کورش منشی
kourosh_e_kabir@yahoo.com
0912-2129404

شبکه‌های کامپیوتری ۲

* تفاوت سوئیچینگ مداری و سوئیچینگ بسته‌ای

• مداری: پهنای باند ثابت دارد، در نتیجه تأخیر ثابت خواهد داشت. نرخ خطا هم تقریباً ثابت است. در نتیجه این موارد کیفیت سرویس ثابت است که امتیاز عمده این روش محسوب می‌شود. در این روش ترتیب اطلاعات نیز حفظ می‌شود.
اشکال عمده این است که پهنای باند در اختیار دو طرف ارتباط است که اگر استفاده نشود، منابع هدر می‌رود. اشکال دیگر این است که وقتی مدار مورد نظر کامل نشود، ارتباط برقرار نخواهد شد.
در این نوع سوئیچینگ باید حتماً توسط گره‌های میانی، ارتباط مداری بین دو سر (host) برقرار شود.

• بسته‌ای: اینگونه است که برای ارسال اطلاعات، هر بسته را به همراه آدرس مقصد و احتمالاً مبدأ و موارد دیگر تحویل شبکه می‌دهیم و از آن زمان مسئولیت حمل بسته تا مقصد بر عهده شبکه است.
مثال عینی این نوع سوئیچینگ، شبکه پست است.
گره‌های میانی در لحظه‌ای که قصد فرستادن اطلاعات را دارند تصمیم می‌گیرند که از کدام خروجی بفرستند و این به معنای مسیریابی بهینه است.
دلیلی وجود ندارد که بسته‌ای که زودتر فرستاده شد، زودتر هم برسد.
مزیت عمده این نوع سوئیچینگ این است که از منابع شبکه بهتر استفاده می‌شود.
اشکال عمده این است که به دلیل تأخیرهای زیاد، کیفیت سرویس پایین می‌آید.

تأخیر:
- تأخیر پردازش برای مسیریابی بهتر ← تقریباً قابل پیش‌بینی
- تأخیر قرار گرفتن در صف تا به پردازش برسد ← تأخیر صف قابل پیش‌بینی نیست.

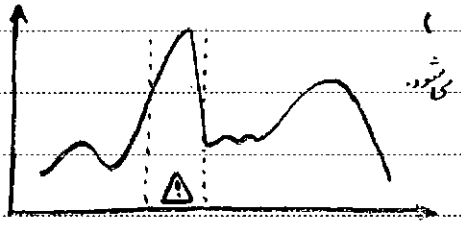
تأخیر ارسال، متفاوت خواهد بود و این مناسب نمی‌باشد (واریانس تأخیر ← jitter).
ترتیب بسته‌ها حفظ نمی‌شود ← بهترین راه رفع این اشکال، شماره گذاری است.
مزیت دوم این روش این است که زمان setup کردن مداری را ندارد.

Subject:
Year. Month. Date. ( )

* دلایل استفاده از سوئیچینگ بسته ای در شبکه های داده

- ماهیت Application های داده تولید می کنند، دارای نرخ بیت متغیر است
  در مخابرات معمولی صدا به صورت PCM تبدیل می شود، نرخ بیت 64 Kbps.

- شبکه های داده ماهیت انفجاری (Burst) دارند، یعنی در بعضی مواقع حجم استفاده داده زیاد می شود.



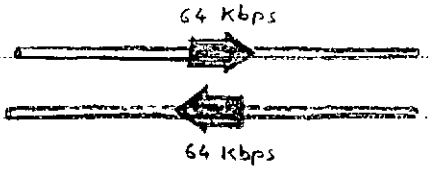
- بعضی از Application های داده، از پیام های کوتاه (short message) استفاده می کنند که بسیار حساس هستند (اکثراً کنترلی هستند) لذا چون تأخیر کلی سوئیچینگ بسته ای برای این بسته ها کمتر از مداری است، از سوئیچینگ بسته ای استفاده می شود.

- در حال استفاده از سوئیچینگ بسته ای، کیفیت سرویس از دست می رود، احتمال از دست رفتن پیام به دلیل ازدحام در بسته ای هم به دلیل ازدحام و هم به دلیل خطا، ولی در مداری بیشتر به دلیل خطاست.

- می خواهیم تلفنهای که استفاده کنیم QoS بالا برود.

- تلفنهای سوئیچینگ مداری در حال منسوخ شدن هستند و طراحی شبکه های کنونی بر مبنای سوئیچینگ بسته ای است.

- در سوئیچینگ مداری، برای ارتباط دو نفر مثلاً یکی حرف می زند دیگری گوش می دهد لذا 50% منابع را از دست می دهیم.



Subject: 2

Year. Month. Date. ( )

* برای تبدیل صدت چون صدا سیگنال پیوسته است می توان با در نظر گرفتن فاصله بین دو نمونه برداری حجم صدت را بسیار بسیار کم کرد.

64 Kbps ⟶ 8 Kbps

↳ 16 bit ↳ 3 bit

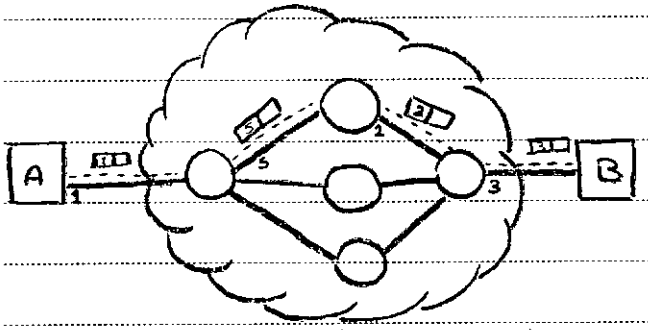
با این روش بجای ۸ بیت، ۱ بیت حذف می شود.

☑ ارتباط بین المللی

* Connection Oriented Packet Switching

در این روش گیرنده اطلاع دارد که اطلاعاتی برای او فرستاده می شود و در جایی که نخواهیم تصحیح و کنترل خطا داشته باشیم استفاده می شود، در واقع لیبل ندارد که مسیر انتقال ثابت باشد. در این روش پس از برقراری ارتباط، مسیر انتقال داده مشخص می شود.

ابتدا یک پیام setup فرستاده می شود، سپس مسیریابی شده، پس از دریافت ack مدار کامل ایجاد شده است. به طور کلی این ارتباط شامل ۳ فاز است:

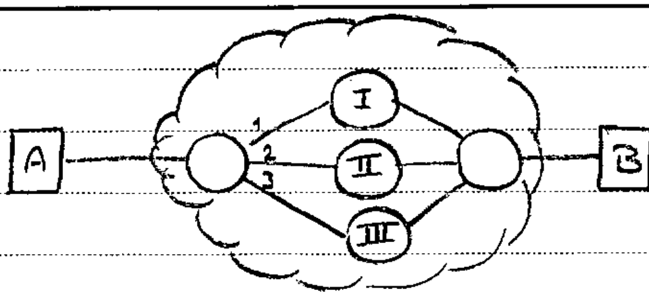


Incoming		Outgoing	
A	1	II	5

Setup Phase
Data Transfer
Release Setup

در این روش کل کانال در اختیار می باشد و منابع را اختصاص داده ایم. مزیت آن این است که مسیریابی را ساده کرده ایم. به این ترتیب که مداری که ایجاد شده است را شماره ای می گذاریم که به Virtual Circuit Identifier معروف است. از این به بعد به شماره مدار را می دهیم و نه آدرس را.

* Connection - Less Packet Switching



در این روش از جدول مسیریابی استفاده می‌شود. Host ها باید آدرس مقصد را به بسته ضمیمه کنند و هر Node وقتی بسته‌ای را می‌گیرد باید آن را با Routing Table مقایسه کند و بهترین مسیر را برای بسته پیدا کند. Next Hop معمولاً یا شماره Port متناظر است.

Destination	Next Hop
B	II (2)

*

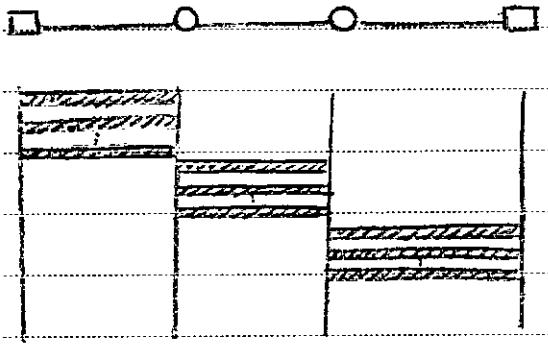
Packet Switching
- Connectionless (Datagram)
  - ☑ IP Networks
- Connection Oriented (Virtual Circuit)
  - ☑ ATM Networks
  - X.25
  - Frame Relay

* Message : اطلاعاتی که در لایه Application ردوبدل می‌شود که می‌تواند مثلاً به صورت فایل باشد.

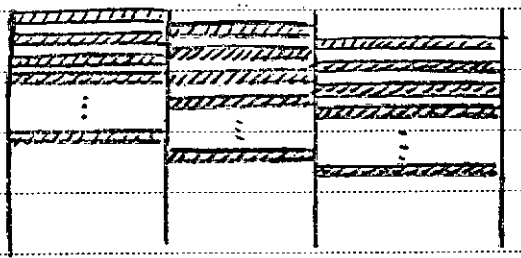
* Packet : واحد اطلاعاتی که در شبکه جابجا می‌شود. اندازه آن محدودیت دارد. یک Packet می‌تواند یک Message باشد، بخشی از آن.

* Frame : در لایه Data link طرح می‌شود. در واقع یک قالب است که داده‌ها داخل می‌روند. ابتدا و انتهای این قالب توسط Flag ها نمایش داده می‌شود.

* Message Switching : نحوه‌ی میانی بسته‌ها را در نظر نگرفته، به محض کامل شدن یک message، مسیریابی کرده، آنرا به گره بعدی می‌دهد. از این روش معمولاً استفاده نمی‌شود. در واقع اصلاً پیاده‌سازی نشده است.



* Packet Switching : در این روش به ازای هر بسته پردازش صورت گرفته، آنرا مسیریابی شده به گره بعدی فرستاده می‌شود.

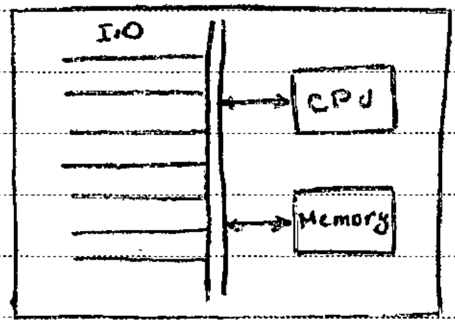


* Cut Through - Packet Switching : در این روش می‌دانیم که طول بسته‌ها چقدر است. لذا سرعت پردازش مشخص است. در نتیجه به محض رسیدن بسته کامل مسیریابی انجام می‌شود و در واقع صبر نمی‌کند که همه بسته برسد. این روش به شبکه‌ی Virtual Circuit گرایش بیشتری دارد.

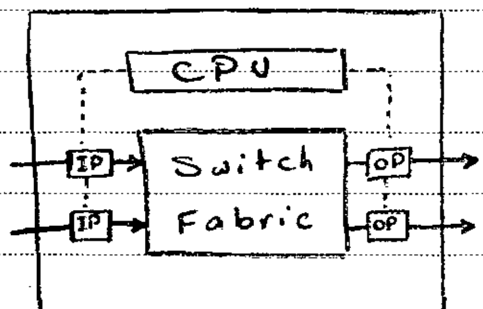
* به گره‌های میانی Switch و Router گفته می‌شود. هر سوئیچ یا روتر تعدادی پورت ورودی و خروجی دارد.

Subject:
Year. Month. Date. ( )

* برای پردازش بسته ۴ احتیاج به پردازنده، حافظه و I/O می‌باشد.



• ایراد معماری بالا، استفاده از Bus است. در این صورت اگر چند بسته با هم بیاید کارایی پایین می‌آید.



• در این معماری بهبود برای حذف حافظه هم دارد و عمل کنترل سوئیچ توسط CPU انجام می‌شود.

* بهترین مسیر برای ما به عنوان مصرف کننده، مسیری است که کیفیت سرویس بالا باشد. مثلاً برای صدا، تاخیر باید کم باشد. برای داده، مسیری که قابل اطمینان تر باشد. برای آگهی بازرگانی کم خرج ترین باشد.

از دید مدیر شبکه، بهترین مسیر مسیری برای است که بتواند بیشترین سرویس را بدهد. فرض کنیم یک بسته که باید تاخیر کمی داشته باشد را می‌خواهیم عبور دهیم. برای اینکار به نوعی یک مبنایی برای تاخیر عددی را نسبت می‌دهیم و با توجه به این امتیاز (cost) بهترین مسیر را انتخاب می‌کنیم.

* ویژگیهای الگوریتم مسیریابی خوب

- حجم پردازش کم شده، سریعتر جواب دهد.
- اگر گره‌ای کم یا زیاد شود باید خود را با تغییرات توپولوژی تطبیق دهد.
- از ازدحام در گره‌ها تا حد ممکن جلوگیری کند.
- دقیق باشد و بهینه جواب بدهد.
- سربار محاسباتی کمی داشته باشد، حافظه کمی بخواهد.

الگوریتم مسیریابی داخلی در حال اجرا است (در واقع هنگام رسیدن Packet) ، وضعیت شبکه را بررسی می‌کند و جدول مسیریابی‌ها را در صورت نیاز به روز می‌کند.

* الگوریتم مسیریابی

- Static : مدیر شبکه وضعیت را محاسبه کرده و آنرا در سیستم بار می‌کند. قابلیت عوض شدن خودکار را ندارد.
- Dynamic : خودشان را با تغییرات شبکه وفق می‌دهد.

Centralized : یک بخش مرکزی دارد که تغییرات را بررسی کرده به بخش‌های دیگر اطلاع می‌دهد.

Distributed : گره مرکزی وجود ندارد. اگر مثلاً لینک قطع شد گره‌های مجاور آن لینک باخبر شده به گره‌های همسایه خود اطلاع می‌دهند و همین طور تا آخر.

* در شبکه‌های داده سعی در استفاده از مدل distributed است. چون وابسته به گره مرکزی نخواهیم بود.

* چون بهر لینک cost اختصاص می‌دهیم، می‌توانیم با استفاده از الگوریتم‌های کوتاه‌ترین مسیر عمل مسیریابی را انجام دهیم. که به دو گروه تقسیم می‌شوند.

Distributed:
- Distance Vector : الگوریتمی می‌سنجد که هر گره تغییرات شبکه (قطع یک لینک) را در مجاور که جدول مسیریابی‌اش تغییر می‌کند بفرستد مجاور خودش، جدول مربوط به خود را می‌دهد.
- Link State : برای هر لینک cost در نظر می‌گیرند. اگر وضعیت یک لینک عوض شود، گره مجاور به اطرافیان خود اطلاع می‌دهد. هر گره توپولوژی کل شبکه را می‌داند. لذا می‌تواند با استفاده از یک الگوریتم مسیریابی را انجام دهد.

Subject:

Year. Month. Date. ( )

* الگوریتم Bellman - Ford

شامل ۲ مرحله است:

I) در این مرحله مقداردهی اولیه انجام می‌شود. به این صورت که هزینه گره‌ها به مقصد را «∞» فرض می‌کنیم، به جز گره مقصد که «صفر» قرار می‌دهیم.

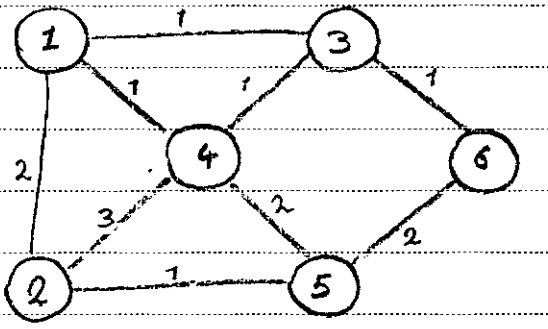
II) در این مرحله عمل به‌روزرسانی انجام می‌گیرد (updating)

$$D_i = \min_{(i \neq j)} \{ C_{ij} + D_j \} \quad \forall i \neq j$$

Repeat step until no more changes.

☑ در این مثال مسیرها متقارن هستند.

$$C = \begin{bmatrix} 0 & 2 & 1 & 1 & \infty & \infty \\ 2 & 0 & \infty & 3 & 1 & \infty \\ 1 & \infty & 0 & 1 & \infty & 1 \\ 1 & 3 & 1 & 0 & 2 & \infty \\ \infty & 1 & \infty & 2 & 0 & 2 \\ \infty & \infty & 1 & \infty & 2 & 0 \end{bmatrix}$$



Iteration	D_1	D_2	D_3	D_4	D_5
Initial	(-1,∞)	(-1,∞)	(-1,∞)	(-1,∞)	(-1,∞)
1	(-1,∞)	(-1,∞)	(6,1)	(3,2)	(6,2)
2	(3,2)	(5,3)	(6,1)	(3,2)	(6,2)
⋮					

جدول مسیریابی گره ⑥

$$D_1 = \min \{ C_{12} + D_2, C_{13} + D_3, C_{14} + D_4, C_{15} + D_5, C_{16} + D_6 \}$$

① ← ① هزینه + ② ← ⑥ هزینه

(Last Node, Distance)

* الگوریتم‌های Distance Vector و از جمله آن الگوریتم Bellman - Ford به علت اینکه تنها اطلاعات نود‌های مجاور خود نیاز است تا بتوانند distributed عمل کنند.

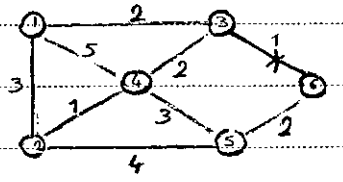
Step 1: Initialization: $D_{ij} = \infty \quad \forall i,j \quad i \neq j$

$D_{ii} = 0$

Step 2: Updating:

$$D_i = \min_j \{ C_{ij} + D_j \} \quad \forall j \neq i$$

* این الگوریتم در مقابل تغییر توپولوژی مقاوم است



d = 6

iteration	Node 1	Node 2	Node 3	Node 4	Node 5
Initial	(-1,∞)	(-1,∞)	(-1,∞)	(-1,∞)	(-1,∞)
1	(-1,∞)	(-1,∞)	(6,1)	(3,3)	(6,2)
2	(3,3)	(4,4)	(6,1)	(3,3)	(6,2)
3	(3,3)	(4,4)	(6,1)	(3,3)	(6,2)

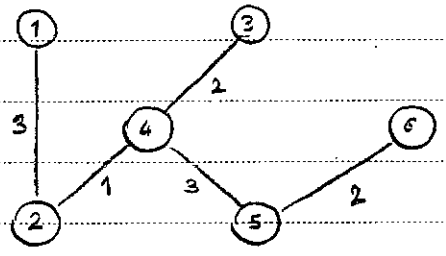
حال لینک 3 - 6 دچار خرابی می‌شود و گره‌های 3، 6 از آن طریق می‌فهمند تغییر پیدا کرده و این را به گره‌های مجاور اطلاع داده می‌شود.

$$D_{36} = \min_K \{ C_{3K} + D_{K6} \} \quad \forall K \neq 3$$

$$D_{36} = \min \{ C_{31}+D_{16}, C_{32}+D_{26}, C_{34}+D_{46}, C_{35}+D_{56}, C_{36}+D_{66} \}$$

$$D_{36} = \min \{ 5, \infty, 5, \infty, \infty \} = 5$$

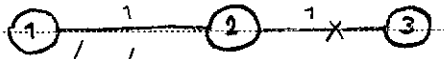
iteration	Node 1	Node 2	Node 3	Node 4	Node 5
1	(3,3)	(4,4)	(4,5)	(3,3)	(6,2)
2	(3,7)	(4,4)	(4,5)	(2,5)	(6,2)
3	(3,7)	(4,6)	(4,7)	(2,5)	(6,2)
4	(2,9)	(4,6)	(4,7)	(5,5)	(6,2)
5	(2,9)	(4,6)	(4,7)	(5,5)	(6,2)



Subject:

Year. Month. Date. ( )

* الگوریتم فوق دچار loop بی نهایت خواهد شد:



در این حالت در iteration های متوالی هزینه سلول زیاد خواهد شد، تنها هزینه یک مسیر از مجموع هزینه های شبکه (و یا حد بالای مشخص شده در شبکه) بالا رفت، آنها با بی نهایت جایگزین خواهیم کرد.

* یکی دیگر از الگوریتم های مسیریابی Dijkstra است که از الگوریتم های link state است الگوریتم bellman-ford در زمان تغییر محلی است و Packet ها را به مسیر نادرست منتقل کند. الگوریتم های link state وضعیت تمام link های شبکه را دارد و هر گره وضعیت تمام link های خود را دارد.

Dijkstra's Algorithm

1. Initialization

$N = \{S\}$

$D_j = C_{sj} \quad \forall j \neq s$

$D_s = 0$

2. Finding The Next Closest Node

Find node $i \notin N$ such that $D_i = \min D_j \; , j \notin N$

Add i to N.

If N contains all the nodes, stop!

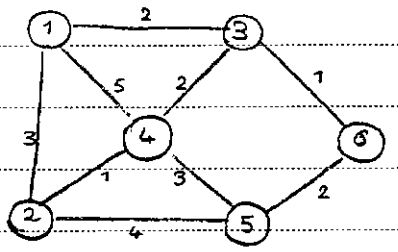
3. Updating Minimum Costs

For each node $j \notin N$, $D_j = \min \{D_j, D_i + C_{ij}\}$

Go to step 2

Subject: 6

Year. Month. Date. ( )



Iteration	N	D_2	D_3	D_4	D_5	D_6
Initial	{1}	3	2	5	∞	∞
1	{1,3}	3	2	4	∞	3
2	{1,2,3}	3	2	4	7	3
3	{1,2,3,6}	3	2	4	5	3
4	{1,2,3,4,6}	3	2	4	5	3
5	{1,2,3,4,5,6}	3	2	4	5	3

• در هر مرحله باید از طریق گره تازه اضافه شده update شدن را چک کرد.

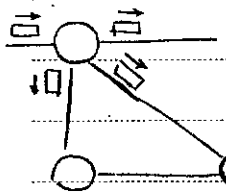
• این الگوریتم باید به صورت متمرکز اجرا شود.

• در شبکه‌ای با n نود، تعداد تکرار این الگوریتم «n-1» بار است.

