

Itsestabiloiva bysanttilainen yhteisymmärrys

Timo Virkkala

Ongelma

- Päätöksenteko
 - Yksi lähettää arvon
 - Kaikki yrittävät päästä yhteisymmärrykseen
- Transientit virheet
 - Ratkaisu: Itsestabilointi
- Bysanttilaiset virheet
 - Ratkaisu: Enemmistö

ss-Byz-Agree -protokolla

- Vikamalli
 - Aluksi rajaton määrä mitä tahansa virheitä
 - Mikä tahansa ”alkutila”
 - Lopulta siedettävä virhetaso riittävän pitkään
 - $3f < n$, missä f virheiden määrä, n solmujen määrä
 - Päätös ajassa $O(f')$ kommunikaatiokierrosta
 - f' samanaikaisten virheiden todellinen määrä

ss-Byz-Agree -protokolla

- Verkko ja kommunikointi
 - n solmua
 - Autentikoidut lähettäjät
 - Ei takuuta viestien järjestyksestä
 - Ei globaalia kelloa, yhteistä pulssia
 - Paikallisten kellojen nopeusvirhe ρ
 - (*Bounded drift*)

Algoritmin yleiskatsaus

ss-Byz-Agree -protokolla

Q. INITIATOR-ACCEPT(G, m)

Alustus/esihyväksyminen (seur.kalvo)

R1. **if** I-accept $\langle G, m', \tau_q^G \rangle$ **and** $\tau_q - \tau_q^G \leq 4d$ **then**
R2. $value := \langle G, m' \rangle$;
R3. MSGD-BROADCAST($q, value, 1$);
R4. **stop and return** $\langle value, \tau_q^G \rangle$.

Arvon hyväksyminen

S1. **if** τ_q mennessä vastaanotettu r erillistä viestiä $(p_i, \langle G, m'' \rangle, \tau_i, i)$
 (missä $\tau_q \leq \tau_q^G + (2r + 1) \cdot \Phi$ ja $\forall i, j : (1 \leq i \leq r) \wedge (p_i \neq p_j \neq G)$) **then**
S2. $value := \langle G, m'' \rangle$;
S3. MSGD-BROADCAST($q, value, r + 1$);
S4. **stop and return** $\langle value, \tau_q^G \rangle$.

Arvon hyväksyminen

T1. **if** τ_q mennessä $|broadcasters| < r - 1$ (missä $\tau_q > \tau_q^G + (2r + 1) \cdot \Phi$) **then**
T2. **stop and return** $\langle \perp, \tau_q^G \rangle$.

Lopetus ilman arvoa

U1. **if** $\tau_q > \tau_q^G + (2f + 3) \cdot \Phi$ **then**
U2. **stop and return** $\langle \perp, \tau_q^G \rangle$.

Lopetus ilman arvoa

siivous:

3d arvon palauttamisen jälkeen nolaa suorituskertaan
liittyvät INITIATOR-ACCEPT ja MSGD-BROADCAST;

Poista kaikki $(2f + 3) \cdot \Phi + 3d$ aikayksikköä vanhemmat arvot ja viestit.

Initiator-Accept

K1. **if** $\tau_q - last_ \tau_q > 7d$ **and if** hetkellä $\tau_q - d$ $initiator[G, _] = \perp$ **then**
K2. lähetä $(support, G, m)$ kaikille;
K3. $initiator[G, m] := \tau_q - d$;

Kenraalilta saadun
arvon kannatus

L1. **if** vastaanotettu $(support, G, m)$ vähintään $\geq n - 2f$ erilliseltä solmulta
 $\alpha \leq 4d$ aikayksikön sisällä toisistaan **then**
L2. $initiator[G, m] := \max[initiator[G, m], (\tau_q - \alpha - 2d)]$;
L3. **if** vastaanotettu $(support, G, m)$ vähintään $\geq n - f$ erilliseltä solmulta
 $2d$ aikayksikön sisällä toisistaan **then**
L4. lähetä $(ready, G, m)$ kaikille;

