

Óraszinkronizáció szenzorhálózatokban

Összeállította:
Orosz György
BME-MIT

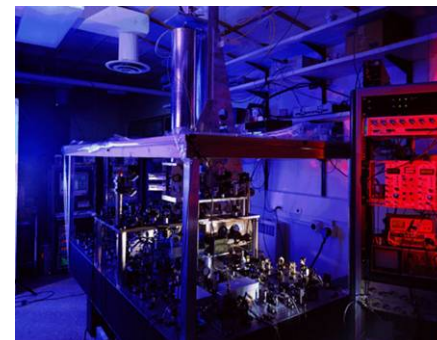
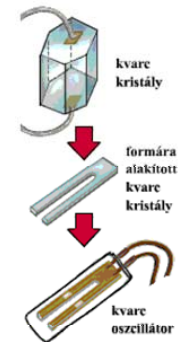
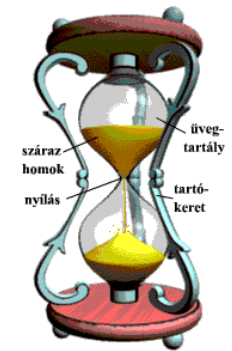
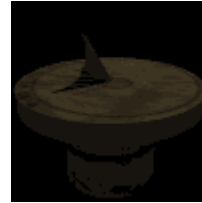
2011. 10. 12.

- Referenciák:

- [1] K. Kömer, P. Blum, L. Meier, “Time Synchronization and Calibration in Wireless Sensor Networks,” in *Handbook of Sensor Networks: Algorithms and Architectures*, Ivan Stojmenović, Ed., Wiley, 2005.
- [2] D. L. Mills, “Internet Time Synchronization: The Network Time Protocol,” *IEEE Transactions on Communications*, COM 39, No. 10, Oct. 1991., pp. 1482–1493.
- [3] S. Ganeriwal, R. Kumar, M. B. Srivastava, “Timing-sync Protocol for Sensor Networks,” *First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, California, USA, Nov. 2003.
- [4] J. Elson, L. Girod, D. Estrin, “Fine-grained Network Time Synchronization Using Reference Broadcasts,” *Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA. Dec. 2002.
- [5] M. Maroti, B. Kusy, Gy. Simon, A. Ledeczi, “The Flooding Time Synchronization Protocol,” Technical Report ISIS-04-501, Institute for 96 Software Integrated Systems, Vanderbilt University, Nashville Tennessee, 2004.
- [6] K. Römer, “Time Synchronization in Ad-Hoc Networks,” *ACM Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc 01)*, Oct. 2001.

Időmérés

- Feladatok összehangolása
- Jelenségek mérése
 - Idő mérés
 - Időtartam mérés
- Ókortól (középkorban is):
 - Természeti jelenségek
 - Napóra
 - Vízóra
 - Homokóra
- Mechanikus óra (középkor)
- Kvarcóra (~1sec/hét)
- Atomóra (~0,2nsec/nap)



Időalapok és szabványok

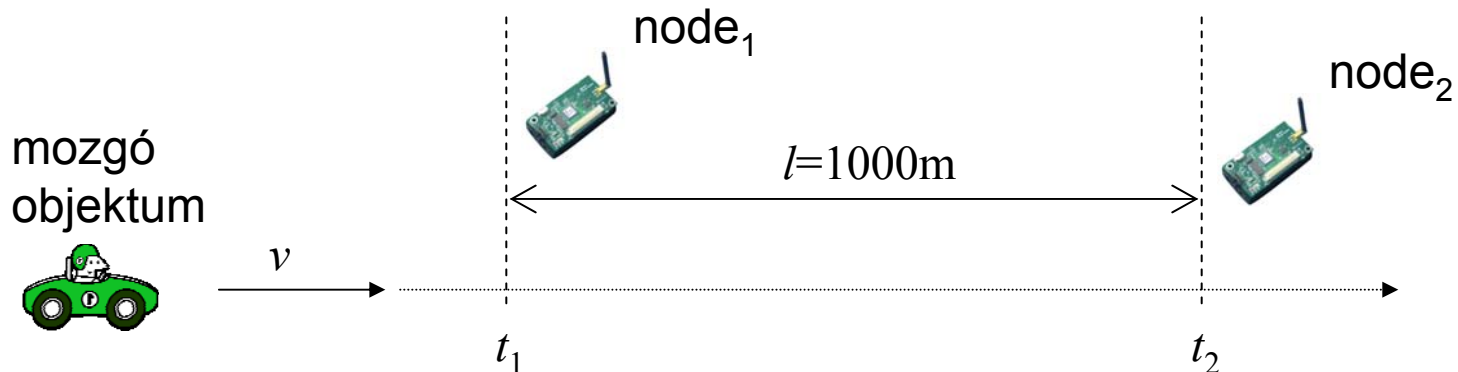
- 1 másodperc a 133-as tömegszámú cézium izotóp úgynevezett hiperfinom átmeneti rezgési periódus-idejének 9.192.613.770-szerese.
- International Atomic Time (TAI)
 - 300 atomórához szinkronizált időalap
 - Nem kapcsolódik a Föld keringési idejéhez
- Universal Time (UT)
 - Föld keringési idejéhez szinkronizált
 - GMT modern megfelelője
- Coordinated Universal Time (UTC)
 - Föld keringési idejét veszi 1 napnak de az időalapot pontosan 1sec-nek veszi: UT-hez képest max. $\pm 0,9$ s
 - Szökőmásodpercek beiktatása
 - Unix time: 1970. január 1. óta eltelt másodperc (átfordulás: 2036)
 - GPS-ek szinkronizálása
- Lehetséges források
 - GPS (UTC-hez szükséges információ)
 - Rádiós órareferenciák, pl.: DCF77, 77.5kHz, Frankfurt

Időalap jellemzése

- Általános időalap: kvarcoszcillátor
 - Olcsó
 - Viszonylag pontos és stabil
 - Kisméretű
- Pontosság
 - Rövid- / hosszú távú stabilitás
 - Pillanatnyi frekvenciát befolyásolhatja (rövid távú stabilitás)
 - Hőmérséklet
 - Rezgés
 - Mágneses hatás
 - Tápfeszültség
 - Hosszú távú stabilitás függ
 - Gyártási tolerancia
 - Öregedés
 - Ár ↔ pontosság
 - Fogyasztás ↔ pontosság (pl. fűtött / termosztátos kvarc)
 - Nagy pontosság: a szinkronizálás gyakoriságát nyújtja, de nem szünteti meg a szükségességét

