شبکه‌های مخابراتی

سید حمید صفوی
دانشکده فنی و مهندسی
دانشگاه محقق اردبیلی
نیمسال دوم ۹۸-۹۹

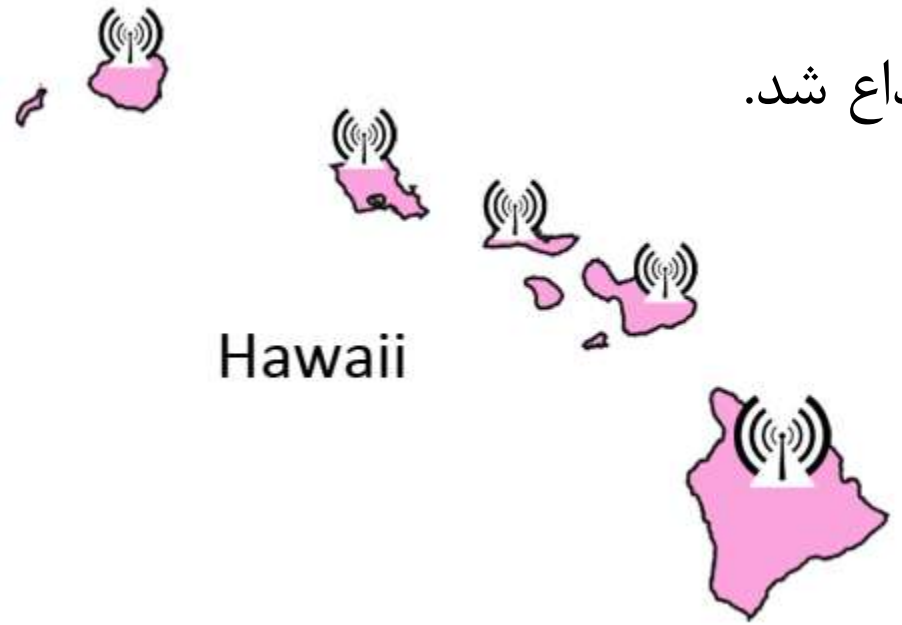
دسترسی چندگانه تصادفی

مفروضات کلی در دسترسی چندگانه تصادفی

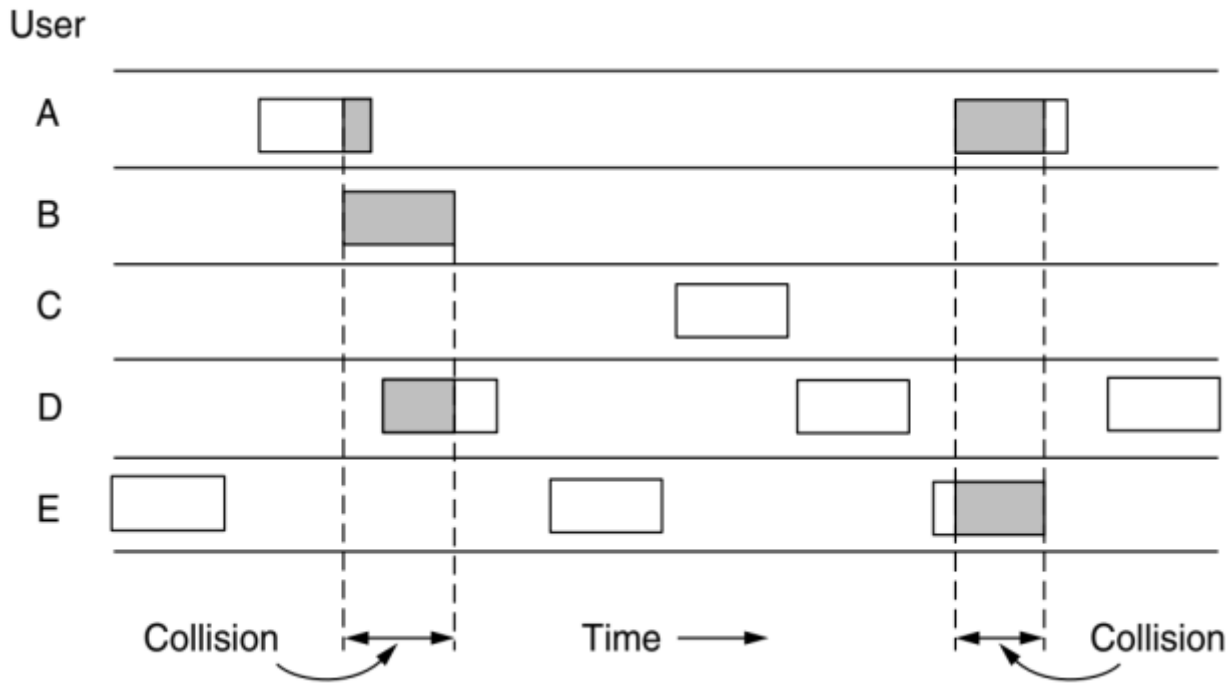
- N ایستگاه یا کاربر مستقل در بازه Δt با احتمال $\lambda \Delta t$ که λ نرخ فریم است، فریم تولید می‌کنند.
- یک کانال فیزیکی موجود است.
- تصادم (Collision) زمانی رخ می‌دهد که ۲ فریم و یا بیشتر، همزمان ارسال شوند.
- دو مدل برای محور زمان در نظر گرفته می‌شود:
  - پیوسته است.
  - گسسته است. (Slotted)
- ایستگاه‌ها و یا کاربرها ممکن است سخت‌افزار لازم برای تشخیص موارد زیر را داشته باشند:
  - تصادم فریم‌ها
  - حضور سیگنال کاربران دیگر بر روی کانال (Carrier Sense)

شبکه ALOHA

- شبکه کامپیوتری اولیه در اواخر دهه ۱۹۶۰ سلسله جزایر هاوایی را در اقیانوس آرام به هم متصل کردند.
  - چه زمانی گره‌ها باید ارسال کنند؟
  - یک پروتکل جدید توسط Norm Abramson ابداع شد.



پروتکل ALOHA



- ایده ساده:
  - گره‌ها فقط زمانی که ترافیک داده دارند، ارسال می‌کنند.
  - اگر تصادم رخ دهد (تایید دریافت نشود)، **زمان تصادفی نامشخصی** را برای ارسال مجدد صبر می‌کنند.
- برخی از فریم‌ها از بین می‌روند، اما بسیاری به مقصد می‌رسند...
- آیا ایده خوبی است؟

تحلیل عملکرد پروتکل ALOHA

- مفروضات:
  - تعداد زیاد کاربران و یا همان ایستگاه‌ها
  - طول مساوی فریم‌ها
  - زمان ارسال هر فریم نرمالیزه به یک
  - کاربران می‌توانند تصادم را تشخیص داده و در صورت تصادم مجدداً ارسال کنند.
- توزیع پوآسون:
  - از روی توزیع دوجمله‌ای حاصل می‌شود.
  - اگر رخدادی با احتمال P به وقوع بپیوندد، احتمال اینکه آن رخداد X بار در N بار آزمایش به وقوع بپیوندد برابر است با:

$$P(X)=\binom{X}{N}P^{X}(1-P)^{N-X}$$

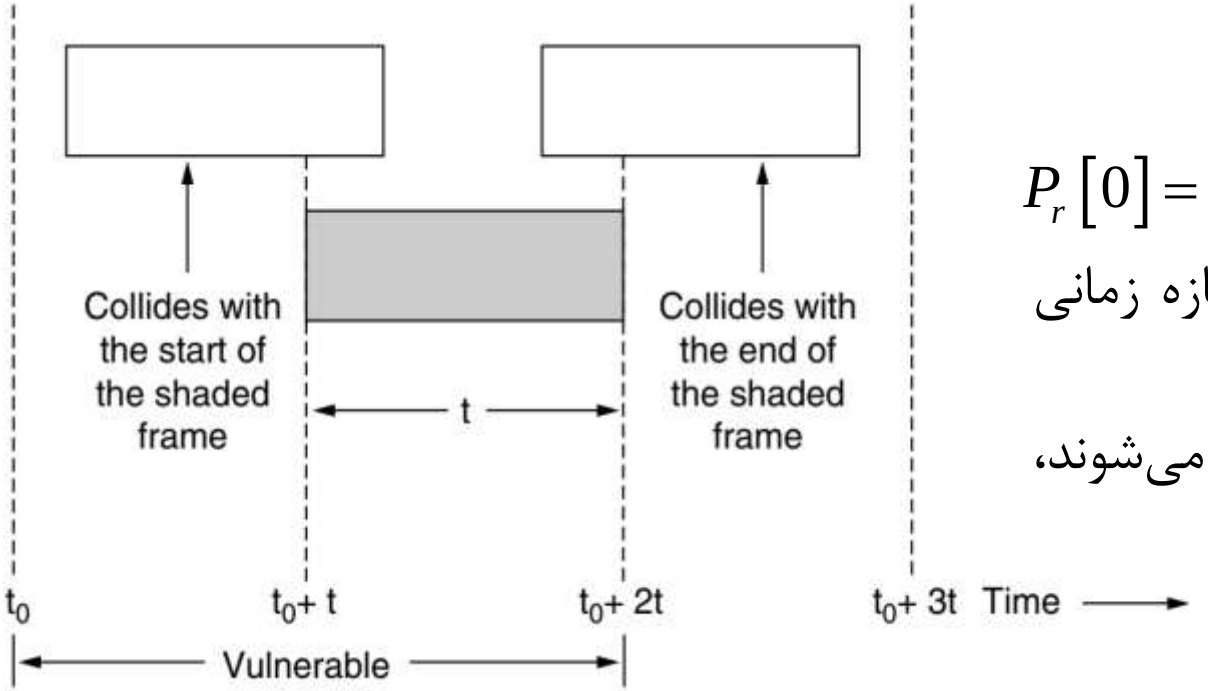
  - اگر N بزرگ باشد و P نزدیک 0 و 1 نباشد، توزیع نرمال حاصل خواهد شد.
  - اگر N بزرگ باشد و P مقادیر خیلی کوچک نزدیک به 0 داشته باشد، توزیع پواسون خواهیم داشت.
  - توزیع پوآسون با متوسط G: احتمال k رخداد در بازه زمانی داده شده:

$$P_r(k)=\frac{G^k e^{-G}}{k!}$$

تحلیل عملکرد پروتکل ALOHA

- همه ایستگاه‌ها با همدیگر به طور متوسط S فریم جدید در هر بازه زمانی فریم (Frame Time) تولید می‌کنند.
- واضح است که اگر $S>1$ باشد، تصادم رخ خواهد داد. بنابراین برای عملکرد مطلوب $0<S<1$ باشد.
- هر ایستگاه تا زمانی که فریم قبلی موفقیت آمیز ارسال نشده‌باشد، فریم جدید را ارسال نمی‌کند.
- احتمال K تلاش در هر بازه زمانی فریم توسط همه ایستگاه‌ها، برای فریم‌های جدید و قدیم یک فرآیند پوآسون با میانگین G در هر بازه زمانی فریم است.
  - بار کم ← به ندرت تصادم رخ می‌دهد ← $S=G$
  - بار زیاد ← تصادم بیشتری رخ خواهد داد ← $G>S$
- **بنابراین بازدهی برابر است با G ضربدر احتمال اینکه تصادم رخ ندهد. $S=G\times P0$**

تحلیل عملکرد پروتکل ALOHA (۲)



- احتمال تولید k فریم در بازه زمانی فریم توزیع پواسون دارد.