* پرتغیر در توپولوژی، تنها یکبار الگوریتم Dijkstra را اجرا می‌کند و به جواب می‌رسد. پس بسیار سریع به تغییرات پاسخ می‌دهد که نسبت به Bellman-Ford مزیت است.

* از آنجایی که Dijkstra به صورت متمرکز هستند، هر نود باید دارای Link State Table کل شبکه باشد و از کل شبکه باید مطلع باشند. ولی هر نود نیاز به پردازش و حافظه خواهد داشت.

* الگوریتم دیگری برای مسیریابی وجود دارد به نام Flooding Routing Algorithm. هر نود بسته‌ای دریافت کند، روی تمام یال‌های خود بجز یالی که بسته را دریافت کرده می‌فرستد. بالاخره بسته از کوتاه‌ترین مسیر به مقصد می‌رسد. این الگوریتم نیازی به هیچ اطلاعی از شبکه ندارد.



و برای زمان راه‌اندازی شبکه مناسب است. هر بار این روش نیاز است بسته‌ها بی‌نهایت loop بیایند. برای حل این مسئله ۲ روش پیشنهاد شده است.

① Hop Count: در Header بسته عددی به عنوان حد بالای جلوتر در شبکه قرار می‌دهیم و به ازای عبور از هر نود یکی از آن کم می‌شود. بسته‌ای که Hop Count=0 دارای می‌شود و گره باید discard شوند.

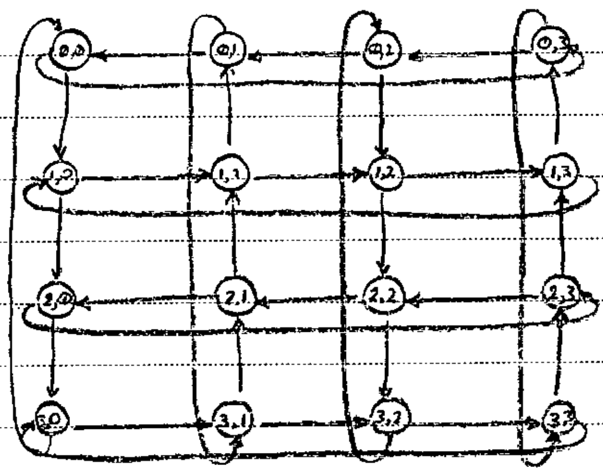
Subject:
Year. Month. Date. ( )

② هر نود جدید متوجه شود که بسته کجاها را از آن عبور کرده است. به این صورت که ID خود را در Header هر بسته
اضافه می‌کند و هنگامی که بسته‌ای دریافت کرد، اگر ID خود را در Header آن پیدا کرد، بسته
را discard می‌کند. در این روش Header بسته مدام بزرگ و بزرگتر می‌شود.

③ هر تولیدکننده بسته یک ID شانزده یا ۳۲ بیتی به بسته‌ی خود بدهد. هر نود ID و گره مبدا هر
بسته را گرفته و در جدولی در خود یادداشت می‌کند. اگر این زوج (ID و گره مبدا) در جدول
حاضر باشد، بسته discard خواهد شد. پس از مدتی ID ها تمام می‌شوند و دوباره از اول ID
به بسته‌ها اختصاص می‌یابد. این زمان مشخص است و پس از آن، آن ID از جدول نود پاک خواهد
شد.

PAPCO

* الگوریتم دیگری برای مسیریابی Deflection Routing است. معمولاً در شبکه‌ی منظم استفاده می‌شود و برای نودهای فاقد حافظه مناسب است. مسیرها در این شبکه‌ها از پیش تعیین شده‌اند.

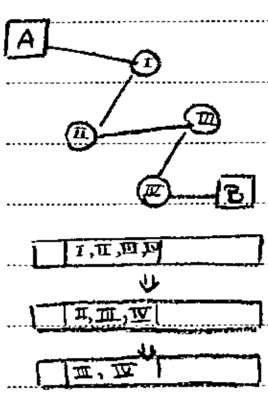


اگر یک link مشغول باشد (busy)، مسیر از همان جا تغییر می‌کند و نود مقصد می‌رسد. نودها برای فرستادن پاکتها پردازشی انجام نمی‌دهند. هر نود ۲ پورت ورودی و ۲ پورت خروجی دارد و نیازی به بافر ندارد. مثلاً در شبکه‌های نوری که بافرینگ پیچیده است کاربرد پیدا خواهد کرد. مثال دیگر multi-processorها هستند.

ترافیک روی 00 → 01 : 03 → 33 → 23 → 22 → 21 → 11 ✓

03 → 02 → 01 →(busy) 00 → 10 → 11 ✗

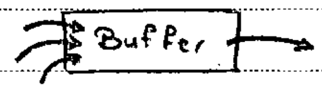
* الگوریتم دیگر Source Routing هستند. در این الگوریتم نودهای مبدأ بسته اطلاعات کل شبکه را دارند و خود مسیریابی کرده و در Header بسته آدرس تمام نودهای میانی نوشته می‌شود. وقتی بسته به هر نود می‌رسد، آن نود خود را از Header حذف می‌کند. در این روش نودهای میانی نیازی به Routing Table ندارند. در صورت خرابی در شبکه و یا تغییر ترافیک شبکه، بسته‌ها آگاه نیستند. این از معایب این روش است که به تغییرات لحظه‌ای پاسخ نمی‌دهد. حسن دیگر این روش حفظ ترتیب بسته‌های یک جریان اطلاعاتی به دلیل یکسان بودن مسیر است. از نمونه‌های این روش در اینترنت در IPv4 و IPv6 است که نودهای میانی نیز قابلیت Source Rou. را دارا است.



Subject:
Year. Month. Date. ( )

* Traffic Management یکی از ابزاری است که QoS را بالا می‌برد. این روش تلاش می‌کند
بدون پایین آوردن ترافیک و در نتیجه کارآیی سیستم، و با تحمل load بالا، QoS را
حفظ کند. تکنیک‌های این Traffic Mangement چند تا اند:

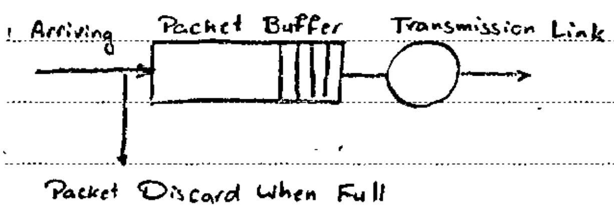
- Routing
- سرویس‌دهی متناسب با نوع سرویس ورودی:



در هر نود برای خروج چند بسته از یک پورت
صفی ایجاد می‌شود که به آنها سرویس داده می‌شود
نحوه قرار دادن بسته‌ها در صف به چند روش انجام می‌گیرد که به این نحوه‌ی قرارگیری
Packet scheduling گویند. به نحوه سرویس‌دهی به صف Queue
Management گوییم که وظیفه آن حفظ فضای buffer و مدیریت است

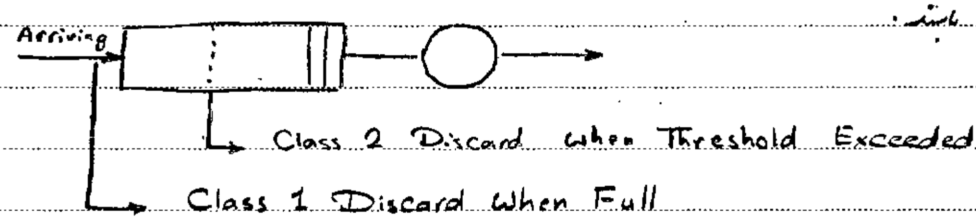
* روشهای Packet Scheduling:

I. First Come First Served

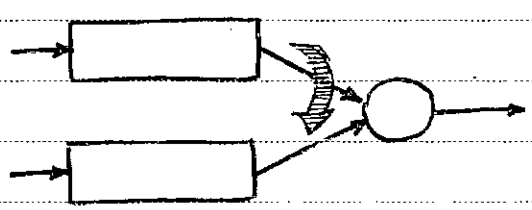


II. Periory Queuing

برای اولویت‌بندی، اولویت‌های پایین‌تر را
به سرویس می‌دهیم که Packet Loss کمتری
داشته باشد.



III. Head Offling (HoL) Periority Queuing

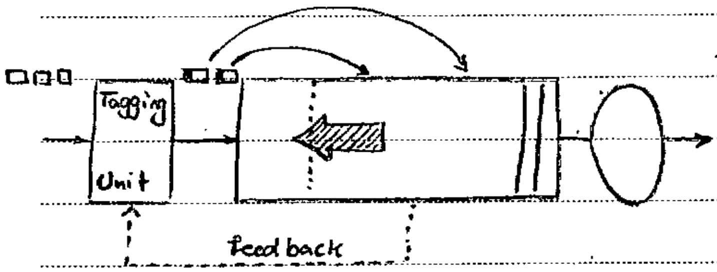


موقعی که صف با اولویت بالا خالی
باشد سراغ صف پایین‌تر می‌رود. بعد
از پایین آمدن زمان سرویس به
اولویت‌های بالاتر است.
ایرادهای این عبارتند از:

① starvation ② پایین آمدن بهره‌وری از فضای buffer

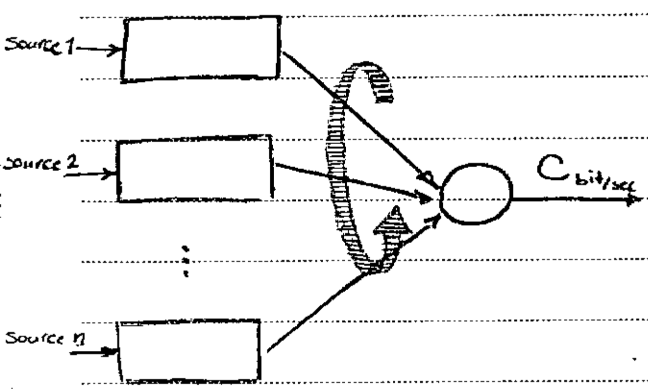
③ بالا رفتن متوسط زمان delay ④

IV) Sorting Packet Buffer



در این روش یک واحد Tagging اولویت بسته‌ها را مشخص می‌کند و آنها در جای خود در buffer قرار می‌دهد. در هنگام پر شدن buffer، با شیفت دادن بسته‌ها، اولویت پایین‌تر را از بین می‌برد. روش دیگر دادن طول عمر به بسته‌های سرویس‌های مختلف (مثلاً طول عمر کوتاه‌تر برای سرویس صوتی) است.

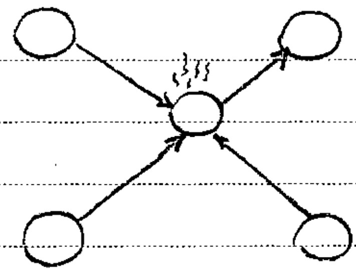
V) Fair Queuing



به ترتیب به صف‌ها نگاه می‌کنیم و به روش Round Robbin به آنها سرویس می‌دهیم. طوری این کار را انجام می‌دهیم که با توجه به طول بسته‌ها متوسط زمان سرویس به همه صف‌ها برابر باشد. این روش تضمین می‌کند که به هر صف باند سرویس به اندازه بزرگتر یا مساوی «$\frac{C}{n}$» است.

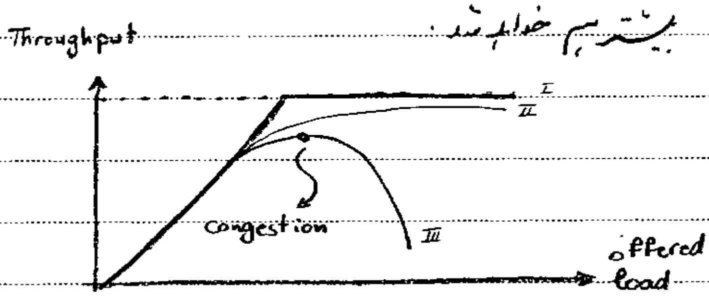
VI) Weighted Fair Queuing

در صورت اولویت داشتن صف‌ها، به هر کدام وزنی اختصاص می‌دهیم (برعکس وزن‌های یکسان در روش V) و چرخش زمانی روی آنها به همان نسبت از آن قرار می‌دهیم.



* از مسائل مهم دیگر، Congestion Control است.

این مسئله زمانی است که حجم ورود اطلاعات به یک نود از پهنای باند خروج آن بیشتر می‌شود. در صورت داشتن یک الگوریتم تشخیص خطای end-to-end این ازدحام بیشتر هم خواهد شد.



این وضعیت congestion می‌تواند منجر به deadlock شود و Throughput = 0 خواهد شد.

روش‌های کنترل ازدحام هدف مواجه شدن نود با ازدحام و در نتیجه پایین آوردن packet loss و بالا بردن QoS است.

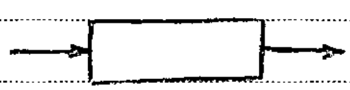
I > Perfect

II > Controlled

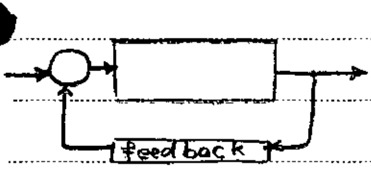
III > Uncontrolled

روش‌های کنترل ازدحام بر دو دسته‌اند:

- Open Loop Congestion Control (preventive)
- Closed Loop Congestion Control (reactive)



روش‌های Open Loop به این صورت است که ورودی را کنترل می‌کنیم و آن را در حالت پایدار نگه می‌داریم.

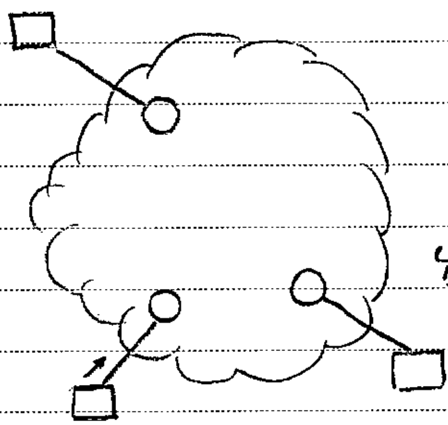


روش‌های Closed Loop به این صورت است که ورودی آزاد است و در صورت بروز ازدحام feedback می‌گیریم و از میزان ورودی می‌کاهیم.

Subject: 9
Year. 84 Month. 7 Date. 19 (5)

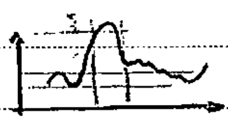
* نتایج congestion
- انباشتگی بافرها ← packet loss
- انباشتگی بافرها ← بالا رفتن delay
- پایین آمدن troughput
- پایین آمدن QoS

* در روش open loop تضمین می‌کنیم که QoS بالا باشد. در روش closed loop با توجه به feed back میزان ورودی کنترل می‌شود ولی packet loss وجود خواهد داشت و QoS را تضمین نمی‌کند. در شبکه‌های ATM از open و در اینترنت از closed استفاده می‌کند.



* در روش open loop سوئیچهای شبکه باید ترافیک شبکه را کنترل کنند تا (ازدحام) به وجود نیاید. این سوئیچها دارای واحدی به نام Call Admission Control (CAC) هستند که اجازه عبور را به بسته‌ها می‌دهد. این واحد برای میزان عبور دارای دو معیار متناقض هستند:

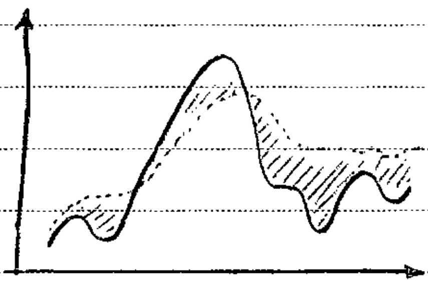
۱- بالا بردن Quality Of Service (از دید user)
۲- حداکثر بردن Troughput (از دید provider)



الگوریتم خاصی برای بهترین حالت وجود ندارد، چون:

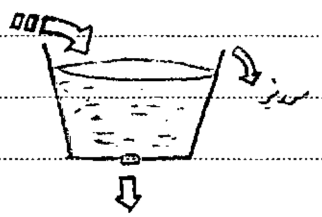
۱- نرخ تولید بسته‌ها متغیر است و به application بستگی دارد. ☑ ارسال مقادیر (نرخ بیت متغیر)
۲- CAC با هر application قراردادی دارد که از سابقه تسلسل آن نتیجه می‌شود. اگر هر واحد از قرارداد خود تخلف کند، براساس policy خود با آن برخورد می‌کند. واحدی به نام Usage Parameter Control (UPC) وظیفه نظارت و سازماندهی را بر عهده دارد که اولاً باید تخلف را تشخیص می‌دهد ثانیاً با تخلف چه برخوردی بیاورد.

• هر بسته‌ی اضافه را حذف می‌کند ← ↓QoS
• بسته‌ای تخلف را Tag می‌زند و اجازه عبور می‌دهد، و در هر سوئیچی که دچار ازدحام شود این بسته‌ها اول حذف می‌شود ← ↑Throughput



• روش سوم که به Traffic Shaping موسوم است با تغییر وضعیت اطلاعات نسبت Peak به بد کردن بسته‌ها اقدام می‌کند. از آنجایی که همه اطلاعات قابل تغییر شکل نیستند ممکن است باعث ایجاد تأخیر و اختلال در داده‌ها و پایین آمدن QoS شود.

از مکانیسم‌های تشخیص تخلف می‌توان به موارد زیر اشاره کرد:



• Leaky Bucket

ورودی متغیر

خروجی ثابت

ورودی متخلف از شکل سرریز می‌شود که می‌توان با آن به روشهای قبلی برخورد کرد.

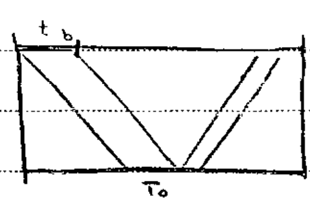
* الگوریتم‌های Closed Loop هیچ کنترلی بر ورودی ندارند پس ازدحام می‌تواند رخ دهد:

• در شبکه‌های TCP/IP مکانیسمی وجود دارد که در زمان Congestion ورودی را کاهش می‌دهد که TCP Congestion Control نام دارد.

APP.	
TCP	UDP
IP	
N.I	

این شبکه‌ها معمولاً برای ارسال Data طراحی شده‌اند که TCP این کار را انجام می‌دهد. UDP حجم کمتری نسبت به TCP دارد و زیاد تولید ازدحام نمی‌کند. (البته امروزه UDP نیز 20% ازدحام را به خاطر سرویس‌های multi media تولید می‌کند و به طور کلی Real Time Protocol‌ها.)

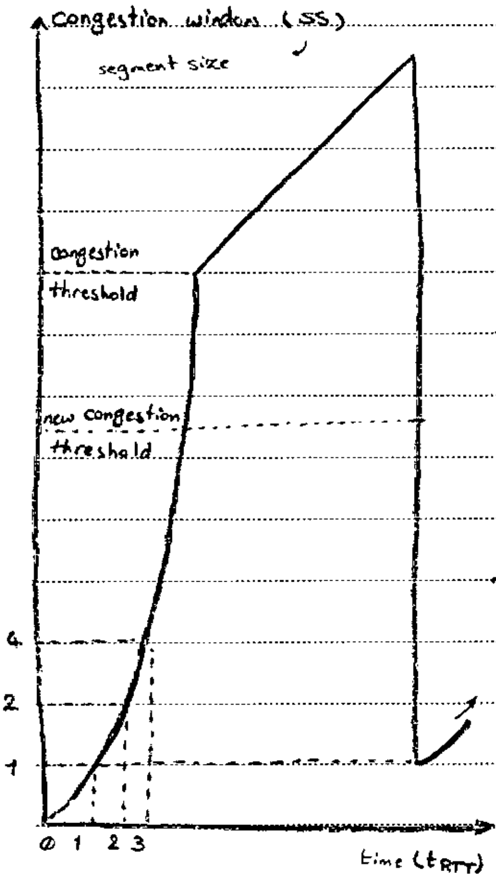
Internet Engineering Task Force (IETF) امروزه برای UDP هم مکانیسم‌هایی پیشنهاد کرده است که به آنها TCP Friendly می‌گویند.) همچنین TCP عمل کنترل خطای End-To-End دارد و از روش ARQ :: Selective Repeat استفاده می‌کند. این روش که به sliding window نیز معروف است دارای بافرهایی است که داده‌های تأیید نشده را نگه می‌دارد. اگر پنجره را کوچک بگیریم حجم اطلاعاتی که وارد شبکه می‌شود کمتر خواهد بود.



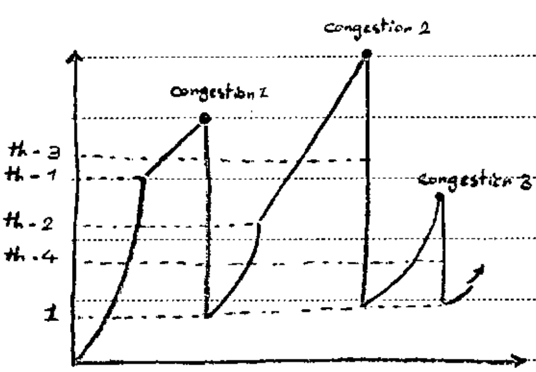
$R_{eff} = \frac{w \cdot n_b}{T_o}$ $\quad w < \frac{t_c}{t_b}$

این مکانیسم با تغییر اندازه پنجره (w) میزان ورودی را تنظیم می‌کند.

حجم اطلاعاتی که در TCP فرستاده می‌شود segment است. TCP سرویس byte stream است. در TCP اندازهٔ segment قابل تعریف است که بیشینه آن 64KB است. وقتی حجم اطلاعات به اندازهٔ یک segment size می‌رسد اطلاعات فرستاده می‌شود. RTT یا



Round Trip Time زمانی است که یک بسته به طرف مقصد رسیده و ACK بگیرد. اندازه پنجره به صورت نمایی افزایش می‌یابد که حداکثر آن عددی به نام congestion-threshold است تا throughput را بالا ببرد.
این فاز به slow start معروف است.
در این فاز مکانیسم کنترلی خاصی را نیز اعمال نمی‌کنیم.
فاز بعدی به congestion avoidance نام دارد که در آن اندازه پنجره یکی یکی افزوده می‌شود تا شبکه با ازدحام مواجه شود. فاز بعدی به congestion occur نام دارد. در این فاز TCP اندازهٔ پنجره را بلافاصله به یک تبدیل می‌دهد و میزان Treshold را به نصف اندازه‌ای که ازدحام رخ داد تغییر می‌دهد و به فاز slow start بر می‌گردد. با این روش خیلی سریع از حالت ازدحام خارج می‌شویم.