Példa

- Két egység sebességet mér



Órák indításának időpontja: $t_0 = 0$ (egyszerre indulnak)

Órák hibája: $\varepsilon = 10\text{ppm}$. Worst case: $node_1$ siet (+10ppm), $node_2$ késik (-10ppm)

Objektum sebessége: $72\text{km/h} = 20\text{m/s} \rightarrow 50\text{sec}$ alatt teszi meg az l távot

Objektum érkezése $\tau = 1$ nappal (86400sec) az órák indítása után

$$\text{Ekkor: } t_1 = (1 + \varepsilon) 86400\text{sec} = 86400.8640\text{sec}$$

$$t_2 = (1 - \varepsilon) 86450\text{sec} = 86449.1355\text{sec}$$

$$\Delta t = (t_2 - t_1) = 48.2715\text{sec}$$

$$v = l / \Delta t = 20.716\text{ m/s} = 74.58\text{ km/s (hiba: } 2.58\text{km/h} = 3.6\% \text{ hiba)}$$

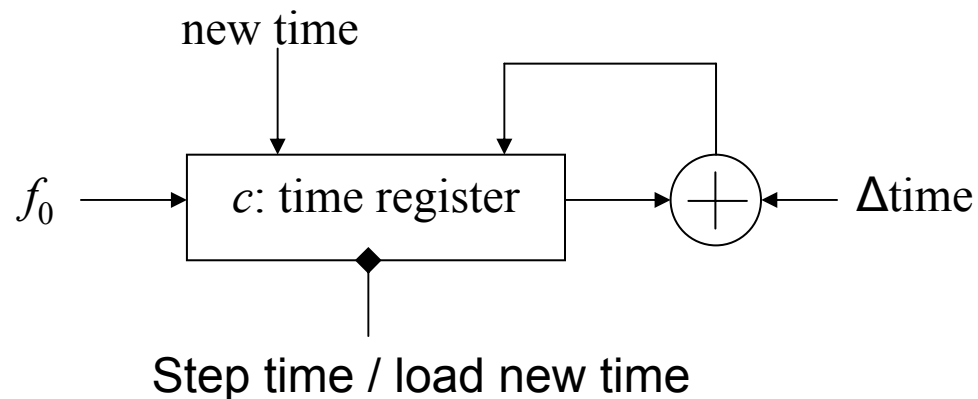
Szenzorhálózatok szinkronizációs sajátosságai

Szinkronizáció új probléma? Nem, de más szempontok.

- **Szenzorhálózatok szinkronizációja**
- Ad-hoc felépítés
- Erőforrás korlát
 - Energia
 - Számítás
 - Memória
 - Kommunikáció
- Eltérő követelmények
- Pontosság tág korlátok között (μsec - sec)
- Nincs lehetőség emberi felügyeletre
- **Hagyományos rendszerek (Internet alapú, pl. NTP [2])**
- Fix infrastruktúra (állandó szerverek).
- Nincs erőforrás probléma
 - Folyamatosan futtat
 - Bármikor jöhet kérés
 - Hálózat folyamatosan használható
- Általános szolgáltatás mindenkinek.
- Maximális pontosság ($\sim 100 \mu\text{s}$) biztosítása mindenkinek.
- Emberi felügyelet.

Óramodell

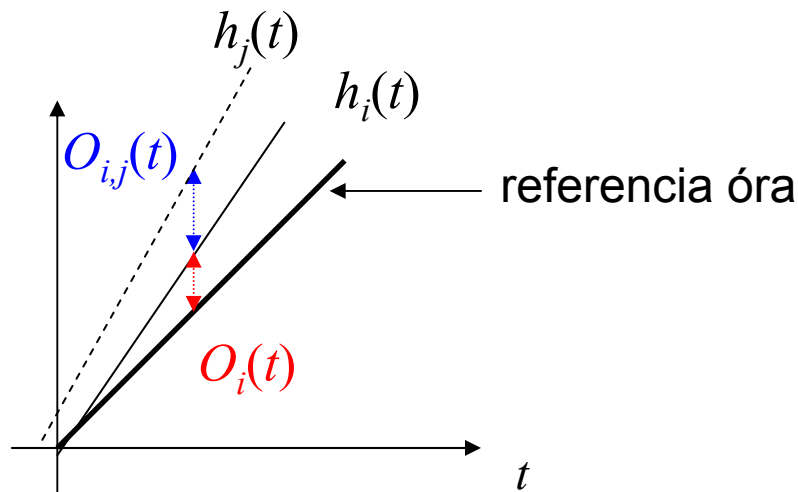
- Időmérés: diszkrét felbontás



- Megvalósítás számlálóval
- f_0 : időalapként szolgáló frekvencia
- Idő léptethető vagy felülírható
 - Óra léptetése (step)
 - Óra beállítása (load)
- Δtime is változtatható paraméter