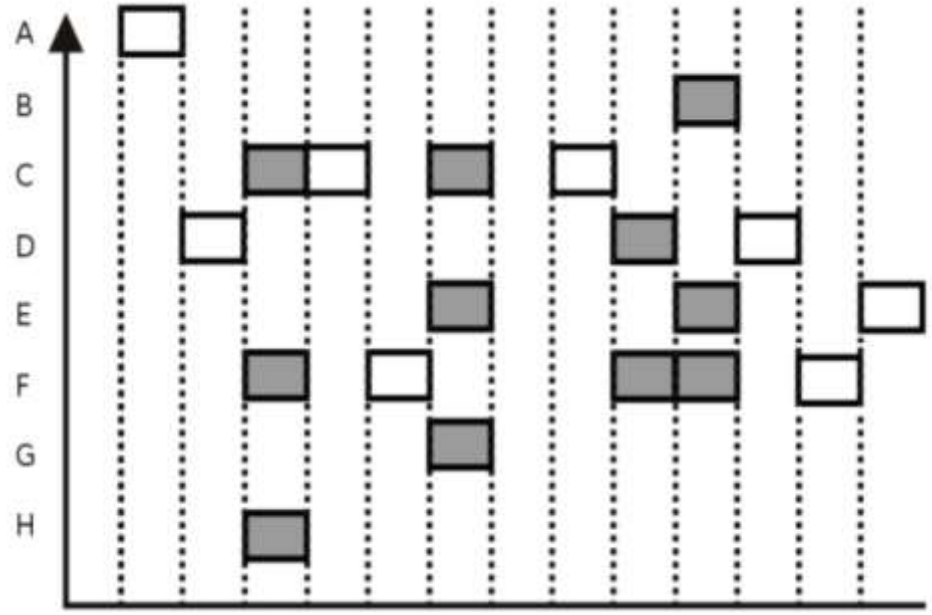
$$P_r[k] = \frac{G^k e^{-G}}{k!}$$

- احتمال اینکه هیچ فریمی تولید نشود برابر است با: $P_r[0] = e^{-G}$
- تصادم، زمانی رخ نمی‌دهد که هیچ فریم دیگری در بازه زمانی دو فریم ایجاد نشود.
- متوسط تعداد فریم‌هایی که در دو بازه زمانی ایجاد می‌شوند، برابر 2G است. بنابراین

$$P_r[k] = \frac{(2G)^k e^{-2G}}{k!} \Rightarrow P_r[0] = e^{-2G}$$

- کارآمدی ALOHA: $\boxed{S = Ge^{-2G}}$

پروتکل Slotted ALOHA

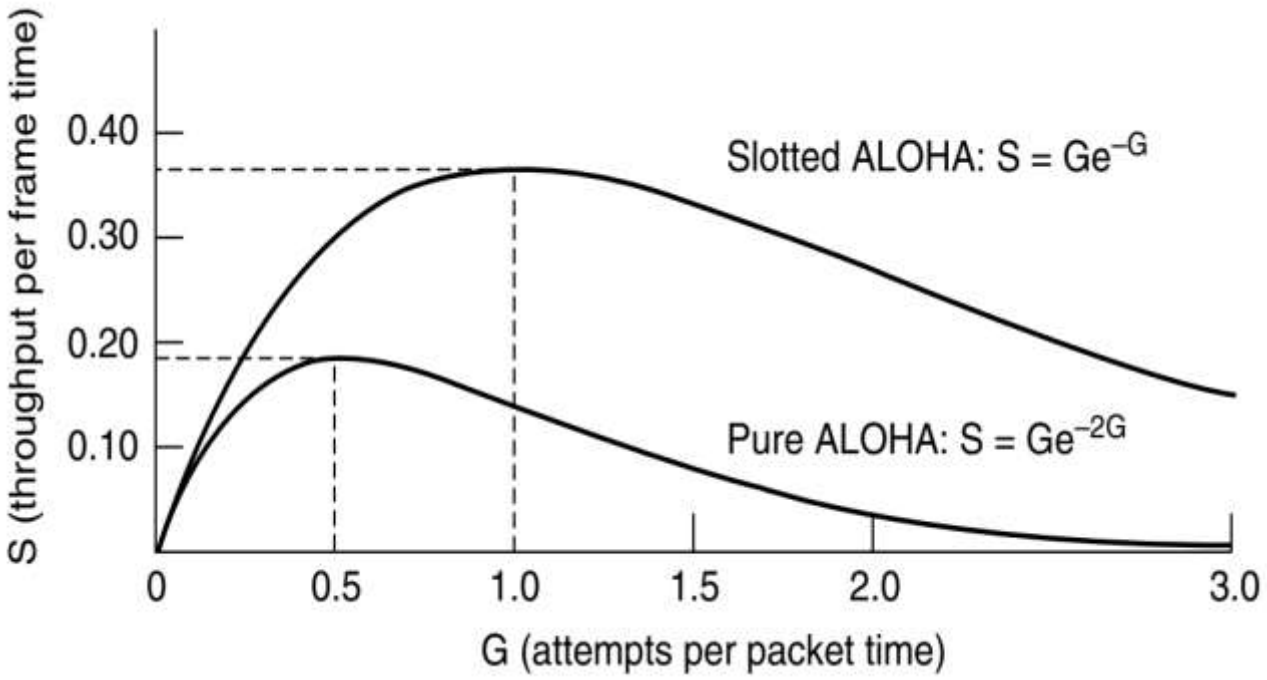


- محور زمان را به بخش‌های گسسته تقسیم می‌کنیم.
- کاربران باید بر روی مرزهای اسلات زمانی توافق کنند.
- در هر اسلات زمانی، یک یا چند کاربر فریم‌های خود را ارسال می‌کنند.
- احتمال ارسال موفق (هیچ فریم دیگری در بازه زمانی داده شده تولید نشود) برابر است با:

$$P_r[k] = \frac{G^k e^{-G}}{k!} \Rightarrow P_r[0] = e^{-G}$$

- کارآمدی روش Slotted ALOHA: $S = Ge^{-G}$
- بیشینه بهره‌وری کانال:
  - روش Pure ALOHA: ۱۸.۴ درصد
  - روش Slotted ALOHA : ۳۶.۸ درصد

جمع‌بندی پروتکل‌های ALOHA



- پروتکل ساده و غیرمتمرکز که در **بار کم** خوب کار می‌کند.
- اما در بار زیاد بازده خوبی ندارد.
  - بررسی‌ها نشان می دهد که نهایتاً ۱۸٪ بازدهی دارد.
  - بهبود: با تقسیم زمان به چند بازه می‌توان راندمان را به ۳۶٪ رساند.

اثر بار بر روی پروتکل ALOHA

- احتمال شکست در ارسال یک فریم: $P = 1 - e^{-G}$
- احتمال اینکه یک ارسال نیاز به k بار تلاش داشته باشد: $P_k = e^{-G}\left(1 - e^{-G}\right)^{k-1}$
- متوسط تعداد ارسال برای هر فریم: $E = \sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} ke^{-G}\left(1 - e^{-G}\right)^{k-1} = e^{G}$

- **نتیجه اخلاقی:** افزایش اندک بار شبکه (G)، می‌تواند به صورت نمایی تعداد ارسال‌ها را زیاد کند. این افزایش منجر به موارد زیر می‌شود:
  - کاهش بازدهی سیستم
  - صف‌های طولانی در لایه MAC که منتظر ارسال می‌مانند.
  - افزایش مقدار متوسط تأخیر هر فریم

تحلیل تأخیر در پروتکل ALOHA

- مدت زمان ارسال یک فریم: X
- مدت زمان انتشار یک فریم: t_{prop}
- متوسط زمان Back-off یا همان زمانی که باید به صورت رندم صبر کرد: B
- متوسط تعداد تلاش برای ارسال یک فریم: $G/S = e^{2G}$
- متوسط تعداد تلاش ناموفق برای ارسال یک فریم: $G/S - 1 = e^{2G} - 1$
- ارسال موفق نیاز به مدت زمان $X + t_{prop}$ دارد و هر ارسال مجدد نیاز به مدت زمان $2t_{prop} + X + B$ دارد.
- بنابراین متوسط تأخیر لایه MAC در روش ALOHA برابر است با:

