

**پاسخ سوال یک :**

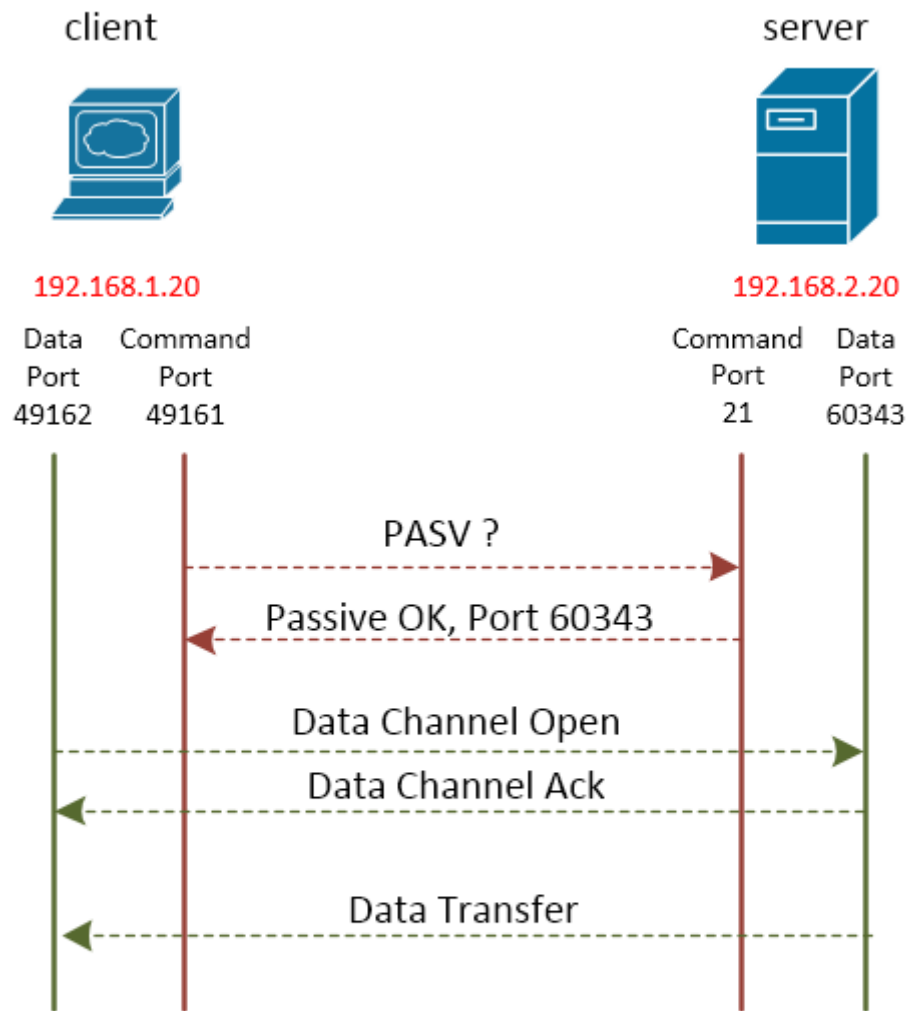
ftp به صورت کلی از طریق دو کانال بین کلاینت و سرور ارتباط برقرار میکند. کانال فط فرمان(command channel) و کانال داده(data channel). این دو کانال کانکشن های جدایی برقرار میکنند، کانال فط فرمان برای دادن دستورات لازم می باشد و کانال داده ها برای انتقال فایل ها می باشد.

در حالت active، کلاینت با کانال فط فرمان ارتباط برقرار میکند (پورت کلاینت میتواند X باشد و پورت سرور ۲۱ می باشد) اما سرور، کانال داده ها را از پورت ۲۰ برقرار میکند (پورت کاربر هرچی میتواند باشد).

در حالت passive کلاینت ارتباط با هر دو کانال رو همزمان برقرار میکند. در این صورت سرور به کلاینت میگه که از کدوم پورت برای کانال داده استفاده کنه.

