

SMTP (Simple Mail Transfer Protocol)

SMTP یک پروتکل لایه ی Application است که از آن برای انتقال امن ایمیل بر بستر TCP استفاده میشود.

SMTP بصورت Client\Server عمل میکند. در هنگام ارسال ایمیل از سوی کلاینت (User Agent)، این بخش در نقش SMTP Client و Mail Server او در نقش SMTP سرور عمل میکند. اما در هنگام ارسال ایمیل از سمت ایمیل سرور کاربر (Sender's Mail Server) به سمت ایمیل سرور گیرنده ی ایمیل (Recipient's Mail Server) به ترتیب در نقش های SMTP Client و SMTP Server عمل میکنند. SMTP پروتکلی قدیمی تر از HTTP است و میتوان گفت که در قلب ساختار سرویس Email قرار دارد. در SMTP محتوای اصلی یا بدنه ایمیل (Body) محدود به کاراکترهای 7 بیتی Ascii است. این ساختار در اوایل دهه 1980 که به تازگی سرویس ایمیل به مرحله کاربرد و پیاده سازی رسیده بود، معقول و طبیعی بود اما در جهان امروز که نیازمند ارسال فایل های ویدیویی، صوت، عکس و ... بر بستر ایمیل هستیم، دیگر پاسخگوی نیاز ما نیست. با توجه به ساختار پروتکل SMTP که عملیات ارسال ایمیل بر عهده ی آن است، مجبور هستیم که این دسته از محتواهای باینری را به کاراکترهای اسکی Code کنیم و سپس آنها را ارسال کنیم. هنگامی که کاربر درخواست Fetch کردن یک ایمیل را میدهد، سرور آنرا در قالب Mime (Mime Type) به Mail Agent کاربر تحویل میدهد و پس از Decode شدن، به کاربر تحویل داده میشود. به بیانی دیگر، طرف گیرنده نیز پس از دریافت این محتوا، آنها را Decode کرده و از محتوای اصلی آن استفاده خواهد کرد. (به این متد Code گذاری Base64 میگویند).

((لازم به ذکر است که HTTP برای انتقال محتوای ویدیویی و امثال آن نیازی به Code کردن محتوا ندارد)).

در انتقال ایمیل از یک Mail Server به Mail Server دیگر، یک اتصال TCP مستقیم بین دو سرور برقرار میشود. اگر سرور مقصد Down شده باشد، ایمیل در داخل سرور مبدا باقی میماند و پس از عملیاتی شدن Mail Server مقصد، از نو تلاش میشود که بین آنها Connection مورد نیاز ایجاد شده و ایمیل تحویل آن گردد.

بررسی مکانیزم عملکردی SMTP :

1- SMTP Client یک TCP Connection روی پورت 25 (پورت پیش فرض برای SMTP) با ایمیل سرور خود برقرار میکند-2- سپس کلاینت و سرور عملیات Handshake لایه ی Application را انجام میدهند که در این گام خود را به همدیگر معرفی میکنند-3- پس از پایان عملیات handshake کلاینت ایمیل را به سمت Smt Server ارسال میکند-4- اگر Smt Client قصد ارسال ایمیل هایی دیگر را داشته باشد، آنها را روی همان Session به سمت سرور ارسال میکند، در غیر این صورت به پروتکل TCP دستور میدهد تا Connection را Close کند.

در زیر یک مثال از تبادل داده بین SMTP Client و SMTP Server را مشاهده میکنیم. در این مثال S:Server و C:Client را مشخص میکند. کلاینت سعی دارد که متنی را از سمت Mail Server خود با نام Crepes.Fr به سمت Mail Server مقصد با نام Hamburgure.edu ارسال کند :

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

کلاینت در این مثال از 5 دستور Quit, Data, Helo, Mail From, RCP To استفاده کرده است. در آخر نیز با یک نقطه که در یک خط واحد وارد میشود، به ارتباط خود با سرور خاتمه میدهد. (CR=Carriage Return + Line Feed)

سرور به ازای دریافت هر دستور یک Response به همراه یک کد (و گاهی نیز پاره ای توضیحات متنی) را برمیگرداند. مثلاً اگر به کمک دستور RCP To قصد ارسال ایمیل به کاربری را داشته باشد ولی برای آن Inbox-Account ی تعریف نشده باشد، سرور پیغام No Such User را با کد 251 را برمیگرداند .

