

PDV 10 2017/2018

Volba lídra (koordinátora)

Michal Jakob

michal.jakob@fel.cvut.cz

Centrum umělé inteligence, katedra počítačů, FEL ČVUT



Příklad

Detaily bankovního účtu jsou replikovány na několika serverech.

Ale pouze jeden by měl být zodpovědný za příjem všech požadavků na čtení a zápis – tím je tzv. **lídr** mezi replikami.

Co kdyby:

- byli dva lídři pro jeden účet?
- se servery neshodly na tom, kdo je lídr?
- kdyby lídr havroval?

Všechny výše uvedené situace by mohl vést k nekonzistencím.

Problém volby lídra

Ze skupiny procesů **vybrat lídra** (který bude řešit specifické úkoly) a **dát vědět všem procesům** ve skupině, kdo je lídrem.

Co se stane, když lídr selže?

- nějaký proces detekuje pomocí detektoru selhání a spustí nové volby

Algoritmus pro volbu lídra musí zajistit:

1. zvolí právě jednoho lídra z bezvadných procesů
2. všechny bezvadné procesy ve skupině se shodnou na tom, kdo je lídr

Systemový model

Skupina N procesů s unikátními identifikátory.

- známe všechny procesy, ale nevíme, které jsou aktivní (bezvadné)

Procesy **mohou havarovat**.

FIFO perfektní komunikační kanál mezi každým párem procesů, tj. zprávy se neduplikují, nevznikají, neztrácejí a jsou doručovány v pořadí odeslání.

Asynchronní systém: neznáma, ale **konečná latence**.

Další požadavky

Každý proces může **vyvolat** volby.

- **jeden proces** může vyvolat v jeden okamžik pouze **jedny volby**.

Více procesů může vyvolat volby **současně**.

- pak požadujeme, aby se nakonec shodly na jednom lídrovi

Výsledek volby lídra by **neměl záviset** na tom, který proces volby **vyvolal**. Po skončení běhu algoritmu volby lídra má každý proces ve své proměnné *ELECTED* identifikátor lídra s nejvyšší hodnotou volebního kritéria.

Volební kritérium:

- typicky nejvyšší identifikátor, tj. IP adresa
- ale taky např: nejvíce RAM, diskového prostoru, nebo např. nejvíce souborů v případě P2P sítí.
- musí být fixní a znám všem procesům při zahájení volby



Bully Algorithmus

Bully algoritmus

Klasický algoritmus pro volbu lídra.

Základní princip: Proces, který **detekoval selhání** dosavadního lídra, **vyzve** procesy s **vyšším ID** ve volbách.

Bully algoritmus

P_i : zahájení voleb (po detekci selhání nebo jako reakce na volby)

```
if  $P_i$  má nejvyšší ID
    Pošli COORDINATOR zprávu
    všem procesům s nižšími
    identifikátory (volby končí).
else // zahájí volby
    Pošli ELECTION zprávu všem
    procesům s vyšším ID
    // následně čekání na odpověď
```

