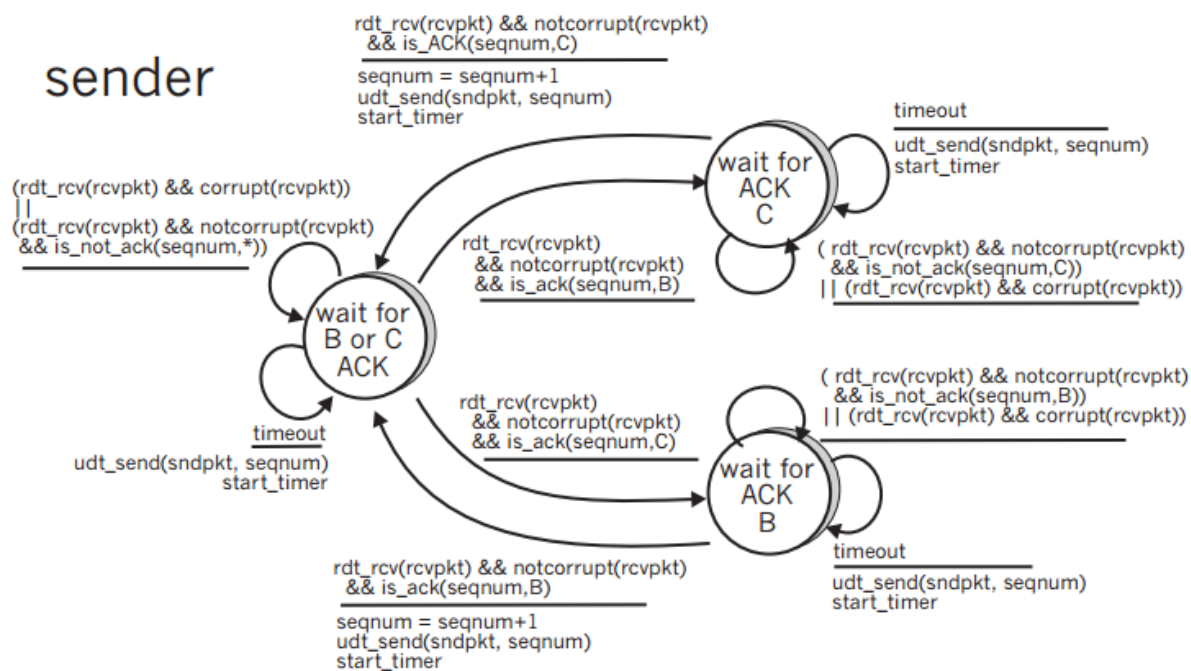
$$util = (0.8) = \frac{0.008n}{30.016}$$
$$n = 3376$$

در نتیجه اندازه پنجره باید حدود ۳۳۷۶ بسته باشد.

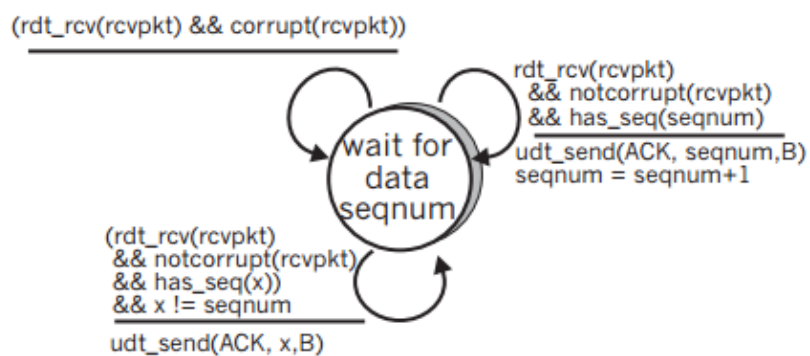
## سوال شانزده

این مساله شبیه همان حالت stop and wait می باشد که باید اندکی آن را تغییر بدهیم. علت آن هم وجود امکان گم شدن بسته در کانال و همچنین ارسال بسته ای از سوی فرستنده است که ممکن است گیرنده آن را قبلا در یافت کرده باشد. در اینجا به sequence number احتیاج داریم که همان حالت 0-bit کافی می باشد.