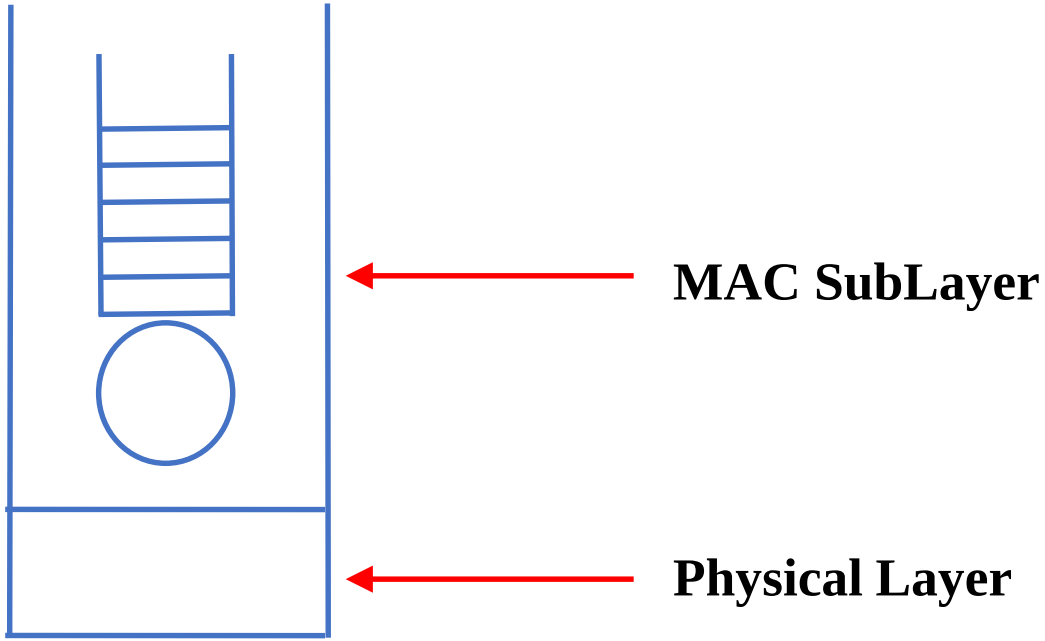
$$E[T_{ALOHA}] = X + t_{prop} + \left(e^{2G} - 1\right)\left(2t_{prop} + X + B\right)$$

- در حالت عادی، زمان backoff به صورت یکنواخت بین 1 و k زمان ارسال packet است. بنابراین

$$B = (k+1)\,X/2$$

مصالحه‌های موجود در پروتکل ALOHA

- نرخ ارسال برحسب bps، محدودیت لایه فیزیکی است برای جابجاشدن بیت‌ها
- فریم‌ها به جز اینکه مدتی به خاطر محدودیت‌های لایه فیزیکی منتظر می‌مانند، مدتی هم به خاطر روش‌های مختلف لایه MAC داخل صف منتظر می‌مانند.



مصالحه‌های موجود در پروتکل ALOHA (۲)

- **Normalized Transfer Delay (NTD):**
  - معیاری برای تأخیر به وجودآمده از روش‌های کنترل دسترسی لایه MAC (تصادم‌ها و ...)

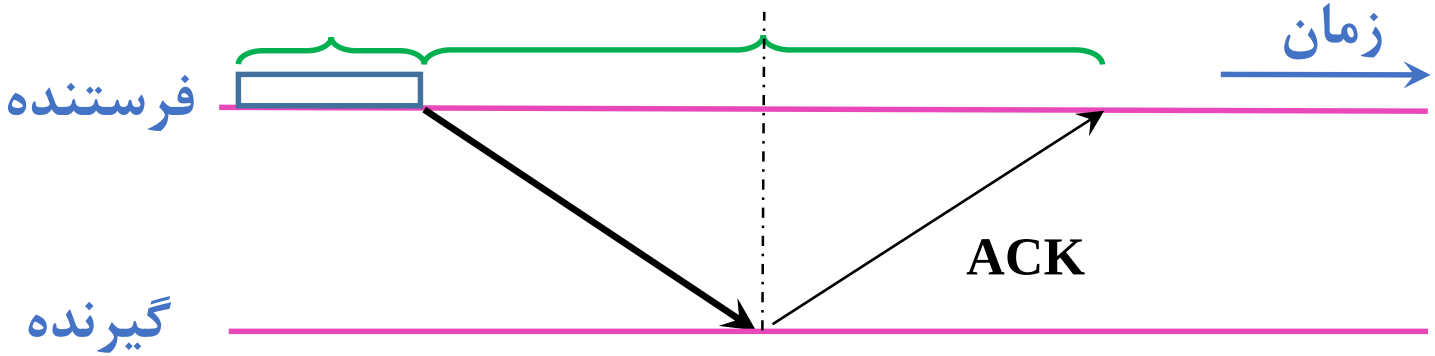
$$NTD = \frac{E[T] \rightarrow \text{Average Transfer Delay}}{E[X] \rightarrow \text{Average Frame Transmission Time}}$$

- برای نرمالیزه کردن، متوسط تأخیر Transfer delay را بر مدت زمانی که بدون روش‌های MAC نیاز به ارسال بود، تقسیم می‌کنیم.
- در اسلایدهای قبلی، متوسط تأخیر Transfer delay را برای ALOHA محاسبه کردیم.

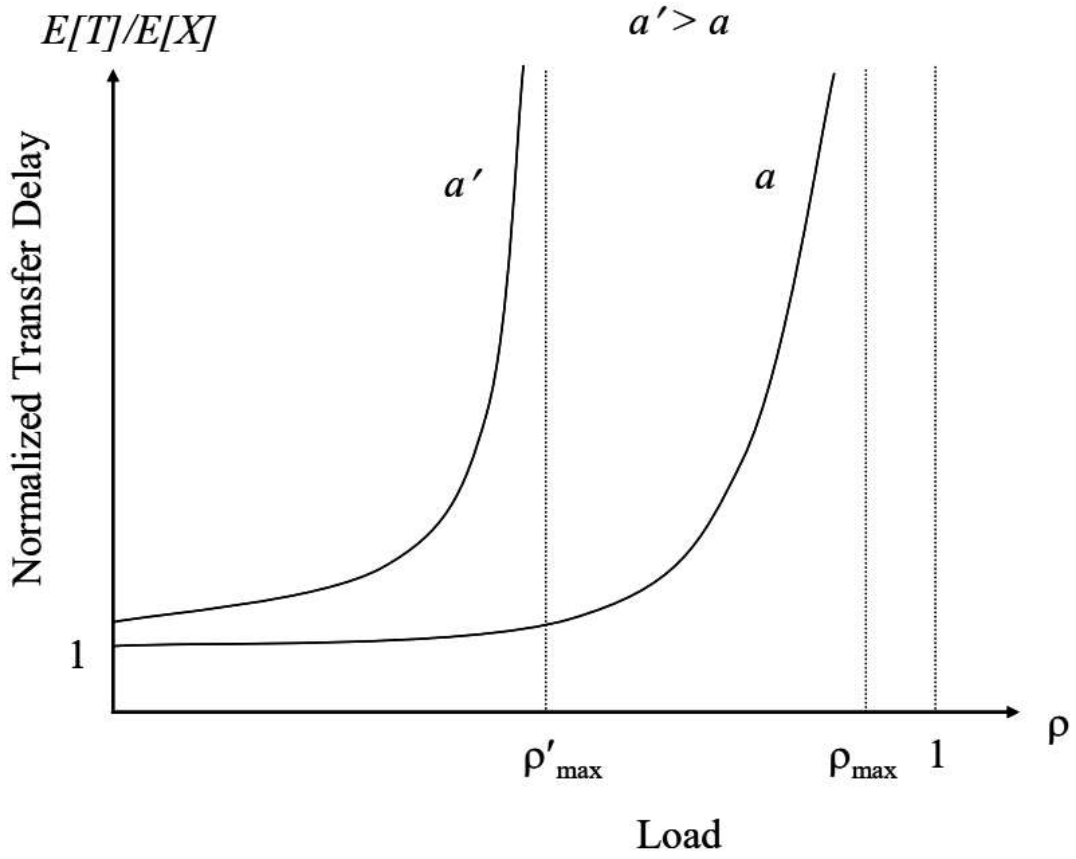
مصالحه‌های موجود در پروتکل ALOHA (۳)

- **Normalized Delay Bandwidth:**
  - تعداد فریم‌های از دست رفته به خاطر تصادم متناسب با a است.

$$a = \frac{\text{One Way Delay} \times \text{Bandwidth}}{\text{Average Frame } \textit{Size}}$$

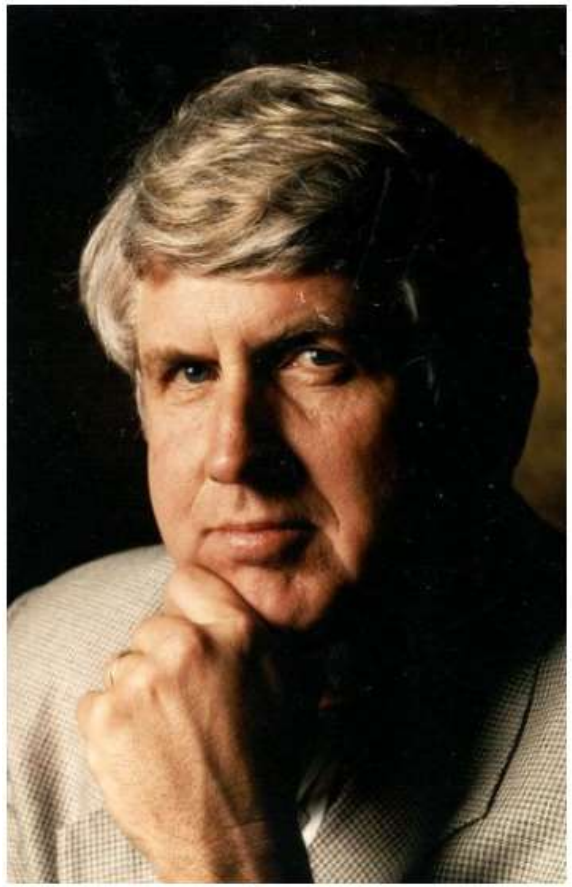


مصالحه‌های موجود در پروتکل ALOHA (۴)