Óramodell

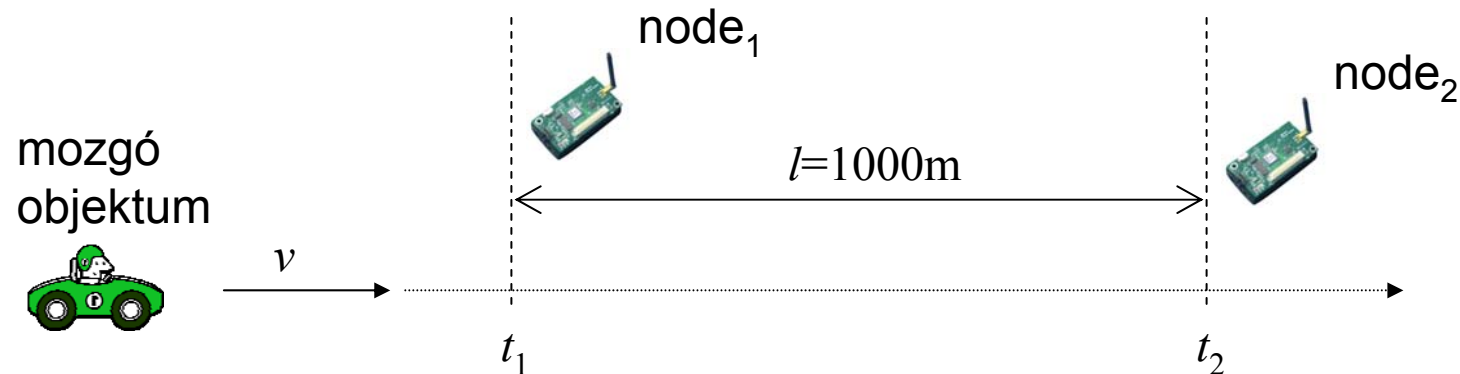
- t : referencia idő, melyet a valódi időként fogadunk el
- N_i : i . node-ot (hálózati elemet) jelöli
- $h_i(t)$: az N_i node lokális ideje t referenciaidőben
- $t^E = t(E)$: az E esemény időpontja
- $h_i^E = h_i(E)$: az E esemény időpontja az N_i node lokális ideje alapján
- $O_{ij}(t) = [h_i(t) - h_j(t)]$, ofszet: lokális órák eltérése egymástól
- $O_i(t) = [h_i(t) - t]$, ofszet: lokális óra eltérése a referenciától
- $c(h_i(t))$: az N_i node órája által mutatott érték t -ben



Óramodell

- Óra gyorsasága: $f_i(t) = dh_i(t)/dt$, referencia óra esetén: $f_i(t) = 1$
- Drift: $\rho_i(t) = dO_i(t)/dt = d[h_i(t) - t]/dt = f_i(t) - 1$
 - Távolodás sebessége a referenciától
 - Ofszet számítható: $O_{i,j}(t) = O_{i,j}(t_0) + \int_{t_0}^t \rho_{i,j}(\tau) d\tau$
 - Ha a drift állandó, akkor $O_{i,j}(t) = O_{i,j}(t_0) + \rho_{i,j}(t - t_0)$
- Referencia óra: $h_{ref}(t) = t$
- Pontos óra: $\rho_i(t) = 0$ tehát $f_i(t) = f_{ref} = 1$
 - Ofszetre nincsen megkötés
- Helyes óra t_0 -ban: $h(t_0) = t_0$
- Konstans sebességű óra (constant rate): $f_i(t) = \text{állandó}$
 - Kvarcoszcillátorokra általában igaz, legalább rövid távon
- Korlátozott driftű (bounded drift) óra: $|\rho_i(t)| < \rho_{\max}$
 - ρ_{\max} : gyártó által megadott paraméter (pl. kvarc: 10ppm...100ppm)
 - ΔT idő alatt ρ_{\max} abszolút értékű drift esetén a referenciaórától való elcsúszás: $|\Delta\tau| \leq \rho_{\max} \cdot \Delta T$, ahol $|\Delta\tau| = |\Delta[h(t) - t]|$
- Korlátozott driftváltozású óra: milyen gyakran kell ρ_i -t számítani

Példa az óramodell alkalmazására



- Referenciaidők az objektum áthaladásakor:
 - $t_1=86400\text{s}$
 - $t_2=86450\text{s}$
- Lokális idők az objektum áthaladásakor:
 - $h_1(t_1) = 86400.8640\text{s}$
 - $h_2(t_2) = 86449.1355\text{s}$
- Az órák paraméterei
 - Drift: $\rho_1(t)=10\cdot 10^{-6}$ és $\rho_2(t)=-10\cdot 10^{-6} \rightarrow \rho_{\max}=10\cdot 10^{-6}$
 - Gyorsaság: $f_1(t)=1.00001$ és $f_2(t)=0.99999$

Szinkronizáció céljai

- A hálózatban található összes óra értéke minél kisebb intervallumon belül legyen
 - $\max(h_i(t) - h_j(t)) \leq M$
- A hálózatban található összes óra értéke minél kisebb intervallumon belül legyen a referenciaidőhöz képest
 - $\max(h_i(t) - t) \leq M$
- Konzisztens becslést adjon az események sorrendiségére:
 - $E1$ és $E2$ események sorrendisége a szenzorok által mért időértékek (h_i^{E1} és h_j^{E2}) alapján egyértelműen meghatározható, tehát:
 $t^{E1} < t^{E2}$ akkor és csak akkor ha $h_i^{E1} < h_j^{E2}$
 - Ha nem lehetséges a mérések alapján a sorrendiséget meghatározni, akkor ezt is meg lehet állapítani: ne legyen téves mérés, legyen egy „sorrend nem meghatározható” állapot is.
- Lehetőleg kis kommunikációs overhead
- Szinkronizáció mint kalibrációs probléma

Időméréshez kötődő feladatok szenzorhálózatokban

- Nincs „a szinkronizációs protokoll”
 - Különböző célok (hőmérséklet, fényerősség vagy hang érzékelése)
 - Különböző eszközök és erőforrások (μ C típus és perifériakészlet, GPS, rádió)
- Hálózati működés időzítése
 - pl.: 1ms/100sec csúszás 10ppm hibánál
- Események kronológiai sorrendje
 - Elosztott érzékelés, központi feldolgozás
 - Ok okozati viszony meghatározása
 - Pl.: először a konyhában majd a folyosón észleltek mozgást
- Állapotkonzisztencia
 - Milyen megfigyelések, mérések tartoznak össze
 - Üzenet felismerése ugyanarról az eseményről: redundancia csökkentés
 - Régi üzenetek eldobása (erőforrás takarékoság)

Időméréshez kötődő feladatok szenzorhálózatokban

- Feladatok koordinációja, időzítése: kapcsolat a felügyeleti szervvel, felhasználóval
 - Pl.: délelőtt az 1. teremben, délután a 2. teremben mérjük a hőmérsékletet
 - Pl.: 18⁰⁰-kor kezdjük a megfigyelést
 - Pl.: milyen esemény történt a lakásban az elmúlt 1 órában
- Időmérésre visszavezethető fizikai mennyiségek mérése
 - Sebesség
 - Lokalizáció (hanghullámok érkezési időkülönbsége)
- Idővariáns jelenség figyelése
 - Változás időállandója kisebb a kommunikációs késleltetésnél
 - Pl. gyorsan mozgó tárgy pozíciójának jelentése a hálózaton belül