دستور (Command)

HELO	Authentication
MAIL	Sender's Name
RCPT	Receiver's Name
DATA	Beginning transmission
SEND	Send directly
SOML	„Send or Mail“
RSET	Reset all Buffers
QUIT	Finnish session

Client → Server

کد بازگشتی

220	Service ready
250	Request mail action ok
354	Start mail input
421	Service not available
450	Request action aborted
500	Syntax Error
550	Requested action not taken
551	User not local
554	Transaction failed

Server → Client

برخی دستورات مورد استفاده در پروتکل SMTP :

Helo : کلاینت خودش را معرفی میکند

Mail From : نامه از کجا ارسال میگردد (مبدأ)

RCP TO : نامه به دست چه کسی برسد (مقصد)

Data : محتویات ایمیل

Reset : بازنشاندن کار (Reset Transaction)

Help : تقاضای راهنمای از گیرنده

Turn: جای فرستنده و گیرنده عوض شود

Quit: خداحافظ!

VERFY: آدرس گیرنده کنترل شود.(آیا وجود دارد؟)

NOOP: گیرنده فعالیتی را انجام ندهد. فقط پاسخ بدهد

EXPN: آدرس داده شده را بسط دهد

برای برقراری ارتباط مستقیم و بی واسطه با سرور میتوان روی پورت 25 به سرور Telnet زد و طبق الگویی ساختار یافته و به ترتیب دستورات را وارد کرد.(یعنی: 25 "Mailservers IP\Name" Telnet >> CMD) پس از وارد کردن دستور Telnet باید فوراً یک Response با کد 220 از سمت Mail Server خود دریافت کنیم.

```
C:\Windows\system32\cmd.exe
C:\Users\941299>telnet mail.mahan.aero 25

Telnet mail.mahan.aero
220 MHN001VW143.MahanAir.Aero Microsoft ESMTP MAIL Service ready at Tue, 21 Mar 2017 21:00:18 +0330
```

SMTP یک پروتکل Push است. همچنین ارتباط TCP از سمت ماشین کلاینت که قصد ارسال ایمیل را دارد، ایجاد میگردد.

در SMTP خطوط Header (Header Lines) با یک خط خالی از Body جدا میشوند.(این خط همان CRLF است). در هر خط Header یک کلمه ی کلیدی وجود دارد و در ادامه اش یک (:) و سپس مقدار داده ای مربوط به آن قرار میگیرد. ("Header: Value" Word).

به عنوان مثال: From: Pouria@Mahan.Aero

توجه شود که Header Line ها از دستورات SMTP مجزا هستند، گرچه ممکن است مشابهت هایی در کلمات استفاده شده در آنها وجود داشته باشد. مثلاً (مثلاً From و Mail From). باید توجه داشت که دستورات SMTP بخشی از TCP Handshake هستند ولی Header Line ها بخشی از خود پیام هستند.

برخی از Header ها برای کاربران آشنا هستند، زیرا در هنگام تدوین و ارسال ایمیل از آنها خواسته میشود که مقادیر مربوط به آنها را وارد کنید.(از جمله ی آنها میتوان به Subject, to اشاره کرد). برخی دیگر از Header ها هستند که بصورت اتوماتیک توسط لایه های زیرین سیستم پر میشوند، مواردی چون Date, Receive و...