Kannatuksen
kerääminen

M1. **if** vastaanotettu $(ready, G, m)$ vähintään $\geq n - 2f$ erilliseltä solmulta **then**
M2. lähetä $(ready, G, m)$ kaikille;
M3. **if** vastaanotettu $(ready, G, m)$ vähintään $\geq n - f$ erilliseltä solmulta **then**
M4. $\tau_q^G := initiator[G, m]$; I-accept $\langle G, m, \tau_q^G \rangle$; $last_ \tau_q := \tau_q$;

Alustuksen
hyväksyminen

siivous:

Poista kaikki $\Delta + 7d$ aikayksikköä vanhemmat arvot ja viestit.
if $last_ \tau_q > \tau_q$ **then** $last_ \tau_q := \perp$.

Msgd-Broadcast

V. lähetä (*init*, *p*, *m*, *k*) kaikille;

Oman arvon ehdotus

W1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + 2k \cdot \Phi$

W2. **if** vastaanotettu (*init*, *p*, *m*, *k*) solmulta *p* **then**

W3. lähetä (*echo*, *p*, *m*, *k*) kaikille;

Ensimmäinen kaiku

X1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + (2k \blacksquare 1) \cdot \Phi$ Typo(?)

X2. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

X3. lähetä (*init'*, *p*, *m*, *k*) kaikille;

X4. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

X5. hyväksy(*p*, *m*, *k*);

Konsensuksen
ehdotus

Y1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + (2k + 2) \cdot \Phi$

Y2. **if** vastaanotettu (*init'*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

Y3. *broadcasters* := *broadcasters* \cup {*p*};

Y4. **if** vastaanotettu (*init'*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

Y5. lähetä (*echo'*, *p*, *m*, *k*) kaikille;

Toinen kaiku

Z1. Millä tahansa hetkellä:

Z2. **if** vastaanotettu (*echo'*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

Z3. lähetä (*echo'*, *p*, *m*, *k*) kaikille;

Z4. **if** vastaanotettu (*echo'*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

Z5. hyväksy(*p*, *m*, *k*);

Hyväksyminen

siivous:

Poista kaikki $(2f + 3) \cdot \Phi$ aikayksikköä vanhemmat arvot ja viestit.

Määritelmiä

Määritelmiä

- Ei-viallinen solmu (*non-faulty node*)
 - Rajallinen poikkeama (*bounded drift*)
 - Nopeus ρ päässä reaaliajan nopeudesta
 - Tottelevaisuus (*obedience*)
 - Noudattaa protokollaa täsmällisesti
 - Rajallinen käsittelyaika (*bounded processing time*)
 - Käsittelee kaikki viestit π aikayksikön sisällä saapumisesta
- Korrekti solmu
 - Jos toiminut ei-viallisesti riittävän pitkään
 - Δ_{solmu} , määritellään myöhemmin

Määritelmiä

- Ei-viallinen verkko (*non-faulty network*)
 - Rajoitettu toimitusaika
 - Kaikki viestit saapuvat δ aikayksikön sisällä
 - Autentikointi
 - Lähettäjän identiteetti ei muutu
 - Viestin sisältö ei muutu
- Korrekti verkko
 - Jos toiminut ei-viallisesti riittävän pitkään
 - Δ_{verkko} , määritellään myöhemmin

Määritelmiä

- Koherentti järjestelmä
 - Päätösvaltainen joukko (*quorum*)
 - Vähintään $n - f$ korrektia solmua
 - Korrekti verkko

Määritelmiä

- Viestien kuljetus- ja käsittelyaika $d \equiv \delta + \pi$
 - δ viestien kuljetusaika verkossa
 - π viestien käsittelyaika solmussa
- $\Delta_{\text{solmu}} \geq 14(2f + 3)d + 10d$
 - Solmun korrektiuteen tarvittava aika
- $\Delta_{\text{verkko}} \geq d$
 - Verkon korrektiuteen tarvittava aika
- n , f ja d vakioita
 - *Eivät voi muuttua virhetilanteissa*

