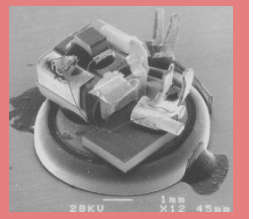
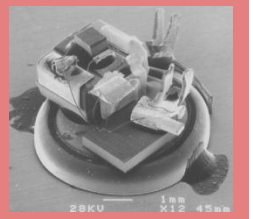


Szenzorhálózatok

Mobilitás (folyt.)
Szállítási és alkalmazási réteg (2011.11.30)

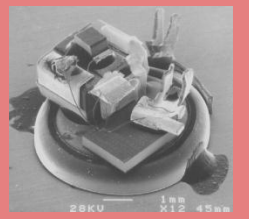
Vidács Attila
Távközlési és Médiainformatikai Tanszék
I.E.325, T:19-25, vidacs@tmit.bme.hu

Miről lesz szó?



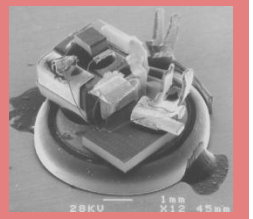
- Mobilitás szenzorhálózatokban
  - Bázisállomás mobilitása
  - Szenzorok mozg(at)ása
- Szállítási és alkalmazási téteg

Mobilitás szenzorhálózatokban (ism.)



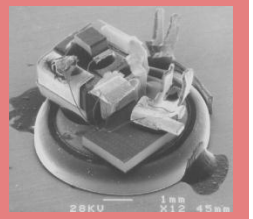
- „Tipikusan” egy szenzorhálózat csomópontjai nem mozognak, de…
  - bizonyos esetekben hasznos lehet a mobilitás, vagy
  - A szenzorok mozgása nem elkerülhető.
- Egy szenzorhálózatban mozoghat…
  - a bázisállomás, és/vagy
  - a szenzorok, és/vagy
  - az „események”, követendő objektumok.
- A mobilitás célja lehet..
  - energiahatékonyság,
  - lefedettség biztosítása,
  - topológia-kontroll,
  - megfigyelés „minőségének” javítása.

Bázisállomás mobilitása (ism.)



- A BS mozg(at)ása lehet…
  - véletlen,
  - predikálható,
  - vezérelt,
  - adaptív.
- Egy BS mozoghat…
  - önerőből, vagy
  - valamilyen hordozó segítségével.

Szenzorok mozgatása

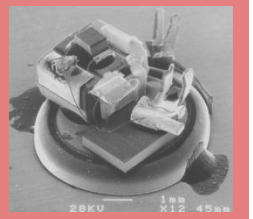


- A BS-sel ellentétben a szenzorok mozgatása kritikus lehet az energiafogyasztás és költség szempontjából!

- A szenzorok mozgása lehet…
  - véletlen, vagy
  - vezérelt.

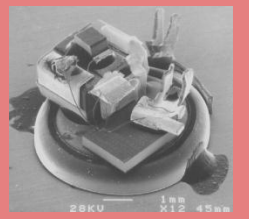
- **Véletlen mozgás** szenzorok esetében tipikus, ha a közeggel ill. környezettel együtt mozognak, passzív módon.
  - Pl. Szenzorok folyókban vagy tengerek felszínén, levegőben, víz alatt, gleccsereken, stb.
  - Pl. Állatokra, járművekre szerelve…

Szenzorok mozgatása



- Szenzorok **vezérelt mozgatása** esetén lehetőség nyílik…
  - a lefedettség növelésére,
  - az összefüggőség biztosítására,
  - az energiafogyasztás kiegyenlítésére.
- Sok esetben a szenzorok mozgatása csak a telepítés után, egyszer történik meg a megfelelő topológia kialakításához.
  - Pl. lefedetlen területek, egyenlőtlen node-sűrűség, kommunikációs „lyukak” kiküszöbölésére (=topológia kontroll).
- Esetenként csak a szenzorok egy (kis) része mozgatható, a többség nem.