به این نکته توجه داشته باشید که User Agent (به آن Message Transfer Agent یا به اختصار MTA نیز گفته میشود) مستقیماً با Mail Server سمت مقصد ارتباط برقرار نمیکند، بلکه ایمیل خود را تحویل سرور ایمیل خود میدهد و Mail Server ما به نیابت از کلاینت و به کمک پروتکل SMTP، ایمیل را به دست Mail Server مقصد میرساند.

در صورتی که Mail Server مقصد Down شده باشد، ایمیل کاربر در صف ارسال ایمیل سرور باقی میماند و (مثلاً) هر 30 دقیقه یکبار مجدداً برای ارسال آن اقدام میکند.

توجه شود که یک کلاینت برای خواندن ایمیل های دریافتی نمیتواند از پروتکل SMTP استفاده کند زیرا این پروتکل بصورت Push کار میکند؛ درحالی که برای خواندن ایمیل از روی Mail Server به پروتکلی بر مبنای Pull احتیاج داریم. برای خواندن ایمیل ها از سرور، از پروتکلهایی چون Pop3-Imap4-Mapi و تنظیم آنها بر روی Mail Agent کاربر (چون برنامه ی Microsoft Outlook) استفاده میکنیم.

زمانی که یک ایمیلی به دست SMTP سرور میرسد؛ سرور به کمک MX رکورد Mail Server مقصد (که با استفاده از سرویس DNS آنرا بدست می آورد)، ایمیل را به Mail Server مقصد ارسال میکند.

در ادامه، چند مثال از ارسال ایمیل به کمک پروتکل SMTP همراه با دستورات و Respose کدهای متناظر و توضیحات آنها را مشاهده میکنیم .

SMTP نمونه ارسال یک نامه

```
helo qolamali
250 aicte.com Hello farsilinux.org [220.57.69.37], pleased to meet you
mail from:qolamali@farsilinux.org
250 2.1.0 qolamali @farsilinux.org... Sender ok
rcpt to: khansari@aicte.com
250 2.1.5khansari... Recipient ok
data
354 Enter mail, end with "." on a line by itself
from: :qolamali@aicte.com
to: khansari@aicte.com
subject: testing
Just For Testing, Please Ignore It.
.
250 2.0.0 e1NMajH24604 Message accepted for delivery
Quit
221 2.0.0 mehr.aicte.com closing connection
Connection closed by foreign host.
```

سلام، من فلانی هستم!
که از farsilinux.org تماس میگیرم

سلام، من aicte.com
از ملاقات شما خوشوقتم

SMTP

نمونه ارسال یک نامه

```
helo qolamali
250 aicte.com Hello farsilinux.org [220.57.69.37], pleased to meet you
mail from:qolamali@farsilinux.org
250 2.1.0 qolamali @farsilinux.org... Sender ok
rcpt to: khansari@aicte.com
250 2.1.5khansari... Recipient ok
data
354 Enter mail, end with "." on a line by itself
from: :qolamali@aicte.com
to: khansari@aicte.com
subject: testing
Just For Testing, Please Ignore It.
.
250 2.0.0 e1NMajH24604 Message accepted for delivery
Quit
221 2.0.0 mehr.aicte.com closing connection
Connection closed by foreign host.
```

می خواهم نامه ای ارسال نمایم

مشکلی در ارسال ندارم!

SMTP

نمونه ارسال یک نامه

```
helo qolamali
250 aicte.com Hello farsilinux.org [220.57.69.37], pleased to meet you
mail from:qolamali@farsilinux.org
250 2.1.0 qolamali @farsilinux.org... Sender ok
rcpt to: khansari@aicte.com
250 2.1.5khansari... Recipient ok
data
354 Enter mail, end with "." on a line by itself
from: :qolamali@aicte.com
to: khansari@aicte.com
subject: testing
Just For Testing, Please Ignore It.
.
250 2.0.0 e1NMajH24604 Message accepted for delivery
Quit
221 2.0.0 mehr.aicte.com closing connection
Connection closed by foreign host.
```

آدرس گیرنده را اعلام می نمایم

بفرست!