TCP از packet lost می‌فهمد که congestion رخ داده است. در نتیجه packet loss یعنی time out را به عنوان ازدحام در نظر می‌گیرد در حالی که تنها عامل packet loss ازدحام نیست.
packet loss → clock expires → time out
این مکانیسم در شبکه با نرخ خطای پایین مناسب است ولی در نرخ خطای بالا (مثلاً wireless) مناسب نیست.

✎ تمرینهای فصل هفتم: گروه دوم (۲۸-۲۹-۳۰-۳۵-۳۹-۵۴-۵۵) + سری اول ← (۸-۹-۱۳-۱۴-۱۵-۱۷-۱۸-۲۲)
1384, 8, 4 — 1384, 8, 10

Subject:
Year. 84 Month. 7 Date. 24 (6)

TCP/IP Networks *

یکی از شبکه‌های packet switching مبتنی بر تکنیک Datagram استفاده می‌کند.
نخستین بار در قالب شبکه Arpanet برای وزارت دفاع آمریکا تشکیل یافت.
مدل لایه‌ای آن حاوی ۴ لایه است:

Application
Transport
Internet
Network Interface

بخش لایه اینترنت که از Inter Networking گرفته شده، اتصال شبکه‌های مختلف مهم است
در حدود سال 81 میلادی انجمنی به نام IAB (Internet Ac) تشکیل شده است که
خود شامل دو دسته است:

IETF : حل و فصل مسائل مهندسی
IRTF : تحقیقات بلند مدت

IETF به سؤالاتی که در زمینه‌های مختلف مطرح می‌شود (Request For Comment) پاسخ می‌دهد
که به آنها شماره‌ای می‌دهد. به این RFC ها در حکم استاندارد و قرارداد نگاه می‌شود و عرفی
هستند چون IETF نهاد رسمی نیست.
لیست RFC ها در سایت www.IETF.org هستند.

RFC → STD : پروتکل استاندارد
RFC → FYI : برای اطلاع همگانی

در استاندارد اینترنت تنها دو لایه Transport و Internet استاندارد شده‌اند و دو لایه
دیگر به دلخواه هستند و می‌توانند از هر پروتکلی تبعیت کنند. این پروتکل‌ها می‌توانند به طور مربوط پیشنهاد
شوند و اگر قبول شدند دیگران نیز از جزئیات آنها اطلاع می‌یابند و RFC تبدیل می‌شوند مانند
سرویس World Wide Web که در سال 90 استاندارد شد (HTTP).

Subject: 11
Year. Month. Date. ( )

* سرویس لایه Internet به لایه بالای خودش شامل ... و اینها است:

• connection - less ← سرویس قابل اطمینانی نیست.

• best - effort ← بیشترین تلاش به معنای عدم اطمینان است.

• packet transfer data ← واحد داده packet است

• کنترل خطا ندارد.

• نام آن Internet Protocol یا IP است

* لایه Transport دو سرویس به لایه بالاتر ارائه می‌دهد که لایه application خود یکی از این دو سرویس را انتخاب می‌کند.

I. • reliable

• connection - oriented

• byte stream transfer data ← سترزم کنترل خطاست.

II. • best effort

• connection - less

• packet (datagram) transfer data

پروتکل I > Transmission Control Protocol ≡ TCP

پروتکل II > User Datagram Protocol ≡ UDP

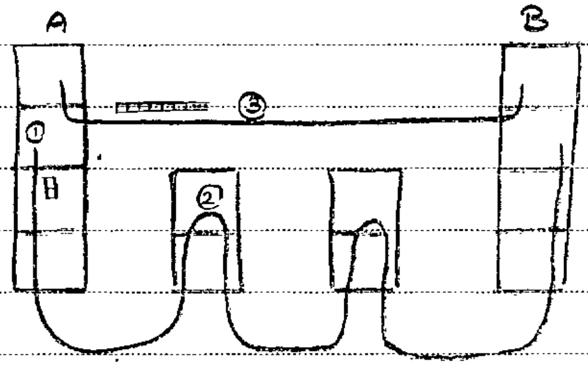
* چون بیشترین استفاده از پروتکل TCP است این مدل TCP/IP خوانده می‌شود.

* مقایسه TCP/IP و OSI

TCP/IP	OSI
Application	Application
	Presentation
	Session
Transport	Transport
Internet	Network
Network Interface	Data Link
	Physical

PAPCO

* Byte Stream :



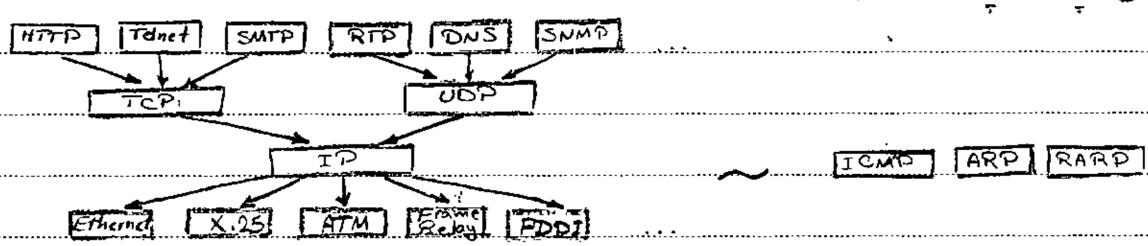
در شبکه‌های با تعداد بند انتهایی (host) و نود میانی (router) احتیاجی به پیاده سازی لایه‌های Transport و Application برای انتقال اطلاعات user نداریم.

① اطلاعات کنترلی برای برقراری ارتباط connection oriented

② مراحل مسیریابی بسته

③ پس از برقراری ارتباط، A و B طوری‌اند که مستقیماً به هم وصل هستند و اطلاعات را مستقیم به هم می‌فرستند. درحالی که سازوکار دیگری برای فرستادن اطلاعات مورد نیاز است چون اتصال TCP خیلی شبیه است به آن TCP Socket نیز می‌گویند، البته این دید مربوط به لایه application است.

* برخی از پروتکل‌ها و پروتکل‌های مرتبط



* TCP چون پروتکل مطمئن است دارای تأخیر است (چون کنترل می‌کند) وقتی زمان مهم است از UDP استفاده می‌شود.

* ICMP : اشکالات در لایه IP را گزارش می‌دهد.

ARP : آدرس IP را به آدرس فیزیکی نگاشت می‌کند.

RARP : ورودی‌ای Disk Less است، آدرس فیزیکی را به آدرس IP نگاشت می‌کند.

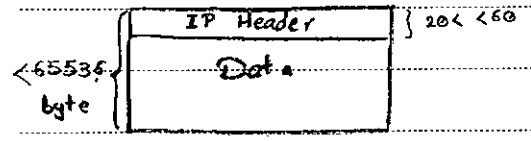
RIP و OSPF و BGP : پروتکل‌های مسیریابی

DHCP : در شبکه host تنظیمات اولیه و اتوماتیک IP در شبکه را بر عهده دارد.

Mobile IP : نگه داشتن IP برای نودهای متحرک را بر عهده دارد.

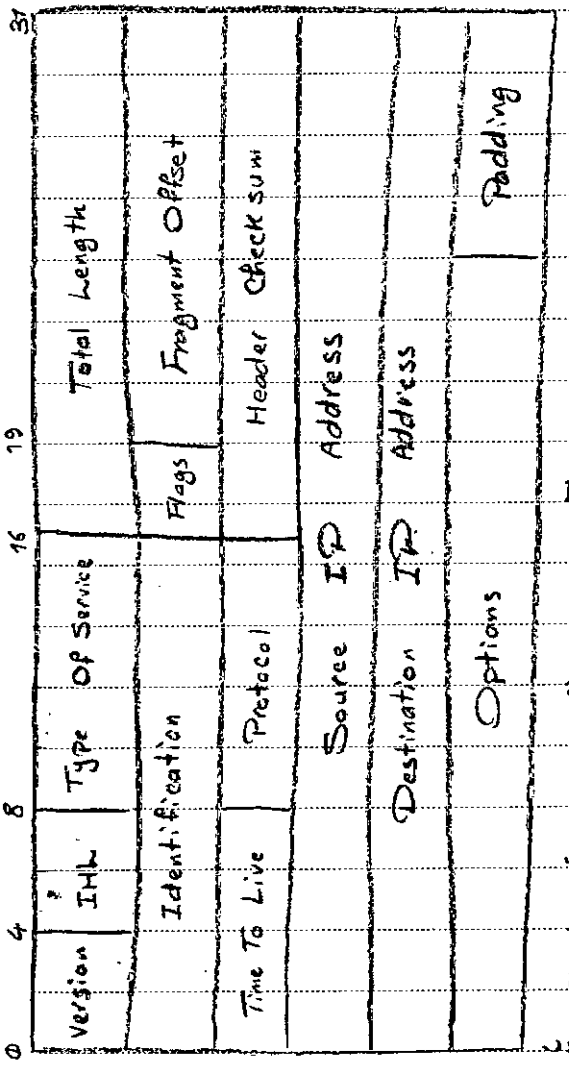
* پروتکل IP

وظیفه این پروتکل جابجایی بسته‌ها به صورت مستقل است. لایه سوم محسوب می‌شود (چون مرجع شماره‌گذاری مدل OSI است)



بسته‌ی IP طول متغیری دارد که حداکثر آن 64 kbytes است. هر بسته شامل دو قسمت header و data است. header دارای طول متغیر است که حداقل 20 byte و حداکثر 60 byte است.

فیلدهای داخل header عبارتند از:



- Version
عدد قرار گرفته در این فیلد نشان‌دهنده فرمت بعدی فیلدهای موجود در آن است. مثلاً روی IPv4 است.

- Internet Header Length
طول هدر را مشخص می‌کند. این فیلد 4 بیت است که 15 عدد را شامل می‌شود. پس واحد آن 32 بیت قرار می‌دهند. حداکثر آن عدد 5 و حداکثر آن عدد 15 خواهد بود. این محدودیت در IPv6 برداشته شده است.

- Total Length
عددی 16 بیتی است و طول کل بسته را نشان می‌دهد که شامل header و data است.

- (TTL) Time To Live
مشخص می‌کند که یک بسته تا چه زمانی در شبکه زنده بماند. عدد آن براساس Hub Count است و هر بار که در شبکه به پیش می‌رود یکی از آن کم می‌شود. این عدد را تولیدکننده بسته در نظر می‌گیرد. وقتی 0 شود بسته نابود می‌شود.

Subject:

Year. Month. Date. ( )

- Protocol

اطلاعاتی که در قسمت data قرار دارند متعلق به پروتکلی هستند که برنامه دارای یک لایه مشترک هستند که این عدد ۸ بیتی در فیلد Protocol ذخیره شده‌اند.

TCP = 6 , UDP = 17 , ICMP = 1 , ...

در IPv4 امکان ذکر لایه IP تا 256 پروتکل در نظر گرفته شده است.

- Source IP Address

هر نود در شبکه یک آدرس IP دارد که با آن معرفی شود. این فیلد حاوی آدرس مبدأ است که عددی 32 بیتی است که برای سادگی نمایش با چهار عدد ۸ بیتی که با نقطه جدا شده‌اند مشخص می‌شوند. ☑ 192.168.31.2

- Destination IP Address

آدرس مقصد که داده حرکت دارد.

- Options

فیلد اختیاری است. چند کاربرد آن در Source Routing و Time Stamp و ... است. چون طول آن متغیر است ممکن است مضرب 32 بیت نباشد.

- Padding

وظیفه آن تکمیل فضای خالی فیلد options برای رساندن طول آن به مضرب 32 است.

- Header Checksum

روی کل header یک فیلد Error Detection قرار می‌دهد.

کل طول هدر را به اعداد 16 بیتی تبدیل می‌کند و عدد X را بدست می‌آورد. منفی عدد X در این فیلد وجود دارد.

$$X = P_1 + P_2 + \dots + P_n \bmod 2^{16-1}$$

وقتی خطا در header تشخیص داده شد بسته از بین می‌رود چون IP یک پروتکل best effort است (خطا: در مقصد نیز X را بدست می‌آورند و header checksum را جمع می‌کنند، نتیجه غیر صفر خطاست)

- Type Of Service

مشخص می‌کند که data از چه نوع سرویس است که نیازمندی سرویس برای حفظ QoS از آن مشخص می‌شود (تأخیر، سیستم صوتی، اعتماد در سرویس داده و ...) این فیلد ۸ بیت است که به صورت

زیر می باشند:

delay	throughput	reliability	cost	periority level	X

- delay ← حساس به تأخیر
- throughput ← بیشترین گذردهی
- reliability ← مطمئن‌ترین مسیر
- cost ← کمترین هزینه

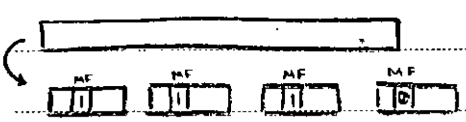
• Flags

لایه Network Interface محدودیتی روی طول بسته اعمال می‌کند که معمولاً کمتر از 64 Kbyte است. پس بسته باید fragment شود و بسته به بسته‌های مستقل تبدیل می‌شود. جمع آوری این تکه‌ها در مقصد صورت می‌گیرد (Re-Assembly). در IPv4 محدودیت اعمال شده توسط لایه N.I. برای بسته‌ها MTU (= Maximum Transfer Unit) است. عمل fragmention می‌تواند در هر جای شبکه اعمال شود.

در فیلد Flags سه بیت حاضر است که عبارتند از:

- DF Flag : Router های میانی حق fragment بسته را ندارند. (Don't Fragment)
- MF flag : نشان دهنده این است که آیا بسته بعدی وجود دارد؟ (More Fragment)
- Reserved : بی استفاده

اگر در مسیر طول بسته بزرگتر از MTU یک router باشد و DF Flag = 1 باشد بسته discard می‌شود و یک خطا خارج می‌دهد.



اگر DF Flag = 0 باشد بسته fragment می‌شود.
MF Flag نشان می‌دهد که تکه بعدی ای نیز وجود دارد.

• Identification

بسته‌ای که fragment شده است دارای شماره خاصی است که در مبدأ به آن تخصیص می‌یابد و تمام قطعه‌های آن، آنها را در خود دارند و ID آنها یکسان است.

• Fragment Offset

نشان دهنده جایی که در یک بسته هنگام بازسازی است. این فیلد 13 بیت دارد ولی آدرس 16 بیت

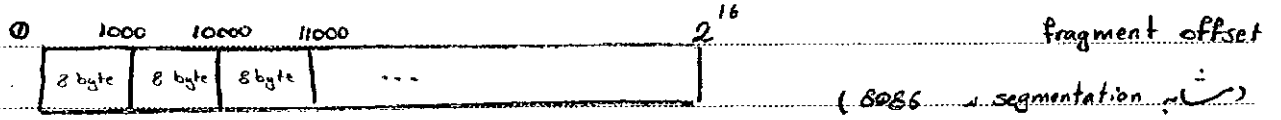
Subject:

Year. Month. Date. ( )

داده برای همین بسته تکه‌ای fragment می‌شود به سمت آخر آدرس آن صفر شده و بقیان بیشتر بالای آدرس آنرا فرستاده.

این قطعه آخر به مقصد رسید (MF = 0) با توجه به Fragment Offset آن می‌فهمیم که چند تکه بوده است، و این زمان مقصد یک زمان سنج راه می‌اندازد، time out شدن آن می‌فهمیم که تکه‌ای در مسیر از بین رفته است و مقصد می‌تواند Reassembly کند. پس بقیه تکه‌های بسته را از بین می‌برد.

DF	MF	Fragment Offset	Meaning
1	x	x	بسته تکه تکه شده است.
0	0	0	بسته تکه نشده است.
0	1	0	تکه اول بسته است.
0	0	غیر 0	تکه آخر بسته است.
0	1	غیر 0	بسته تکه شده است.



fragment offset

(شبیه segmentation در OSG)

F.O ⇒ @1 @10 @11 ...

block size = L + user information in cells P + cell header = H *

$$\text{efficiency} = \frac{L}{L + \left[\frac{L}{P}\right] \times H}$$

Subject: 14

Year. 84 Month. 8 Date. 1 (8)

* در Router ها برای یک process نقل دارد که تغییرات ایجاد شده در شبکه را می‌گیرد و جدول مسیریابی را update می‌کند. وجود میلیون‌ها نود در اینترنت باعث افزایش تعداد سطرهای این جدول می‌شود که دو مشکل در پی دارد:
  - حجم جدول زیاد می‌شود.
  - سرچ در جدول زمان زیادی را صرف می‌کند.

برای حل این مسائل راه‌کارهای مختلفی ارائه شده است.

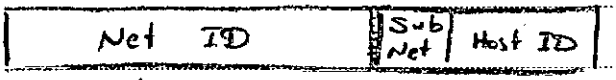
* آدرس‌دهی سلسله مراتبی:

بخشی از آدرس شبکه را مشخص می‌کند و بخش دیگر host درون آن شبکه را مشخص می‌کند. Router‌ها فقط باید آدرس شبکه را داشته باشند و نیازی به ذخیره آدرس همه host‌ها ندارند.

200.2.100.X

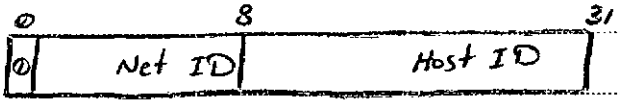
Net ID ↳ host تا 255

این سلسله مراتب می‌تواند سطوح بیشتری داشته باشد. به عنوان مثال ۳ سطح

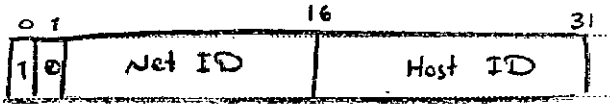


* Net ID، Host ID طول ثابتی دارد. بر این اساس شبکه‌ها را به سه دسته تقسیم می‌کند که به آن‌ها Classful IP Address می‌گویند:

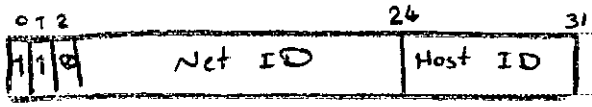
- Class A
  شبکه‌های بزرگ



- Class B
  شبکه‌های متوسط



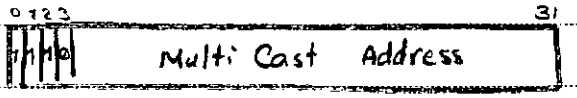
- Class C
  شبکه‌های کوچک



Subject:

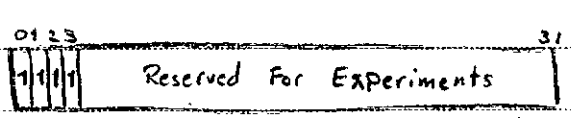
Year. Month. Date. ( )

Class D



برای Multi Cast Address

Class E



برای کارهای آزمایشی

* برخی از سرویس‌ها در اینترنت گروهی است. مانند تلویزیون، پخش رادیویی و ...
تمام آی‌پی‌های استفاده از سرویس با یک آدرس گروهی (Multi Cast) به آن assign شوند
و هنگامی که کارت‌ها ثبت شد این آدرس از آن‌ها گرفته شود.

* به مرور زمان تعداد شبکه‌های کلاس B و C زیاد شدند (مثلاً ۱۰۰۰ تا شدند) برای سرویس به آن:
- کلاس B آدرس‌های زیادی را هدر می‌داد.
- چند کلاس C حجم جدول‌های آدرس را زیاد می‌کند.

برای همین در سال 93 شرکت IETF روش جدیدی بنام Classless Inter Domain Routing
(CIDR) طراحی کرد که اندازه Net ID و Host ID در آن بر حسب شبکه تغییر
می‌کند. در این روش در هر سطح Routing Table احتیاج داریم تا ببینیم در آن چند بیت
Net ID است و چند بیت Host ID، سپس عدد Net Mask می‌زنیم. در آن به ازای
بیت‌های Net ID، "1" می‌گذاریم و بقیه صفر. این عدد را با آدرس گره‌ی ورودی And
می‌کنیم تا ببینیم آیا متعلق به شبکه هست یا خیر.

مثلاً اگر ۲۲ بیت Net ID داریم:

22 10
1 1 1 ... 1 0 0 0 ... 0

معمولاً IP, Mask را اینگونه نمایش می‌دهند: 200.198.3.1 / 22 ← تعداد بیت‌های Net ID

گاهی نیز Mask را کامل می‌نویسند: 255.255.252.0

* هر Network Interface یک آدرس IP دارد و یک host می‌تواند چند N.I داشته باشد.

Subject: 15

Year. Month. Date. ( )

* در Router و Gateway چند N.I دارد پس آدرس Next Hub در پرسته نشان می دهد که بسته باروی کدام N.I حرکت کند. در Routing Table یک entry به نام default وجود دارد که اگر آدرس در جدول وجود نداشته باشد بسته باری قرار می دهد. اگر این entry وجود نداشته باشد بسته در صورت پیدا نکردن آدرس خود در جدول حذف می شود و یک پیغام خطا (Host Unreachable) ایجاد می کند.

* یکی از آدرس های لوپ 127.x.x.x است که Loop Back نام دارد که آدرس میان host است. بسته به میان آدرس برمی گردد. معمولاً آن به صورت 127.0.0.1 استفاده می کنند.

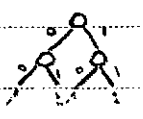
* آیدی CIDR

☺ از محدوده IP Address بخوبی استفاده می کنیم.

☺ محدودیت روش Classful را حل می کند و اگر دو شبکه ای دو آدرس کلاس C متوالی داشتیم، با ادغام آن ها می توانیم یک آدرس Classless بدست آوریم. به این عمل Supernetting گویند.

200.100.0.X
200.100.1.X
} 200.100.0.0 / 23 ☑

☹ در Net ID، Prefix نیز می بینید و حل پذیرش بسته Prefix Matching نام دارد. در CIDR ممکن است آدرس های IP موجود باشند که زیر مجموعه هم باشند پس در Prefix Matching دچار مشکل می شود چون با چند Prefix تطبیق می کند. در اینجا پدیده Longest Prefix Matching استفاده شود تا بیشترین تطبیق به عنوان Next Hub انتخاب شود. در این روش باید با حل جدول تمایل صورت بپذیرد که پیچیدگی زیادی دارد. پس از روش های مختلف برای سرعت دادن به آن، استفاده از ساختار Binary Tree است که حجم آن زیاد است. روش های دیگری نیز پیشنهاد شده است.