Szenzorok mozgatása

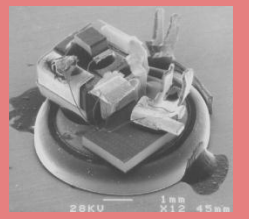


- Eseményvezérelt hálózatok esetében a szenzorok **adaptívan elmozoghatnak** az „érdekesebb” területek felé.

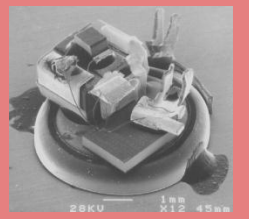
- Szenzormozgatás startégiája lehet…
  - **reaktív**: az eseményekre reagálva mozognak a szenzorok
    - Pl. közelebb húzódnak az érdekesebb területekhez
  - **proaktív**: megpróbálnak optimálisan (egyenletesen) elhelyezkedni.

- Egy lehetséges algoritmus-család a szenzorok mozgatására a **potenciálmezőn alapuló** megoldás
  - Pl. elektrosztatikus mező utánzása, a szenzorháló „magától” szétterül egyenletesen.

„Virtuális” mozgás szenzorhálózatban

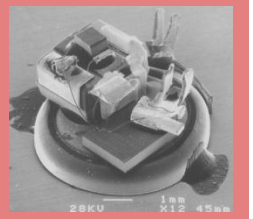


- **Mobil ágensek**: Olyan kódrészletek, amelyek a hálózaton belül mozognak, adott területen hajtódnak végre.
- Egyenlőtlen vagy időben változó (pl. esemény-vezérelt) felajánlott forgalom esetében megéri pl. az adatokat helyben feldolgozni.
- A hálózat átkonfigurálható a BS-ek virtuális áthelyezésével is.



Szállítási és alkalmazási réteg

Szállítási és alkalmazási réteg



Szenzorhálózatok architektúrája

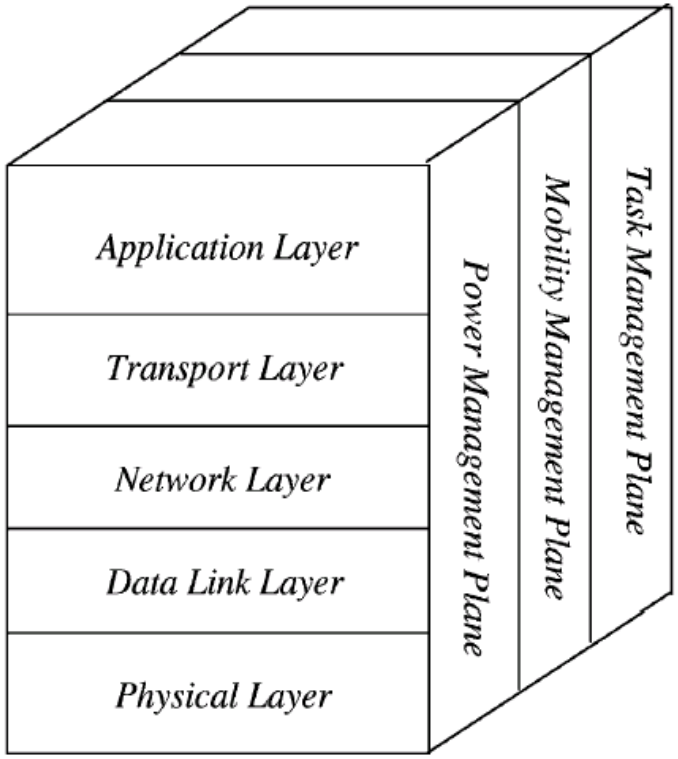
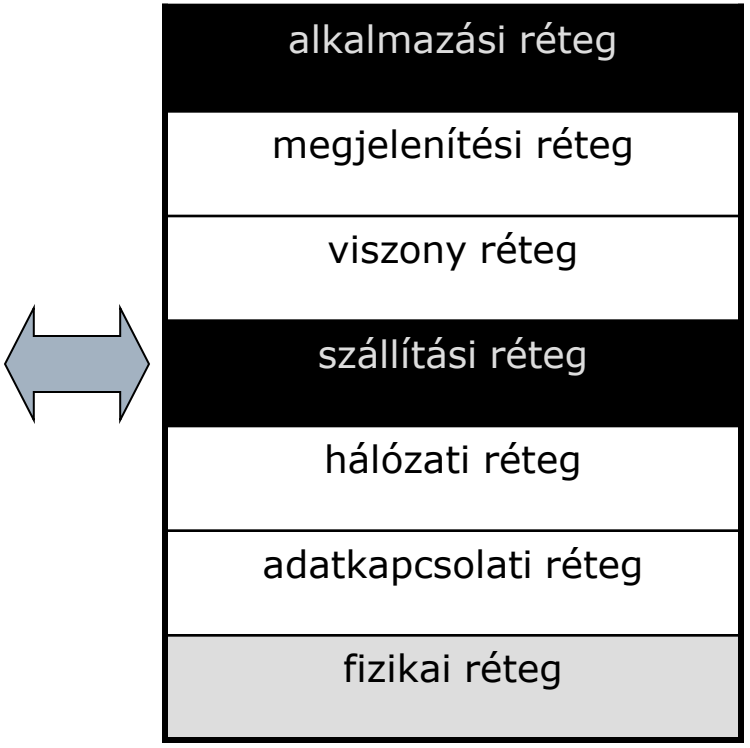
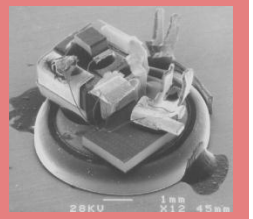