SMTP

نمونه ارسال یک نامه

```
helo qolamali
250 aicte.com Hello farsilinux.org [220.57.69.37], pleased to meet you
mail from:qolamali@farsilinux.org
250 2.1.0 qolamali @farsilinux.org... Sender ok
rept to: khansari@aicte.com
250 2.1.5khansari... Recipient ok
data
354 Enter mail, end with "." on a line by itself
from: :qolamali@aicte.com
to: khansari@aicte.com
subject: testing
Just For Testing, Please Ignore It.
.
250 2.0.0 e1NMajH24604 Message accepted for delivery
Quit
221 2.0.0 mehr.aicte.com closing connection
Connection closed by foreign host.
```

می خواهیم متن نامه را بنویسیم

نامه را بنویس ولی یادت باشد که
آخر نامه، " " را وارد کنی

SMTP

نمونه ارسال یک نامه

```
helo qolamali
250 aicte.com Hello farsilinux.org [220.57.69.37], pleased to meet you
mail from:qolamali@farsilinux.org
250 2.1.0 qolamali @farsilinux.org... Sender ok
rept to: khansari@aicte.com
250 2.1.5khansari... Recipient ok
data
354 Enter mail, end with "." on a line by itself
from: :qolamali@aicte.com
to: khansari@aicte.com
subject: testing
Just For Testing, Please Ignore It.
.
250 2.0.0 e1NMajH24604 Message accepted for delivery
Quit
221 2.0.0 mehr.aicte.com closing connection
Connection closed by foreign host.
```

نامه ام تمام شد

نامه ات را برای ارسال پذیرفتم

SMTP

نمونه ارسال یک نامه

```
helo qolamali
250 aietc.com Hello farsilinux.org [220.57.69.37], pleased to meet you
mail from:qolamali@farsilinux.org
250 2.1.0 qolamali @farsilinux.org... Sender ok
rcpt to: khansari@aietc.com
250 2.1.5khansari... Recipient ok
data
354 Enter mail, end with "." on a line by itself
from: :qolamali@aietc.com
to: khansari@aietc.com
subject: testing
Just For Testing. Please Ignore It.
.
250 2.0.0 e1NMajH24604 Message accepted for delivery
Quit
221 2.0.0 mehr.aietc.com closing connection
Connection closed by foreign host.
```

ممنون، خدا حافظ

خدا حافظ، ارتباط را قطع می کنم

((برای تحقیق بیشتر در خصوص پروتکل SMTP میتوان به RFC 1425 و نسخه های توسعه یافته ی این پروتکل (ESMTP مراجعه نمود)))

POP3 (Post Office Protocol-Version 3)

این پروتکل ساختار بسیار ساده ای دارد و در RFC-1939 تعریف شده است. از این پروتکل برای دانلود ایمیلها از روی سرور استفاده میشود. کارایی و قابلیت های این پروتکل به نسبت محدود میباشد. در ابتدا User Agent با ایجاد یک Connection روی پورت 110 (پورت پیش فرض)، 3 گام را به ترتیب پشت سر میگذارد. 1-Authorization-2-Transaction-3-Update.

در گام نخست از این پروتکل، کاربر با ارسال Username\Password خود بصورت Clear خود را به سرور Authenticate میکند. در گام دوم کاربر ایمیلهای موجود در Inbox خود را Retrieve (بازیابی) مینماید. کاربر میتواند برخی ایمیل ها را برای Delete شدن Mark کند و یا آماری از وضعیت-سایز و تعداد ایمیل های موجود بر روی Inbox خود را دریافت کند. 3-این گام (Update) وقتی صورت میگیرد که Client دستور Quit را صادر کند تا به Session خاتمه بدهد. در این لحظه سرور ایمیل هایی را که برای Delete شدن علامت گذاری شده اند را پاک مینماید.