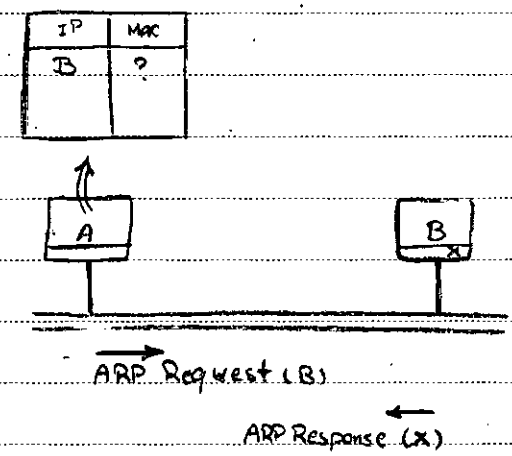


* یکی دیگر از مسائل IP تخصیص آدرس IP به آدرس Physical است. آدرس Mac عددی شش بایتی است (در Ethernet). حل مسئله تخصیص این آدرس ها ARP نام دارد.

Subject:
Year. Month. Date. ( )

دارد (Adress Reservation Protocol). این پروتکل به N.I بستگی دارد.

* ARP در شبکه Ethernet



یک ARP Table در نودها به صورت متقابل وجود دارد. اگر آدرس IP بسته خاصی به آدرس فیزیکی خاصی مربوط نباشد، نود بسته‌ای به نام ARP Request را Broadcast می‌کند که آن آدرس IP گیرنده است. host دارای آن شماره IP، بسته را گرفته و بسته ARP Response حاوی آدرس MAC خود به فرستنده باز می‌گرداند. این entry در جدول دارای طول عمری بین 5 تا 30 دقیقه خواهد بود تا اگر دستگاهی از شبکه خارج شد و IP آن به دیگری منتقل شد، بتوان این تغییر را recover کرد.

* RARP (Reverse ARP) عکس ARP است. شبکه‌ی مورد نیاز است که host های بدون دیسک هستند و وقتی boot می‌شوند باید IP خود را دریافت کنند. برای این کار درخواستی می‌دهند و یک سرور مخصوص اینکار که در شبکه وجود دارد پاسخ آنها را می‌دهد.

* آدرس MAC آدرسی 6 بایتی است که 3 بایت اول آن کد شرکت تولیدی آن (مثلاً برای شرکت cisco داریم 00-00-0C) و 3 بایت بعدی شماره سریال آن است (2^{24} شماره).

* پروتکل ICMP Internet Control Message Protocol

در مدل‌های لایه‌ای، هر لایه وظیفه‌ای دارد که امکان دارد در اجرای آن با مشکل روبرو شود که این خطا باید به نحوی handle شود.

مثلاً: TTL = 0

No match found in routing table

Error in IP checksum

پروتکل فوق به منظور این است که اگر خطایی در لایه IP رخ داد، این خطا به مبدأ گزارش شود. در حقیقت برای انتقال خطا در لایه IP است.

از وظایف دیگر این لایه امتحان اتصال بین دو نقطه (Connectivity) است. این کار با بسته‌ای به نام Echo Request رخ می‌دهد. از برنامه‌هایی که اینکار را می‌کنند برنامه Ping است. و به کمک آن می‌توان نودهای موجود در شبکه را پیدا کرد. با دادن محدوده IP های شبکه و ping کردن تک تک می‌توان Network Discovery کرد.

برنامه Trace Route می‌تواند با اضافه کردن یکی یکی شماره TTL مسیر بسته از مبدأ به مقصد را پیدا کند و بدین طریق از توپولوژی شبکه نیز می‌توان اطلاع یافت.

پیغام‌های ICMP بسته توسط بسته‌ی IP حمل می‌شود (نه TCP و UDP) و در آن «Protocol ID = 1» است. این پیغام خود دارای ساختاری است که در RFC آن موجود است. یکی از فیلدهای آن نوع پیغام را مشخص می‌کند.

* IPv6

IPv4 کاربرد بسیاری یافت که رشد بزرگ آن موجب بروز مشکلاتی در آن شد. اصلی‌ترین مسئله آن تمام شدن محدوده IP Address بود. در سال 1994 انجمن IETF با توجه به این مسئله IPv6 را پیشنهاد کرد که مزایای آن نسبت به IPv4 عبارتند از:

- Longer Address Field

  32 bit → 128 bit : 3.4×10^{38} address

- Simplified Header Format

  فیلدهای مربوط به Fragmentation دیگر لازم نیستند و حذف شده‌اند.

Subject:

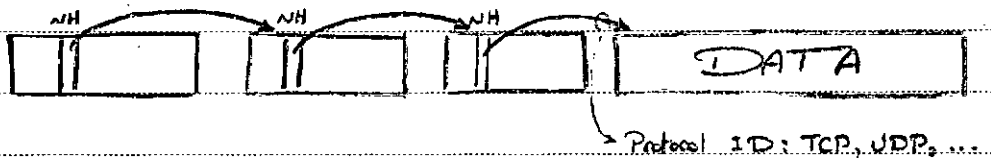
Year. Month. Date. ( )

به جدید تمام NIC های امروزی Error Checking دارند ، نیازی به Header Checksum نباید.

تست Header Length حذف شده است یعنی طول آن همواره ثابت است.

- Flexible Support of Options

هر Header فیلدی به نام Next Header وجود دارد که به Header بعدی اشاره می‌کند. در آن نوع Option مشخص می‌شود. این زنجیره نهایتاً به یک پد به پایان می‌رسد که در آن تست Next Header آن Protocol ID قرار می‌گیرد.



- Flow Label Capability

مشخص می‌کند بسته متعلق به چه جریانی است تا بتوان جریان مورد نظر را دنبال کرد و این عمل QoS را بالا می‌برد. این نیاز با زیاد شدن ارتباطات multi media اهمیت بیشتری پیدا شد.

- Security

عملیات رمزنگاری و تصدیق هویت در IPv4 با اضافه شدن IP Sec حمایت می‌شود در حالی که در v6 به صورت درون سازی شده وجود دارد.

- Large Packets Support

این بسته‌ها بزرگتر از 64 KB ، Jumbo Packet نام دارند. (Byte = octet)

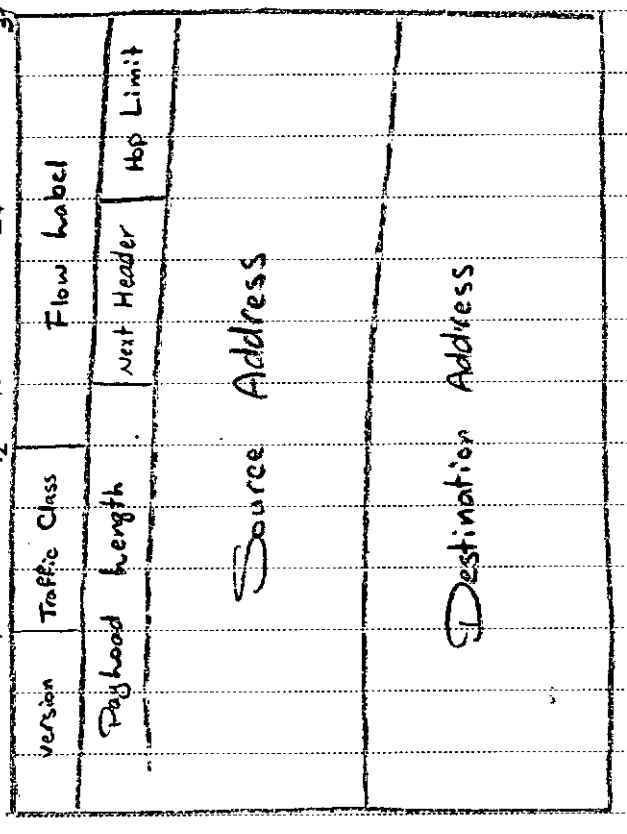
- Fragmentation at Source Only

این عمل وظیفه تقدیم‌های میانی را ساده‌تر می‌کند که این عمل کارایی را بالا می‌برد.

- No Checksum Field

Subject: 17

Year. Month. Date. ( )



* قالب IPv6 به صورت روبرو است:

Traffic Class – معادل Type Of Service است.

Payload Length – طول قسمت Data (< 64KB) را بر حسب بایت

Hop limit – معادل TTL است.

header در v6 از لحاظ اندازه بزرگترند ولی

بسیار ساده‌تر هستند.

برای پشتیبانی از Jumbo یک header دیگر

header اصلی ترتیب می‌کند که در

Payload length صفر می‌باشد، مشخص

می‌کند که بسته Jumbo است.

* معادل لایه Transport در مدل OSI شامل دو پروتکل TCP و UDP است.

* UDP (User Datagram Protocol)

این پروتکل بسیار ساده است و یک پیغام به طول حداکثر 64 KB را از لایه IP دریافت می‌کند و به لایه بعد می‌دهد. ضرورت وجود این لایه برای عملیات زیر است:

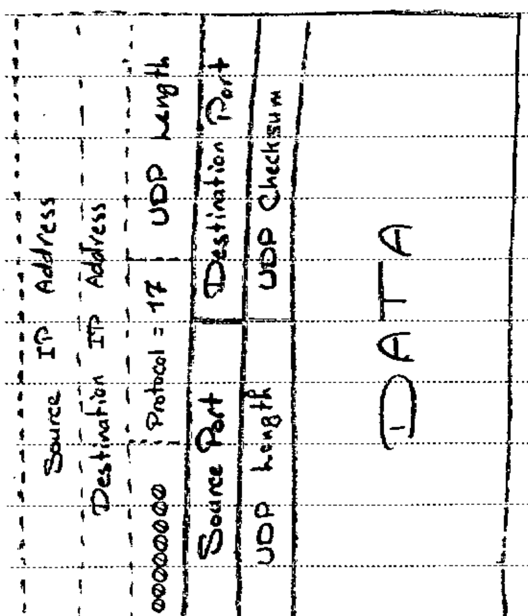
- TCP یک پروتکل مطمئن لایه‌ی دو به دو کنترلهایی که اعمال می‌کند زمانبر است. پس سرویسهایی که زمان برای آنها اهمیت دارد و سرویسهای Real Time و نیز سرویسهایی که پیچیدگی کدهای ایجاد می‌کند و سربار TCP برای آن زیاد است از UDP استفاده می‌کنند مانند: RTP، SNMP، DNS و ...

Subject:
Year. Month. Date. ( )

• IP برای داده‌ها checksum نمی‌گذارد. به همین دلیل UDP اینکار را انجام می‌دهد تا بتواند انواع خطاها (و نیز خطای بیتی) را تشخیص دهد.

• اگر برنامه‌های Application بیشتری وجود داشته باشد که بخواهند از UDP استفاده کنند، خود UDP آنها را multiplex می‌کند و در مقصد UDP آنها را demultiplex می‌کند به این صورت که به کمک فیلد Port Number، Applicationها را از یکدیگر جدا می‌کند (در حالی که در لایه IP، پروتکل‌ها از هم جدا می‌کند).

• بیشتر سراغ Applicationهایی می‌روند که سرعت برای آن مهم‌تر از بی‌خطا بودن است. مانند سرویس‌های صوتی و ویدیویی و ....



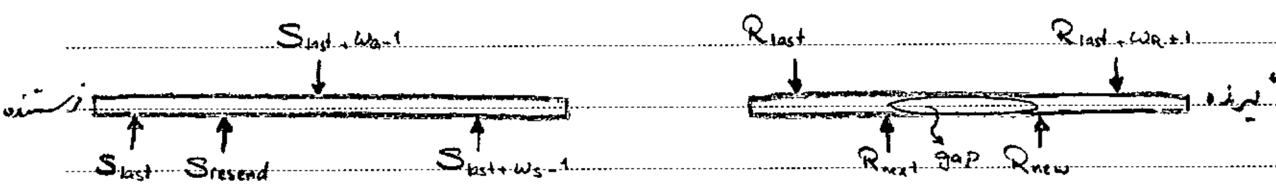
فرمت کلی UDP header به صورت روبرو است:
مبدأ در بالای header قسمت مجازی psudo header را ایجاد می‌کند که به تناسب آن با header می‌تواند ایراد را کشف کند. این قسمت را همراه بسته می‌فرستد و نیز خود نگه می‌دارد تا اگر بسته دچار خطایی در لایه Transport شد آن را بفهمد:

• طول بسته اشتباه باشد.
• بسته از مسیر اشتباه آمده است.
• بسته به مقصد اشتباه آمده است
• پروتکل بسته غیر از UDP باشد.
• ...

UDP Checksum سربار محاسباتی دارد، می‌توان با حذف آن سرعت را بالا برد، با صفر قرار دادن آن در مبدأ به مقصد می‌فهمانیم که نمی‌خواهیم checksum را محاسبه کنیم. اگر خود محاسبات checksum در مبدأ برابر صفر شود، شکل دیگر آن (111 ... 11) را می‌فرستیم.

* پروتکل رایج در لایه Transport ، TCP (Transmission Control Protocol) است.
در سرویس reliable ، connection-oriented ، byte stream data transfer است. TCP
برای ارائه سرویس مطمئن نیاز به روش کنترل خطا دارد که از روش ARQ: Selective Repeat
استفاده می‌کند.

در این روش در فرستنده و گیرنده بافرهایی وجود دارد که فرستنده بدون دریافت ack به میزان پنجره بسته
می‌فرستد و گیرنده آنها را به ترتیب دریافت کرده و درخواست بسته‌های باقی مانده را می‌کند. در روشهای
ARQ اطلاعات کنترلی قبل از ارسال Data تبادل می‌کنند، پس تمام سرویسهای reliable ما
connection-oriented است. در لایه فرستنده هنگام فرستادن اطلاعات TCB ایجاد
می‌شود (Transmission Control Block) که بافرها در آن تعریف می‌شود.



Send Window　　　Receive Window

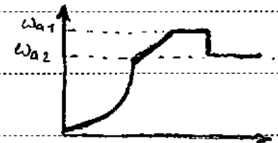
شماره‌ترتیب به دلیل بودن byte-stream روی بایت‌ها تعریف می‌شوند نه segment ها.
S_{last} : آخرین بایتی که تأییدیه آن را گرفته است.
S_{recent} : آخرین بایتی که فرستادیم ولی ack آن را نگرفته‌ایم.
$S_{last+Wa-1}$: اطلاعاتی که application به لایه TCP می‌دهد.

$S_{recent} - S_{last} < S_{last} + W_s - 1$

Wa : حداکثر میزان پنجره‌ای است که گیرنده توان دریافت آن را دارد و وابسته به flow control است.
Ws : طول بافر فرستنده است.
R_{last} : آخرین بایتی که به ترتیب دریافت کردیم و ack را فرستادیم.
R_{next} : آخرین بایتی که application آن را خوانده است.
R_{new} : آخرین اطلاعاتی که به ترتیب دریافت کردیم.
Wa اندازه جای خالی گیرنده برابر $(R_{last} + W_R + 1) - (R_{next})$ است که به صورت app ابتدای بیت

گیرنده همواره در ack خود W_A را به فرستنده اعلام می‌کند. اندازه پنجره از W_A و W_S کوچکتر است.

پس :



در حالت کنترل ازدحام

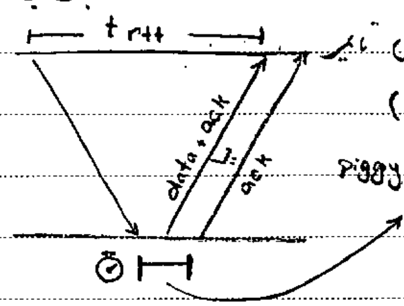
$$W_A = W_R - (R_{new} - R_{last})$$

↳ advertis

تعداد بایتی که ارسال شده اما ack نگرفته است:

$$S_{recent} - S_{last} \leq W_A$$

TCP یک ارتباط full duplex است. به روش piggy backing دیتا را پیاده می‌کند. در آن در مسیر برگشت داده‌ها ack نیز به مقصد حمل می‌شوند و هزینه خاصی ندارد. البته با ارسال هر segment یک تایمر روشن می‌شود که اگر در مدت آن داده‌ای برای برگشت نباشد، بسته‌ای ack خالی را برای ack می‌فرستد. این تایمر با ارسال ack خاموش می‌شود (به طریق: سوار بر data یا جدا). اگر این تایمر expire شود، ack بدون piggy backing فرستاده می‌شود.



برای محاسبه تایمر time out داریم:

$$T_{timeout} = t_{Rtt} + k\,\sigma_{Rtt}$$

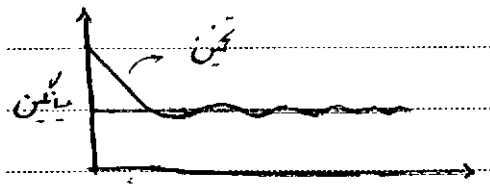
انحراف معیار زمان رفت و برگشت بسته   که متوسط زمان رفت و برگشت بسته

$$T_{Rtt} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

این روش محاسبه T_{Rtt} مناسب نیست چون به همه داده‌ها در پایان ارسال دسترسی داریم. برای تخمین T_{Rtt} از روش‌های مختلفی تخمین می‌زنند و مقدار اولیه را مثلاً به صورت دستی مقدار پیش‌فرض تنظیم می‌کنیم. یکی از روش‌های تخمین این است:

$$T_{Rtt}(new) = \alpha\, T_{Rtt}(old) + (1-\alpha)\, T_n \quad , \quad 0 < \alpha < 1$$

در TCP برای α مقدار "$\alpha = 7/8$" را تخصیص می‌دهند. علت انتخاب این کسر، آسانی پیاده‌سازی "$\alpha \cdot 1/8$" با یک عمل شیفت متوالی است. مقدار α قابل تغییر است.



حال باید α_{RTT} را محاسبه کرد (تخمین زد) که TCP به صورت زیر محاسبه می‌کند:

$$\alpha_{RTT}(new) = \beta\,\alpha_{RTT}(old) + (1-\beta)\,|T_n - t_{RTT}| \quad , \quad 0<\beta<1$$

که β در TCP مقدار $3/4$ در نظر گرفته می‌شود.

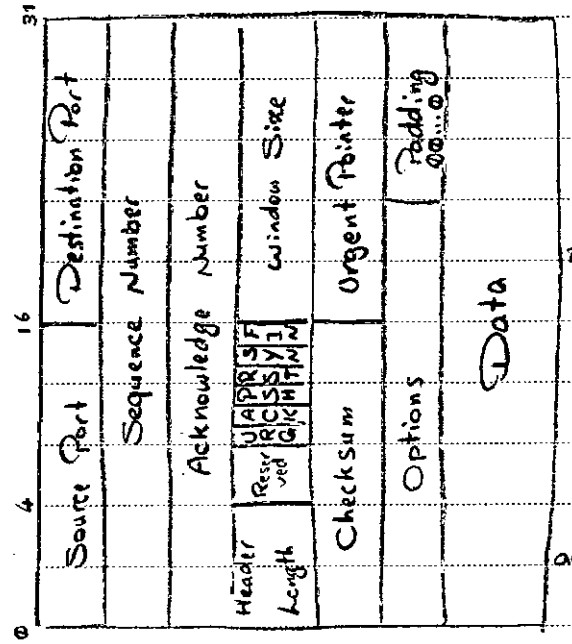
در TCP مقدار K برابر 4 در نظر می‌گیریم:

$$t_{out} = t_{RTT} + 4\,\alpha_{RTT}$$

* برای انجام کنترل جریان TCP هر header به لیست اضافه می‌کند که در حد یک segment است، بنابراین محدود به 64KB است. هنگامی که اطلاعات به اندازه یک segment size شد TCP آن را می‌فرستد، مگر در دو حالت:

• Application درخواست ارسال سریع اطلاعات را دارد و به اصطلاح آن را Push کرده است (از دستور push استفاده کرده است).

• از تایمر data مدتی گذشته ولی میزان آن = segment size نرسیده است.



TCP header به صورت روبرو است:

- ack # مربوط به ack در حال برگشت است
- طول TCP header متغیر است از ۲۰ تا ۶۰ بایت
- $w_a \equiv$ window size
- برای محاسبه checksum احتیاج به pseudo header دارد ولی Protocol ID آن همیشه ۶ است.
- اگر طول header مضرب 16 نباشد در انتها padding صفر اضافه می‌کنیم.
- بیت SYN برای برقراری ارتباط است.
- بیت ACK برای ارسال ack است، ack # معنادار می‌شود.

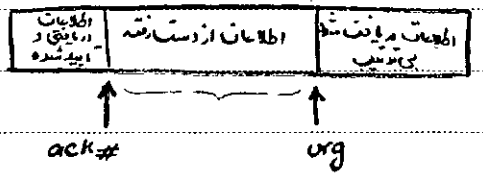
Subject:
Year. Month. Date. ( )

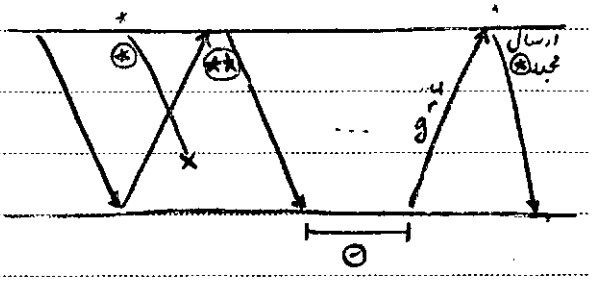
- PSH اطلاعاتی که توسط app فرستاده push شده‌اند.
- FIN برای خاتمه ارتباط است (توسط app)
- URG برای selective reapeat مرتبی است که می‌فهمیم اطلاعات دوباره ارسال شود. اطلاعات مورد نظر با شماره Urgent Pointer مشخص می‌شود. این آدرس نسبی است و از جایی که ack دریافت شده است جابجایی را مشخص می‌کند.
- RST وقتی فرستاده می‌شود که به دلیل غیر طبیعی ارتباط قطع شود.
- برای window size 16 بیت اختصاص دادیم که به طرف مقابل داده 64 KB محدود می‌کند. راه‌حل برای افزایش این عدد window scaling است که از option بیشماره می‌رود. در آن واحد را می‌توان 1 Byte ، 2 Byte ، 4 Byte و ... قرار داد.

تمرین سری سوم : فصل ۸

۳ - ۵ - ۸ - ۱۳ - ۱۴ - ۱۵ - ۲۳ - ۲۵ - ۲۸ - ۳۰

* اگر اطلاعات در TCP از این حد بیشتر بودن می‌توان با URG Pointer استفاده کرد.