حالت passive عموماً در شرایطی استفاده میشه که سرور FTP توانایی برقراری کانال داده ها رو نداشته باشه. یکی از دلایل اصلی برای برقرار نشدن ارتباط استفاده از firewall می باشد. ممکن است شما قانون های تعریف کنید که به شما اجازه باز کردن ftp.asklinux.ir رو بده. (یعنی کانال فط فرمان برقرار بشه) ولی سرور به دلیل استفاده از دیوار آتش توانایی ارسال اطلاعات از کانال داده ها و عبور آنها از دیوار آتش رو نداشته باشه. در این حالت ftp passive میاد هر دو کانال رو طرف خود کاربر باز میکند که مشکلی از فایروال بوجود نیاد. در این حالت همواره هر دو کانال باز هستند.

## FTP PASSIVE MODE



Response Code: **Entering Passive Mode (227)**

Response Arg: **(192,168,2,20,235,183)**

Passive IP address: 192.168.2.20

Passive TCP port: **60343** = **235** \* (256) + **183**

*FTP passive mode*

## پاسخ سوال دو :

پروتکل (tcp) transmission control protocol یک پروتکل ارتباطی (connection oriented) است و عملکرد آن بدین صورت است که برای هر پکت ارسالی توسط کامپیوتر مبدا باید یک پکت ack از سرور مقصد مبنی بر دریافت صحیح و بدون نقص پکت دریافت کند . اگر طی زمان مشخصی این پیام ack توسط مبدا دریافت نگردد فرایند ارسال پکت مجدداً تکرار خواهد شد و کاربر آن بیشتر در مواردی است که نیاز به اطمینان از صحت انتقال اطلاعات داریم مانند پروتکل های http , SMTP , pop3 , FTP , Telnet , ...

پروتکل (UDP) User datagram protocol یک پروتکل بدون ارتباطی (connection less) است . بر خلاف tcp در این پروتکل هیچ گونه پیام ack مبنی بر دریافت پکت از سوی سرور ارسال نشده و بیشتر در مواردی مانند انتقال صوت voip یا ویدئو که پهنای باند در این موارد از اهمیت بالایی برخوردار است بکار می رود زیرا در صورت استفاده از پروتکل tcp جهت انتقال این ترافیک هر پیام ack به ازای دریافت پکت خود باعث اشغال پهنای باند خواهد شد .

FTP مخفف File Transfer Protocol است. FTP هم نام یک سری از برنامه هاست است و هم روشی برای انتقال اطلاعات بین کامپیوتر هاست. Anonymous FTP قابلیتی است که در نرم افزارهای FTP وجود دارد و به استفاده کننده این اجازه را می دهد تا فایلها را از هزاران کامپیوتر Host (اجاره دهنده فضای وب سایت) روی اینترنت به کامپیوتر شخصی خود انتقال دهند. سایتهای FTP شامل کتابها، مقالات، نرم افزار، بازیهای کامپیوتری، عکس، موزیک، فیلم و ... می باشند.

این سرویس تقریباً از طریق تمامی میزبانان وب در اختیار کاربران خود گذاشته می شود تا آنها بتوانند از طریق نرم افزارهای مخصوص این کار، برای انتقال فایل ها استفاده نمایند.

نکته :

• پروتکلی که طبق آن سایت و یا فایل ها را بر روی هاست آپلود می کنند .

• در سال ۱۹۷۰ در اینترنت توسعه یافت.

مانند HTTP که محتوای وب را منتقل می کند یا SMTP که ایمیل ها را منتقل می کند FTP هم ساده ترین راه برای تبادل فایل از یک کامپیوتر به کامپیوتر دیگر می باشد.

پروتکل HTTP یا Hyper Text Transfer Protocol اصطلاحاً به پروتکلی گفته می شود که برای ایجاد ارتباط، دریافت، و ارسال داده ها بین سرور و کلاینت استفاده می شود. این پروتکل از پروتکل TCP/IP برای بستن پلی میان سرور و کلاینت استفاده می کند. طریقه کار ارتباط کلاینت با سرور، با استفاده از پروتکل HTTP به این ترتیب است که داده ها، از طریق بسته های اطلاعاتی، بین سرور و کلاینت رد و بدل می شود. به این ترتیب که برای برای ارسال داده ای به سمت مقصد، در ابتدا، داده، به بخش های کوچکتری شکسته می شود و سپس از هر کدام به سمت مقصد و با ترتیب مشخص ارسال می شوند.