کلاینت در پروسه ی Transaction بین User Agent و Pop3 Server یک سری دستورات را اجرا میکند که برای هر دستور یک Response تولید خواهد شد. این Response میتواند +Ok یا Err - باشد.

در مثال زیر روی پورت 110 به Mail Server خود Telnet زده ایمو با رایه ی Username\Password خود را به سرور Authenticate کرده ایم.

```
telnet mailServer 110
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

در مثال زیر ،پس از برقراری ارتباط Telnet با سرور،نمونه تبادلات داده ای در فاز Transaction را در پروسه ی لیست کردن و خواندن ایمیلها و سپس Mark کردن ایمیلها برای Delete شدن از روی سرور را مشاهده میکنیم.

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: (blah blah ...
S: .....
S: .....blah)
S: .
C: dele 1
C: retr 2
S: (blah blah ...
S: .....
S: .....blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

توجه شود که پروتکل Pop3 را میتوان در دو Mode مختلف Download-and-Keep و Download-and-Delete تنظیم کنیم. در مثالی که در بالاتر دیدیم، بر طبق متد Download-And-Delete عمل شده بود.

مشکلی که در حین کار کردن با پروتکل POP3 و در حالت Download-And-Delete مشاهده گردد این است که کاربر تنها بر روی یک سیستم قادر به خواندن اطلاعات است و پس از خوانده شدن اطلاعات، ایمیل از روی سرور پاک میگردد و در صورتی که بخواهد از چند سیستم مختلف برای خواندن ایمیلهايش استفاده نماید، قادر به این کار نخواهد بود.

در طول یک Pop3 Session, بین User Agent و Mail Server, سرور دسته ای از اطلاعات وضعیتی کاربر (State Information) را نگهداری میکند. (مشخصاً چک میکند که کلاینت کدامیک از ایمیل‌هایش را برای پاک کردن Mark کرده است). سرور Pop3 این اطلاعات را روی همان Pop3-Session برقرار شده منتقل نمیکند. این مشخصه, پیاده سازی آن را بسیار راحت تر کرده است.

لازم به ذکر است که پروتکل Pop3 نسبت به پروتکل‌های دیگر Mail Access قدیمی تر بوده و بر روی سیستم‌هایی که قدیمی بوده و از پروتکل‌های جدید چون IMAP4 یا MAPI پشتیبانی نمیکنند, میتوان از آن استفاده کرد. Pop3 از Flag هایی چون Read/Urgent/Unread پشتیبانی میکند.

IMAP4(Internet Message Access Protocol-Version 4)

با استفاده از پروتکل IMAP جهت دریافت ایمیل‌ها, این امکان میسر است که آنها را بر روی چند سیستم (MTA) مختلف (جدای از مرورگر وب), دریافت نمود. زمانی که کاربر از پروتکل Pop3 استفاده میکند, این امکان را دارد که تنها پس از دانلود ایمیل‌ها بر روی ماشین خود, پوشه ای ایجاد کرده و ایمیل‌هایش را دسته بندی و Folder بندی کند (یا آنها را Delete نماید). همچنین خواهد توانست در بین ایمیل‌های دریافتی خود بر اساس نام Subject\Sender به جستجوی ایمیل‌هایی خاص بپردازد. گاهی کاربران مایلند که ساختار پوشه بندی (و دسته بندی) ایمیل‌ها را روی سرور انجام دهند و از هر سیستمی که به سرور وصل میشوند, بتوانند این ساختار را حفظ نمایند. به عبارتی پوشه بندی و دسته بندی ایمیل‌ها را روی سرور انجام دهند. این کار از عهده ی