Szinkronizáció paramétere

- Maximális offset:
 - $\max(h_i(t) - h_j(t))$
 - $\max(h_i(t) - t)$
 - Precíz időszinkronizáció (pl. lokalizáció) ↔ Kis erőforrás
- Frekvenciahiba / drift kompenzációja: milyen gyakran kell újraszinkronizálni
- Skálázhatóság és hangolhatóság
 - Pl.: pontosság / kommunikáció-energiafogyasztás
 - Paraméterek akár futás közben is hangolhatóak
 - Átültethető legyen más struktúrájú hálózatra is
- Beállási idő
 - Offsetre és driftre vonatkozóan
 - Függ a pontosságtól
 - Pl. 10usec pontosság 10sec, 1us pontosság 2min
- Élettartam
 - Meddig tart a szinkronizált állapot
 - Alkalmazásfüggő, pl.:
 - Hőmérséklet mérése folyamatosan
 - Hőmérséklet mérése amikor bekapcsol a fűtés
 - Energiatakarékosságot segíti

Szinkronizáció paramétere

- Hatókör
 - Fizikai távolság (pl. épületen belül)
 - Logikai távolság
 - Single-hop / multi-hop (hibaterjedés)
 - Multihop rendszerek: átjárókban mért távolság
- Kommunikáció igény
 - Hány üzenetváltás kell (node páronként, csoportonként)
 - Üzenetváltás gyakorisága
 - Unicast (pont-pont)
 - Multicast (több fogadó)
- Energiafogyasztás
 - Kommunikációval összefügg
 - Mennyit kell a csatornát figyelni, üzenetet küldeni, ébren lenni...
- Robosztusság
 - Hogyan tolerálja a hibás egységeket?
 - Hogyan tolerálja az új egységek belépését?
 - Hogyan tolerálja a struktúra változását?
- Speciális HW / SW igény
 - Pl.: GPS, precíziós oszcillátor, HW időbélyegezés
 - Költségek

Szinkronizáció paramétere

- Érvényesség
 - A hatókörön belül milyen egységekre teljesül a szinkronizáció
 - Pl.: adott szenzorral rendelkező node, adott címtartományban vagy geometriai pozícióban található node
- Szimmetrikus / Aszimmetrikus kommunikáció
 - Gond az aszimmetrikusnál: round-trip delay nem mérhető (lásd később)
 - Pl.: bázisállomás: nagyobb adóteljesítmény megengedett (aszim.)
- Egymásnak ellentmondó paraméterek:
 - Pontos szinkronizáció ↔ gyors beállítás
 - Gyors beállítás ↔ alacsony fogyasztás (pl. több kommunikáció miatt)
 - Nagyobb hatókör ↔ pontos szinkronizáció
 - Pontosság ↔ alacsony kommunikáció igény
- Nem feltétlenül független paraméterek
 - Bizonyos hatókörön (csoporton) belül pontos de egy nagyobb hálózatot tekintve kevésbé pontos
 - Pl.: a hálózatot csoportokra osztjuk. Egy objektum mozgását érzékeljük. Az egyes csoportokon belüli node-ok mérik a sebességet: pontos szinkronizáció kell. Csoportok között elég pontatlanabb, ha pl. csak a mozgás iránya fontos (melyik területre lépett be előbb).

Szinkronizáció típusai

- Implicit/explicit

- Implicit

- Az adattovábbítást használják szinkronizációra
 - Előny: nem jelent nagy extra terhet
 - Hátrány: ritka adatcserénél pontosság romlik

- Explicit

- Önálló csomagok szinkronizációra
 - Előny: nem igényli a folyamatos adatáramlást
 - Hátrány: extra teher a hálózat számára (csatornafoglaltság, energia)

- Külső/ Belső szinkronizáció

- Külső szinkronizáció (globális)

- Globális időreferenciához szinkronizáltak (pl. UTC-hez GPS-szel)
 - $h_i(t) = t$
 - Feladatok összehangolása (pl. 5-kor találkozunk)
 - A hálózat élettartamán túl is érvényes adatnak kell lennie
 - Adatbázisokban események rögzítése

- Belső szinkronizáció (lokális)

- Egységek egymáshoz képest szinkronizáltak
 - Nincs információ valamilyen globális időreferenciáról (pl.: UTC)
 - Pl.: üzenetküldés összehangolása

Szinkronizáció típusai

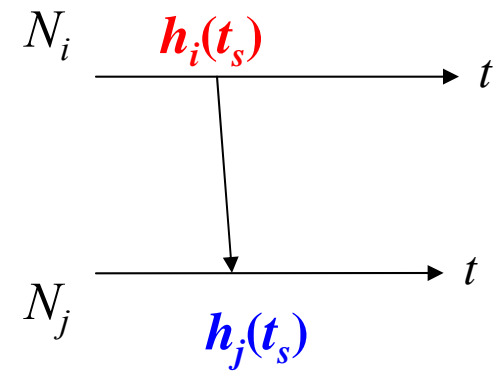
- Abszolút / Relatív szinkronizáció
 - Abszolút szinkronizáció
 - $h_i(t) = h_j(t)$
 - Minden óra ugyanazt az értéket mutatja (nulla offset)
 - Események sorrendjének meghatározása
 - pl.: Hálózat bekapcsolása óta eltelt idő
 - Relatív szinkronizáció
 - Pontos időalap
 - $f_i(t) = 1$
 - Fizikai mennyiségek mérése (pl. sebesség), ahol fontos a pontos érték is.
 - Megegyező időalap
 - $f_i(t) = f_j(t)$, de $f_i(t) = 1$ nem feltétlenül szükséges
 - Együtműködés a hálózatban: pl. egyszerre „ébredjenek” T idő múlva
 - pl.: Minden node valamilyen időmérésre visszavezethető jelenséget mér és azokat kell összehasonlítani (minden node méri egy autó sebességét)
 - Megfigyelés ugyanolyan gyakorisággal

