

حاصل شده و علامت جلسه ۱۵

چگونه می توان Resource Allocation و congestion control را از برای سیستم

و کیفیت عملکرد آن را چگونه بررسی کنیم؟ $\&$ throughput و delay (از نظر

efficiently (کارآمدی) - و معیار دیگری Fairness است.

درستی می خواهیم Resource Allocation انجام دهیم معیار را تابع هدف برای

شبه شکل می دهیم. که این تابع هدف می تواند یک تابع جمع شونده باشد $\sum U_i(n_i)$

باشد که آن ها می توانند flow ها و یا user ها باشند.

و ما می خواهیم بدانیم Resource Allocation تحت این object اصلی شود

حقیر بنظر می آید. جواب: مهم این است که U_i (Utility) چگونه

تعیین شود. که این U_i می تواند به شکل های مختلف تعیین شود که گویا متفاوت

باید Fairness را ایجاد کند:

① Proportional Fairness: $U_i(n_i) = \log(n_i)$

(۲) weighted proportional fairness : $U_i(x_i) = w_i \log(x_i)$

(۳) α - Fairness : $U_i(x_i) = \frac{x_i^{1-\alpha}}{1-\alpha}$

(۴) Max-min fairness; overall objective : $\max \min x_i$

* α - Fairness کلی تر از مقیاس و این object می توان مقیاس کرد.

* ما می دانیم که Fairness بودن ، معنی مساوی بودن نیست ، اما فقط این مقیاس را داریم!

(برای محاسبه)

$$F(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2}$$

* Jain's Fairness Index

فرمان سید / Resource Allocation

انجام می دهیم که در هر Allocation سهم هر کس برابر است با x_i .

مثلاً در شبکه برای تخصیص فضای بزرگ (در خروجی روتر) تقسیم buffer space

هر کدام از این سهم ها می تواند x_i باشد.

* اولین موضوع صرفاً، انحصاری برای Computer Network نیست و می توان در جاهای دیگر هم استفاده کرد.

* خاصیت دوم Index: این شاخص است و نکته مهم اینست وقتی

کوچک شود بهترین وضعیت Fairness را دارد (همه n ها به هم مساوی هستند)

در حالت ایده آل وقتی هیچگاه صفر نمی شود و minimum آن $\frac{1}{n}$ است

وزنی، $\frac{1}{n}$ می شود زمانی است که بیشترین عدم انصاف (زمانی که یکی از n ها برابر 0 باشد)

باشند و بقیه n ها برابر صفر باشند (بدقیقت resource نامیده می شود)

ما از این Index با فرض یکسان بودن x ها استفاده می کنیم

* این Index بیشتر برای اندازه گیری ^{فقط} fairness استفاده می شود

Example 1: consider a set of resource allocations as $x = (1, 0, 1) \Rightarrow$ Jain's Index = $\frac{2}{3}$

Example 2: ~~request~~ consider a set of resource allocations as $x_i = 1$, $i = 1, \dots, k$, and 0 oth.
Jain's Index = $\frac{k}{n}$

\Rightarrow if $y\%$ of users treated fairly (equally resourced) and others starved (no resourced), the Index will be equal to $y\%$

Example 3: Consider a set of resource allocations as 50, 30, 50 while the optimal ones are 50, 10, 10.

در این مورد به صورت نسبی انجام می شود و سپس می توانیم به مقدار optimal مقایسه ایالات ایجاد شده است.

Note: An approach is to first normalize the allocations according to optimal ones, and then compute the index:
 $n_s (50/50, 30/10, 50/10) = (1, 3, 5)$

* برای اندازه گیری Pair می توان از مقادیر زیر استفاده کرد:

۱) Mean : $\mu = \frac{1+3+5}{3} = 3$

۲) Variance: $\sigma^2 = 2$

۳) Max-min ratio : $5/1 = 5$

۴) Normalized distance from the optimal:

Normal Distance = 0.17

در Normalized فاصله از مقدار optimal را با نسبت به مقدار optimal مقایسه می کنیم و تقسیم بر مقدار optimal می کنیم.
 اول : distance - بعد : می توان از این ریشه گیری کنیم که مقادیر صاف را داشته باشند.
 و منفی نیست و وجود نداشته باشد.

خصائص Jain's Fairness Index :

- ۱) محدود بودن bounded
- ۲) ~~scale~~ independent : اگرچه اعداد را در یک مقیاس تغییر می‌دهیم
- ۳) مستقل از scale است.
- ۴) Continuous : پیوسته است.
- ۵) Direct Relationship : رابطه مستقیم با fairness دارد.

congestion control : Queuing Disciplines

فرض می‌کنیم یک روتر داریم، که می‌خواهیم تقسیم کنیم buffer space بین روتر
چگونه و به چه ترتیبی بپذیریم و چه کسی را drop کنیم؟

۱) **FIFO** : وقتی packet های روتر آن به سرانجام ارسال می‌شود، روتر
رسیده است. و برای drop کردن می‌توان بر روی های مختلفی بررسی کرد.
مثلاً اگر ترافیک جدید آمد drop شود و یا اینکه ممکن است
آن packet تازه رسیده، مهم باشد، آنگاه ~~packet~~ packet دیگری
اولویت کمتری دارد و حذف کنیم.

~~نکته~~ **FIFO** : روشی برای مدیریت بافر و نحوه ارسال است. و با drop کردن
متفاوت است. اما معمولاً ~~FIFO~~ FIFO در کنار tail drop می‌آید.

tail drop : وقتی وقتی packet جدید وارد می‌شود و ترافیک آن را drop می‌کنیم

و معیار $tail\ drop$ این است، هر کس که دیرتر آید را $drop$ می کند

« شکل توی جزوه رو برای نحوه $drop$ کردن نگاه کن »

بهترین موردی که در روترها استفاده می شود FIFO است.

$priority\ queuing$ در $FIFO$: می توان به جای یک بافر چندین بافر داشتیم

و هر بافر را برای هر کلاس که ارجحیت دارد قرار دهیم. و نگاه کن $flow$ های مختلف

جدید وارد می شوند به عنوان یک کلاس با آنها برخورد می شود و آن Pb_w که بالاترین

$priority$ را دارد تا زمانی که صف آن خالی نشده باشد، اول ارسال می شود.

« شکل توی جزوه »

اما اگر می بایست خودش را ~~از~~ ارجح تر معرفی کند باید اولاً مکانیزم های که

این مورد را بررسی می کنند مستقیماً آن را کرد. (نمودار در شبکه های عاری این کار انجام نمی شود)

کستری

در شبکه های عاری همه $flow$ ها یکسان هستند مثلاً Pb_w های ~~کستری~~

روماً : ساز کردن با یول : که در شبکه های عاری رخ نمی دهد معمولاً در $back\ bone$ ها

در ISP ها رخ می دهد.