Pop3 ساخته نیست. برای حل این مشکل و پاره ای دیگر از نواقص موجود در Pop3 پروتکل IMAP4 در RFC 3501 معرفی شد. قابلیت‌های IMAP4 خیلی بیشتر از Pop3 است و به مراتب پروتکلی پیچیده تر است. ابتدا که ایمیل‌ها دریافت میشوند به پوشه ای Inbox کاربر منتقل میشوند اما کاربر میتواند پوشه ای را ساخته و با تعریف Rule مربوط به آن, به IMAP Server دستور بدهد که ایمیل‌های دریافتی متناظرش را مستقیماً به داخل آن پوشه منتقل کند. برخلاف Pop3 این پروتکل اطلاعات وضعیتی کاربر را روی همان Session که برای ارتباط با سرور Establish شده است نگهداری و تبادل میکند. (مثلاً نام پوشه و اینکه چه ایمیل‌هایی به چه پوشه‌هایی مربوط میشوند).

یکی از ویژگی‌های مهم IMAP4 این است که به کاربر این امکان را میدهد تنها بخش‌هایی از پیام چند بخشی را (Multi part Mime) بجای خواندن کل پیام, دریافت کند. این ویژگی مخصوصاً برای حالت‌هایی که پهنای باند Connection پایین است و ایمیل محتوی Video, Voice, Image و ... است میتواند بسیار موثر و مفید باشد.

📌 در IMAP میتوان پیام‌ها را آزادانه بین MailBox های مختلف متعلق به یک یا چند اکانت Copy, یا Move کرد.

همچنین پیام‌ها را میتوان بین کلاینت و سرور جابجا نمود.

هنگامی که یک ایمیل مبتنی بر IMAP بر روی سرور ذخیره میشود، میتوانیم پیام ها را روی ماشین کلاینت Cache کنیم تا بتوان بصورت Offline از آنها بهره برد (آنها را خواند). در این حالت پاک کردن Cache به معنی پاک کردن خود پیام نیست. کاربر این امکان را دارد که ایمیل های دلخواه خود را به داخل یک Public Folder انتقال بدهد تا سایر کاربرانی که به آن Public Folder دسترسی دارند، بتوانند آن ایمیل را مشاهده نمایند. در ساختار Public Folder ها میتوان Flag هایی را به نحوی تعریف کرد که برای هر User مجزا باشد یا اینکه بین کاربرانی که به آن Public Folder دسترسی دارند Sahre شده باشد. مثلاً اگر ایمیلی برای یک Public Folder ارسال گردد و یک نفر از بین اعضای که آن Public Folder آن ایمیل خاص را دیده باشد، آن ایمیل برای همه ی کاربرانی که به آن Public Folder دسترسی دارند بصورت Read درآید. ((مثلاً در حالتی که یک شرکت به مشتریان خود سرویس خاصی را از طریق ایمیل ارائه میدهد ... مثلاً پشتیبانی فنی و سوال و جواب و ...)).

پروتکل IMAP به علت اینکه ایمیل ها را روی سرور قرار میدهد و روی ماشین کلاینت Cache میکند، فضای ذخیره سازی و بار پردازشی بیشتری رروی Mail Server قرار میدهد.

همچنین دسته ای از Attribute ها را در قالب دستورات دیگر، تبادل میکند و تحویل کاربر میدهد. (Attribute هایی چون سائز پیام Flag، هایی چون Seen, Answerd, Recent). این Flag ها برای Sync کردن کلاینت (User Agent) و سرور کاربرد دارد. ممکن است که بعدها کلاینت با صدور فرمان Expunge همه ی ایمیل های قبلی خود را از پوشه ی Trash\Deleted Items بطور کامل (یعنی از روی سرور) پاک کند. این اتفاق باید به واسطه ی یک Flag به اطلاع سرور برسد.

در IMAP از دستورات مختلفی بهره برد که در زیر تعدادی از آنها ذکر شده اند :