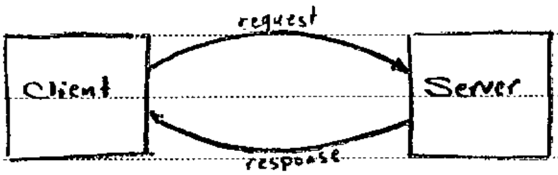




دریافت ** گیرنده می‌فهمد که * نرسیده است و تا بزرگی را کاری می‌اندازد تا منتظر * باشد. پس از انقضای آن درخواست urg می‌فرستد تا * مجدداً ارسال شود.

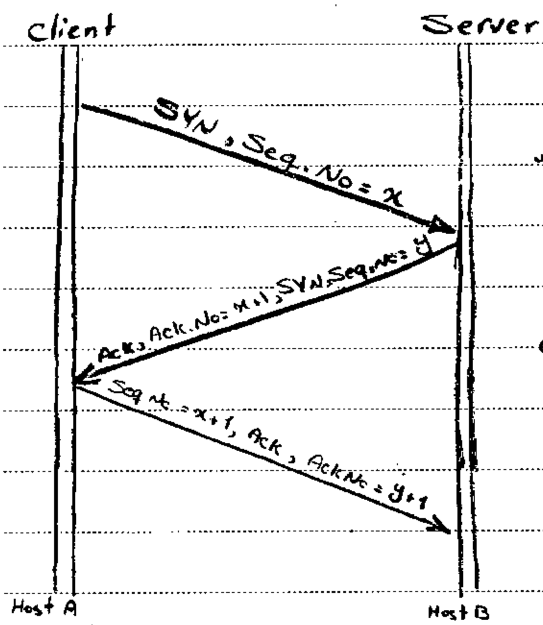
* تمام پروتکلهایی که در شبکه مشخص می‌شوند یک state machine دارند. TCP نیز از این قاعده مستثنی نیست. در شکل 28-8 کتاب شکل آن آمده است. برای تغییر state نیاز به رخ دادن یک event است. معمولاً برای نمایش این پروتکلها زبانی به نام SDL وجود دارد (Specification & Description lang.) و سطح بالاست و توسط ITU استاندارد شده است (International Telecommunication Union). SDL گاهی با نام System Design lang نیز خوانده می‌شود.

* Client – Server : server همیشه آماده سرویس دادن هستند ولی تنها در زمانی که درخواستی از طرف client دریافت کنند برای آن اطلاعات را ارسال می‌کنند.



☑ web server — browser

* TCP برای بازکردن connection از مدل تلفنی استفاده می‌کند ولی پس از آن کاری به این مدل ندارد. کامپیوتری که نقش server را دارد یک TCP Port را باز می‌کند و در حالت listen قرار می‌گیرد. client نیز یک TCP Port باز می‌کند. روند برقراری ارتباط به کمک بسته‌های کنترلی و به روش 3way handshaking صورت می‌گیرد.



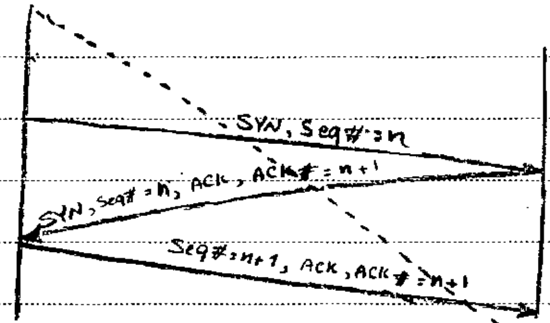
Connection Establishment

• Seq. No از یک عدد تصادفی آغاز می‌شود. علت آن است که در صورت قطع شدن ارتباط قبلی اگر دوباره از Seq. No x آغاز شود، با برقراری ارتباط جدید ممکن است دریافت بسته‌های ارسالی ارتباط قبلی شماره‌های تکراری ایجاد کنند. پس Seq No باید طوری عوض شود که اندازه پنجره دریافت نتواند بسته‌های قبلی را بپذیرد.

$Seq\ No = 4\ GB = 2^{32}$

$Window\ Size = 2^{20}$

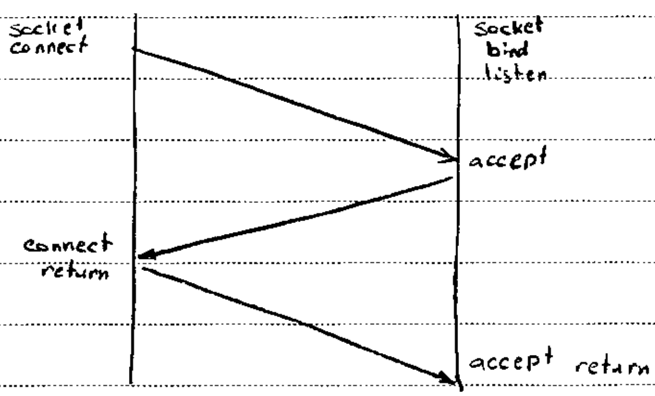
Subject:
Year. Month. Date. ( )



* بین هر دو لایه یک interface است. بین دو لایه نرم افزاری TCP و Application یک interface نرم افزاری (API) وجود دارد که امکاناتی برای استفاده لایه بالاتر می‌دهد مثلاً:

- socket
- bind : port# خاصی را باز می‌کند. مثلاً برای server ، 80 است.
- listen
- read
- write
- close

Connection Establishment ☑



* فازهای ارتباط

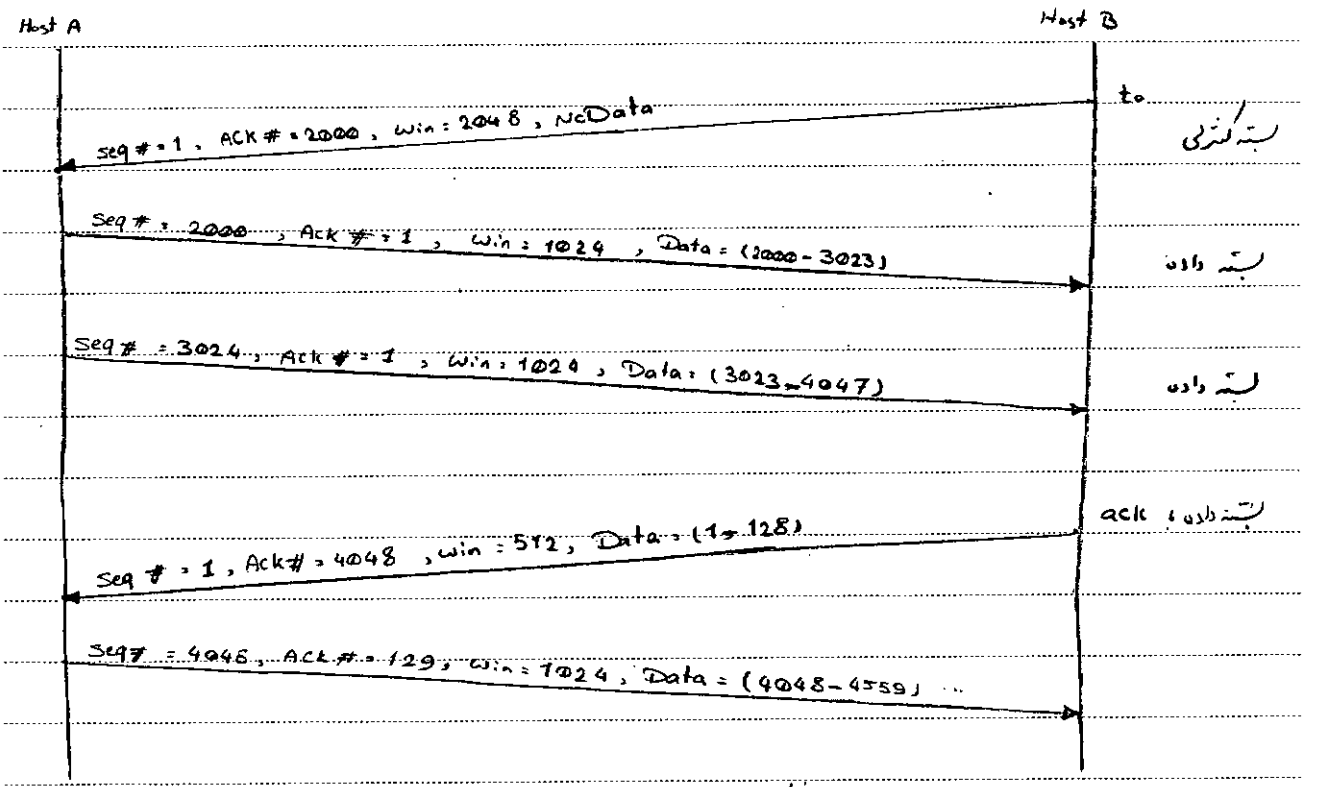
Connection Setup ⇨ Data Transfer ⇨ Connection Release

* فاز Data Transfer

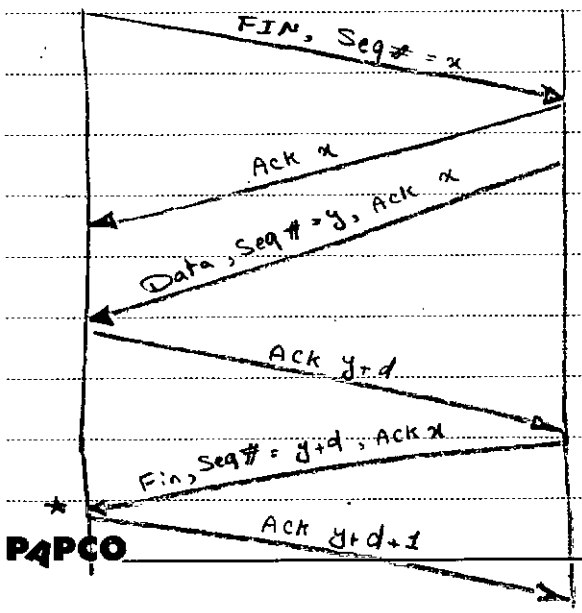
الان ارتباط برقرار است. با زدن win ، عمل flow Control صورت می‌گیرد. در واقع ارسال بیهوده‌ای در TCP نداریم.

Subject: 21

Year. Month. Date. ( )



* TCP چون در ارتباط برقرار می‌کند (B→A , A→B) باید به نحوی مستقل ببندد و زمان بستن آن بستگی به App دارد. برای خاتمه ارتباط TCP پس از خالی کردن بافر TCP فرمان finish را می‌فرستد که دستور آن توسط تابع close صادر می‌شود. این امر توسط بسته کنترلی حاوی FIN انجام می‌دهد، پس seq# نیز به جلو می‌رود.



* در اینجا بلافاصله بعد از connection بسته می‌شود چون ممکن است بسته‌هایی در شبکه وجود داشته باشند. این فاصله زمانی بسته، خاتمه ارتباط برابر است با:

2 MSL ≈ 4 min

↳ Maximum Segment Lifetime

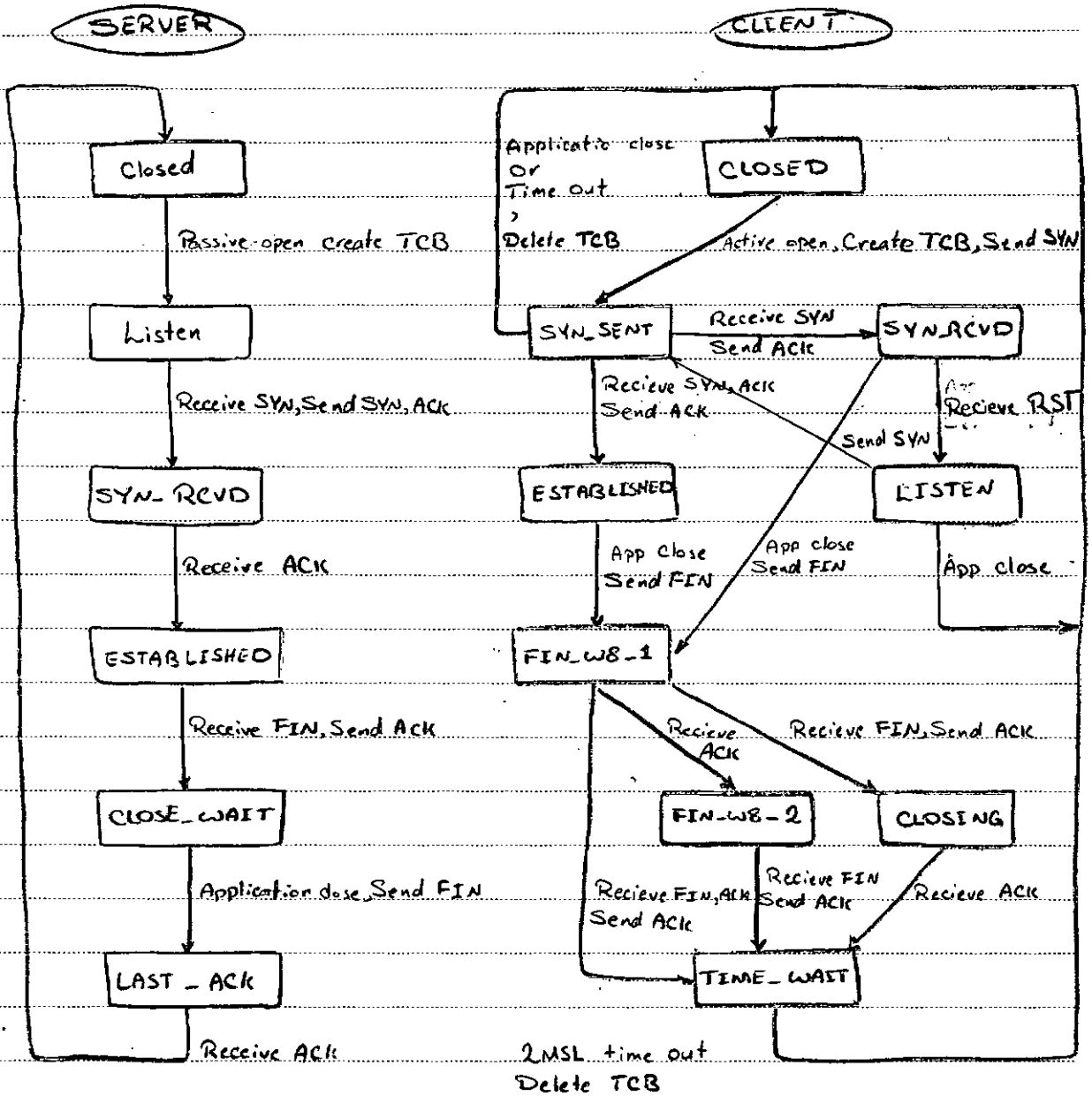
یک سیستم وقتی restart می‌شود به اندازه MSL اجازه برقراری هیچ اتصالی را نمی‌دهد.

P4PCO

Subject:

Year. Month. Date. ( )

TCP State Diagram



* چون در لایه IP ارتباط connectionless است ترتیب دریافت FIN, ACK و حالت FIN_WB_1 نیز

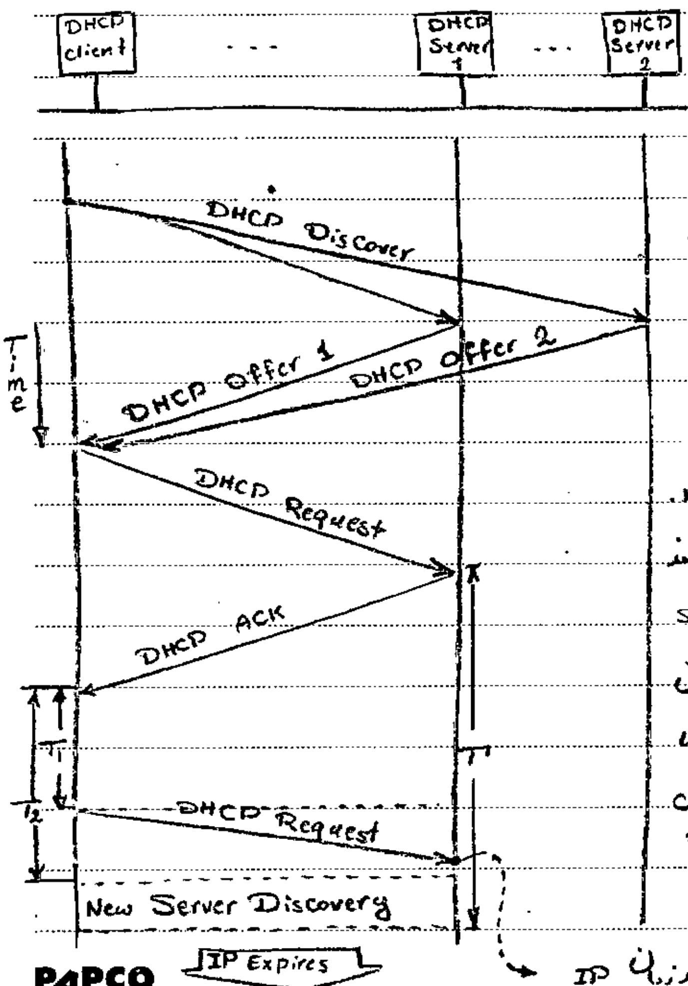
است به حالت اولی، دومی و یا هر دو باشد.

* پروتکل DHCP

برای افزودن یک node به شبکه یک سری تنظیمات (Configuration) لازم است.

- این کار می‌تواند به صورت دستی و توسط افراد انجام گیرد (IP ثابت) اما در این صورت احتمال خطا زیاد می‌شود.
- اگر بخواهیم یکی از کامپیوترها را جابجا کنیم مجدداً تنظیمات باید انجام شود.
- فرض کنید در شبکه N تا کامپیوتر و $N/2$ تا IP معتبر داریم، در این صورت همزمان نمی‌توان از همه آنها استفاده کرد.
- زمانی که ارتباط با اینترنت از طریق ISP انجام می‌گیرد، یک IP به host تخصیص داده می‌شود که این مکانیزم به صورت اتوماتیک است.

برای رفع مشکلات فوق پروتکلی به موجب تعریف پروتکل « پیکربندی نیز با آن به صورت اتوماتیک » شده است.
((Dynamic Host Configuration Protocol))



یک server در داخل شبکه این سرویس را ارائه می‌دهد که حداقل یک DHCP Server در شبکه لازم است. عمل ارسال پیام‌های DHCP از طریق پورت‌های 67 و 68 که مربوط به UDP است صورت می‌گیرد. در این مرحله client براساس یک مکانیزم مشخص یکی از offerها را می‌پذیرد.

— اولین درخواست را می‌پذیرد (معمولاً).
— به صورت تصادفی درخواست‌ها را دریافت می‌کند یکی را می‌پذیرد.

وقتی client یک server را انتخاب می‌کند به server یک درخواست می‌فرستد. server به client یک ACK می‌فرستد که در آن زمان T وجود دارد که client این IP را مدت T می‌تواند در اختیار داشته باشد. client در حدود دو تایمر T_1 و T_2 روشن می‌کند. معمولاً $T_1 = 0.5T$ است. در این زمان client

درخواست تمدید زمان IP

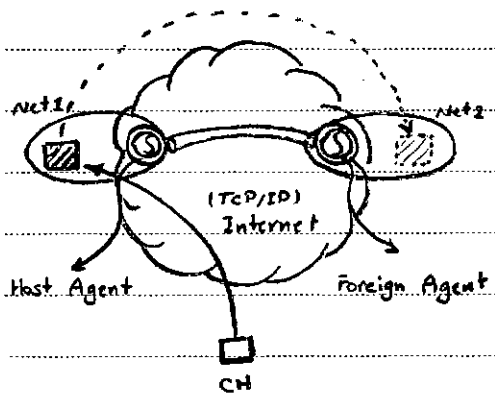
Subject:

Year. Month. Date. ( )

Server درخواست می‌کند تا مدت قرارداشتن IP را تمدید کند. در صورتی که از Server پاسخ ACK نیاید، در زمان T_2 که معمولاً $T_2 = 0.625\,T$ (قابل تنظیم) درخواست Discovery می‌دهد. به هر حال در زمان T این IP توسط پروتکل رها می‌شود. $(T_2 .. T_2)$ زمان انتظار برای ACK است.

Mobile IP

شبکه TCP/IP برای route کردن بسته‌ها به سمت مقصد از NetID استفاده می‌کند. اگر یک host قابلیت حرکت داشته باشد و از شبکه خود خارج شود (home net) و وارد شبکه دیگر (foreign net) شود، سه مرحله‌ای زیر را می‌باید:



- عوض کردن IP مقصد Mobility نیاز، ولی بسیار آسان است. نرم CH این را می‌کند ولی
- می‌فهمد و اگر سرویس به آن ارائه می‌دهیم قطع می‌شود.
- پروتکل Mobile IP امکان عدم تغییر IP را می‌دهد در صورتی که بسته شبکه home
- home agent را به Agent foreign آن حمایت کند. برای اینکار به
- foreign agent نیاز است که معمولاً روی Gateway پیاده‌سازی می‌شود (می‌تواند)

1. Agent Discovery

هر gateway (FA) معمولاً در فواصل زمانی معین پیام‌های خود مانند پیام Hello Message را ارسال می‌کند که MH‌های وارد شده به آن Net آنها را متوجه شده، پاسخ می‌دهند.

2. Address Assignment

نرم متحرک از شبکه مقصد آدرس می‌گیرد که به آن Care-of-adress می‌گویند که می‌تواند از طریق DHCP Server یا Foreign Agent آنرا بگیرد. این آدرس می‌تواند داخل Mobile Host یا FA نگهداری شود.