Ominaisuudet

- Yhteisymmärrys (*agreement*)
 - Kaikki korrektit solmut päätyvät samaan arvoon
 - ...jos tämä arvo on epätyhjä
- Validiteetti (*validity*)
 - Jos korrektit solmut aloittavat protokollan kenraalin G lähettämälle arvolle, niin kaikki korrektit solmut hyväksyvät tämän arvon
- Päättyminen (*termination*)
 - Protokolla päättyy äärellisessä ajassa
- Oikea-aikaisuus (*timeliness*)
 - Monimutkainen määritelmä

Algoritmin toiminta

Aloitus

- Kenraali G lähettää kaikille (*Initiator*, G , m)
 - Mikä tahansa solmu voi toimia kenraalina
 - Useita käynnistyksiä samanaikaisesti
 - Korrektit kenraalit lähettävät rajoitetusti
- Solmut aloittavat protokollan
 - Jos vastaanotettu suora viesti kenraalilta
 - Kutsutaan *Initiator-Accept*(G , m)
 - Jos havaitaan, että muut aloittaneet
 - Suoritetaan osia *Initiator-Accept*:sta

Esihyväksyntä

Q. INITIATOR-ACCEPT(G, m)

R1. **if** I-accept($\langle G, m', \tau_q^G \rangle$) **and** $\tau_q - \tau_q^G \leq 4d$ **then**
R2. $value := \langle G, m' \rangle$;
R3. MSGD-BROADCAST($q, value, 1$);
R4. **stop and return** $\langle value, \tau_q^G \rangle$.

S1. **if** τ_q mennessä vastaanotettu r erillistä viestiä $(p_i, \langle G, m'' \rangle, \tau_i, i)$
 (missä $\tau_q \leq \tau_q^G + (2r + 1) \cdot \Phi$ ja $\forall i, j : (1 \leq i \leq r) \wedge (p_i \neq p_j \neq G)$) **then**
S2. $value := \langle G, m'' \rangle$;
S3. MSGD-BROADCAST($q, value, r + 1$);
S4. **stop and return** $\langle value, \tau_q^G \rangle$.

T1. **if** τ_q mennessä $|broadcasters| < r - 1$ (missä $\tau_q > \tau_q^G + (2r + 1) \cdot \Phi$) **then**
T2. **stop and return** $\langle \perp, \tau_q^G \rangle$.

U1. **if** $\tau_q > \tau_q^G + (2f + 3) \cdot \Phi$ **then**
U2. **stop and return** $\langle \perp, \tau_q^G \rangle$.

siivous:

$3d$ arvon palauttamisen jälkeen nolaa suorituskertaan
 liittyvät INITIATOR-ACCEPT ja MSGD-BROADCAST;
Poista kaikki $(2f + 3) \cdot \Phi + 3d$ aikayksikköä vanhemmat arvot ja viestit.

Initiator-Accept

- Primitiivin **Initiator-Accept**(G, m) tarkoitus
 - Yhteisymmärrys kenraalin lähettämästä arvosta
 - ”Esihyväksyminen”
 - Kandidaattiarvo tulevalle lopulliselle hyväksynnälle
 - Yhteisymmärrys aloituksen ajankohdasta
 - Suhteessa omaan kelloon

Initiator-Accept

K1. if $\tau_q - last_ \tau_q > 7d$ and if hetkellä $\tau_q - d$ $initiator[G, _] = \perp$ then
K2. lähetä $(support, G, m)$ kaikille;
K3. $initiator[G, m] := \tau_q - d$;

Kenraalilta saadun
arvon kannatus

- Suoritetaan solmussa q
 - Lohko K suoritetaan vain, jos q vastaanottanut G :n suoran viestin
- Paikallisaika τ_q
- Tietorakenne $initiator[G, m]$
 - Säilyttää tiedon kenraalin G aloittaman instanssin arvioidusta aloitusajasta τ_q^G
- Lähetetään muille tieto omasta tilanteesta
 - $(support, G, m)$
 - ”Kannatan G :n lähettämää arvoa m ”