Login, Authenticate, Select, Examine, Close, Logout, Fetch, Store, Delete, Expunge

همچنین سرور میتواند Response های مختلفی را در پاسخ به دستور صادر شده از سمت کاربر داشته باشد، مواردی چون No به معنی اینکه کلاینت مجوز اجرای آن دستوری که وارد کرده است را ندارد، یا Bad به معنی اینکه دستور با فرمت درستی وارد نشده است و یا Ok داشته باشد.

Web Based Email

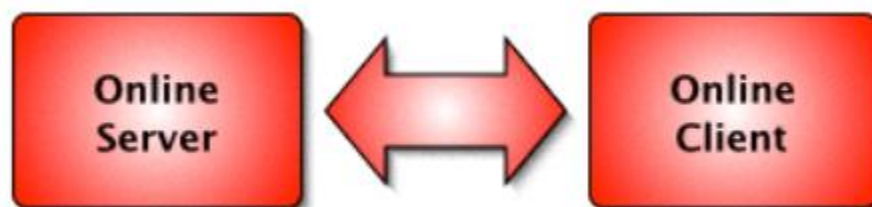
در دنیای امروز استفاده از مرورگرهای اینترنتی برای مشاهده ی ایمیلها امری رو به رشد است. در این حالت UserAgent یگ مرورگر بوده و کاربر به کمک پروتکل HTTP/HTTPS با سرور ارتباط برقرار کرده و محتویات Inbox خود را مشاهده میکند. (بجای ... Pop3\Imap4). در این حالت، هنگامی که User ی قصد ارسال ایمیلی را دارد، این ایمیل را به واسطه ی پروتکل

HTTP\HTTPS (بجای استفاده از SMTP) به سمت کاربر نهایی ارسال میکنند اما پس از دریافت ایمیل کاربر به Mail Server او، عملیات انتقال این ایمیل از سرور مبدأ به سرور مقصد به کمک پروتکل SmtP انجام میپذیرد.

Email Delivery Models

هریک از پروتکل های Mail Access امکانات خاص به خود را دارند و در خصوص نحوه ی تحویل ایمیل به کاربر میتوانند در یک یا بیش از یکی از مدلهای Online,Offline,Disconnected کار کنند.

Online: در این حالت MTA به سرور وصل شده و بصورت آنلاین باقی میماند (چون Mapi) و میتواند ایمیل ها را مشاهده نمود و Folder و نیز Public Folder های درون Inbox خود را تغییر داده یا پاک نماید و یا اینکه به آنها بیفزاید (add کند). کاربران متعدد میتوانند بصورت همزمان به یک Public Folder وصل شوند. در این حالت همه ی ایمیلها بر روی سرور باقی میماند. تصویر زیر نیز این نکته را نمایش میدهد که کلاینت و سرور بصورت پیوسته و دو طرفه به هم متصل میباشند.



Offline: در این حالت کلاینت به حالت آنلاین درآمده و به سرور وصل میشود و ایمیلها را دانلود میکند و سپس به حالت Offline در می آید. همه ی ایمیل ها و Attachment ها بر روی ماشین کلاینت ذخیره میشوند و از روی سرور پاک میشوند. همه ی کارهای خواندن، پردازش کردن و یا پاک کردن ایمیلها در حالتی اتفاق می افتد که کلاینت Offline است. تصویر زیر نیز بیانگر این مشخصه است که ایمیل ها بصورت یک طرفه از سرور بر روی ماشین کلاینت منتقل میگردد.



Disconnected: این مدل به کاربر این امکان را میدهد که ایمیل ها را دانلود کند و آنها را در حالتی که **Disconnect** است ویرایش نموده و بعدا که مجدداً به سرور **Connect** شد، آنها را بر روی سرور **Restore** کند. مشابه مدل آنلاین، این مدل، سرور را محل رسمی و اصلی قرار گیری ایمیلها میدانند. این مدل از شناسه هایی یکتا (**Unique Identifiers**) استفاده میکند تا مطمئن شود که هنگام آپلود ایمیل ها (**Restore** کردن ایمیل ها روی سرور) پس از هربار تغییر (**Modification**) به درستی صورت گرفته است. همانطور که در تصویر زیر مشاهده میکنیم در این مدل ایمیل ها بین کلاینت و سرور بصورت دو طرفه تبادل میگردد.

(پروتکل IMAP میتواند در هر سه مدل Offline-Online-Disconnected کار کند.)