P_i : reakce na ELECTION

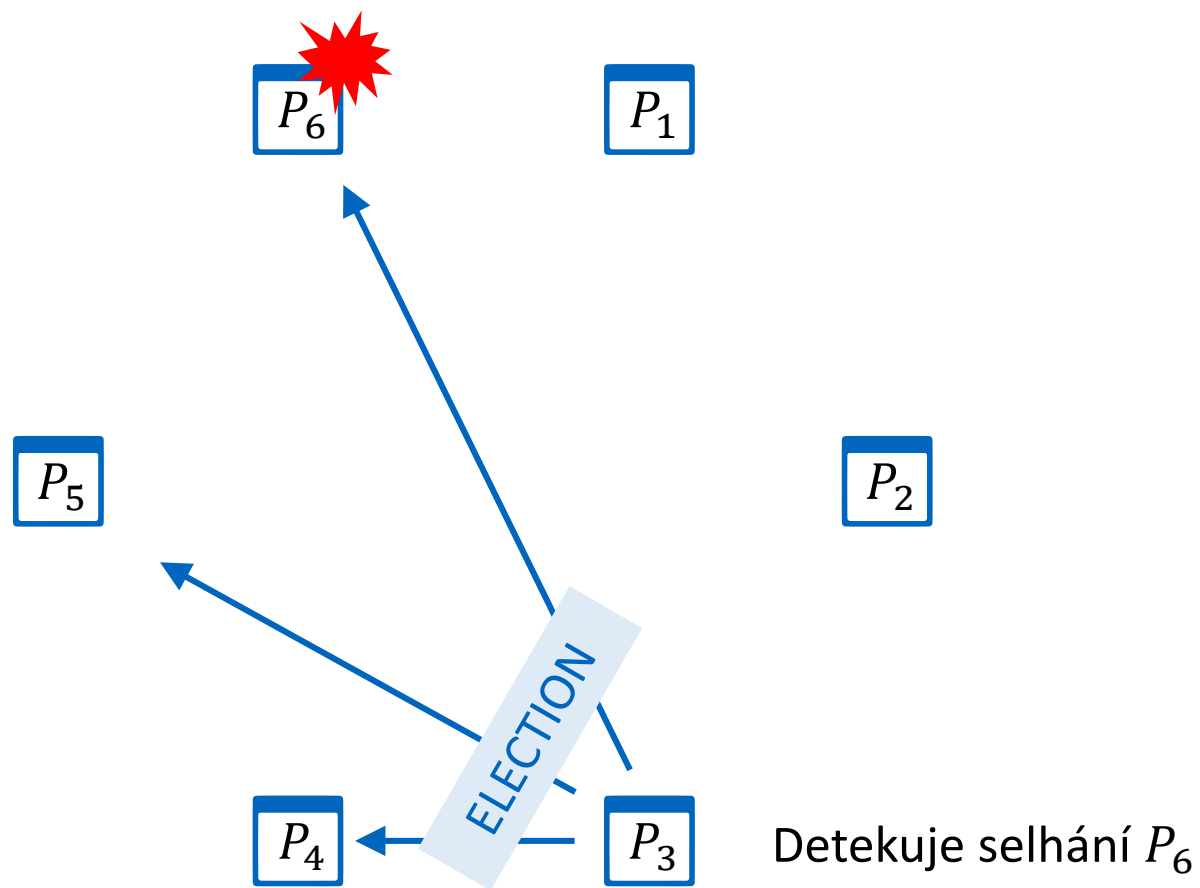
```
Odpověz OK
If pokud  $P_i$  dosud nezahájil volby
    zahaj volby
```

P_i : čekání na odpovědi
(po vyvolání voleb)

