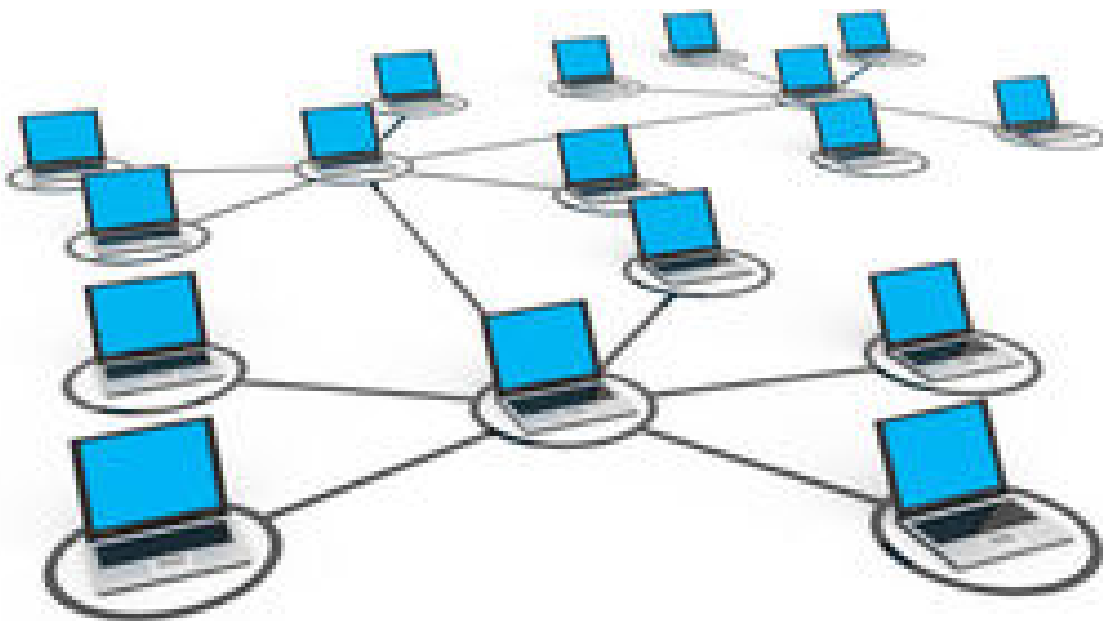


PIE és PVPIE hálózati ütemező algoritmusok

Előadás vázлата

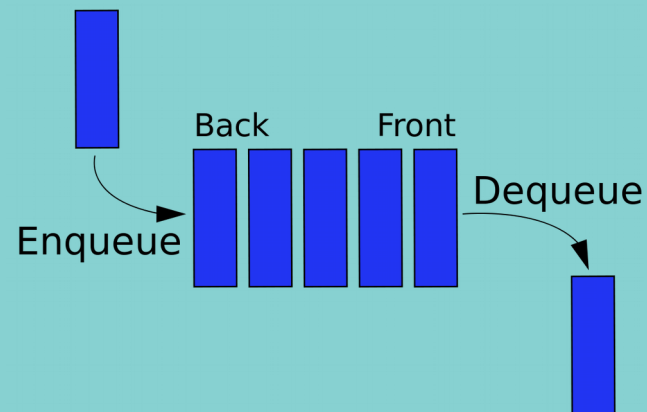
- 1) Általános fogalmak
- 2) PIE
- 3) PVPIE

Hálózati alapfogalmak



© Can Stock Photo

- hálózat
- hálózati csomópont
- hálózati kapcsolat
- hálózati csomagok
- hálózati protokoll



Protokoll

Szabályrendszer a kommunikáció:

