

# LC PROJECT

TERM 13991 1399/11/20

*Radmehr karimian 98103556*

## بخش اول:

- 1) بخش preamble شامل 7 بایت می باشد که در واقع به ما نشان می دهد که ورودی و فریم اینترنت در حال آمدن است. هر بایت preamble در واقع شامل 10101010 می باشد و به دستگاه ما کمک میکند که تشخیص دهد ورودی ما ایترت می باشد..
- 2) بخش بعدی ما SFD می باشد که در واقع به ما نشان میدهد که دیگر بخش header فریم ما در حال اتمام است و در بیت های بعدی مک آدرس های ما دریافت خواهند شد.  
این بخش شامل یک بایت بوده و دارای الگوریتم شبیه preamble بوده با این تفاوت که در بیت آخر دچار تفاوت هستند یعنی بیت ما به شکل 10101011 خواهد بود.
- 3) بخش مک آدرس های ما شامل دو بخش است.  
بخش نخست: destination mac address که در واقع شامل شیش بایت می باشد و به ما میگوید که فریم ارسال شدی برای چه دیوایس و دستگاهی می باشد. اگر دستگاهی که فریم را دریافت کرده مک آدرس مشابهی داشته باشد به بررسی ادامه فریم می پردازد در غیر این صورت فریم توسط دستگاه رد می شود.  
بخش دوم : source mac address که در واقع شامل شیش بایت می باشد و به ما میگوید که یک اطلاعات ما توسط چه دستگاهی فرستاده شده است.  
در واقع این مک آدرس ها برای سینک کردن کلاک های دو دستگاه به ما کمک می کند.

4 و 5) بخش ethertype ما شامل دو بایت بوده و در واقع به ما مشخص میکند که چه نوع فریم اینترنتی فرستاده شده و از چند بایت تشکیل شده است. در سیستم های جدید و امروزی به صورت کلی از استاندارد های ipv4 و ipv6 استفاده می شود. هر چند نوع های بسیاری از فریم های اینترنت وجود دارد.

به طور کلی به وسیله ethertype میتوان اندازه و نوع فریم را مشخص کرد و با آن تعداد بایت های موجود در payload را شناسایی کرد.

6. fcs شامل 4 بایت می باشد که در واقع به ما در تشخیص و پیدا کردن خطا کمک میکند.

7. فرض کنید داده ما 1101011111 و چند جمله ای ما  $x^4+x+1$  باشد

آنگاه داریم:

در واقع با توجه به اینکه باقی مانده دیتا ما با fcs برابر است خواهیم یافت که آیا دچار ارور شده است یا نه

که این ارور معمولاً با بخشی به عنوان tcp بررسی و رفع می شود اما با فرض وجود یک خطا می توان آن را بدست آورد و تصحیح کرد.

Datu:

1101011111

poly:

$$x^4 + x^3 + 1 = 10011$$

← 10011

Data : 110101111100000

10011 | 110101111100000

10011

10011

10011

011110

10011

11010

10011

10010

10011

fcs

← 10 = R

00010 ← 00000  $\xrightarrow{N}$  10011

→ R(10011 | 110101111100010) = 0

توضیحاتی در مورد بخش دوم:

به طور کلی در ابتدا کد را میگیریم بررسی میکنیم که آیا ما 7 بار ورودی 10101010 را داده ایم یا خیر و اگر نداده بودیم شمارش را دوباره از اول انجام میدهیم (اینکار به کمک متغیر i انجام می شود).

سپس برای بررسی 10101011 ابتدا بررسی میکنیم که آیا ابتدا 7 بار 10101010 را ورودی گرفته ایم یا نه اگر گرفته بودیم به بررسی می پردازیم. لازم به ذکر است که اگر در اینجا نیز کد دیگری دریافت کنیم فوراً I را صفر میکنیم. و اگر دریافت کردیم I را یک میکنیم.

سپس در مرحله بعد چنانچه مقدار I درست بودند مک آدرس ها را دریافت میکنیم و پس آن ethertype را.

حال بعد از دریافت ethertype با توجه به مقدار و طول آن داده دریافت میکنیم و سپس fcs را نیز دریافت میکنیم.

سپس پس از آن داده ها را به ترتیب از بیشترین به کمترین ارزش مرتب کرده و عملیات تشخیص و تصحیح را انجام میدهیم.

لازم به ذکر است که چون ما میدانیم که یک بیت خطا داریم میتوانیم با تغییر بیت به بیت و پیدا کردن باقی مانده به روش crc خطا را تصحیح کنیم.

توضیحات بیشتر در خود کد داده شده است.