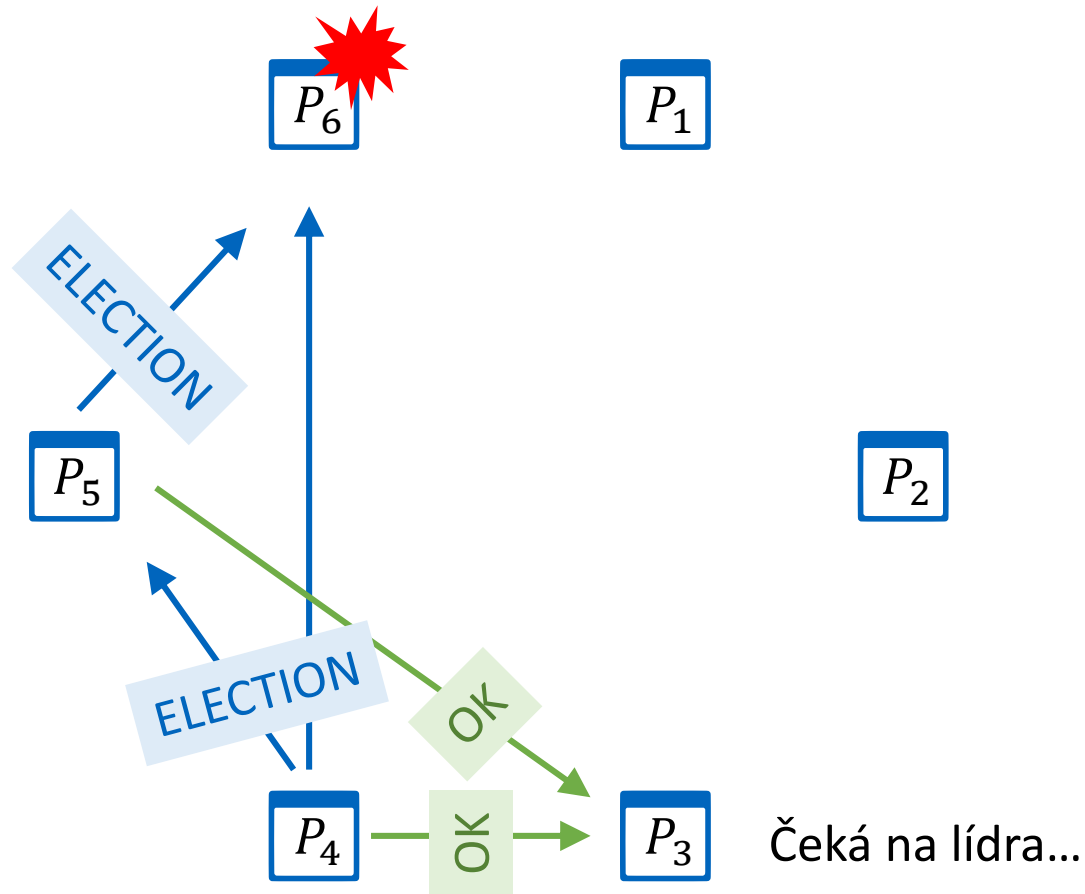
```
if nedorazí žádná odpověď v
časovém limitu
    prohleš se jako  $P_i$  za lídra;
    pošli COORDINATOR zprávu
    všem procesům s nižším ID;
    // volby skončily

else // existuje aktivní proces s
vyšším ID než  $P_i$ 
    čekej na zprávu
    COORDINATOR;
    pokud nedorazí v časovém
    intervalu, iniciuj nové volby
```