- szintaxisához
- szinkronizációjához
- hibaelhárításhoz

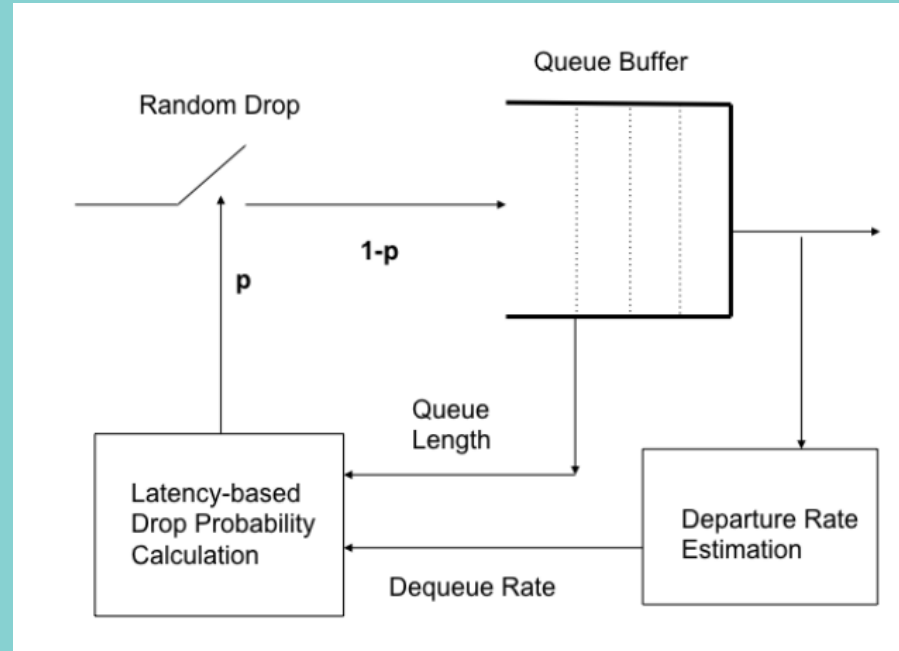
Protokoll:TCP

- Nagyon gyakori protokoll.
- Az internet nagy része TCP-t vagy UDP-t használ.
- Addig növeli a küldési sebességet, amíg nem érzékel csomagdobást. Ekkor nagyon nagy mértékben visszaveszi a küldési sebességet.
- „Bufferbloat” probléma

„Bufferbloat”

- Az a jelenség, hogy a hálózati csomópontokban lévő nagy buffer(memória) nagy várakozási időt okoz.
- TCP nagy kihasználtságra (high link utilization) törekszik, ami miatt különösen érzékeny.
- FIFO ütemezéssel ez probléma.

Megoldás: PIE! (vagy bármilyen AQM)



1. rész: PI

The PIE algorithm updates the drop probability periodically as follows:

- estimate current queueing delay using Little's law:

$$cur_del = \frac{qlen}{avg_drate};$$

- calculate drop probability p as:

$$p = p + \alpha * (cur_del - ref_del) + \beta * (cur_del - old_del);$$

- update previous delay sample as:

$$old_del = cur_del.$$

2. rész: (PI)E

if $p < 1\%$: $\alpha = \tilde{\alpha}/8$; $\beta = \tilde{\beta}/8$;
else if $p < 10\%$: $\alpha = \tilde{\alpha}/2$; $\beta = \tilde{\beta}/2$;
else: $\alpha = \tilde{\alpha}$; $\beta = \tilde{\beta}$;

$\tilde{\alpha}$

3. rész: továbbküldési sebességbecslés

Upon packet departure

1. Decide to be in a measurement cycle if:

$$qlen > dq_threshold;$$

2. If the above is true, update departure count dq_count :

$$dq_count = dq_count + dq_pktsize;$$

3. Update departure rate once $dq_count > dq_threshold$ and reset counters:

$$dq_int = now - start;$$

$$dq_rate = \frac{dq_count}{dq_int};$$

$$avg_drate = (1 - \varepsilon) * avg_drate + \varepsilon * dq_rate$$

$$start = now.$$

$$dq_count = 0;$$

4. rész: „burst”-ök kezelése

Upon packet arrival

1. If $burst_allow > 0$

enqueue packet bypassing random drop;

Upon dq_rate update

2. Update burst allowance:

burst_allow = burst_allow - dq_int;

3. if $p = 0$; and both cur_del and old_del less than $ref_del/2$, reset $burst_allow$,

burst_allow = max_burst;

PVPIE: Packet Value aware PIE

- PPV(Per Packer Value)+PIE
- Különböző szolgáltatásoknak különböző erőforrásigénye van
- Két részből áll: csomagértékek hozzárendelése a csomagokhoz tetszőleges célhoz illő szabály szerint és a csomagok továbbítása, eldobása a csomag értéke alapján a hálózat belsejében

PVPIE komponensei