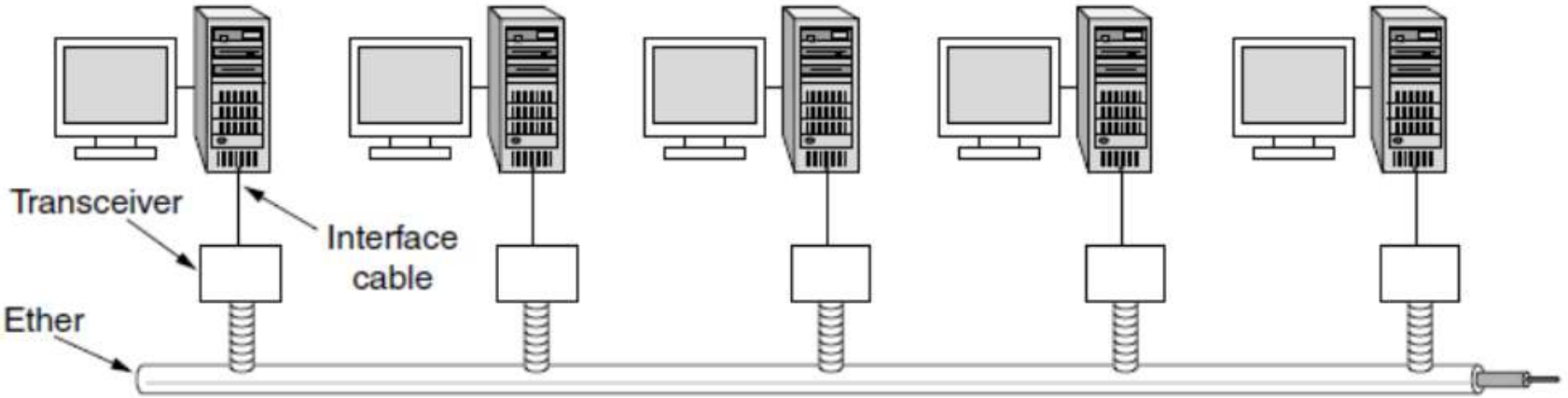
- برای روش‌های مشابه ALOHA داریم:
  - هنگامی که بار کم است، NTD نزدیک 1 است. با افزایش بار، NTD نیز به طور پیوسته و به صورت نمایی افزایش می‌یابد.
  - بیشینه بار قابل دسترس خیلی کمتر از 1 است.
  - بیشینه بار قابل دسترس با افزایش مقدار a، کاهش می‌یابد.
  - معنی عبارت فوق این است که زمانی که تعداد کاربران زیادی (از یک تعدادی بیشتر) برای به دست آوردن کانال می‌جنگند، تعداد تصادم زیادی رخ می‌دهد و هیچ کدام نمی‌توانند از کانال استفاده کنند.
  - با افزایش مقدارa، تحت بار کمتری سیستم موقعیت بحرانی خواهد داشت.

اترنت کلاسیک



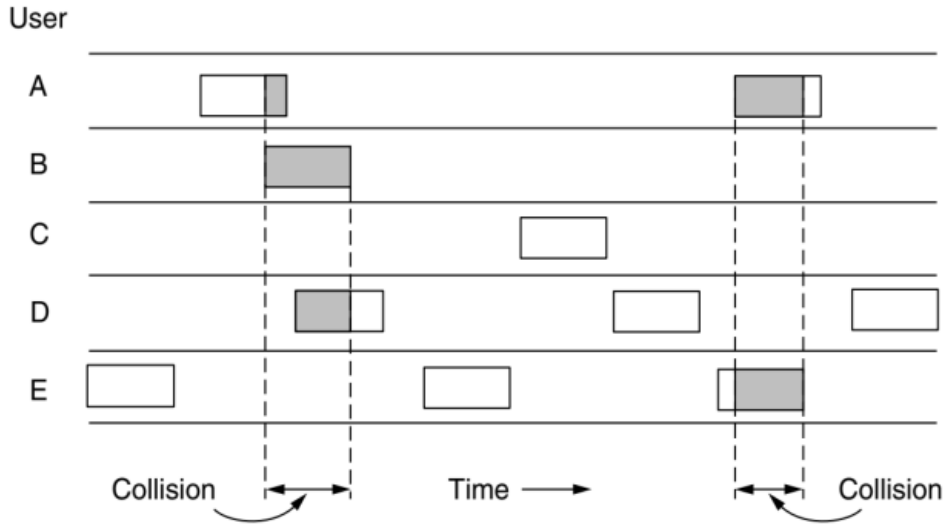
: © 2009 IEEE

- در سال ۱۹۷۳، Bob Metcalfe با الهام گرفتن از **ALOHA** اترنت را برای LAN ها اختراع کرد.
  - گره‌ها، کابل‌های کواکسیال با سرعت ۱۰ مگابیت بر ثانیه (Mbps) را اشتراک‌گذاری می‌کنند.
  - اترنت کلاسیک در دهه‌های ۱۹۸۰ و ۱۹۹۰ بسیار محبوب بود.



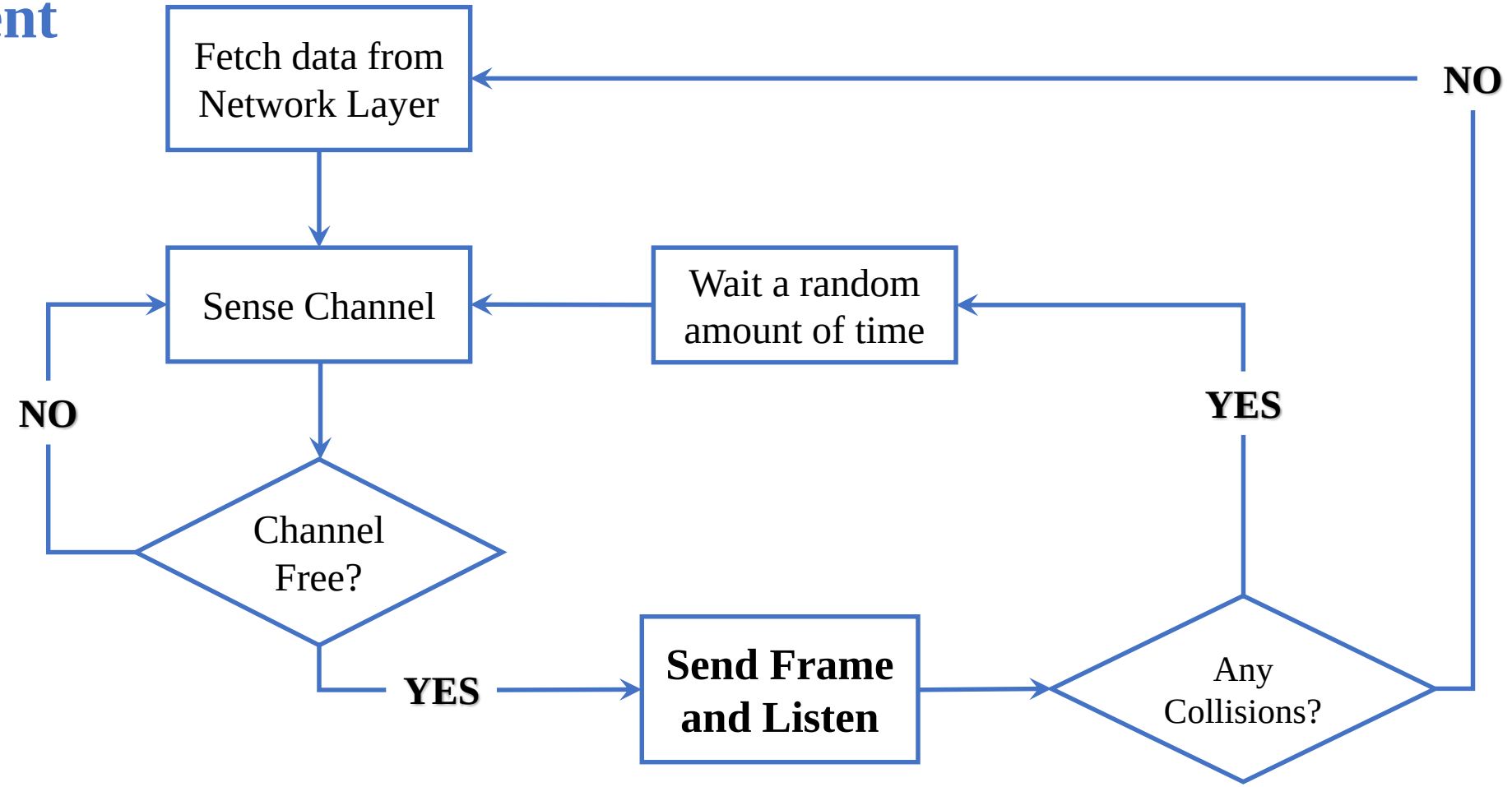
CSMA(Carrier Sense Multiple Access)

- **ایده اصلی: تعداد تصادم می‌تواند با گوش دادن به کانال قبل از ارسال کاهش یابد.**
- بنابراین، اگر لایه فیزیکی بتواند این سرویس را به لایه مک بدهد که زمانی که کسی در حال ارسال است را تشخیص دهد، می‌تواند بازدهی را بهبود دهد.
  - این کار به راحتی با سیم انجام می‌شود اما در بی سیم نه!
- آیا این کار تصادم را برطرف می‌کند؟
  - چرا بله یا چرا نه؟



CSMA(Carrier Sense Multiple Access)

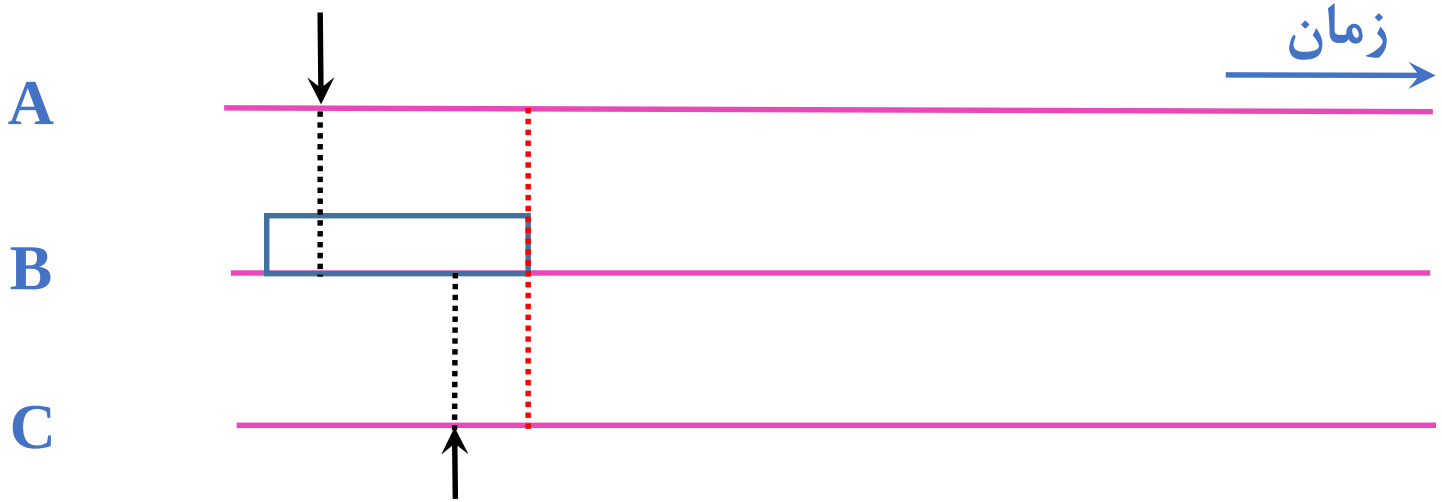
- **1- Persistent**



CSMA(Carrier Sense Multiple Access)

- **روش 1-Persistent:**
  - به کانال گوش کن و به محض اینکه کانال خالی شد، ارسال کن.
  - اگر تصادم رخ داد، مقدار زمان تصادفی منتظر باش و سپس ارسال کن.
  - این روش خیلی حریص است. نامگذاری 1-Persistent به خاطر آن است که هنگام خالی بودن کانال با احتمال یک ارسال می‌کند.