مقایسه مدلهای مختلف تحویل ایمیل (Email Delivery) را در جدول زیر مشاهده میکنیم :

Feature	Delivery Models		
	Offline	Online	Disc
Multiple client support	NO	YES	YES
Minimum server connection time	YES	NO	YES
Minimum server CPU and disk usage	YES	NO	NO
Minimum client disk usage	NO	YES	NO
Multiple remote mailboxes	NO	YES	YES
Mail processing when not online	YES	NO	YES

MIME(MultiPurpose Internet Mail Extensions)

در نسخه های ابتدایی از پروتکل SMTP ارسال Voice, Video, Image همراه با ایمیل چندان میسر نبود. علت آن طراحی خاص پروتکل SMTP بود که مبتنی بر کد Ascii بود و این چنین محتوایی با آن قابل Code کردن و ارسال نبود. همچنین SMTP از

کاراکترهای خاص چون علائم فنوتیک پشتیبانی نمیکرد؛ در نتیجه سازمان IETF بر آن شد که نسخه های جدیدی از SMTP را معرفی کند تا در ساختار جدید آن، بتوان محتوایی چون عکس و صوت و ... را ارسال کرد. در سالهای بعد که محتواهای فایلی جدیدی ابداع شدن و نیاز ارسال آنها با ایمیل احساس شد IETF بر آن شد که بجای ارایه ی نسخه های جدید از پروتکل SMTP راهکاری ارایه دهد که با کمک آن بتوان با توسعه آن، بدون دستکاری ساختار پروتکل SMTP، ارسال محتواهای جدید بر بستر آن را ممکن ساخت. نتیجه ی این پژوهش ها ابداع پروتکل MIME شد. در واقع MIME تنها تعدادی Header به ابتدای بسته ارسالی محتوی ایمیل اضافه میکند. (به دلیل بهره گیری سرویس ایمیل از پروتکل MIME، امروزه در بسیاری از متون علمی عبارت SMTP\MIME را همراه باهم مشاهده میکنیم). MIME یک پروتکل انتها باز (Open Ended) است و امکان توسعه و افزودن محتواهای جدید به آن جهت ارسال بر بستر SMTP میسر میباشد. این پروتکل برای Code کردن محتوای باینری فایلهایی چون Video, Image ... از الگوریتم Base64 بهره میبرد. از توضیحات این الگوریتم صرف نظر میکنیم اما در زیر نمونه ای از یک پیام که حاوی مقداری متن نوشتاری، یک عکس و یک فایل به فرمت PostScript است، برای درک بیشتر ساختار داده ای MIME آورده شده است. همچنین اصطلاح Multipart بودن پیام های MIME را میتوان در ساختار پیام مشاهده نمود:

```
MIME-Version: 1.0
Content-Type: multipart/mixed;
boundary="-----417CA6E2DE4ABCAFB5"
From: Alice Smith <Alice@cisco.com>
To: Bob@cs.Princeton.edu
Subject: promised material
Date: Mon, 07 Sep 1998 19:45:19 -0400

-----417CA6E2DE4ABCAFB5
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

Bob,

Here's the jpeg image and draft report I promised.

--Alice

-----417CA6E2DE4ABCAFB5
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
... unreadable encoding of a jpeg figure

-----417CA6E2DE4ABCAFB5
Content-Type: application/postscript; name="draft.ps"
Content-Transfer-Encoding: 7bit
... readable encoding of a PostScript document
```

(در آینده لازم است که بر روی پروتکل MAPI از مایکروسافت و نیز مدل Web Based Email تحقیقات بیشتری صورت پذیرد)