Initiator-Accept

```
K1. if  $\tau_q - last\_ \tau_q > 7d$  and if hetkellä  $\tau_q - d$   $initiator[G, \_] = \perp$  then  
K2.   lähetä  $(support, G, m)$  kaikille;  
K3.    $initiator[G, m] := \tau_q - d$ ;
```

Kenraalilta saadun
arvon kannatus

```
L1. if vastaanotettu  $(support, G, m)$  vähintään  $\geq n - 2f$  erilliseltä solmulta  
    $\alpha \leq 4d$  aikayksikön sisällä toisistaan then  
L2.    $initiator[G, m] := \max[initiator[G, m], (\tau_q - \alpha - 2d)]$ ;  
L3. if vastaanotettu  $(support, G, m)$  vähintään  $\geq n - f$  erilliseltä solmulta  
    $2d$  aikayksikön sisällä toisistaan then  
L4.   lähetä  $(ready, G, m)$  kaikille;
```

Kannatuksen
kerääminen

- L1 ja L2 toistetaan, kunnes I-accept
- Muut suoritetaan korkeintaan kerran
 - Kun esiehto pätee
- Jos vastaanotettu riittävä kannatus
 - Lähetään $(ready, G, m)$
 - ”Olen valmis esihyväksymään arvon m ”

Initiator-Accept

```
K1. if  $\tau_q - last\_tau_q > 7d$  and if hetkellä  $\tau_q - d$   $initiator[G, \_] = \perp$  then  
K2.   lähetä  $(support, G, m)$  kaikille;  
K3.    $initiator[G, m] := \tau_q - d$ ;
```

Kenraalilta saadun
arvon kannatus

```
L1. if vastaanotettu  $(support, G, m)$  vähintään  $\geq n - 2f$  erilliseltä solmulta  
    $\alpha \leq 4d$  aikayksikön sisällä toisistaan then  
L2.    $initiator[G, m] := \max[initiator[G, m], (\tau_q - \alpha - 2d)]$ ;  
L3. if vastaanotettu  $(support, G, m)$  vähintään  $\geq n - f$  erilliseltä solmulta  
    $2d$  aikayksikön sisällä toisistaan then  
L4.   lähetä  $(ready, G, m)$  kaikille;
```

Kannatuksen
kerääminen

```
M1. if vastaanotettu  $(ready, G, m)$  vähintään  $\geq n - 2f$  erilliseltä solmulta then  
M2.   lähetä  $(ready, G, m)$  kaikille;  
M3. if vastaanotettu  $(ready, G, m)$  vähintään  $\geq n - f$  erilliseltä solmulta then  
M4.    $\tau_q^G := initiator[G, m]$ ; I-accept $\langle G, m, \tau_q^G \rangle$ ;  $last\_tau_q := \tau_q$ ;
```

Alustuksen
hyväksyminen

- Jatketaan esihyväksynnän ehdottamista
- Jos riittävästi solmuja valmiina hyväksymään
→ Esihyväksyntä **I-accept** $\langle G, m, \tau_q^G \rangle$

Varsinainen yhteisymmärrys

- Kun esihyväksyntä on tehty
 - Määritetty kenraali G , arvo m , aloitusaika τ_q^G
- Aika muodostaa varsinainen yhteisymmärrys
 - Käytetään primitiiviä $\text{Msgd-Broadcast}(p, arvo, k)$
 - Pyrkii hyväksymään viestin $arvo = \langle G, m \rangle$
 - Lähettäjäsolmu p
 - Viestintäkierros k
- Jokainen lohko seuraavilla kalvoilla suoritetaan korkeintaan kerran
 - Jos esiehto pätee

Varsinainen yhteisymmärrys

Q. INITIATOR-ACCEPT(G, m)

R1. **if** I-accept $\langle G, m', \tau_q^G \rangle$ **and** $\tau_q - \tau_q^G \leq 4d$ **then**
R2. $value := \langle G, m' \rangle$;
R3. MSGD-BROADCAST($q, value, 1$);
R4. **stop and return** $\langle value, \tau_q^G \rangle$.