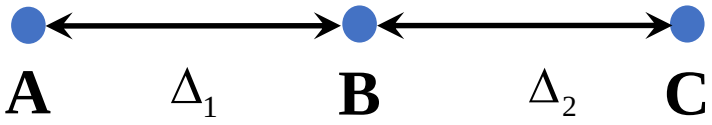
CSMA(Carrier Sense Multiple Access)

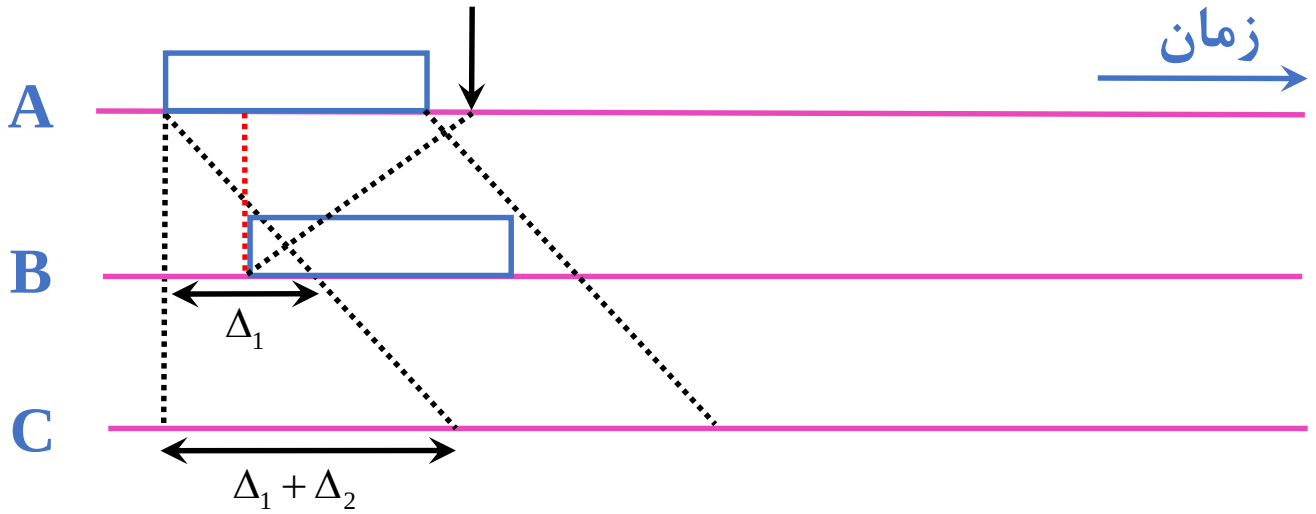


- کاربر A در لحظه مشخص شده به کانال گوش می‌دهد و متوجه می‌شود کانال پر است. به محض خالی شدن کانال شروع به ارسال می‌کند.
- کاربر C نیز در لحظه مشخص شده به کانال گوش می‌دهد و متوجه می‌شود کانال پر است. به محض خالی شدن کانال شروع به ارسال می‌کند.
- لحظه‌ای که کاربرهای A و C شروع به استفاده می‌کنند، تصادم رخ می‌دهد.

CSMA(Carrier Sense Multiple Access)

- فرض کنید مدت زمان مشخصی طول می‌کشد تا خالی و یا پر بودن کانال توسط کاربران sense شود.

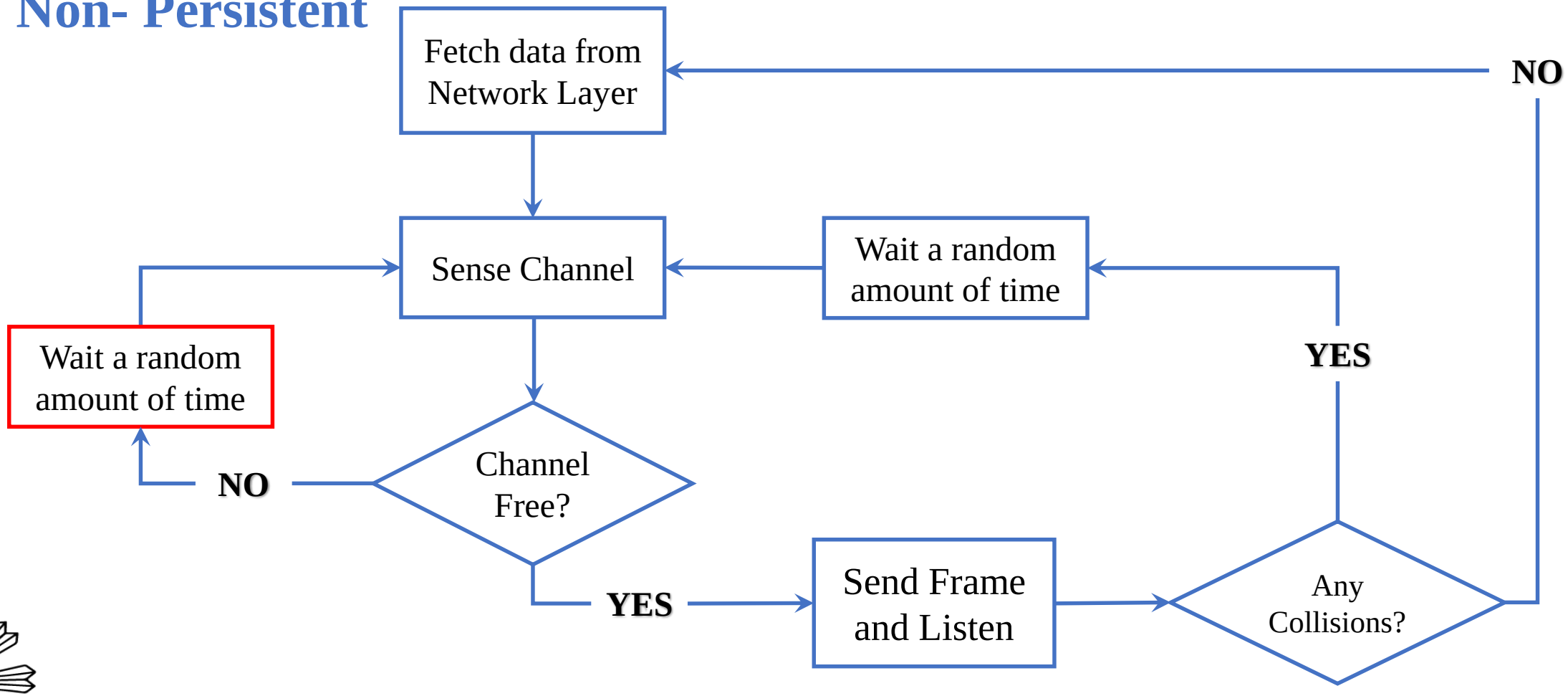




- **نکته:** از تأخیر انتشار طولانی باید اجتناب شود. این امکان وجود دارد که حتی پس از اینکه یکی از ایستگاه‌ها شروع به ارسال کرد، ایستگاه دیگری به دلیل تأخیر انتشار، کانال را خالی sense کند و شروع به ارسال کند.

CSMA(Carrier Sense Multiple Access)