3. Registration

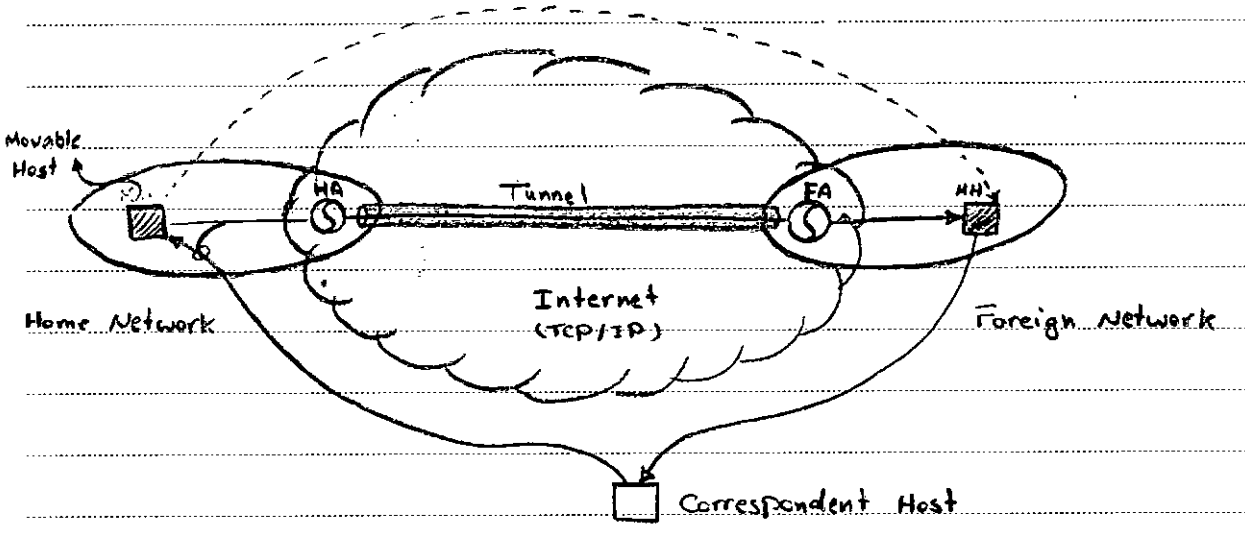
در داخل HA یک جدول ایجاد می‌شود که در آن آدرس MH و Care-of-Adress هم نگاشت می‌شود.

Subject: 23

Year. Month. Date. ( )

4. Tunneling

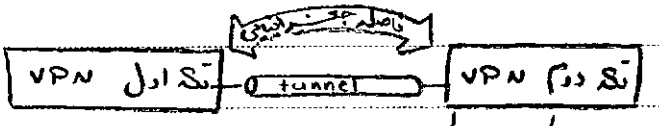
بین HA و FA یک مسیر مجازی ایجاد می‌شود که بسته‌هایی که از آن عبور می‌کنند مقدشان مخفی است. اگر MH در home net بود از آدرس معنی آن استفاده می‌کند. اگر MH در foreign net باشد از طریق tunnel به آن می‌رسد. پس مسیریابی در اینترنت متفاوت است که این روی QoS تأثیری ندارد.



* Tunneling

پروتکلی است که از IP استفاده می‌کند و دارای "ID=4" است. وقتی می‌خواهیم Packet را از تونل عبور دهیم آنرا به عنوان Data در بسته‌ای قرار می‌دهیم که مبدأ آن HA و مقصد آن FA است. در حقیقت بسته‌ها در بسته‌های تونل encapsulate می‌شوند.

کاربرد آنها در VPN (Virtual Private Network) است.



محاسن آن هزینه کم (به خاطر بهره‌برداری از لینک فیزیکی است)، سرعت نسبتاً بالا و تغییرات آن راحت است (مثلاً تغییر Bandwidth)، معایب آن امنیت کم و تأثیرپذیری QoS به خاطر عمومی بودن لینک می‌باشد. در حقیقت VPN بجای استفاده از لینک فیزیکی مخصوصی با استفاده از شبکه TCP/IP عمومی Private Network را می‌سازد.

کاربرد دیگر آن در IP Sec است که بسته‌ها به صورت رمزگذاری شده در داخل بسته‌های دیگر می‌فرستند.

Subject:

Year. Month. Date. ( )

* سیستم دچار کل tunneling در Mobile IP در صورت جابجایی زیاد MH، تعداد agent های واسطه زیاد می‌شود و در نتیجه تأخیر افزایش می‌یابد. در هر مرحله می‌توان با فرستادن یک Binding Message از طرف HA به CH، آدرس جدید MH را به آن داد تا خود با Foreign Net یک Tunnel برقرار کند.

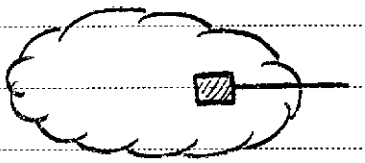
* در شبکه‌های IP مسیریابی به صورت پویا انجام می‌شود. پروتکل‌هایی که اطلاعات مربوط به جداول مسیریابی را بین گره‌ها انتقال می‌دهد، پروتکل‌های مسیریابی می‌نامند.

* شبکه اینترنت مربوط به هیچ سازمان و نهاد خاصی نیست، ولی از چندین شبکه و سازمان مختلف تشکیل شده است.

* Autonomous System مجموعه‌ای از Router‌ها هستند که تحت مدیریت یک سازمان باشند که می‌توانند از یک پروتکل مسیریابی استفاده کنند یا از چند پروتکل. به طور معمول از یک پروتکل استفاده می‌شود.

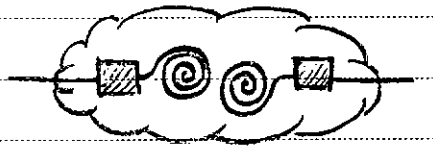
* Autonomous System:

I. Stub A.S.



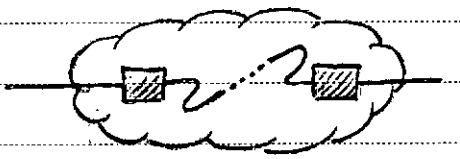
شبکه‌هایی که از طریق فقط یک Router به بیرون متصل هستند.

II. Multihomed A.S.



چند Router دارند به شرطی که ترافیک از خود عبور ندهند.

III. Transit A.S.



چند Router برای انتقال به جهان بیرون دارند و ترانزیت هم دارند (یعنی ترافیک را از خود عبور می‌دهند.)

* Stub A.S. احتیاجی به A.S. Number ندارند چون تنها یک Gateway دارند (اگر بیشتر داشته باشند می‌توان از آن‌ها برای توزیع به شبکه استفاده کنند). A.S Number برای پروتکل‌های مسیریابی مورد استفاده قرار می‌گیرند.

Subject:

Year. Month. Date. ( )

* مسیریابی بین AS ها برای پیدا کردن مسیرهای رسیدن به مقصد ← BGP

Exterior Gateway Protocol (EGP)

مسیریابی داخل AS ها برای پیدا کردن بهترین مسیر ← RIP ، OSPF

Interior Gateway Protocol (IGP)

RIP : Routing Information Protocol

OSPF : Open Short Pass First

BGP : Border Gateway Protocol

* RIP از روش Distance Vector استفاده می‌کند، بنابراین اطلاعات رد و بدل شده، جداول مسیریابی هستند که اینبار از طریق پورت 520 توسط UDP انجام می‌شود. در این روش هر گره هزینه‌ای (metric) برابر یک دارد.

در RIP حداکثر هزینه 16 است. هزینه‌های بزرگتر مساوی 16 را ∞ در نظر می‌گیرند (در RIP سرعت و طول بین گره‌ها تأثیر ندارد.) اطلاعات هر ۳۰ ثانیه یکبار به گره‌های مجاور می‌رود. چون از UDP استفاده می‌شود، ممکن است پیامها نرسد. لذا هر گره اگر بیش از ۱۸۰ ثانیه هیچ پیام اطلاعاتی را دریافت نکند، خطر را برآن می‌گذارد و لینک مربوط قطع شده و هزینه آنرا ∞ قرار می‌دهد. یک گره می‌تواند از فاصله‌ی خود بخاطر نداشته باشد تا گره خاص را برگزیند، مثلاً دقتی زمان زیادی است update نشده یا پایی با مبدأ نامشخص داریم.

0	8	16	32
Command	Version	Zero	
Address Family ID		Zero	
IP Address			
Zero			
Zero			
Metric			
⋮			

Routing Information Message = RIP entries

(max ی‌توان ۲۵ تا entry داشت - حجم پیام ۲۰ بایت)

Command ← در پیام RIP از چه نوعی است. دو نوع داریم: اگر 1 باشد Request است هر ۳۰ ثانیه یکبار response می‌آید. اگر 2 باشد Response

Subject: 25

Year. Month. Date. ( )

- version نسخه RIP را مشخص می‌کند. امروزه دو نسخه 1 و 2 موجود است.
- Address Family Identification: از آنجایی که RIP فقط برای IP طراحی نشده است (هر چند تا کنون فقط برای IP استفاده شده است) در این فیلد می‌گوییم که در کجا استفاده می‌کنیم. برای IP عدد 2 را داریم.
- IP Add آدرس گره مقصد است.
- Metric: عددی بین 1 تا 15 است. 16 به بعد ∞ در نظر گرفته می‌شود.

نکته: به تعداد سطرهای جدول routing در فیلد آخر نگاه می‌شود (حداکثر 25 تا)

نکته: version 1 از CIDR پشتیبانی نمی‌کند. ضمن از لحاظ security هم ایراد دارد.
version 2 از CIDR پشتیبانی می‌کند.

* OSPF از روش Link State استفاده می‌کند. پس اطلاعات رد و بدل شده، وضعیت لینک‌ها می‌باشد. در این روش یک پایگاه داده Link State از لینک‌های داخلی یک A.S دارد و با استفاده از یک الگوریتم مسیریابی متمرکز مثل Dijkstra می‌توان بهترین مسیر را پیدا کرد (این عمل به طور مستقل توسط هر node اجرا می‌شود.)

مزیت آن نسبت به RIP این است که اهمیت را تعریف کرد یعنی نظر hop count نیست. (در واقع وضعیت یک لینک) قابل تعریف است. هر پیام که می‌رسد با توجه به لینک چه خاصیتی برایش مهم است به دنبال جدول مربوط به آن می‌گردیم و بهترین مسیریابی را انجام می‌دهیم. برای آن یک عدد 16 بیتی داریم، لذا می‌توانیم حداکثر 2^{16} جدول مختلف برای کاربردهای مختلف داشته باشیم. از CIDR پشتیبانی می‌کند. از security حمایت می‌کند. اگر مسیرهایی پیدا شده که cost آنها یکی باشد، این پروتکل خاصیت آن را دارد که ترافیک را بین آنها تقسیم کند.

پیام‌های OSPF مربوط به مسیریابی را مستقیماً در IP Packet قرار می‌دهند، Protocol ID = 89

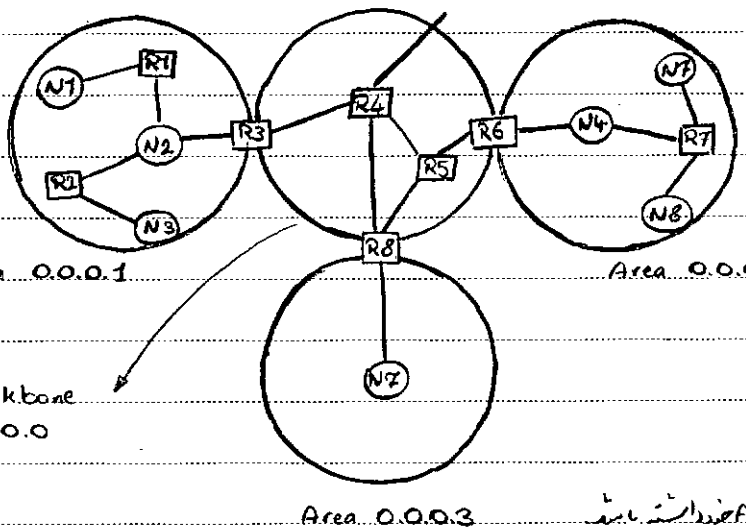
* در RIP اطلاعات به همه گره‌ها Broadcast می‌شود ولی در OSPF اینگونه نیست و انتخاب می‌کند که مسیریابی با کدام Router باشد (در Multi Access) و اگر P2P باشد میان Router انتخاب می‌شود.

* معمولاً در شبکه‌ها سه لایه در نظر می‌گیرند:
  - لایه Access
  - لایه Distribution
  - لایه Backbone

*

- Routing Protocol
  - IGP
    - RIP
    - OSPF
  - EGP — BGP

* برای اینکه تبادل اطلاعات مسیر در داخل A.S. محدود می‌شود و حجم آن زیاد نباشد در داخل A.S. نیز یک تقسیم بندی صورت می‌گیرد:



این ناحیه به صورت Admin توسط static تعیین می‌شود.

حالت سلسله مراتبی در دو مرحله (لایه) با استفاده از این روش تنها لازم است پرینت اطلاعاتی در مورد Area خود داشته باشد

- Internal Router
- Area Border Router
- Backbone Router
- Autonomous System Boundary Router

Subject: 26

Year. Month. Date. ( )

* در اولین گام، OSPF، هر Router، Router‌های همسایه خود را شناسایی کند.
Router همسایه (Neighbor) هستند که N.I. مشترک داشته باشند (از این رو N.I. هم متصل باشند).
در فاصله زمانی مشخص پیام‌های «Hello Message» برای Router‌های همسایه فرستاده می‌شود.

Subject:

Year. Month. Date. ( )

* وقتی Router با هم سنکرون می‌شوند آن‌ها را Adjacent می‌نامیم. در لینک‌های Multi-acess وقتی تعداد نودهای neighbor زیاد باشد و چند پیغام Hello Message باید رد و بدل شود یکی از آن‌ها به عنوان اصلی و دیگری به عنوان back up انتخاب می‌شود.

* عملیات OSPF:

I. پیدا کردن نودهای neighbor

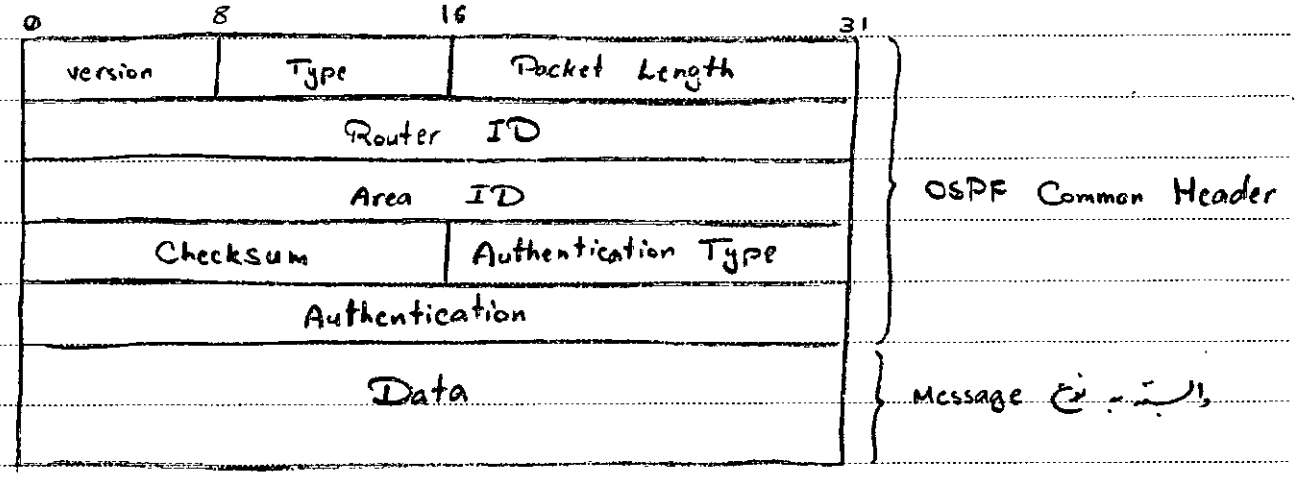
II. adjacency

III. اطلاع رسانی تغییرات به وسیله Advertising Msg ← Link-state update

* عملیات OSPF توسط پیام‌هایی اطلاع داده می‌شوند که عبارتند از:

1. Hello
2. Database Description
3. Link-State Request
4. Link-State Update
5. Link-State Acknowledgement

فرمت این پیام‌ها متفاوت است ولی دارای یکسانی هستند به فرم زیر است:



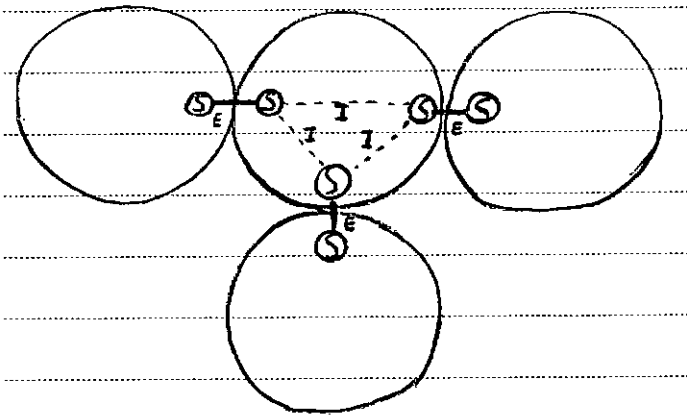
* Hello Msg مانند ضربان قلب است. اگر پیام آن دقیقه ایجاد شد، ارتباط متصل برقرار است و اگر قطع شد ارتباط از بین رفته است.

Subject: 27

Year. Month. Date. ( )

* پروتکل مسیریابی BGP:

این پروتکل از نوع ——— است و نقطه به نقطه ای gateway ها برای A.S ها اجرا می شود. به همین دلیل آن ها را border ... می نامند.



اطلاعات خود را از طریق TCP منتقل می کنند به مدخل هایی که از BGP استفاده می کنند، BGP Speaker گفته می شود که اطلاعات مسیریابی بین آن ها جریان است که به دلیل استفاده از TCP این اطلاعات مطمئن منتقل می شوند.

* انواع BGP:

- EBGP (external): انتقال بین gateway های دو ناحیه
- IBGP (internal): انتقال بین gateway های یک ناحیه

* انواع پیام های BGP عبارتند از:

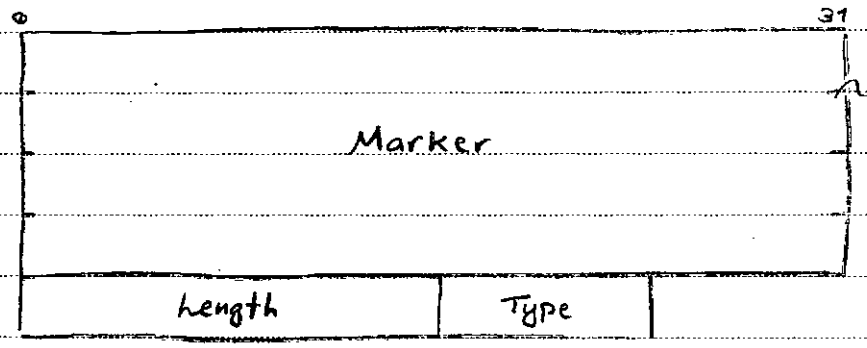
1. OPEN
2. UPDATE
3. NOTIFICATION
4. KEEP ALIVE

پس از برقرار شدن انتقال TCP، open فرستاده می شود.
تغییرات در BGP توسط update اطلاع داده می شود.
خطاها و ایرادات توسط notification هشدار داده می شود.
تست کردن ارتباط، مانند keep alive، hello msg ارسال می کند.

Subject:

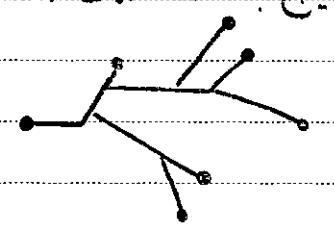
Year. Month. Date. ( )

* فرمت کلی Header در BGP مشترک است و عبارتست از :



* Multicast Routing برای راه حلهای مختلفی است :

- در ابتدا به تعداد مقصد از داده کپی گرفته شده، فرستاده شود که این بدترین راه حل است.
- داده به صورت درخت پشته گسترده می‌شوند. احتیاج به Multicast Server



داده با به کار بردن بسته بندی و تخلیص آدرس به آنها به این فرآیند کمک کند.

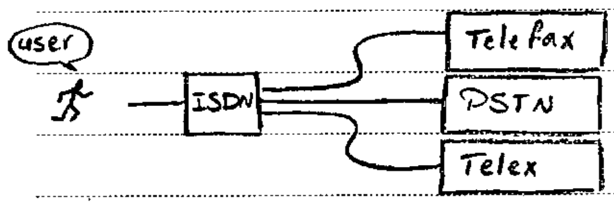
* انواع پخش پیام (Routing):

- Uni Cast (تک پخشی)
- Multi Cast (چند پخشی)
- Broad Cast (همه پخشی)

Subject: 28
Year. 84 Month. 9 Date. 8 (15)

* شبکه‌ی ATM

• اوایل دهه‌ی ۸۰ تنوع سرویسهای مخابراتی زیاد شده بود. این تنوع سرویس احتیاج به Interface ی زیادی داشت. ITU (سابق CCITT) برای اجتماع این سرویسها به ایده‌ی ISDN متوسل شد (Integrated Service Digital Net) تا این Interface این شبکه‌ها با کاربر را استاندارد کند (یعنی User to Network Interface را استاندارد کرد). با اختراع فیبر نوری ایده‌ی Broad Band ISDN (B-ISDN) مطرح شد. این سرویسها دارای نرخ بیت متغیر است (Variable Bit Rate) دارا



که ISDN نرخ بیت ثابتی دارد. برای حل این ناسازگاری دو راه حل مطرح شد:

• Time Division Multiplexing ← complexity ↑ (پیچیدگی سخت‌افزاری)

• Packet Switching ← delay ↑ , ✓VBR

ITU برای استفاده بهینه از هر دو راه، راه حلی بینابینی پیدا کرد که مزایای TDM را برای بهینه کردن Packet Switching استفاده می‌کند که همان ATM است.

* ATM از نوع Virtual Circuit Packet Swit... است.

	Variable Bit Rate	Delay	Bursty Traffic	Processing
TDM	Multirate Only	Low, Fixed	Inefficient	Minimal, very high Speed
Packet	Easily Handled	Variable	Efficient	Header and Packet Processing Required
ATM				

Subject:
Year. Month. Date. ( )

* طول بسته‌ها ثابت ← مدار سخت‌افزاری ← سرعت ↑

حجم پردازش بدی بر بسته ثابت است. ← pipe lining (موازی کردن عملیات ذاتاً سری)

Virtual Packet Sw. ، VER را حمایت می‌کند.

Virtual Packet Sw. ، ترتیب منطقی را نیز بخوبی سرویس می‌دهد.

استفاده از Virtual Circuit احتیاج به جستجو در جدول مسیریابی را از بین می‌برد.