### مزایای بیشتری از پروتکل HTTP

لازم به ذکر است که این پروتکل از Handshaking برای ارسال و دریافت داده ها استفاده می کند. در Handshaking برای شروع و پایان عملیات دریافت و ارسال بسته های اطلاعاتی بین سرور و کلاینت، سؤالاتی از مقصد پرسیده می شود.

بعنوان مثال فرض کنید کلاینت A می خواهد از سرور B داده هایی را دریافت نماید. برای این منظور A در ابتدا درخواستی را به B می فرستد تا آمادگی B را برای ارسال پرسوجو نماید. در اینمالت A صبر خواهد کرد تا B جوابی را به این پرسش ارسال نماید. پس از آنکه B آمادگی را خود اعلام کرد، A پیامی را سمت B می فرستد و از او می خواهد تا داده های اطلاعاتی را بفرستد. B پیام را دریافت می کند و اطلاعات را آماده می کند و آن را به بسته های کوچکتری تقسیم می کند و توسط پروتکل Handshaking آنها را پشت سر هم و به ترتیب ارسال می کند.

لازم به ذکر است که پروتکل TCP/IP که در پروتکل HTTP مورد استفاده قرار می گیرد، امنیت داده ها و تضمین ارسال آنها را تماماً بر عهده گرفته است. این پروتکل بر روی لایه هایی قرار دارد و معماری این پروتکل بصورت چند لایه است.

for Server-Client :  $D_{cs} = \max \left\{ \frac{NF}{\mu_s}, \frac{F}{d_{min}} \right\}$  (a)

for P2P :  $D_{p2p} = \max \left\{ \frac{F}{\mu_j}, \frac{F}{d_{min}}, \frac{NF}{\mu_j + \sum_{i=1}^N \mu_i} \right\}$

Client - Server :

$\mu \backslash N$	10	100	1000
300 Kb	7680	51200	512000
700 Kb	7680	51200	512000
2 Mb	7680	51200	512000

(b)

P2P :

$\mu \backslash N$	10	100	1000
300 Kb	7680	25904	47559
700 Kb	7680	15616	21525
2 Mb	7680	7680	7680

### • مقیاس پذیری معماری P2P

برای مقایسه معماری مشتری- سرور با رقیب آن یعنی پروتکل نظیر به نظیر (P2P) و نیز مشخص نمودن خود مقیاس پذیری درونی (ذاتی) P2P، از یک الگوی کمیته ساده برای توزیع یک فایل به یک مجموعه ثابت از نظیرها برای هر دو نوع معماری بهره می گیریم. همانگونه که شکل ۲- ۲۴ نشان می دهد سرورها و نظیرها با استفاده از پیوندهای دستیابی به اینترنت به یکدیگر متصل شده اند. اگر نرخ یا سرعت بارگذاری پیوند دستیابی سرور را  $\mu_s$  بنامیم و نرخ بارگذاری پیوند دستیابی  $i$  آمین نظیر را  $\mu_i$  و نیز سرعت بارگیری پائین پیوند دستیابی همان نظیر را  $d_i$  و سرانجام اگر اندازه ی یک فایل که بایستی توزیع گردد  $F$  بیت و تعداد نظیرهایی که مایل به نسخه برداری از فایل می باشند  $N$  در نظر گرفته شود. زمان توزیع فایل عبارت است از مدت زمانی که نسخه ای از فایل به همگی  $N$  نظیر خواهد رسید. در بررسی زمان توزیع برای هر دو معماری مشتری- سرور و نظیر به نظیر از یک فرض ساده و کلی نسبتاً دقیق استفاده شده است و آن این موضوع است که هسته ی اینترنت را از لحاظ پهنای باند (سرعت) وافر می داند. به عبارت دیگر تنگراه ها را فقط در دستیابی شبکه می داند. همچنین فرض بر آن است که سرور و مشتری ها در هیچیک از کاربردهای شبکه شرکت ندارند، بنابراین همه ی سرعت عملیات بارگذاری به طرف بالا و بارگذاری های پائین فقط به توزیع این فایل اختصاص یافته است. ابتدا زمان توزیع فایل برای معماری مشتری- سرور که  $D_{cs}$  نامگذاری شده است، محاسبه می گردد. در معماری مشتری- سرور هیچیک از نظیرها در توزیع فایل کمک کار نخواهند بود البته موارد ذیل نیز در ملاحظات قرار می گیرند:

- سرور بایستی یک نسخه از فایل را به هر یک از تعداد  $N$  نظیر انتقال دهد. بنابراین حجم ارسالی معادل  $NF$  بیت می باشد. از آنجائیکه نرخ بارگذاری سرور  $\mu_s$  است زمان توزیع فایل حداقل  $\frac{NF}{\mu_s}$  می باشد.