- **Non- Persistent**

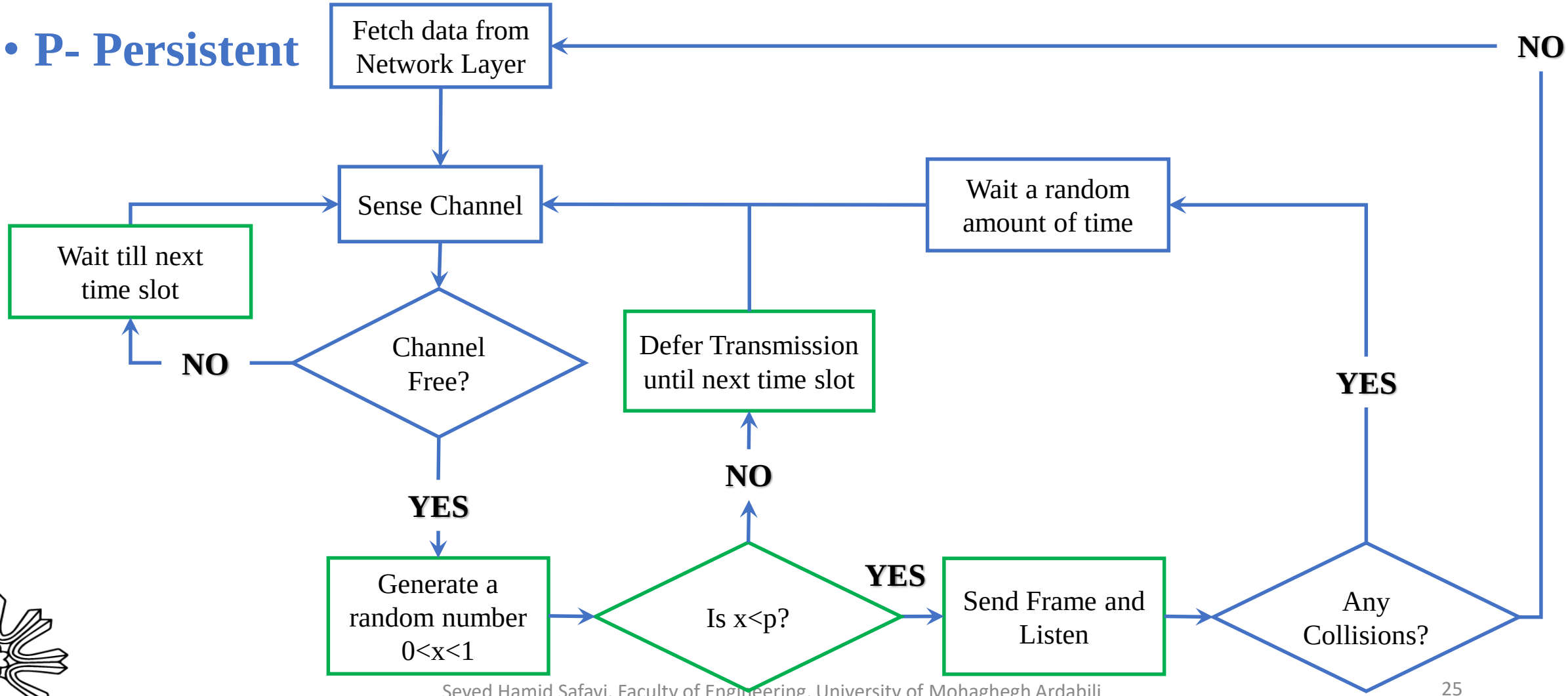


CSMA(Carrier Sense Multiple Access)

- **روش Non-Persistent:**
  - به کانال گوش کن؛ اگر کانال خالی بود، ارسال کن و اگر کانال خالی نبود، قبل از گوش دادن به کانال، زمان تصادفی صبر کن.
  - استفاده بهتر اما تأخیر طولانی‌تر نسبت به 1-Persistent

CSMA(Carrier Sense Multiple Access)

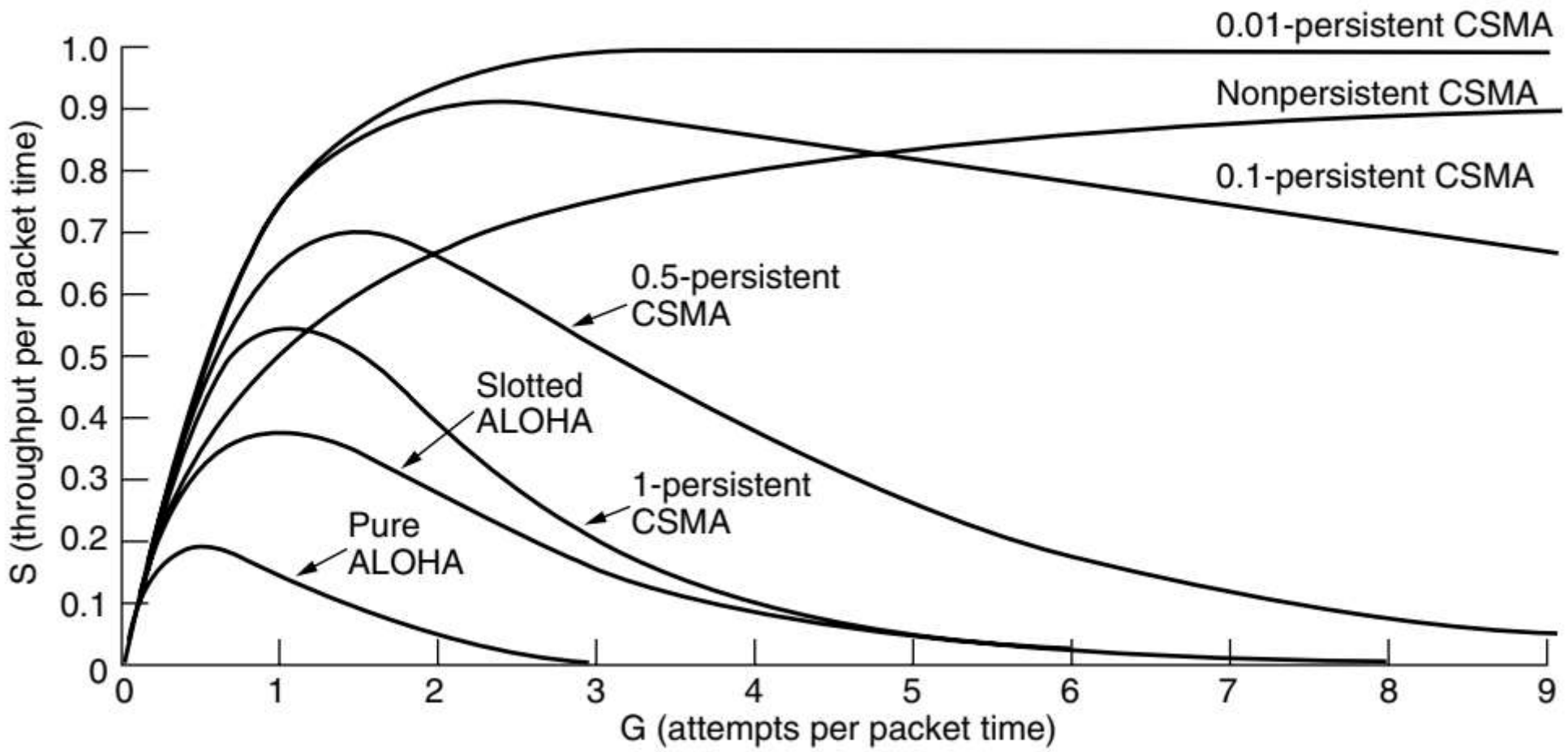
- **P- Persistent**



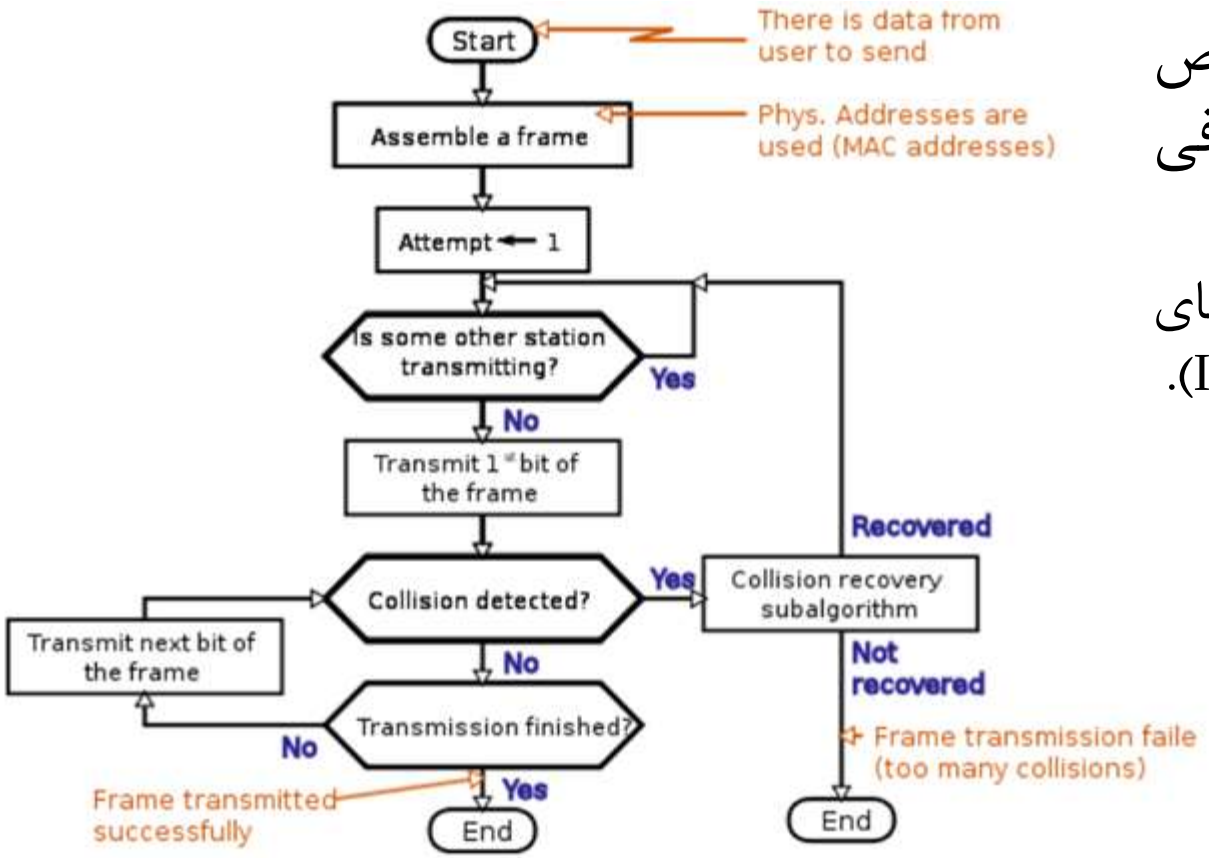
CSMA(Carrier Sense Multiple Access)

- **روش P-Persistent:**
  - به کانال گوش کن؛ اگر کانال خالی بود، با احتمال P ارسال کن و با احتمال 1-P به اسلات بعدی موکول کن.
  - اگر اسلات زمانی بعدی هم خالی بود، الگوریتم فوق را تکرار کن.
  - اگر کانال اشغال بود، تا اسلات زمانی بعدی صبر کن.
  - استفاده بهتر اما تأخیر طولانی‌تر نسبت به 1-Persistent