Szinkronizáció típusai

- Folytonos / igény szerinti
 - Folytonos
 - Maximális pontossággal a hálózat teljes életciklusa alatt
 - Folytonos működés: felesleges energiafogyasztás ☹
 - Igény szerinti
 - A működés adott időszakában aktív
 - Pl. rövid időtartamú mérések:
 - Amikor egy jármű közeledik, nagyobb pontosságot igényel, mert a sebességét kell mérni, ha távol van, elég ha a jelenlétét érzékelik.
 - Autós példa: az, hogy mikor érte el a sebességmérő pontokat elég perc felbontással megadni: kevésbé pontos szinkronizáció is megfelelő. Sebességmérés: ekkor már pontos szinkronizáció kell
 - Event triggered (eseményvezérelt)
 - Post-facto szinkronizáció
 - Time triggered
 - Előre tervezett
 - Megfigyelés előtt durva szinkronizáció (el ne tévesszék az időpontot)
 - Megfigyelés közelében finomít
- Valós / virtuális
 - Fizikai / időbélyeg konverzió
 - Lásd később

Órák eltérésének mérése

- Szinkronizációs referenciapontok gyűjtése
 - Órák eltérésének meghatározásához
 - Egyazon időponthoz tartozó időbélyegek különböző egységeken.
 - Lehet pl. ismert időkülönbséggel jellemezhető időpontok gyűjtésével is, de végeredményben ezeket is áttranszformáljuk közös időponttá.
 - N_i és N_j node-ok esetén szinkronizációs referenciapont a $\{h_i(t_s), h_j(t_s)\}$ időpár: amikor az N_i node-on az idő $h_i(t_s)$, akkor az N_j node-on az idő $h_j(t_s)$
- Szinkronizációs pontok gyűjtési módszere
 - Rádiós üzenetek segítségével
 - Mindennapi példák:
 - Óraegyeztetés
 - „Nálad mennyi az idő? Nálam 17:00”
 - „Nálam 17:01”
 - Tehát az ofszet 1perc
 - Több mérés alapján a drift is számítható
 - Déli harangszó: mindenki nagyjából egyszerre hallja



Szinkronizáció megvalósítása

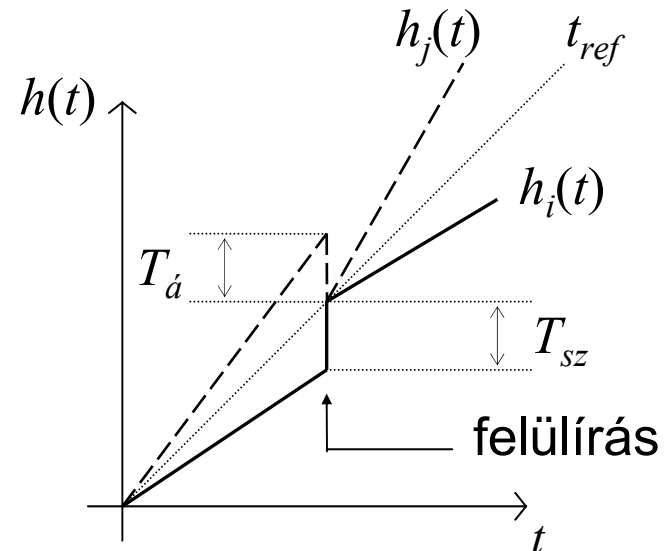
Fizikai szinkronizáció

- Fizikai szinkronizáció óraátállítással

N_i : referencia időhöz viszonyítva lassú node. Lokális ideje: $h_i(t)$

N_j : referencia időhöz viszonyítva gyors node. Lokális ideje: $h_j(t)$

- Óra értékének felülírása
- Folyamatos ofszetkorrekció
- Előny
 - Egyszerű megvalósítás
 - Elvileg egyből a valós időre áll be
- Hátrány
 - A felülírás késleltetése nem ismert minden esetben
 - Kimaradnak/ismétlődnek időpontok
 - Pl. UTC-nél szökőmásodperc
 - Pl.: téli/nyári óraátállítás



T_a : átlapolódás az időben

T_{sz} : szakadás az időben

Szinkronizáció megvalósítása

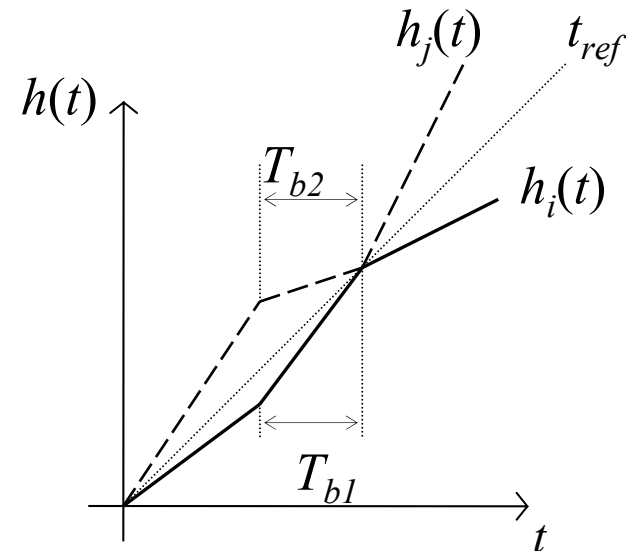
Fizikai szinkronizáció

- Fizikai szinkronizáció óra sebességének hangolásával

N_i : referencia időhöz viszonyítva lassú node. Lokális ideje: $h_i(t)$

N_j : referencia időhöz viszonyítva gyors node. Lokális ideje: $h_j(t)$

- Óra sebességének állítása
- Előny
 - Nincsen kimaradt időpont
 - Az óra átírás késleltetése nem számít
 - Korrigálható a sebesség is nem csak az ofszet
- Hátrány
 - Az óra nem egyből veszi fel a helyes értéket
 - Bonyolultabb módszer
- PLL elvhez hasonlít