• اگر  $d_{\min}$  نرخ بارگیری پائین نظیر با کندترین سرعت ممکن باشد. به عبارت دیگر  $d_{\min} = \{d_1, d_p, \dots, d_N\}$  در نتیجه نظیر با کندترین سرعت نمی‌تواند همه‌ی  $F$  بیت فایل را در زمان کمتر از  $\frac{F}{d_{\min}}$  دستیابی نماید. بنابراین حداقل زمان توزیع در کمترین مقدار ممکن خود برابر  $\frac{F}{d_{\min}}$  می‌باشد.

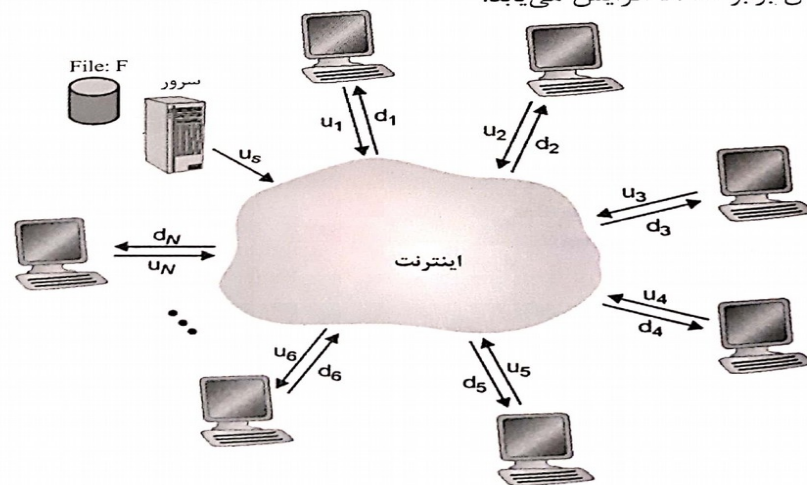
با قرار دادن این دو مورد در کنار یکدیگر رابطه‌ی زیر را می‌توان تعریف نمود:

$$D_{cs} \geq \max \left\{ \frac{NF}{\mu_s}, \frac{F}{d_{\min}} \right\} \quad (۱-۲)$$

این رابطه یک الزام حداقل را برای کمترین میزان زمان توزیع در معماری سرور- مشتری تعریف می‌کند. اگر این نیاز حداقل به‌عنوان زمان توزیع واقعی در نظر گرفته شود آنگاه:

$$D_{cs} = \max \left\{ \frac{NF}{\mu_s}, \frac{F}{d_{\min}} \right\} \quad (۲-۲)$$

از معادله‌ی (۲-۲) چنین نتیجه‌گیری می‌شود که اگر  $N$  به اندازه‌ی کافی بزرگ باشد، زمان توزیع مشتری- سرور از طریق رابطه‌ی  $\frac{NF}{\mu_s}$  بدست می‌آید و بنابراین زمان توزیع به گونه‌ی خطی با تعداد نظیرهای  $N$  افزایش می‌یابد. به‌عنوان مثال اگر تعداد نظیرها از یک هفته به هفته‌ی بعد یک هزار برابر افزایش پیدا کند یعنی مثلاً از ۱۰۰۰ به یک میلیون برسد، آنگاه زمان لازم جهت توزیع فایل به همه‌ی نظیر با یک مقیاس برابر ۱۰۰۰ افزایش می‌یابد.