Error Control را بر می‌داریم چون اکثر سرویسهای Multimedia نیاز ندارند (جایش end 2 end)

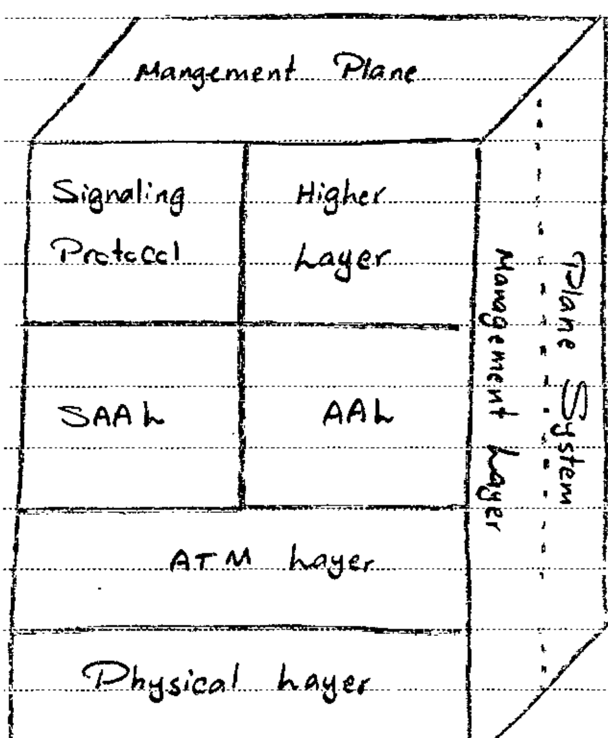
Flow Control هم با استفاده از متدهای Open Loop صورت می‌گیرد.

برعکس TDM در ATM تخصیص منابع نداریم پس روی بسته‌های خروجی صف داریم، که تأخیر را تحمیل می‌کند.

* مدل لایه‌ای ATM

ITU مدل لایه‌ای ۴ سطحی روبرو را برای ATM در نظر گرفته است.

I. Higher Layer
II. ATM Adaptation Layer
III. ATM Layer
IV. Physical Layer

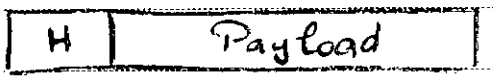


لایه دوم بسته به نوع سرویس مورد استفاده می‌تواند متنوع باشد. لایه اول تنها در مودهای پایانی ارائه می‌شود و اطلاعات user را جابجا می‌کند. اطلاعات دیگر در این شبکه اطلاعات کنترلی (Signaling) است، ارتباط را کنترل کنند (برقراری، انتقال، قطع) که Signaling Protocol آن را بر عهده دارد و لایه دوم SAAL (Signaling ATM Adaptation Layer)

آن را برقرار می‌کند. این دو وجه جانبی باید در تمام نودهای مسیر نیز حاضر باشند.
تمام عملیات توسط قسمت Management کنترل می‌شوند که بر عملکرد سه لایه نظارت دارد.
خطاهای فیزیکی نیز توسط این قسمت مدیریت سیستم نظارت می‌شود.

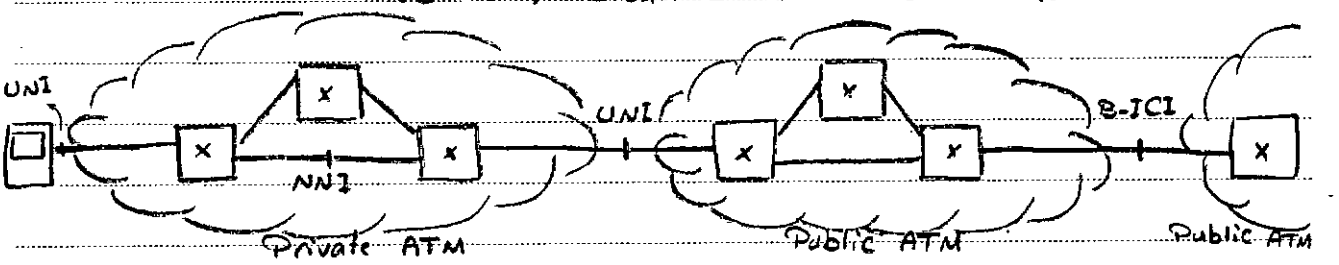
* لایه ATM استقلال از سرویس است یعنی به Higher layer کاری ندارد. تنها وظیفه آن برآوردن QoS مناسب آن سرویس است. چون سرویس‌های مختلف، انتظارات متفاوتی دارند.

* لایه Physical در شبکه‌ی ATM به دو قسمت است:
• شامل دو لایه
— لایه اول وظیفه رسانه فیزیکی، Line Coding و ... است.
— لایه دوم وظیفه Framing را برعهده دارد. شبکه‌ی ATM طول ثابتی دارند که اصطلاحاً cell خوانده می‌شود. طول آن 53 Byte است که 5 بایت آن header است، 48 بایت داده‌ی مختلف payload است.



این لایه Convergence Sublayer ندارد.

* در شبکه‌ی ATM سه نوع Interface وجود دارد:
I> User 2 Network (UNI)
II> Network 2 Network (NNI)
III> Broad Band Intercareer (B-ICI)



Subject:
Year. Month. Date. ( )

وقتی یک Private ATM می‌خواهد سیگنال Public ATM منتقل شود برای آن یک
مرزبند user خواهد بود با UNI به هم متصل می‌شوند. اتصال دو شبکه Public
با B-ICI در سرعت 155 Mbps صورت می‌گیرد.

* سرعت‌ها در ISDN

B: data (64 Kbps), D: signaling

Basic Rate Interface: 2B+D = 144 Kbps

Primary Rate Interface: 30B+D = 2 Mbps معادل E1 خطوط

* خطوط مخابراتی به نام trunc شناخته می‌شوند:

E1 •

1 " کانال برای سنکرون سازی

1 " کانال برای Signaling

30 " کانال برای انتقال data

این کانال‌ها به روش TDM با هم Multiplex شده‌اند.

E1 استاندارد "اروپایی" نیز خوانده می‌شود.

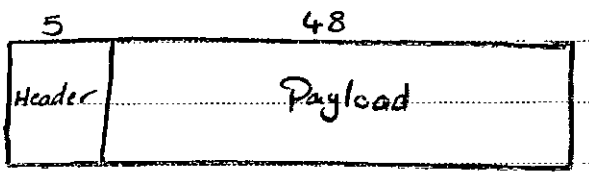
T1 •

23 " کانال برای انتقال data +

T1 استاندارد "آمریکایی" نیز خوانده می‌شود.

Subject: 30
Year. 84 Month. 9 Date. 15 (16)

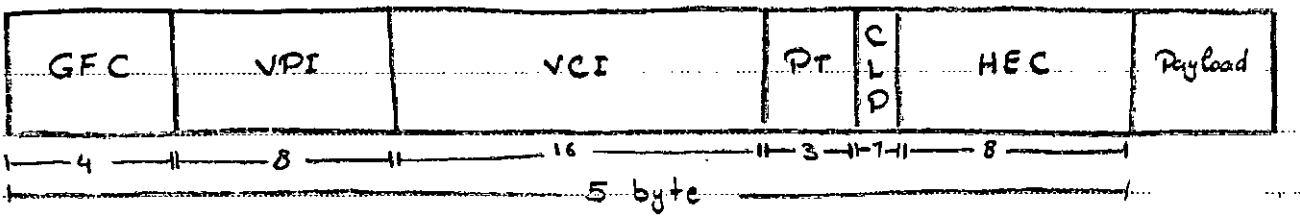
ATM Cell *



کوچکترین واحد لایه ATM ، عمل forwarding در ATM بر اساس virtual circuit است و چون Header از آن استفاده می‌کند، لذا کوچکترین بخش سلول شماره شناسایی مدار مجازی است.

فرمت لایه‌های ATM در مرجع Interface متفاوت است. user 2 net و net to net

* فرمت سلول ATM برای UNI به صورت زیر است:



GFC : Generic Flow Control

VPI : Virtual Path Identifier

VCI : Virtual Channel Identifier

PT : Payload Type

CLP : Cell Loss Pericrity

HEC: Header Error Control

* برای فرمت NNI از سلولی مانند فرمت UNI استفاده می‌کنند. با این تفاوت که در آن GFC حذف شده و VPI به 16 بیت افزایش می‌یابد.

* مجموعه VPI + VCI را تشکیل می‌دهند Virtual Circuit Identifier این بخش بندی (Circuit = Path + Channel) امکان آدرس‌دهی سلسله مراتبی را فراهم می‌کند. معمولاً در نوع ATM Switch داریم:

① VPI Switch ② VCI Switch

Subject:
Year. Month. Date. ( )

* Payload Type

I ، اطلاعات user و انواع آن

II ، اطلاعات کنترلی و انواع آن (OAM (Operation & Administration & Maintenance)

* Cell Loss Periority (CLP)

اگر «1» باشد در حالت ازدحام اولویت بالاتری برای حذف شدن دارد. در ابتدای تولید بسته همواره «0» است. اگر تولید کننده از قبل داد حذف تخلف کند در واحد اعمال Policy در ATM (که open loop است)، «1» کردن این بیت آنرا tag می‌زند.

* Header Error Control (HEC)

برای اطمینان از درست بودن header پاره‌سازی شده است (CRC 8).
در ATM کنترل خطا داریم ولی برای header حتماً باید اعمال شود.

* Generic Flow Control (GFC)

بخش flow control که در ATM بخاطر استفاده از open loop باید کنترلی
بر user interface شبکه اعمال شود تا بار بیش از حد بر شبکه وارد نشود و ازدحام ایجاد
نشود. به این دلیل در UNI وجود دارد ولی در NNI وجود ندارد.

* حالتهای Payload Type

b3 : اطلاعات user است یا خیر b3 = 0 → user

b2 : هنگام تولید سلول ATM این بیت صفر است. در هنگام عبور از مسیر اگر به نودی برخورد کند
که دچار ازدحام باشد، این بیت «1» می‌شود تا به مقصد آنرا اطلاع دهد.

b1 : از وظایف لایه ATM Adaptation است. شامل segmentation، reassembly
سلول‌ها با اندازه ثابت 48 byte می‌سازد. از انواع AAL، 4 AAL و AAL نوع 5 است که
از این بیت استفاده می‌کند. در این نوع اگر b1 = 0 باشد، قطعه دریافت شده، قطعه شروع و
میانی یک segment است (قطعه‌ها به ترتیب بخاطر virtual circuit) و اگر «1» باشد قطعه
پایانی است.

PqPCO

Subject: 31

Year. Month. Date. ( )

b_3	b_2	b_1	
0	0	0	user data, congestion not experienced, SDU Type = 0
0	0	1	user data, congestion not experienced, SDU Type = 1
0	1	0	user data, congestion experienced, SDU Type = 0
0	1	1	user data, congestion experienced, SDU Type = 1
1	0	0	OAM F5, segment associated cell
1	0	1	OAM F5, end-to-end associated cell
1	1	0	Resource Management (RM) cell → Available Bit Rate سرویس
1	1	1	Reserved for future use

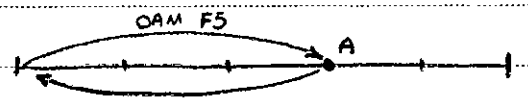
* انواع تقسیمات مدیریتی

OAM
- F4: virtual path (سطح مسیر)
- F5: virtual channel (سطح کانال)

(F1، F2، F3 روشهای فیزیکی استفاده می‌شوند)

هرکدام از آنها می‌توانند برای segment یا end-to-end باشند.

connectivity برای تشخیص ☑



یک سلول OAM F5 با آدرس "A" فرستاده و loop back دریافت می‌شود.

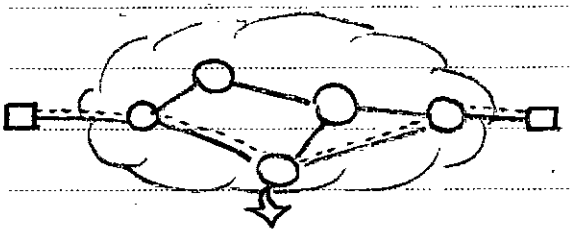
* ATM از نوع Virtual Circuit است.

هر نود یک Routing Table دارد.

انواع Virtual Circuit عبارتند از:

- Permanent (PVC) — این برای همیشه setup می‌کند.
- Switched (SVC) — برای مدت زمانی که می‌خواهیم یک مسیر ایجاد کنیم.

PVC ☑ می‌توان VPN درست کرد.



Destination Node	VCI	Next Hop Node	VCI

Subject:

Year. Month. Date. ( )

* شبکه‌های ATM سعی QoS را تضمین می‌کند کیفیت سرویس با تعدادی پارامتر معرفی می‌کند. این پارامترها در ATM به شرح زیر است که در شروع برقراری ارتباط کاربر به شبکه می‌دهد که یک "V.C" و این مشخصات می‌خواهد (هر سرویس نیازی نسبت به همه را مشخص کند):

• Cell Error Ratio

شبکه مسیریابی می‌کند و می‌بیند آیا می‌تواند درخواست "X" را رعایت کند. این کار به کمک CAC انجام می‌شود.

$$X = \frac{\text{تعداد سلول‌های دارای خطا}}{\text{تعداد سلول‌های ارسال شده}}$$

• Cell Misinsertion Rate

نرخ ارسال اشتباه سلول‌ها به سایر گیرنده نباید از این حد بیشتر باشد. (واحد $cell/sec$)

• Severely-errored Cell Block Ratio

اگر تعدادی سلول پشت سر هم (یک block) خراب شدند، در برقراری ارتباط و QoS تأثیر بدی دارد. به عنوان مثال اگر اندازه بلوک را "M" در نظر گیریم و از "K" سلول بیشتر دچار خطا باشد، بلوک را خطا در نظر می‌گیریم.

$$X = \frac{\text{تعداد بلوک‌های دارای خطا}}{\text{تعداد بلوک‌های سلولی ارسال شده}}$$

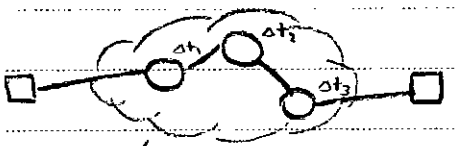
• Cell loss Ratio

از دست رفتن کمتر از این مقدار در ارتباط قابل چشم‌پوشی است.

$$X = \frac{\text{تعداد سلول‌های از دست رفته}}{\text{تعداد سلول‌های ارسال شده}}$$

• Cell Transfer Delay

میانگین تأخیر ارسال در این حد قابل تحمل است.



$$t_d = \Delta t_1 + \Delta t_2 + \Delta t_3$$

این تأخیر در دو قسمت خروجی (بخاطر ترافیک، queuing و ...) را با یک مدل ریاضی مدل می‌کنند و امید ریاضی آنرا میانگین تأخیر ارسال در نظر می‌گیرد.

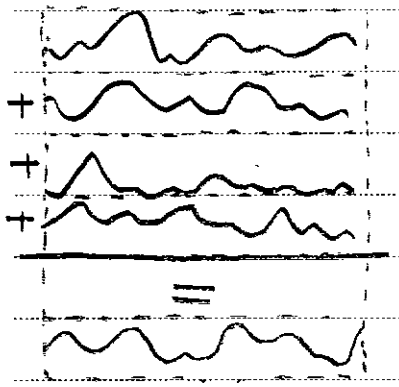
• Cell Delay Variation

واریانس تأخیر ارسال نسبت به میانگین است که آنرا jitter می نامند (فاصله از میانگین)

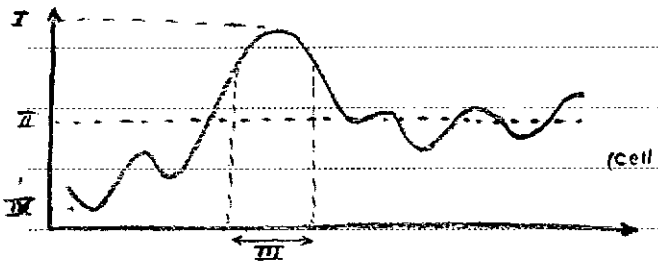
☑ بخش تصاویر روی اینترنت به میانگین ارسال حساس نیست ولی jitter برای آن مهم است.

☑ ارتباط تلفنی که interactive است علاوه بر واریانس تأخیر به میانگین نیز نیاز دارد.

* BISDN کیفیت سرویس را تضمین می کند. بنابراین user باید پارامترهای کیفیت مد نظر خود را اعلام کند. شبکه در صورتی که توانایی دریافت و ارائه چنین کیفیتی را داشته باشد، اجازه برقراری ارتباط را از طریق واحد CAC می دهد. این واحد با جمع آماری ترافیک می بیند که آیا می تواند ترافیک جدید را برای ارائه سرویس بپذیرد یا خیر.



* هر ترافیک با یک سری مشخصات آماری مشخص می شود. در شبکه ATM به چهار شاخص آماری ترافیک توجه می شود (Traffic Descriptors)



I، Peak Cell Rate

II، Sustainable Cell Rate

III، Maximum Burst Size (cell واحد)

IV، Minimum Cell Rate

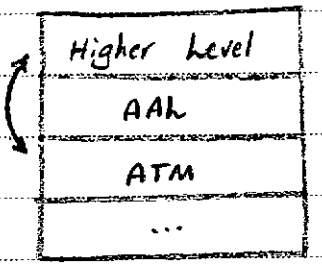
واحد CAC این اطلاعات را دریافت می کند و اجازه برقراری ارتباط را می دهد یا نمی دهد.

واحد UPC پس از برقراری ارتباط به روش open loop به ارتباط نظارت می کند تا از قرارداد منعقد شده براساس شاخص های بالا تخلف نکند.

Subject:

Year. Month. Date. ( )

* لایهٔ AAL (جمع بندی)



لایهٔ بالا می‌تواند شامل هر App باشد، پس داده‌ی آن می‌تواند stream یا packet باشد. این لایه وظیفه Segmentation و Reassembly را بعهده دارد.

وظیفه اصلی این لایه، استقلال لایهٔ ATM و لایهٔ App از سرویس ارائه‌ی کند است. اینکار نظیر کنندهٔ QoS خواهد بود. ITU سرویس را به چهار کلاس A تا D تقسیم می‌کند و برای هر کدام یک AAL خاص طراحی کرده است:

* Class A ⟶ AAL 1
  - Constant Bit Rate (CBR) ← مکالمه صوتی - روش PCM
  - timing
  - connection - oriented

* Class B ⟶ AAL 2
  - variable Bit Rate (VBR) ← تصاویر ویدیویی MPEG
  - timing
  - connection - oriented

؟ روش MPEG یک روش تناظلی است. نرخ بیت آن 2-6 Mbps است. بین هر دو تصویر اختلاف به روشهای گراف ذخیره شده که یک ماتریس خاصت (sparse) است که حجم آن متغیر است.

Subject: 33

Year. Month. Date. ( )

* Class C → AAL 3/4 , AAL 5
  - VBR — X.25 ☑
  - no-timing
  - connection-oriented

* Class D → AAL 3/4 , AAL 5
  - VBR — IP Packets ☑
  - no-timing
  - connection less

علاوه بر این چهار نوع AAL ، نوع دیگری نیز برای انتقال اطلاعات کنترلی وجود دارد که با نام SAAL (Signaling ATM Adaptation layer) معرفی است.

* تولید کننده‌های تجهیزات مخابراتی مثل AT&T ، Alcatel و ... مجمع ATM Forum آنها دسته‌بندی کرده‌اند:

I. CBR	: constant bit rate	(AAL 1 ≡ class A)
II. rt-VBR	: real time variable bit rate	(AAL 2 ≡ class B)
III. nrt-VBR	: non real time variable bit rate	(AAL 3,4 ≡ class C, D)
IV. ABR	: available bit rate	(AAL 2)
V. UBR	: unspecified bit rate	(AAL 5)

* ABR : دارای timing است ولی نرخ بیت در دسترس را به سرویس تخصیص می‌دهد برای کاربردهایی است که در آن سرویس می‌تواند QoS خود را تطبیق دهد مانند انتقال تصاویر که امکان تغییر نرخ بیت شبکه بستگی دارد.

* UBR : تصمیم‌گیری CAC نسبتاً محافظه‌کارانه است و همواره میزانی از نرخ بیت شبکه آزاد می‌ماند

تا کیفیت سرویس را تضمین کند. UBR بدون تضمین QoS و به صورت best effort
این پهنای باند را به سرویسی تخصیص می دهد. (چون CAC از جمع آماری استفاده می کند
پس گاهی فضا بیشتر از استفاده می شود و UBR سوء استفاده می کند!)

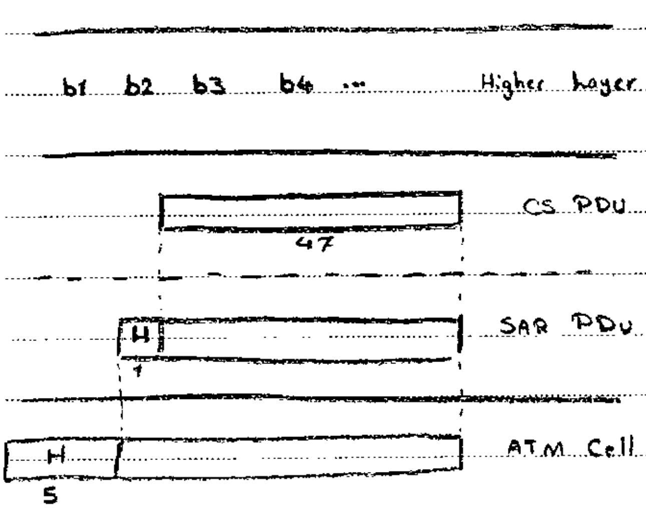
* لایه AAL را می توان به دو زیر لایه تقسیم کرد:

I. SAR (Segmentation And Reassembly)
قطعه قطعه کردن داده به تکه های 48 بایتی در فرستنده
بازسازی قطعات در گیرنده

II. CS (Convergance Sublayer)
تعیین مشخصات ترافیکی

common part (SSCS) ← روی تمام داده ها اعمال می شود.
service specific (CPCS) ← وابسته به سرویس و احتیاجات کاربر

* AAL 1



higher layer داده ها به صورت stream با زمانبندی ثابت به CS
می دهد. CS هر 47 بایت را
یک CS PDU تبدیل می کند.
آنها به SAR می دهد. SAR با
افزودن 1 بایت header آن را
به SAR PDU تبدیل می کند
و آنها به لایه ATM می دهد و آن
با افزودن 5 بایت header تبدیل
به ATM cell می کند. header
لایه SAR برای بازسازی قطعات در
مقصد است.

header لایه SAR به شرح زیر است:

Convergence Sublayer Indicator	Sequence Count	Sequence Number Protection
1 bit	3 bit	4 bit

CSI ← 0 : 47 بایت متعلق به کاربر

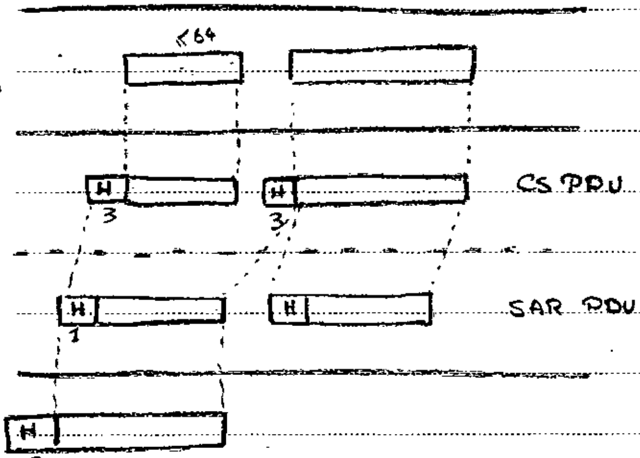
1 : 46 بایت متعلق به کاربر + 1 بایت اشاره گر لایه CS

متعلق به عملیات همگام سازی CS در طرف ها

بهره وری در این روش: $\frac{47}{47+6} = \frac{47}{53} = 89\%$

سرباری معادل 11% برای بدست آمدن timing می پردازیم

* AAL2



زمانبندی بسته ی higher level ثابت است ولی طول آن متغیر است (حداکثر می تواند 64 byte باشد) پس سرویس VBR است ولی timing دارد. در قسمت CS به بسته ی دریافتی 3 بایت header اضافه می کنیم و سپس بسته ها را پشت هم می چسبانیم تا یک stream شوند و سپس آنها را 47 بایت 47 بایت می کنیم و به آن SAR، 1 بایت اضافه می کند ...