T_{b1} : szinkronizálódás ideje

T_{b2} : szinkronizálódás ideje

Szinkronizáció megvalósítása

Időbélyeg konverzió

- Virtuális szinkronizáció időbélyeg konverzióval

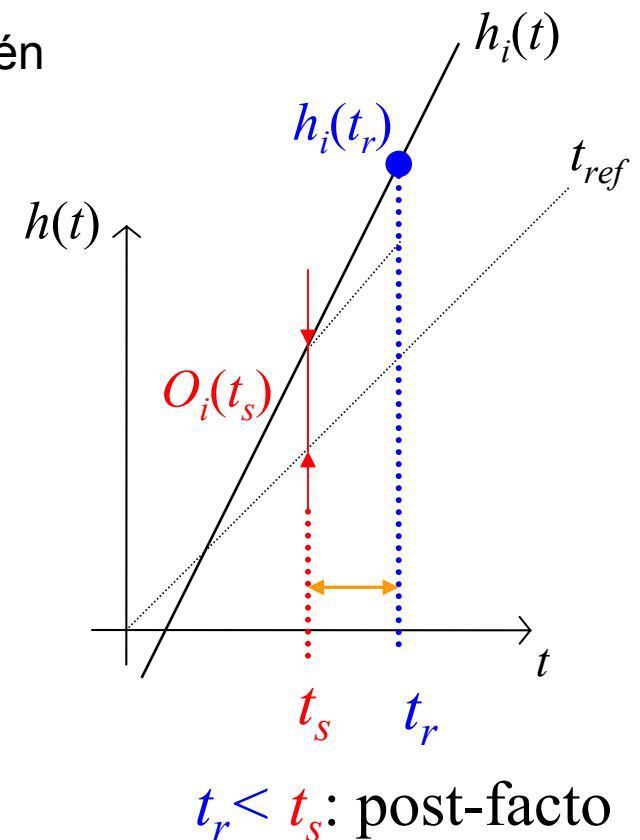
- Nem kell az órát hangolni, hanem a lokális idő alapján becsli a referencia időt
 - Pl. ha tudom, hogy az óram adott hónap elején 5 percet késelt [$O_i(t_s)=5\text{min}$], és tudom, hogy hetente 1 sec-t késik [$\rho_i=1\text{sec/hét}$], akkor bármikor kiszámíthatom a pontos időt.

- Bármilyen időpont konvertálható

- Pl. post-facto szinkronizáció: extrapoláció időben visszafele

- Szinkronizáció menete

- $O_i(t_s)$ ofszet mérése
- ρ_i drift mérése
- Időbélyeg konverzió:
 - $h_i(t_r) = t + O_i(t_s) + (t_r - t_s) \rho_i$
 - Tehát: $t = h_i(t_r) - O_i(t_s) - (t_r - t_s) \rho_i$



Szinkronizáció megvalósítása

Időbélyeg konverzió (Példa)

- Két óra (node₁, node₂) által mutatott időpontokat feljegyeztük a következő időpontokban:
 - 2000. jan. 10. 15:45:00 (node₁) → 2000. jan. 10. 15:47:01 (node₂)
 - 2000. jan. 15. 15:45:00 (node₁) → 2000. jan. 15. 15:47:03 (node₂)
- Node₂ 2000. jan. 29. 18:57:20-kor észlel egy eseményt. Mi ennek az eseménynek az időpontja node₁ órája alapján?
 - Paraméterek (Legyen h_1 a viszonyítási alap):
 - $h_1(t_1) = 2000. \text{ jan. } 10. \text{ 15:45:00 (node}_1 N_1)$
 - $h_2(t_1) = 2000. \text{ jan. } 10. \text{ 15:47:01 (node}_2 N_2)$
 - $h_1(t_2) = 2000. \text{ jan. } 15. \text{ 15:45:00 (node}_1 N_1)$
 - $h_2(t_2) = 2000. \text{ jan. } 15. \text{ 15:47:03 (node}_2 N_2)$
 - $h_2(t_3) = 2000. \text{ jan. } 29. \text{ 18:57:20 (node}_2 N_2)$; Kérdés: $h_2(t_3) = ?$
- **Ofszet:** $O_{2,1}(t_1) = 2\text{min } 1\text{sec}$, $O_{2,1}(t_2) = 2\text{min } 3\text{sec}$
- **Drift becslése:** $\rho_{2,1} \approx [O_{2,1}(t_2) - O_{2,1}(t_1)] / [t_2 - t_1]$
 $\rho_{2,1} \approx [3\text{sec} - 1\text{sec}] / [5\text{nap}] = 2/432000 = 4.63\text{ppm}$
- $t_3 - t_2 \approx 14\text{nap } 3\text{óra } 10\text{perc } 20\text{sec} = 1221017\text{sec}$
- $h_1(t_3) = h_2(t_3) - O_{2,1}(t_2) - \rho_{2,1} (t_3 - t_2) = h_2(t_3) - 2\text{min } 3\text{sec} - 4.63\text{ppm} \cdot 1221017\text{sec}$
- $h_1(t_3) = 2000. \text{ jan. } 29. \text{ 18:55:11.35}$

2min 8.65sec

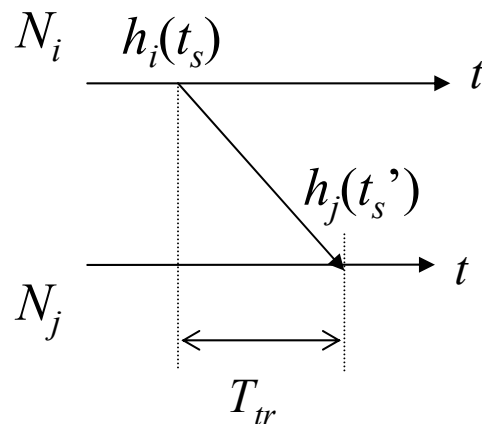
Szinkronizáció megvalósítása

Időbélyeg konverzió (Példa)

- Egyszerűsített megoldás:
 - 5 nap alatt 2sec-et csúszik el az óra (jan. 10-től 15-ig)
 - Jan. 29-én bekövetkező eseménykor kb 14,125nap telik el az utolsó szinkronizációs pont óta (14nap 3h)
 - 14,125 nap alatt $14,125/5*2\text{sec}=5.65\text{sec}$ -et csúszik
 - Tehát a 2min3sec ofszethez 5.65sec adódik, ami 2min 8.65sec ofszetet eredményez
- Vigyázni kell az előjelekre: siet vagy késik (nő az ofszet vagy csökken)