شکل ۲-۲۴ نمادی از چالش توزیع فایل

( دلیل اضافه کردن این اسکرین شات ها به جای توضیح شفصی شده ، صرفا ضیق وقت است : )

### پاسخ سوال پنج :

اکنون تحلیل مشابهی را برای معماری P2P انجام می‌دهیم. در این حالت هر کدام از نظیرها به تنهایی می‌توانند سرور را در توزیع فایل کمک نمایند. هنگامی که یک نظیر مقداری از داده‌ی متعلق به







یک فایل را دریافت می‌کند، می‌تواند از ظرفیت بارگذاری خود جهت توزیع آن داده به دیگر نظیرها استفاده نماید. محاسبه‌ی زمان توزیع برای معماری P2P کمی پیچیده‌تر از معماری مشتری-سرور است زیرا زمان توزیع وابسته به رفتار هر نظیر در انتقال بخشی از فایل به نظیرهای دیگر دارد. اما در هر حال استنتاج یک عبارت ساده برای حداقل زمان توزیع امکان‌پذیر می‌باشد. برای دستیابی به چنین عبارتی ملاحظات ذیل در نظر گرفته می‌شود:

- در آغاز توزیع، فقط سرور حامل فایل می‌باشد. برای انتقال این فایل به جمعیتی از نظیرها، سرور بایستی هر یک از بیت‌های فایل را حداقل یکبار به درون پیوند دستیابی تزریق کند. بنابراین حداقل زمان توزیع برابر  $\frac{F}{\mu_s}$  می‌باشد. برعکس روش مشتری-سرور با یک بار ارسال بیت از سوی سرور این احتمال قوی وجود دارد که مجدداً نیاز به ارسال آن بیت از سوی سرور نباشد زیرا نظیرها ممکن است به توزیع بیت در بین خود بپردازند.
- مشابه معماری مشتری-سرور یک نظیر با پائین‌ترین نرخ بارگیری نمی‌تواند همه‌ی  $F$  بیت یک فایل را در زمان کمتر از  $\frac{F}{d_{\min}}$  دستیابی نماید. بنابراین حداقل زمان توزیع برابر است با  $\frac{F}{d_{\min}}$ .
- سرانجام مشاهده این نکته که مجموع سرعت بارگذاری در یک سیستم به طور کلی برابر است با مجموع نرخ‌های بارگذاری سرور و هر یک از نظیرها به تنهایی، به عبارت دیگر:

$$\mu_{\text{total}} = \mu_s + \mu_1 + \dots + \mu_n$$

سیستم بایستی بارگذاری  $F$  بیت را به هر یک از  $N$  نظیر انجام دهد. در نتیجه تعداد  $NF$  بیت را انتقال می‌دهد. این عمل برای سرعت بالاتر از اندازه‌ی  $\mu_{\text{total}}$  امکان‌پذیر نمی‌باشد. در نتیجه حداقل زمان توزیع نیز برابر است با:

$$\frac{NF}{(\mu_s + \mu_1 + \dots + \mu_n)}$$

با ترکیب سه مورد توضیح داده شده در بالا می‌توان حداقل زمان توزیع برای P2P را به صورت ذیل تعریف نمود:

$$D_{P2P} \geq \max \left\{ \frac{F}{\mu_s}, \frac{F}{d_{\min}}, \frac{NF}{\mu_s + \sum_{i=1}^N \mu_i} \right\} \quad (3-2)$$

معادله‌ی (۳-۲) یک الزام حداقل را برای کمترین میزان زمان توزیع در معماری P2P نشان می‌دهد. چنین به نظر می‌رسد که اگر تصور شود هر نظیر قادر به توزیع مجدد یک بیت بلافاصله پس از دریافت آن باشد، آنگاه یک الگوی توزیع مجدد که واقعاً بتواند این حداقل الزام را فراهم نماید، امکان‌پذیر می‌باشد. در واقع اگر بخش‌هایی از یک فایل به جای بیت‌های تک توزیع مجدد شوند معادله ۳-۲ تقریب مناسب و سودمندی را از حداقل زمان توزیع نشان می‌دهد. بنابراین با استفاده از الزام حداقل معادله‌ی ۳-۲ کمترین زمان توزیع واقعی برابر است با:

$$D_{P2P} = \max \left\{ \frac{F}{\mu_s}, \frac{F}{d_{\min}}, \frac{NF}{\mu_s + \sum_{i=1}^N \mu_i} \right\} \quad (4-2)$$

پاسخ سوال شش : به دیگر کاربران