آخرین بسته از SAR PDU برای رسیدن به 47 بایت از padding استفاده می کند.

AAL2 می تواند اطلاعات در higher layer را با هم multiplex کند یعنی با یک ارتباط بفرستد. فیلد channel identifier (CID) در CS header که 1 بایت است به ما کمک می کند این کار را انجام دهیم.

*

Subject:
Year. Month. Date. ( )

اگر به طور فرضی طول بسته‌ی تولیدی کاربر 32 بایت باشد (میانگین):

بهره‌وری (utilization): ضرب بهره‌وری لایه‌ها: $\frac{32}{35} \times \frac{47}{48} \times \frac{48}{53} = 82\%$

سربار: 18%

* AAL 3,4

حداکثر طول بسته‌ی تولیدی در higher level برابر 64KB است. در لایه‌ی CS به آن 4 بایت header و 4 بایت trailer اضافه می‌شود. طول بسته با padding به مضرب ... می‌رسد. در لایه‌ی SAR این CS PDU که 44 بایت است با 2 بایت header و 2 بایت trailer به 48 بایت می‌رسد (در آخر با padding به این اندازه می‌رسد). وظیفه‌ی این AAL انتقال این اطلاعات با طول متغیر است.

قالب کلی CS PDU به صورت زیر است:

Common Part Indicator → معین می‌کند که در header و trailer چه فیلدهایی می‌آید (فعلاً فقط 0)

Begining Tag → فرستنده با انتخاب این ID شماره‌ی بسته را مشخص می‌شود.

Buffer Allocation Size → به گیرنده طول بافر را اعلام می‌کند تا بتواند پیام را بازسازی کند

Alignment → مقدار آن صفر است و برای تنظیم سلول است.

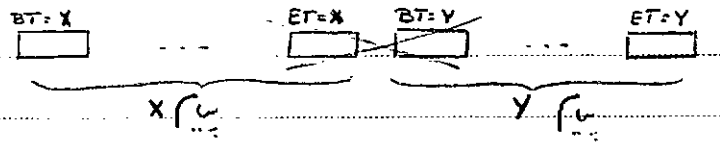
Ending Tag → برابر با B-Tag (این ID در هر سلول با دیگری متفاوت است)

Length → طول بسته را مشخص می‌کند.

Padding → طول trailer را مضرب 4 بایت می‌کند

Subject: 35

Year. Month. Date. ( )



⇒ E-Tag و B-Tag در تشخیص خطای Burst کمک می‌کند.

قالب کلی SAR PDU عبارتست از:

Segment Type → نوع سگمنت که در جدول پایین آمده است.

Segment Number → شماره ترتیب (۰ تا ۱۵) برای تشخیص از بین رفتن خطا

Multiplexing ID → روی یک VC می‌توانند پیام‌های متفاوتی را بفرستند که در این شماره پیام است.

Lenght Indicator → طول واقعی پیام (برای تشخیص طول تا آخر padding دارد)

Seg. Type	Description
0 0	Begining of Message (BOM)
0 1	Continue Of Message (COM)
1 0	End Of Message (EOM)
1 1	Single ... Message (SSM)

بهره وری (اگر طول بسته 64 KB باشد):

$$Eff\ (CS) = \frac{64\ KByte}{(64\ KByte + 8\ Byte)} \simeq 1$$

$$Eff\ (SAR) = \frac{44}{44+4} = \frac{44}{48} = 0.92$$

$$Eff\ (ATM) = \frac{48}{53}$$

$$utilization = \frac{44}{53} = 83\ \%$$

دارای 8% سربار اضافه SAR این AAL 3,4 و کندترین دوسرویس Multiplexing و Error Detection (که هم است) را می‌دهد.

Subject:

Year. Month. Date. ( )

AAL 5 *

به خاطر سربار AAL 3,4 زیاد

مورد پسند سرویس AAL 5 و

سرعت سرویس Multiplexing

بالا بردن utilization بدید

آمد. در لایه CS ، 8 بایت

trailer اضافه می‌کنیم، padding و

طول آن مضرب 48 بایت می‌شود.

لایه SAR این واحدها را به تکه‌های

48 بایتی می‌شکند و سربار بر آن

اضافه نمی‌کند. در پایان 5 بایت

بر آن اضافه می‌شود و سلول ATM را

می‌سازد. برای بازسازی این سلول

دریافت‌کننده سه تا Payload Type موجود در ATM header بازسازی می‌شوند.

قالب trailer به صورت زیر است:

UU (user 2 user) 1 Byte	CPI 1 Byte	Length 2 Byte	CRC 4 Byte

CRC روی کل پیام محاسبه می‌شود و دریافت‌کننده پس از بازسازی پیام خطا مشخص می‌شود.

higher layer می‌تواند یک بایت اطلاعات برحسب نیاز اطلاعات به higher layer

طرف دیگری بفرستد که User 2 User نام دارد.

سربار این روش تنها 5 بایت ATM است و کمینه است (با در نظر گرفتن طول 64 KByte برای اطلاعات):

$$\text{utilization} = \frac{48}{53} = 90\%$$

در حال حاضر AAL5 پیاده‌سازی شده است!!!

Subject: 36
Year. Month. Date. ( )

* Signaling AAL

اطلاعات این AAL توسط تمام گره‌هایی که وظیفه کنترل را دارند پردازش می‌شوند. شکل 20-9 کتاب

این سرویس مطمئن است. در قسمت Service Specific تأمین می‌شود که در آن ارتباط connection-oriented است شبیه AAL5 است. در برقراری ارتباط از روش ARQ: selective Repeat استفاده می‌شود.

* Signaling

عملیاتی که برای کنترل، ایجاد، حفظ و خاتمه ارتباط انجام می‌دهیم عملیات Signaling نام دارد. بر دو نوع است:

① UNI

② NNI

☑ مثال‌های عملی

برداشتن گوشی (UNI) — گرفتن شماره (UNI) — مسیریابی در مرکز مخابرات (NNI) — رزرو کردن منابع (NNI) — ارسال سیگنال زنگ (NNI) — برداشتن گوشی در مقصد (UNI)

در ایران پروتکل Signaling System 7 (SS7) پیاده شده است.

☑ برای سیگنال B-ISDN توسط ITU-T پروتکل‌های زیر تعریف شده است:

UNI: Q.2931 NNI: SS7

* ATM Addressing

دو استاندارد وجود دارد:

Subject:

Year. Month. Date. ( )

E.164 ← I

مانند شبکه مخابرات

حالت سلسله مراتبی

حداکثر 15 رقمی (بدون 0)

1	-	NPA	-	NXX	-	ABCD
country code		area code		office code		subscriber number

ISO NSAP ← II

آدرس پروتکل ATM که عدد 20 بایتی است

(AESP) ATM End System Addressing

IDP	High order DSP	End System ID (ESI)	Selector
6 Byte			1 Byte

Domain ← طول متغیر بسته به

IDP → Inter Domain

High order DSP → Net ID مانند

End System ID → معمولاً شماره MAC دستگاهی

* سیگنالینگ اصلی

- setup
- call proceeding ← شبکه به محض دریافت setup اینرا برای فرستنده (پیام setup دهنده) برگشت می‌دهد
- connect
- connect ack.
- release
- release complete

* روش سیگنالینگ در شبکه‌های ATM معمولاً Source Routing است که در آن مبدأ پیام setup مسیر مخابراتی ایجاد می‌شود.

PqPCO

Subject: 37
Year. 84 Month. 10 Date. 11 ( )

* معماری های پیشرفته

از سال 1990 شبکه های IP از لحاظ سخت افزاری و برنامه های کاربردی پیشرفت زیادی کرده اند.
به همین دلیل کاربرد زیادی دارند.

معایب IP عبارتند از:

I > پذیرفتن ارسال IP (connectionless - best effort) در سرعت بالا به عنوان طرف عمل می کند و باعث کاهش سرعت می شوند.

II > فقدان اطمینان در ارسال

امروزه با رفتن به سمت شبکه های سرعت بالا و نیز کاربرد گسترده IP، سنده هینه کردن IP مطرح می شود.

برای انجام این روش دو نظر گرفته شده است:

I > روش سنتی آدرس دهی IP حفظ شود و عمل مسیریابی با آدرس مقصد ادامه یابد.
در آن باید سرعت پردازش روتر ها (به علت پردازش مداوم ...) بالا برود.
به این روش Pure Destination Based Forwarding می گویند. در آن روتر ها هزینه می شود.

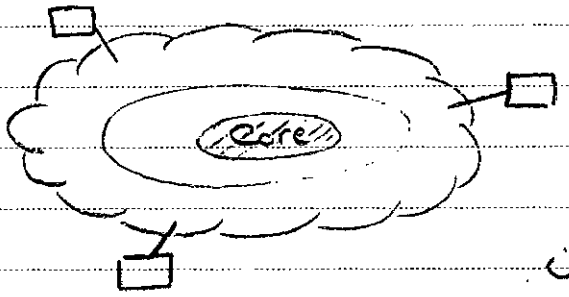
II > در آن براساس برچسبی که به بسته می دهیم (مانند ATM) بسته را به مقصد هدایت کنیم.
به این روش Switched Forwarding می گویند.

* در هر شبکه هرچه به هسته می رویم سرعت رسانه ها بالا می رود و تعداد روتر ها کمتر می شود. از همین رو شبکه را به سه ناحیه تقسیم می کنیم:

I > Access ← پیرامونی ترین

II > Distribution

III > Core



روشهای پیشنهاد شده در لایه Core مطرح می شوند چون سرعت بالا در آن مطرح است (مثل Gigabit Routers)

Subject:
Year. Month. Date. ( )

* روش Switched Routing به دو صورت است.

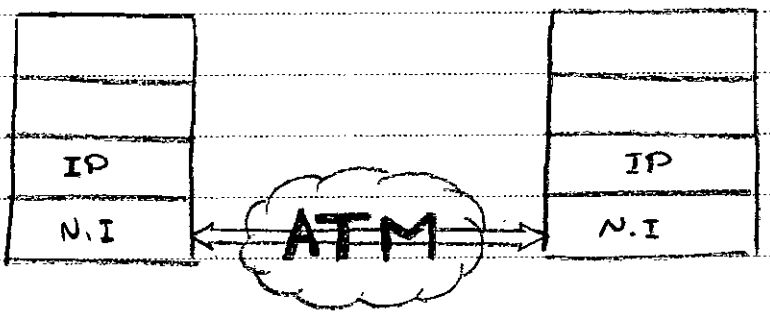
I) Overlay

شبکه ترکیبی ATM Network شبیه‌سازی می‌شود و در حقیقت ATM Core به عنوان شبکه IP Network Interface طرح شده است.

انواع آن عبارتند از:

- CLIP : Classical IP Over ATM
- LANE : LAN Emulation
- MPOA : Multi
- NHRP : Next Hop Reservation Protocol

بزرگترین حسن روش Overlay استقلال شبکه‌ی IP و ATM است که به برنامه قابلیت گسترش و scalability می‌دهد.
ایراد آن اینست که به خاطر احتیاج به آدرس ATM نیاز به یک ARP برای تبدیل آدرس IP به ATM است که پیچیدگی زیادی دارد.



II) Peer

نودهای مرزی، نودهایی هستند که عضو هر دو شبکه ATM و IP هستند و دارای یک آدرس هستند. در آن شبکه ATM مثلاً N.I برای IP نیست، بلکه شبکه‌ای هم‌اندازه و معادل آنست.
به دلیل یکسانی آدرس IP و ATM احتیاجی به ARP نداریم ولی به دلیل وابستگی این دو شبکه به هم محدودیت ایجاد کرده‌ایم.

Subject: 38

Year. Month. Date. ( )

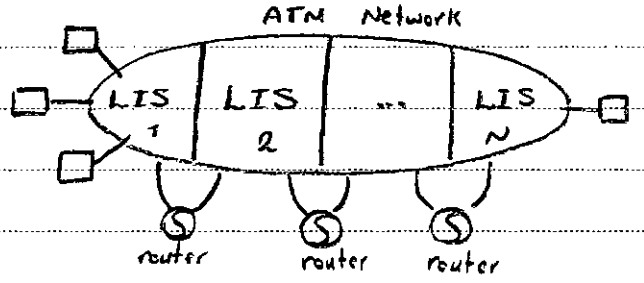
از انواع آن می‌توان ذکر کرد:

Multi Protocol Label Switching : MPLS ●

☑ شبکه ایران از روتر‌های لایه ۳ استفاده می‌کند (بین ۵ تا ۸ تا که کاملاً همبند هستند)

CLIP *

در شبکه‌های متصلی IP از طریق subnet شخص می‌شود که این کارت شبکه به همراه چه range دیگری از کامپیوتر‌ها روی یک NIC قرار دارد و در جدول مسیریابی آنها قرار می‌دهند این روش کلاسیک IP است.



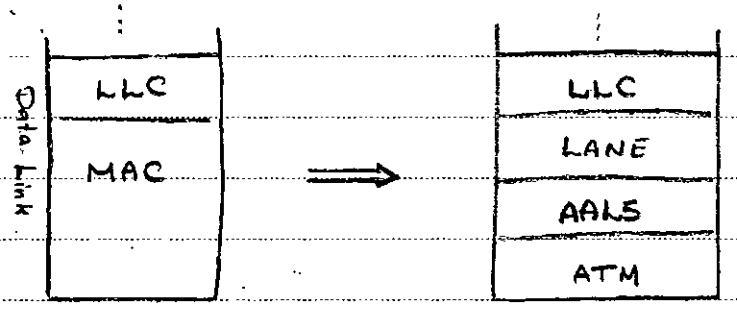
LIS : logical IP Subnet

در CLIP مانند IP کلاسیک (☑ Ethernet) شبکه ATM را بخش‌بندی منطقی می‌کنیم.

ایراد این روش این است که با وجود اینکه تمام host‌ها روی یک شبکه ATM قرار دارند ولی به منظور خروج از هر LIS باید از router عبور تا به LIS بعدی وارد شوند.

LANE *

برتری حسن آن این است که بدون دخالت در لایه Network شبکه سرعت بالا ایجاد می‌کند و مخلق IP نیست.

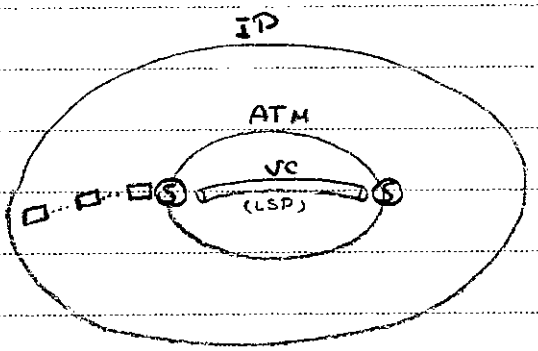


P4PCO ID

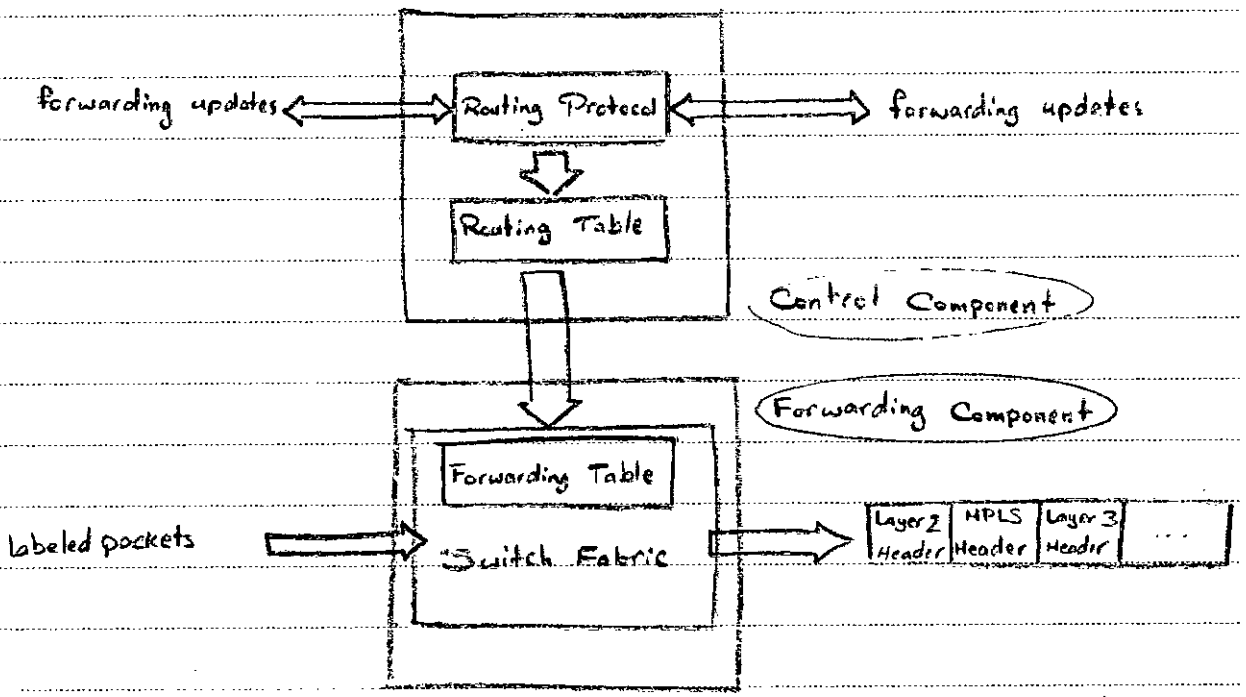
Subject:

Year. Month. Date. ( )

MPLS *

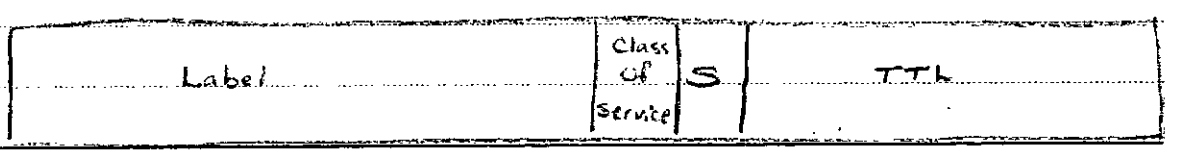


در این روش در قسمت ATM بین دو ترمینال VC ایجاد می‌شود که به آن Label Switch Path یا LSP می‌گویند. اگر VC ها از نوع دائم (Permanent) ایجاد شده باشند می‌توان آن‌ها را از نوع دائم (Permanent) ایجاد کرد. زمان تأخیر فقط مربوط به ایجاد VC ها هستند. نودهای میانی این لایه دارای معماری زیر هستند:



پروتکل بین لایه دوم و سوم است. چون لایه داخلی ATM است Lable ها عبارتند از VCI و VPI.

فرمت MPLS Header به شکل 32 بیتی است:



PqPCO

S: Heirarchical Stack Bit ← مشخص می‌کند که در لایه‌بندی، لایهٔ آخری است یا نه

Quality Of Service *

• سرویس IP یک سرویس best effort است.
برای انتقال برنامه‌های multi media و real time مناسب نیست.
برای تضمین QoS باید از قبل منابع را در نظر بگیریم تا کیفیت سرویس بالا باشد.

• IETF در اوایل دههٔ ۱۹۹۰ گروهی را مأمور تضمین QoS کرد.
این گروه یک معماری به نام Integrated Services ارائه داد که اختصاراً "IntServ" می‌خوانند.
این پروتکل سیگنالینگی تحت نام RVSP دارد.
این پروتکل منابع را از قبل رزرو می‌کند.
این پروتکل قابل پیاده‌سازی نبود چون:
RSVP، منابع در روترهای میانی رزرو می‌شوند، پس صف‌هایی با اولویت تشکیل شده، کیفیت سرویس تضمین می‌شود. برای این منظور تمام روترها باید RSVP را پشتیبانی کنند و باید عوض شوند. در ضمن در لایهٔ Core باید وضعیت میلیون‌ها ارتباط را نگه داشت که عملاً ممکن نیست.
RSVP می‌گوید که باید وضعیت هر جریان را نگه داریم (Per Flow).

• این گروه معماری دیگری به نام Differential Services ارائه داد که اختصاراً "DiffServ" می‌نامند.
در آن سرویس‌ها را کلاس‌بندی کرده و QoS مورد نظر آنها را تأمین می‌کنیم (Per Class).
روترهای میانی با توجه به قابل شناختن سرویس‌ها به آن کلاس سرویس می‌دهند و QoS آنها را تضمین می‌کنند.
هرچه تعداد بیشتری روتر از DiffServ حمایت کنند، QoS بالاتر می‌رود.
بیشتر در روترهای مرزی که سرعت کمتری دارند مؤثر است.

Subject:

Year. Month. Date. ( )