CSMA Throughput



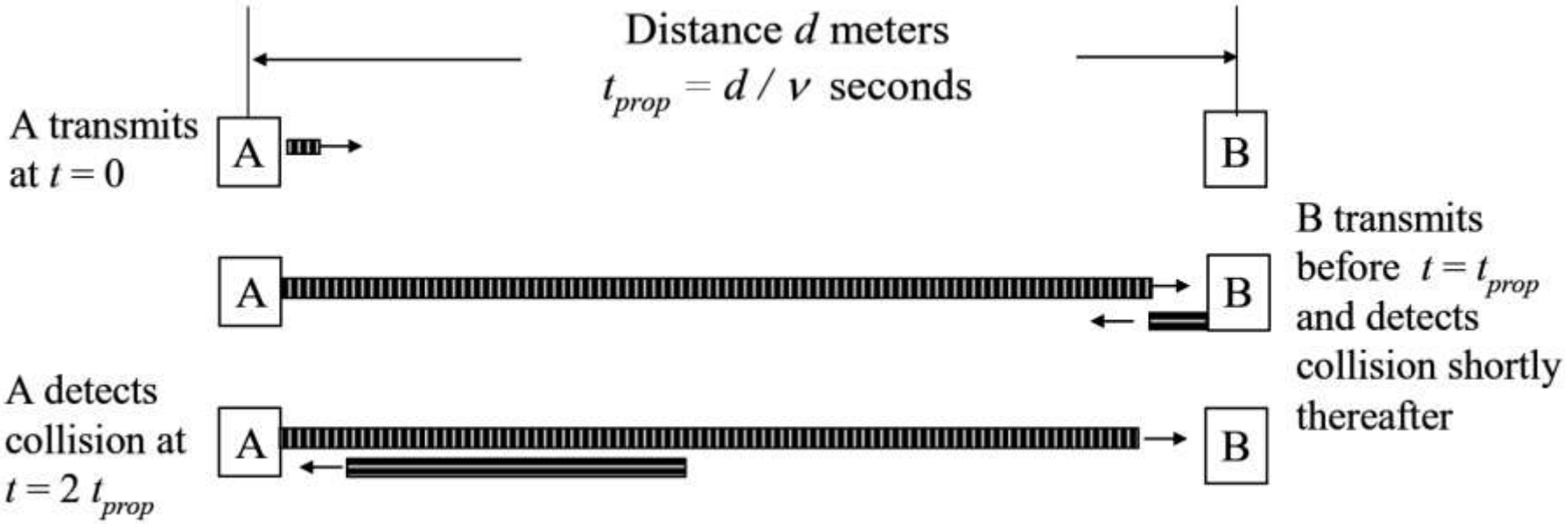
CSMA with Collision Detection



- **ایده اصلی:** دقیقاً پس از تشخیص تصادم ارسال را قطع کن و زمان تصادفی منتظر باش
  - به طور گسترده در لایه MAC شبکه های LAN کاربرد دارد (اترنت و IEEE 802.3).

CSMA with Collision Detection

- چقدر طول می‌کشد تا تصادم تشخیص داده شود؟
  - دو برابر مدت زمان تأخیر انتشار طول می‌کشد تا مطمئن شویم تصادم رخ داده‌است (زمان رقابت Contention Period). (۱۰ میکروثانیه برای یک کیلومتر کابل)

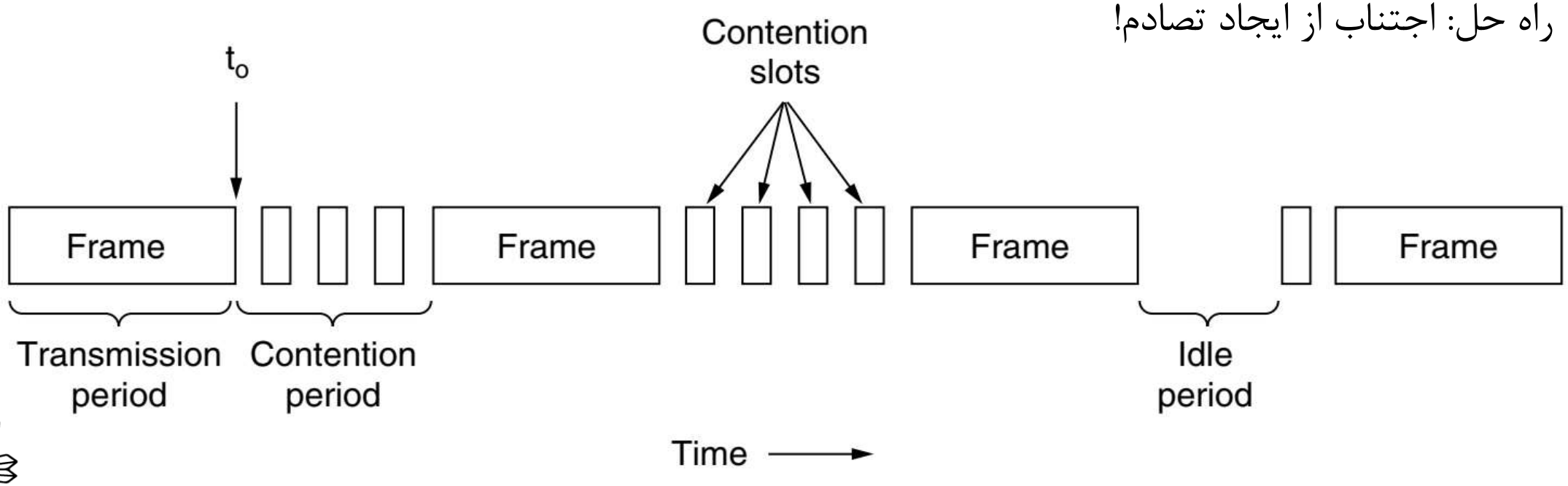


CSMA with Collision Detection

- **سوال:** در سناریوی توضیح داده شده، اصلاً نود A متوجه تصادم می‌شود؟
- برای اینکه متوجه تصادم شود، یک راه حل این است که در لایه لینک دنبال ACK باشد. اما نمی‌خواهیم overhead اضافه کنیم. می‌خواهیم با همین اختیارات لایه مک و لایه فیزیکی مسئله را حل کنیم. راهش انتخاب طول مناسب هر فریم است. اگر طول هر فریم بزرگتر از دو برابر تأخیر انتشار باشد، حتماً نود A متوجه تصادم خواهد شد چون زمانی که ارسالش به پایان نرسیده، از نود B فریم دریافت می‌کند که نشان دهنده تصادم است.
- **نتیجه اخلاقی:** طول فریم را بیش‌تر از دو برابر Propagation Delay در نظر بگیریم تا بتوانیم تصادم را تشخیص دهیم.

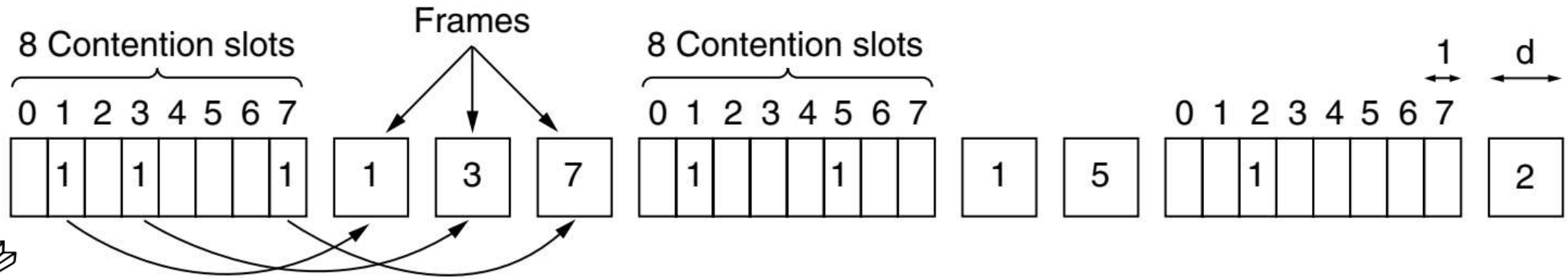
CSMA with Collision Detection

- گرچه روش CSMA/CD از ایجاد تصادم اجتناب می‌کند، اما با این حال در دوره رقابت ناگزیر تصادم رخ خواهد داد.
- Contention period و یا دوره رقابت: زمانی است که نودهایی که می‌خواهند داده بفرستند با همدیگر رقابت می‌کنند.
- دوره رقابت می‌تواند برای کابل‌های طولانی، بزرگ باشد.
- راه حل: اجتناب از ایجاد تصادم!



پروتکل‌های بدون تصادم

- **پروتکل Bit-Map: (رزرو پهنای باند)**
  - اعلام ارسال داده در بازه رقابت
  - کارآیی برابر $\frac{d}{d+N}$ (d برابر طول فریم است.) هنگام بار کم و برابر $\frac{d}{d+1}$ هنگام بار زیاد
  - امکان بهینه شدن با اجازه دادن برای ارسال چندین اسلات. کارآیی نزدیک 1