S1. **if** τ_q mennessä vastaanotettu r erillistä viestiä $(p_i, \langle G, m'' \rangle, \tau_i, i)$
 (missä $\tau_q \leq \tau_q^G + (2r + 1) \cdot \Phi$ ja $\forall i, j : (1 \leq i \leq r) \wedge (p_i \neq p_j \neq G)$) **then**
S2. $value := \langle G, m'' \rangle$;
S3. MSGD-BROADCAST($q, value, r + 1$);
S4. **stop and return** $\langle value, \tau_q^G \rangle$.

T1. **if** τ_q mennessä $|broadcasters| < r - 1$ (missä $\tau_q > \tau_q^G + (2r + 1) \cdot \Phi$) **then**
T2. **stop and return** $\langle \perp, \tau_q^G \rangle$.

U1. **if** $\tau_q > \tau_q^G + (2f + 3) \cdot \Phi$ **then**
U2. **stop and return** $\langle \perp, \tau_q^G \rangle$.

siivous:

$3d$ arvon palauttamisen jälkeen nolaa suorituskertaan
liittyvät INITIATOR-ACCEPT ja MSGD-BROADCAST;
Poista kaikki $(2f + 3) \cdot \Phi + 3d$ aikayksikköä vanhemmat arvot ja viestit.

Msgd-Broadcast

V. lähetä $(init, p, m, k)$ kaikille;

Oman arvon ehdotus

- Lähetetään $(init, p, m, k)$
 - ”Minä p ehdotan viestiä m kierroksella k ”
 - Viesti m sisältää $\langle G, arvo \rangle$
 - $arvo$: Esihyväksynnän kandidaattiarvo (siellä m)

Msgd-Broadcast

V. lähetä $(init, p, m, k)$ kaikille;

Oman arvon ehdotus

W1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + 2k \cdot \Phi$

Ensimmäinen kaiku

W2. **if** vastaanotettu $(init, p, m, k)$ solmulta p **then**

W3. lähetä $(echo, p, m, k)$ kaikille;

- Jos vastaanotettu ehdotus omasta arvosta
 - Kaiutetaan se muille
- Suorituksesta
 - Lohkot suoritetaan vain, kun τ_q^G on määritelty
 - Jokainen viesti lähetetään korkeintaan kerran
 - Vastaanotetut duplikaattiviestit jätetään huomiotta
 - Solmut lakkaavat osallistumasta viestintäkierrokseen $3d$ lopullisen hyväksymisen jälkeen

Msgd-Broadcast

V. lähetä (*init*, *p*, *m*, *k*) kaikille;

Oman arvon ehdotus

W1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + 2k \cdot \Phi$

W2. **if** vastaanotettu (*init*, *p*, *m*, *k*) solmulta *p* **then**

W3. lähetä (*echo*, *p*, *m*, *k*) kaikille;

Ensimmäinen kaiku

X1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + (2k - 1) \cdot \Phi$ Typo(?)

X2. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

X3. lähetä (*init'*, *p*, *m*, *k*) kaikille;

X4. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

X5. hyväksy(*p*, *m*, *k*);

Konsensuksen
ehdotus

- Jos vastaanotettu riittävästi kaikuja
 - Tiedetään montako solmua on mukana
 - Jos vaikuttaa, että voidaan saada enemmistö
 - Ehdotetaan yhteisymmärrystä
 - Jos ylivoimainen enemmistö, hyväksytään heti
 - Silti jatketaan myös seuraaviin lohkoihin

Msgd-Broadcast

V. lähetä (*init*, *p*, *m*, *k*) kaikille;

Oman arvon ehdotus

W1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + 2k \cdot \Phi$

W2. **if** vastaanotettu (*init*, *p*, *m*, *k*) solmulta *p* **then**

W3. lähetä (*echo*, *p*, *m*, *k*) kaikille;

Ensimmäinen kaiku

X1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + (2k \blacksquare 1) \cdot \Phi$ Typo(?)