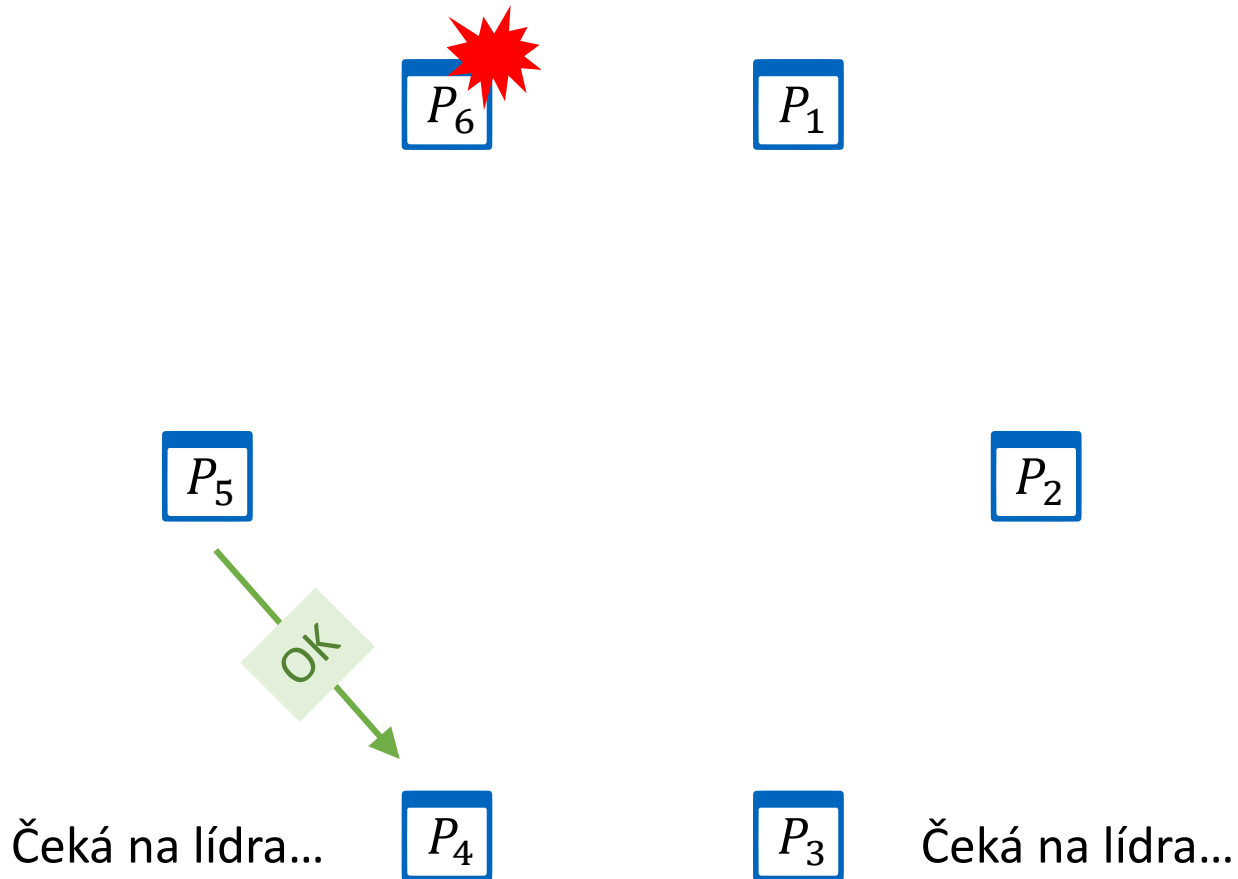

Bully algoritmus



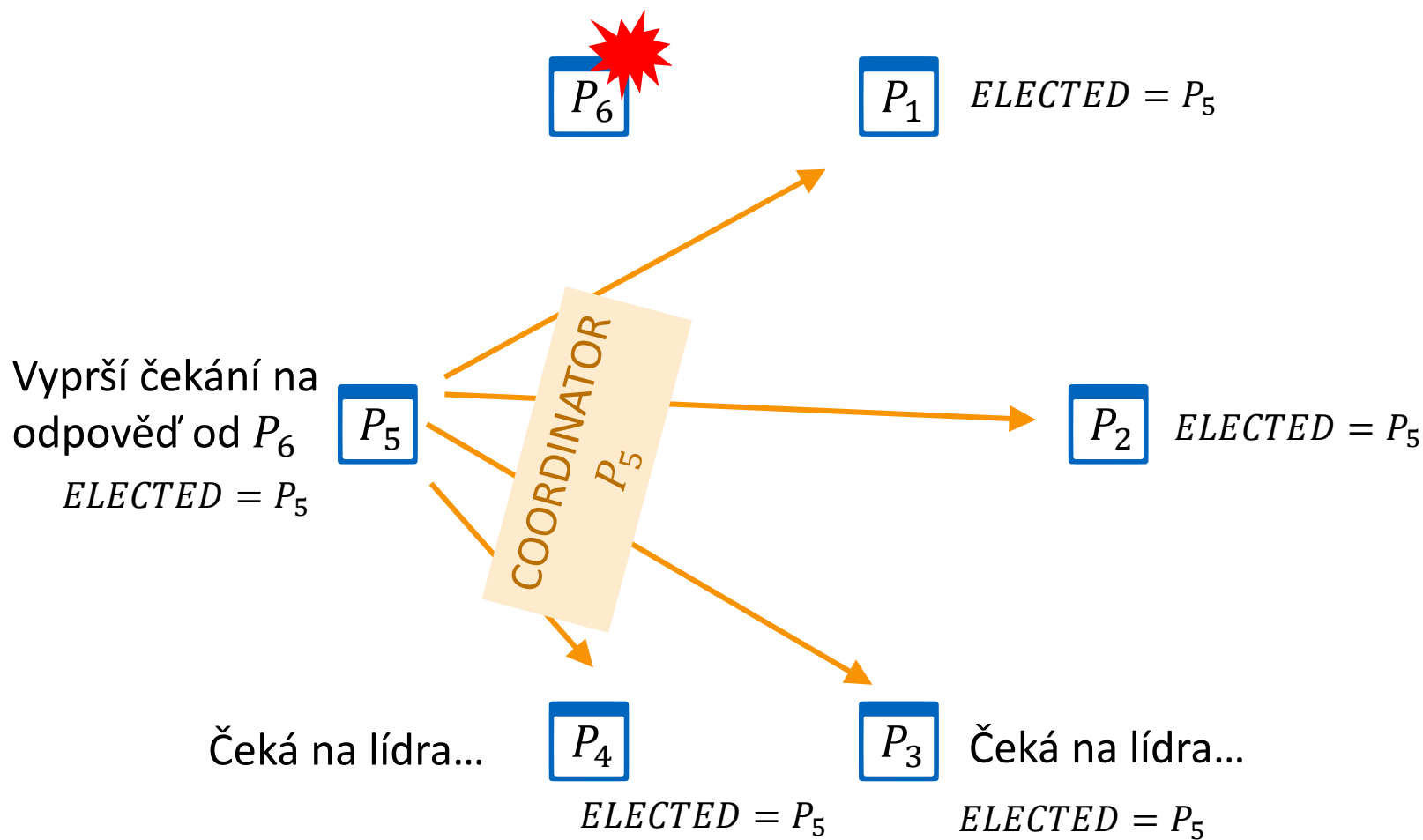
Bully algoritmus



Bully algoritmus

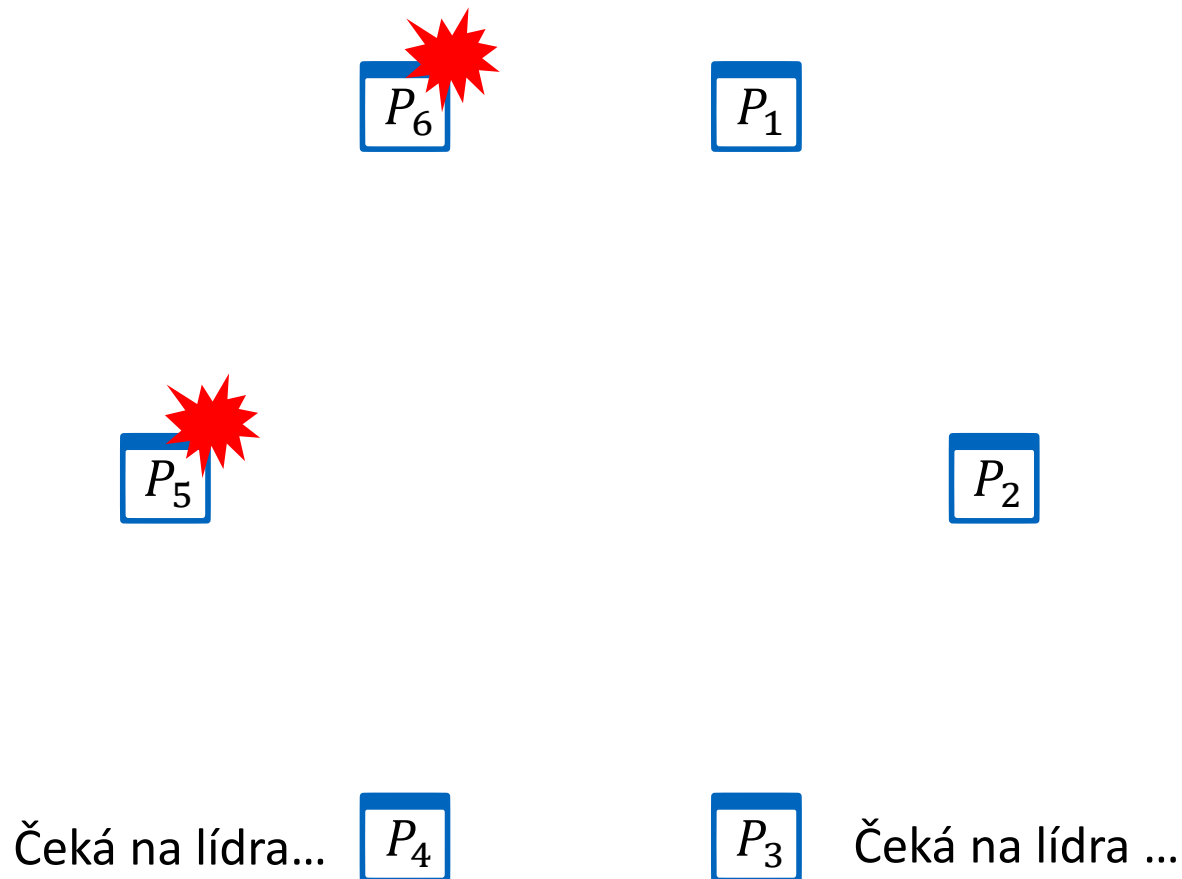


Bully algoritmus

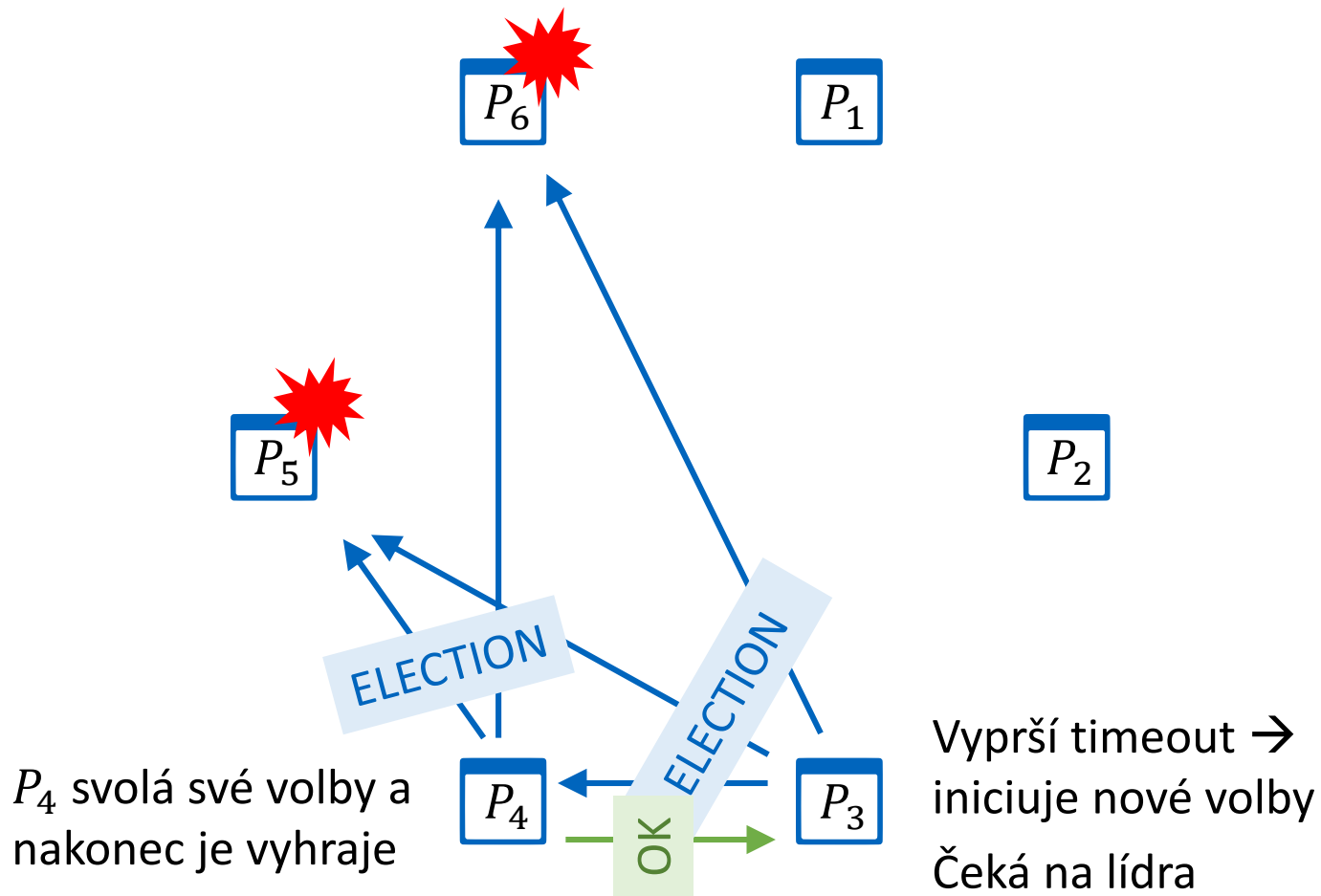


Volby jsou skončeny

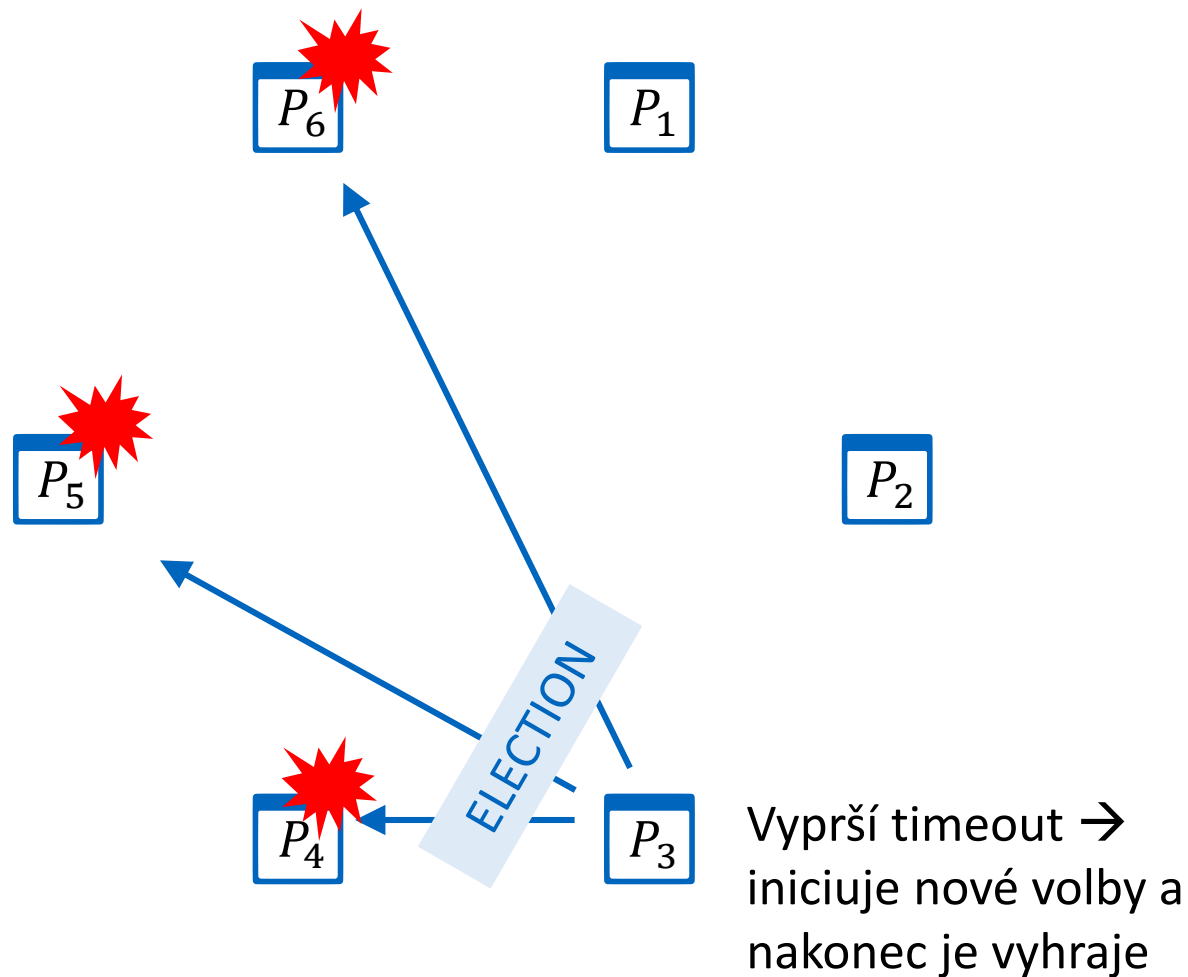
Selhání během volby



Selhání během volby



Selhání během volby



Selhání a timeouty