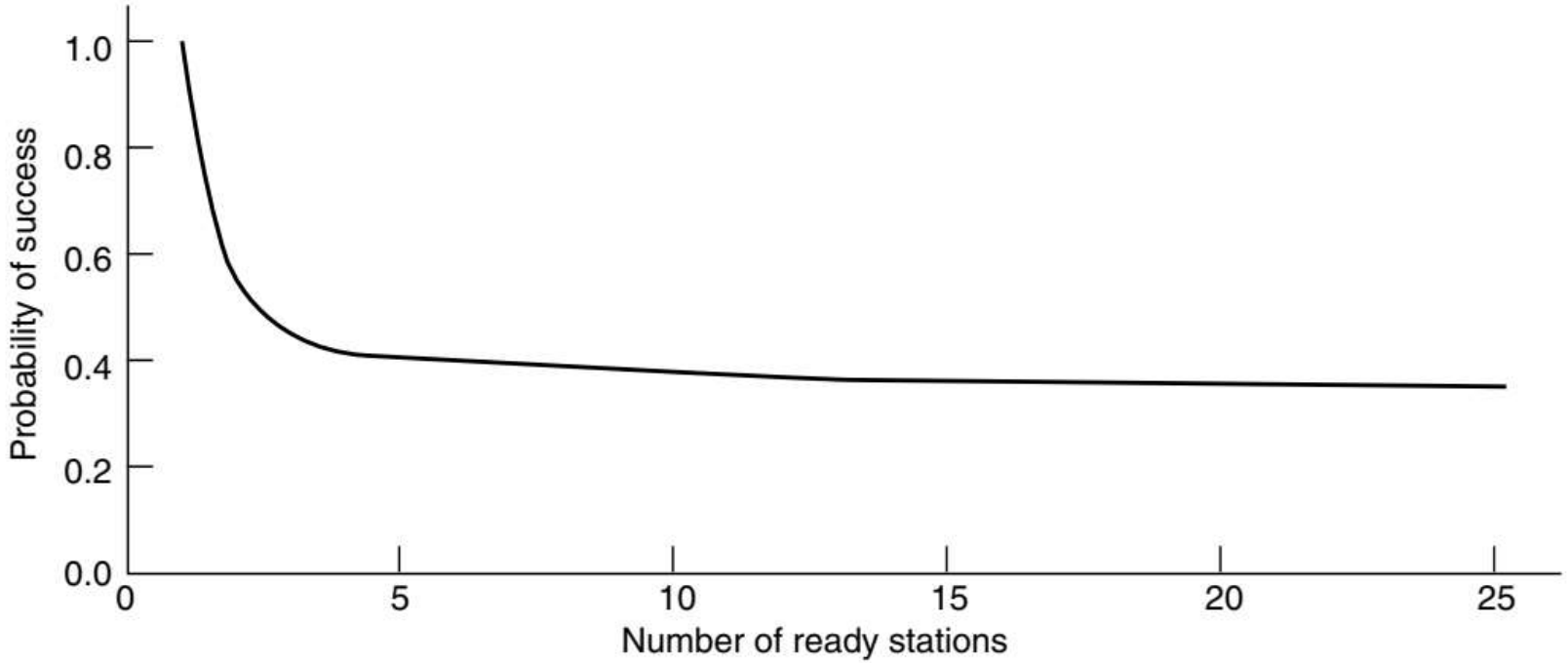
پروتکل‌های تصادم محدود
Limited-Contention Protocols

- تا اینجا:
  - پروتکل‌های مبتنی بر رقابت: تأخیر کم در بار کم
  - پروتکل‌های بدون تصادم: استفاده بهینه از کانال در بار زیاد
  - می‌خواهیم از مزیت‌های هر دو روش استفاده کنیم!
- تعداد K کاربر را در نظر بگیرید که هر کدام با احتمال p، یک فریم را در اسلات زمانی مشخصی ارسال می‌کنند.
- مقدار **بهینه** احتمال اینکه کاربری کانال را در اختیار بگیرد برابر است با

$$P_r[k] = kp(1-p)^{k-1} \Rightarrow p_{optimal} = \frac{1}{k} \Rightarrow P_{r,optimal} = \left(1-\frac{1}{k}\right)^{k-1}$$

- **نتیجه اخلاقی:** تعداد کاربرانی که رقابت می‌کنند (k) را کم کنیم!

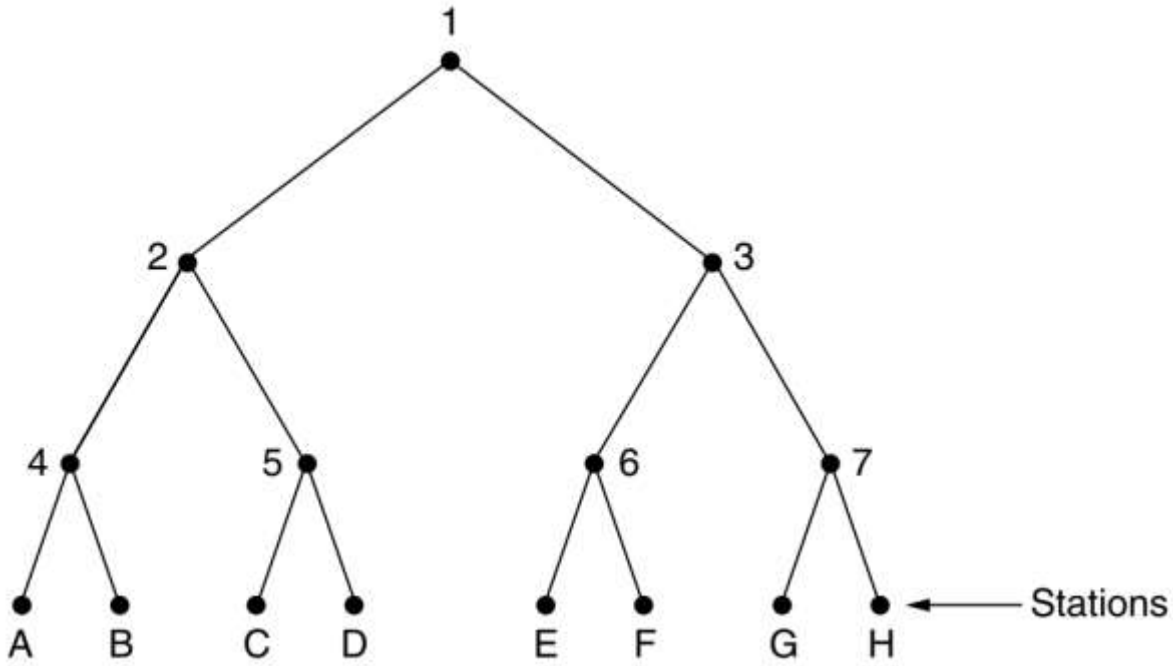
پروتکل‌های تصادم محدود
Limited-Contention Protocols



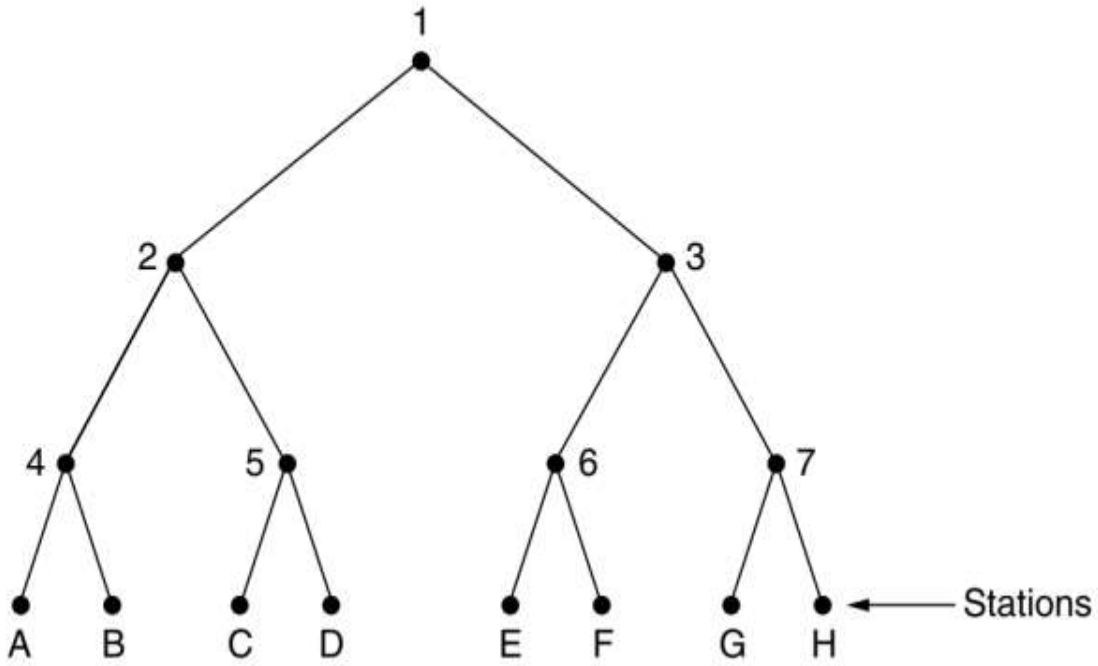
$$\Pr[\text{success with optimal } p] = \left[\frac{k-1}{k}\right]^{k-1}$$

پروتکل‌های تصادم محدود
The Adaptive Tree Walk Protocol

- **هدف: کاهش تعداد کاربرانی است که رقابت می‌کنند.**
- چگونه کاربران را به اسلات‌های زمانی تخصیص دهیم؟



پروتکل‌های تصادم محدود
The Adaptive Tree Walk Protocol



- **قوانین پروتکل:**
- در اسلات زمانی 0، همه کاربران زیرمجموعه گره 1 می‌توانند برای ارسال تلاش کنند.
- اگر تصادم رخ دهد، همه کاربران زیرمجموعه گره 2 می‌توانند در اسلات زمانی 1 ارسال کنند.
- سه حالت می‌تواند رخ دهد:
  - یکی از کاربران موفق شود. اسلات زمانی بعدی را به کاربران زیرمجموعه گره 3 می‌دهیم.
  - تصادم رخ دهد. ارسال را با کاربران زیرمجموعه‌های گره 4 در اسلات زمانی 2 ادامه می‌دهیم.
  - هیچ کسی ارسال نکند! ارسال را با زیرمجموعه‌های گره 6 در اسلات زمانی 2 ادامه می‌دهیم.

پروتکل‌های تصادم محدود
The Adaptive Tree Walk Protocol

- **مثال:** فرض کنید فقط کاربرهای G و H ترافیک برای ارسال داشته باشند.
- در اولین اسلات زمانی 0، همه کاربرانی که عضو درخت هستند اجازه ارسال دارند. چون G و H هر دو ارسال دارند، بنابراین تصادم داریم.
- اگر تصادم رخ داد، در اسلات زمانی بعد فقط نصف افراد می‌توانند برای ارسال رقابت کنند. چون تصادم رخ داده‌است، در اسلات زمانی بعدی، مثلا فقط زیرمجموعه‌های گره 2 رقابت می‌کنند.
- وقتی زیرمجموعه‌های گره 2 رقابت می‌کنند، کانال خالی می‌ماند. چون زیرمجموعه‌های گره 2 فریمی برای ارسال ندارند!
- اسلات زمانی بعد زیرمجموعه‌های گره 6 اجازه رقابت دارند و در این حالت نیز کانال خالی می‌ماند.
- در اسلات زمانی بعدی G ارسال می‌کند و اسلات زمانی بعد از آن H ارسال می‌کند.