X2. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

X3. lähetä (*init'*, *p*, *m*, *k*) kaikille;

X4. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

X5. hyväksy(*p*, *m*, *k*);

Konsensuksen
ehdotus

Y1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + (2k + 2) \cdot \Phi$

Y2. **if** vastaanotettu (*init'*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

Y3. *broadcasters* := *broadcasters* \cup {*p*};

Y4. **if** vastaanotettu (*init'*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

Y5. lähetä (*echo'*, *p*, *m*, *k*) kaikille;

Toinen kaiku

- Jälleen kaiutetaan
- *broadcasters* ei selitetty alkup. artikkelissa
 - Lista mukanaolevista solmuista

Msgd-Broadcast

V. lähetä (*init*, *p*, *m*, *k*) kaikille;

Oman arvon ehdotus

W1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + 2k \cdot \Phi$

W2. **if** vastaanotettu (*init*, *p*, *m*, *k*) solmulta *p* **then**

W3. lähetä (*echo*, *p*, *m*, *k*) kaikille;

Ensimmäinen kaiku

X1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + (2k + 1) \cdot \Phi$ Typo(?)

X2. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

X3. lähetä (*init'*, *p*, *m*, *k*) kaikille;

X4. **if** vastaanotettu (*echo*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

X5. hyväksy(*p*, *m*, *k*);

Konsensuksen
ehdotus

Y1. Hetkellä $\tau_q : \tau_q \leq \tau_q^G + (2k + 2) \cdot \Phi$

Y2. **if** vastaanotettu (*init'*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

Y3. *broadcasters* := *broadcasters* \cup {*p*};

Y4. **if** vastaanotettu (*init'*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

Y5. lähetä (*echo'*, *p*, *m*, *k*) kaikille;

Toinen kaiku

Z1. Millä tahansa hetkellä:

Z2. **if** vastaanotettu (*echo'*, *p*, *m*, *k*) vähintään $\geq n - 2f$ erilliseltä solmulta **then**

Z3. lähetä (*echo'*, *p*, *m*, *k*) kaikille;

Z4. **if** vastaanotettu (*echo'*, *p*, *m*, *k*) vähintään $\geq n - f$ erilliseltä solmulta **then**

Z5. hyväksy(*p*, *m*, *k*);

Hyväksyminen

- Hyväksytään korkeintaan kerran

ss-Byz-Agree -protokolla

Q. INITIATOR-ACCEPT(G, m)

Alustus/esihyväksyminen (seur.kalvo)

R1. if I-accept($\langle G, m', \tau_q^G \rangle$) and $\tau_q - \tau_q^G \leq 4d$ then
R2. $value := \langle G, m' \rangle$;
R3. MSGD-BROADCAST($q, value, 1$);
R4. stop and return $\langle value, \tau_q^G \rangle$.

Arvon hyväksyminen

S1. if τ_q mennessä vastaanotettu r erillistä viestiä $(p_i, \langle G, m'' \rangle, \tau_i, i)$
 (missä $\tau_q \leq \tau_q^G + (2r + 1) \cdot \Phi$ ja $\forall i, j : (1 \leq i \leq r) \wedge (p_i \neq p_j \neq G)$) then
S2. $value := \langle G, m'' \rangle$;
S3. MSGD-BROADCAST($q, value, r + 1$);
S4. stop and return $\langle value, \tau_q^G \rangle$.

Arvon hyväksyminen

T1. if τ_q mennessä $|broadcasters| < r - 1$ (missä $\tau_q > \tau_q^G + (2r + 1) \cdot \Phi$) then
T2. stop and return $\langle \perp, \tau_q^G \rangle$.

Lopetus ilman arvoa

U1. if $\tau_q > \tau_q^G + (2f + 3) \cdot \Phi$ then
U2. stop and return $\langle \perp, \tau_q^G \rangle$.

Lopetus ilman arvoa

- Hylkääminen

- Joko päädytään tyhjään arvoon \perp
- ... tai ei hyväksytä mitään arvoa

Todistukset

Todistukset

... sivuutetaan triviaaleina

... jätetään harjoitustehtäväksi

Kysyttävää?

Hyvää Joulua!