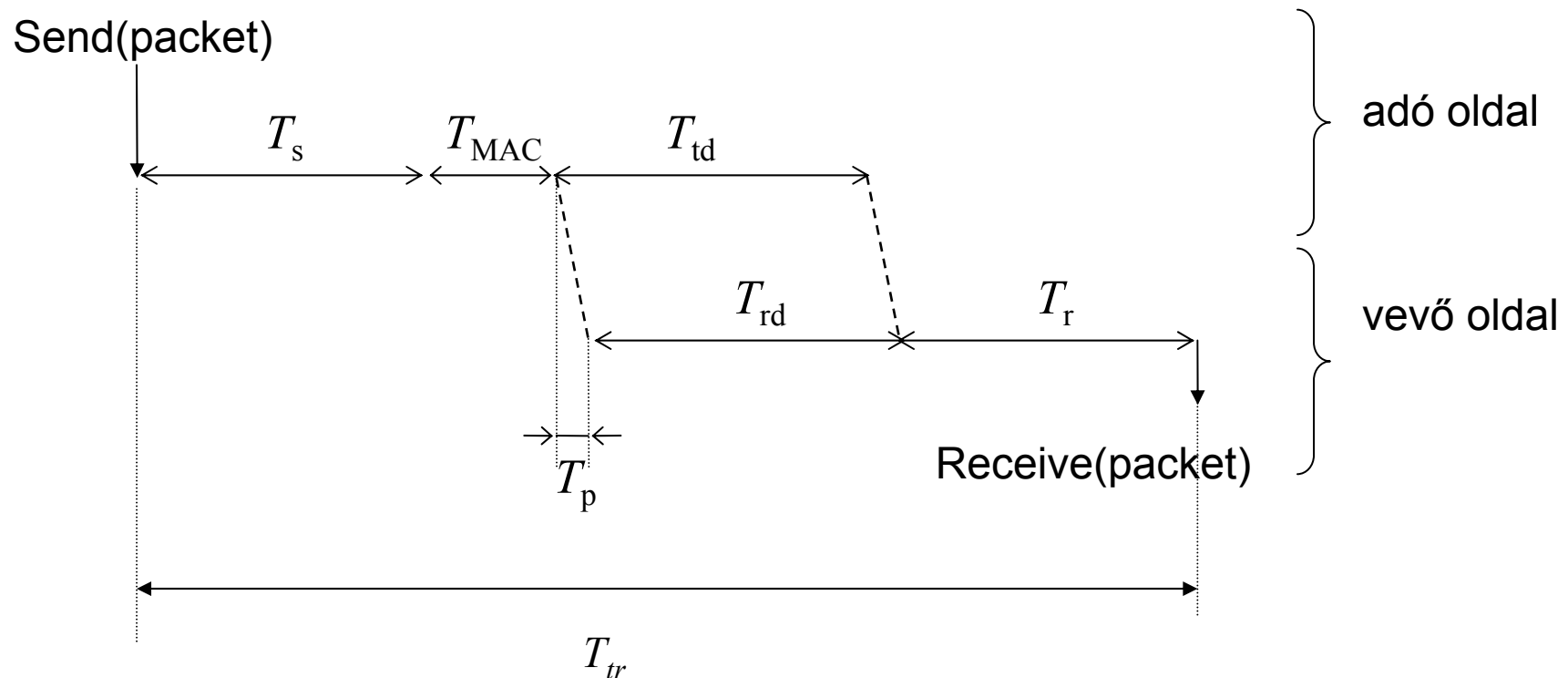
Kommunikációs időzítések

- Szinkronizációs pontok gyűjtése rádiós üzenetküldéssel
- Ideális feltételezés: nulla késleltetés, tehát $t_s = t_s'$
- Valójában: $h_j(t_s') = h_i(t_s + T_{tr})$
 - T_{tr} : üzenetküldés ideje
- T_{tr} nem mindig ismert
 - Mérhető bizonyos esetekben, pl. üzenetváltással
 - Gond: T_{tr} ingadozása (a nem determinisztikus időzítés általában is probléma)



Kommunikációs időzítések

- Késleltetés forrásai
- Általános időzítési diagram
- Sok részlet eszközfüggő



Kommunikációs időzítések

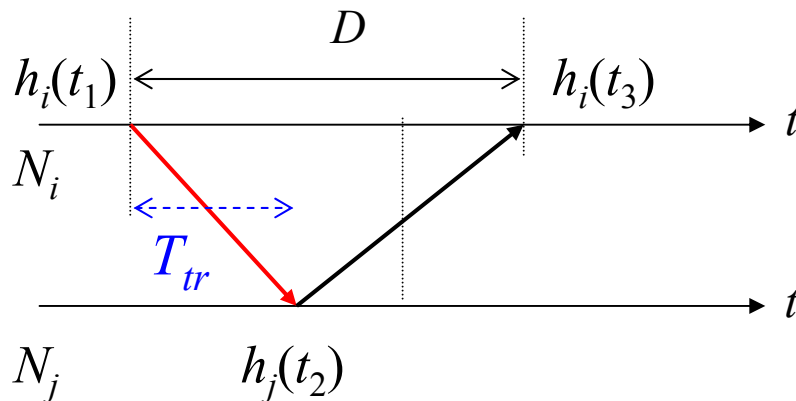
- T_s : Küldés előkészítési idő (Send Time)
 - Csomag elküldését kezdeményező parancs kiadása és az adásra kész állapotba kerülése között eltelt idő.
 - Szoftver késleltetés mértéke függ:
 - Processzor terheltsége ⌚
 - Futó folyamatok száma ⌚
 - Kontextusváltás ideje 🕒
 - Folyamatok prioritása ⌚
 - IT-k kezelésének az ideje (megszakítás vezérelt kommunikáció) 🕒
 - Hardver késleltetés mértéke függ:
 - Kommunikációs IC feléledési ideje 🕒
 - Kommunikációs IC konfigurálása 🕒
 - Kommunikációs adatcsomag betöltése (eszközfüggő) 🕒
 - Ingadozása nagy lehet: 0msec-100msec
- 🕒: fix késleltetés
- ⌚ : változó késleltetés