FSM ها به صورت زیر خواهند بود :



## receiver B



### سوال هجده

A) فرض کنید که گیرنده بسته  $k-1$  را دریافت کرده است و این بسته و تمامی بسته های ماقبل را  $ack$  فرستاده است. اگر همه این  $ack$  ها توسط فرستنده دریافت شود آنگاه پنجره فرستند می شود  $[k, k+N-1]$ . حال فرض کنید هیچ یک از این  $ack$  ها به دست فرستنده نرسد در این صورت سایز پنجره عبارت است از  $[k-N, k-1]$ . در نتیجه شماره ها در رنج  $[k-N, k]$  خواهد بود.

B) اگر گیرنده برای بسته  $k$  صبر کند در این صورت یعنی بسته  $k-1$  را دریافت کرده و آن را  $ack$  کرده همچنین  $N-1$  بسته قبلی را نیز دریافت و  $ack$  کرده است. اگر هیچ یک از این  $ack$  ها دست فرستنده نرسیده باشد یعنی در حال انتشار ند. چون فرستنده بسته های  $[k-N, k-1]$  را فرستاده است در نتیجه  $ack$  برای بسته  $k-N-1$  را دریافت کرده است در نتیجه گیرنده هیچ گاه  $ack$  با شماره کمتر از  $k-N-1$  نمیفرستد در نتیجه رنج  $ack$  های در حال انتشار بین  $k-N-1$  تا  $k-1$  خواهد بود.

## سوال بیست

از آنجایی که کانال A-to-B یک کانال است که ممکن است بسته ها را از دست بدهد ، A نیاز دارد تا timeout بگذارد و retransmit کند . از آنجایی که تاخیر کانال نامشخص و متغیر است ممکن است که A یک بسته را دو بار بفرستد در حالیکه B اولی را دریافت کرده است در نتیجه به sequence number نیز احتیاج داریم. یک شمارنده 1-bit برای این مساله کافی ست.

ماشین حالت آن دارای حالات زیر است :

- Wait for request 0 from above : در این حالت درخواست دهنده اطلاعات منتظر است تا از لایه بالاتر داده بیاید. در این حالت وقتی درخواست را دریافت کرد یک پیام R0 به B ارسال میکند و یک تایمر را ست میکند و به حالت wait for D0 میرود. در این حالت A هر پیام ارسالی از سمت B را ignore میکند.
- Wait for D0 : در این حالت صبر میکند که D0 از سوی A ارسال گردد . اگر پیام D0 از سمت B آمد به حالت wait for request 1 from above میرود. در این حالت اگر D1 را بگیرد آن را ignore میکند.
- Wait for request 1 from above : در این حالت باز دوباره برای درخواست از لایه بالاتر صبر میکند و هر گاه درخواست رسید پیام R1 را به B میفرستد و به حالت wait for D1 میرویم.
- Wait for D1 : در این حالت منتظر یام D1 میماند و اگر دریافت کرد دوباره به حالت wait for request 0 from above میرود. در این حالت اگر پیام D0 را دریافت کند آن را ignore میکند.

حال خود B هم دو حالت می تواند داشته باشد :

- Send D0 : در این حالت صبر میکند تا در خواست R0 بیاید و وقتی آمد داده D0 را ارسال میکند .  
در این حالت وقتی مطمئن شد که درست ارسال شده به حالت بعدی یعنی send D1 میرود.
- Send D1 : در این حالت منتظر دریافت درخواست R1 می باشد و وقتی دریافت کرد پیام D1 را ارسال میکند. هر وقت از درست رسیدن آن اطمینان کرد به حالت send D0 میرود.