Pokud přestanou selhávat další procesy, dojde časem ke zvolení lídra.

Jak nastavit timeout?

Nejhorší případ – selhání lídra detekováno procesem s nejnižším ID.

Pro dokončení volby lídra je v nejhorším případě potřeba **čtyři komunikační latence**:

1. Proces s nejnižším ID pošle ELECTION zprávu ostatním procesům
2. Proces s druhým nejvyšším ID:
 - pošle OK procesům s nižšími ID
 - pošle ELECTION procesu s nejvyšším ID
3. Procesu s druhým nejvyšším ID vyprší timeout při čekání na odpověď procesu s nejvyšším ID
4. Procesu s druhým nejvyšším ID rozešle zprávu COORDINATOR

Analýza

Nejhorší případ – komunikační zátěž

- celkem $N - 1$ procesů pošle zprávu ELECTION všem procesům s vyšším ID
- tj. celkem: $N - 1 + N - 2 + \dots + 1 = \frac{N(N-1)}{2} = \mathbf{O(N^2)}$ zpráv

Nejlepší případ: proces s druhým nejvyšším ID detekuje selhání lídra

- komunikační zátěž: $(N - 2)$ zpráv COORDINATOR
- čas do ukončení: **1 komunikační latence**

Živost

Bully algoritmus pracuje s timeouts → **v asynchronním systému** jeho běh nemusí nikdy skončit → **živost není garantována.**

V synchronním systému:

- lze spočítat nejhorší jednosměrnou latence = doba přenosu zprávy + doba reakce na zprávu.
- pokud timeout nastavíme na násobek nejhorší jednosměrné latence, je **živost garantována.**

Souhrn

Volba lídra důležitý problém v DS.

Bully algoritmus klasický algoritmus předpokládající selhání procesů, ale perfektní FIFO kanál.

V asynchronním systému **negarantuje živost**; v synchronním nebo částečně synchronním systému lze živost dosáhnout vhodným **nastavením timeout**.