Kommunikációs időzítések

- T_{MAC} : Közeghozzáférési idő (Medium Access Time)
 - Szabad csatornára várakozik
 - Függ a csatorna terheltségétől
 - Legkevésbé determinisztikus: <500msec
- T_{p} : Terjedési idő (Propagation Time)
 - Determinisztikus
 - $c = 3 \cdot 10^8 \text{m/s}$, $T_{\text{p}} = l / c < 1 \mu\text{sec}$ (ha $l < 300\text{m}$)
- T_{td} : Adatátviteli idő (Data Transmission Time)
 - Értéke függ:
 - Adatátvitel sebessége
 - Küldött csomag mérete
 - Kódolás és moduláció időtartama ($\sim 100 \mu\text{sec}$)
 - Tipikusan < néhány 10ms
 - Determinisztikus
- T_{rd} : Adatfogadás idő (Data Reception Time)
 - Átlapolódik T_{td} -vel, és
 - T_{td} -hez hasonló tulajdonság (kiv.: kódolás, moduláció \rightarrow demoduláció, dekódolás)
- T_{r} : Csomagfeldolgozási idő (Receiving Time)
 - Csomag vételének fizikai befejezésétől az alkalmazás értesítéséig
 - Befolyásoló paraméterek és nagyságrend: u.a. mint T_{s}

Késleltetések kompenzálása

- Determinisztikus részek kompenzálhatóak
- Időbélyeg készítése:
 - IT rutinban: pl. rádiós egység jelzi az adat érkezését
 - IT rutinban + szoftver kiegészítés (pl. átlagolás több IT-re)
 - Speciális HW segítségével: pl. CC2420 SFD jel
- Round-trip késleltetés mérése
 - N_i szinkronizációt kér, N_j válaszol, az üzenetbe beteszi $h_j(t_2)$ -t
 - Szinkronizációs pont becslése:
 - $h_i(t_2) \approx h_i(t_3) - D/2$; $D = h_i(t_3) - h_i(t_1) \rightarrow \{h_i(t_2), h_j(t_2)\}$



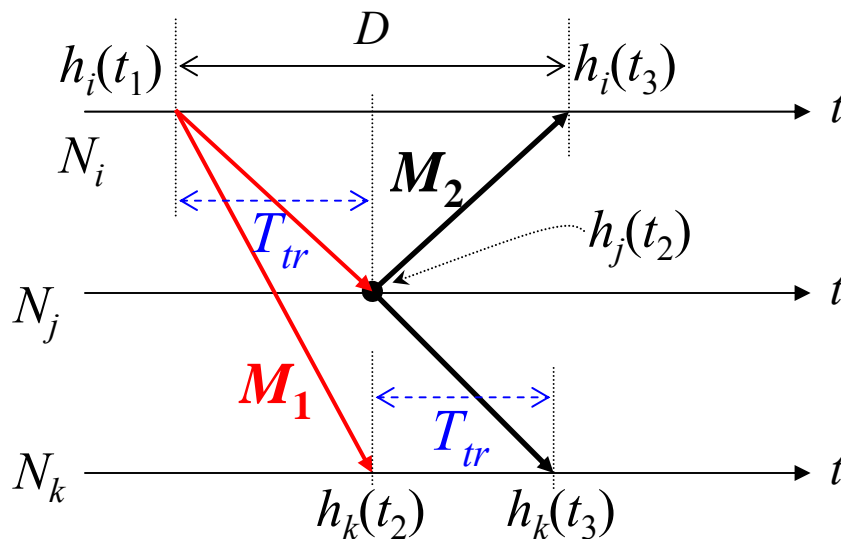
$$0 < T_{tr} < h_i(t_3) - h_i(t_1) = D$$

Becslés (szimmetrikus adatátvitel):

$$T_{tr} \approx \frac{1}{2} [h_i(t_3) - h_i(t_1)]$$

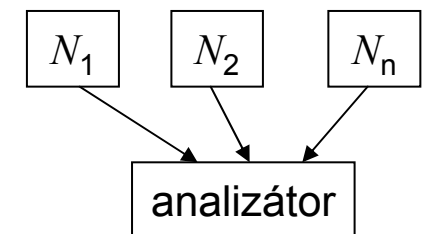
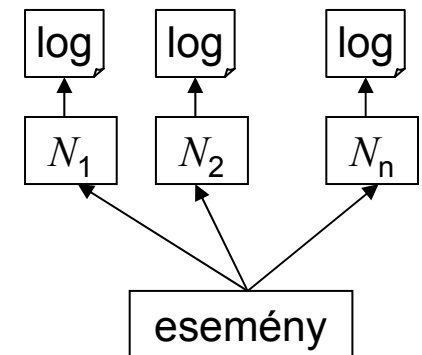
Késleltetések kompenzálása Csökkentett kommunikációs igény

- Referencia node: N_j
- Több node esetén a kommunikációs igény csökkentése: az aktív node-ok szinkronizációs üzenetváltásait használjuk
- N_k figyeli a csatornát és veszi N_i és N_j üzeneteit is (M_1 és M_2 üzenetek).
- N_j az M_2 üzenetben elhelyezi $h_j(t_2)$ -t (az üzenet küldésének időpontja N_j órája alapján).
- Ez az időpont megegyezik az M_1 üzenet érkezési idejével N_k órája alapján.
- Feltételezve, hogy N_j egyből válaszol, N_k becsülni tudja T_{tr} -t: $T_{tr} = h_k(t_3) - h_k(t_2)$
- A becslés alapján szinkronizációs pont generálható: $\{h_k(t_2), h_j(t_2)\}$
- Vegyük észre: N_k node nem küld üzenetet (sajnos a vétel módban is van fogyasztás, viszont nem kell újabb szinkronizációs kört indítani)



Teljesítményparaméterek

- Pontosság mérése
 - Összehasonlíthatóság
 - Órák közötti maximális eltérés
 - Órák közötti átlagos négyzetes eltérés
 - Eltérés a globális időtől
- Mérési alternatívák
 - Az összes node ugyanazon esemény időpontját méri (központi esemény, elosztott mérés)
 - Az összes node egy adott időpontban generál eseményt (elosztott esemény, központi mérés)
 - Egy node-on több algoritmus szimultán, de ezek több hop-on keresztül kapcsolódnak



Pl.: N_1 a referencia, de N_2 , N_3 és N_4 láncon önmagához szinkronizálódik: összehasonlítás