Fig. 3. The sensor networks protocol stack.

ISO OSI



Szállítási réteg



- A szállítási réteg szükséges lehet, ha a szenzorhálózatot az Interneten vagy más külső hálózaton keresztül érjük el.

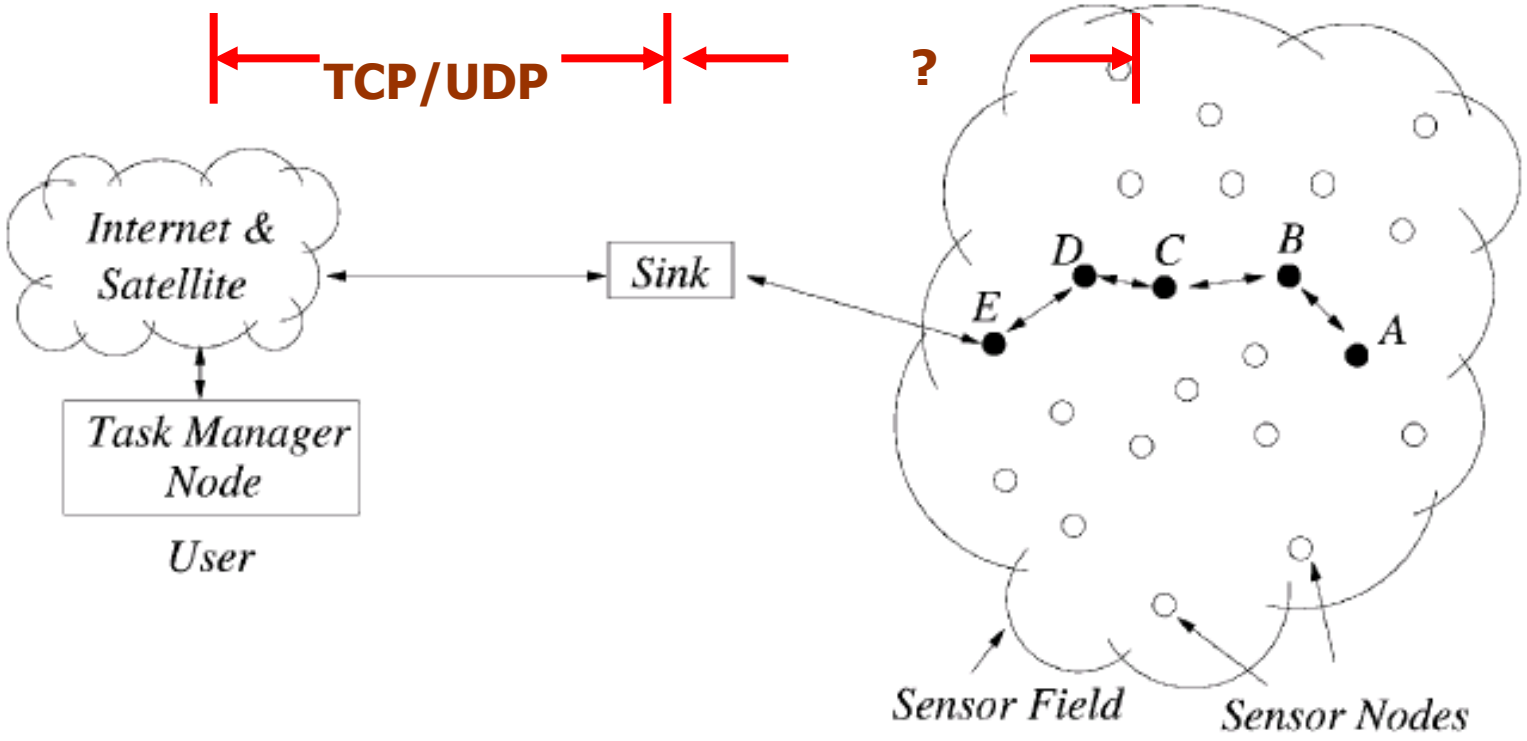
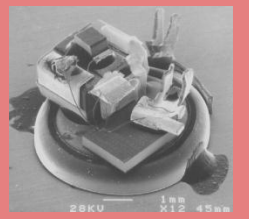


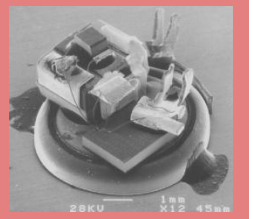
Fig. 2. Sensor nodes scattered in a sensor field.

Szállítási réteg



- A TCP-szerű megoldások nem alkalmazhatóak
  - Nincs mindig egyedi címzés.
  - Nincs elegendő memória a node-okban.
  - Nagy többletköltség a TCP „kézfogás” és az állandó nyugtázás.
  - A torlódásvezérlés szükségtelen, bonyolult.
- Inkább UDP-szerű kommunikáció a kedvező.
- „TCP-splitting”: A TCP kapcsolat a BS-nél befejeződik, a szenzorhálózaton belül datagrammok.
- Jelenleg nincs(!) javaslat vagy kutatási eredmény szenzorhálózatokban is alkalmazható szállítási réteg protokollra!

Alkalmazási réteg

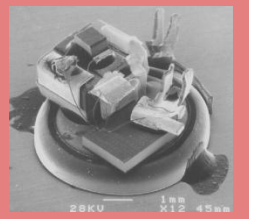


- A javasolt alkalmazások széles skálájával ellentétben az alkalmazási réteg-beli protokollok szenzorhálózatokban még alig kutatott terület!.

- Lehetséges menedzsment protokollok:
  - **SMP – Sensor Management Protocol**
    (Szenzor menedzsment protokoll)
  - **TADAP – Task Assignment and Data Advertisement Protocol**
    (Feladat hozzárendelés és adat hirdetés protokoll)
  - **SQDDP – Sensor Query and Data Dissemination Protocol**
    (Szenzor lekérdezés és adatterjesztés protokoll)

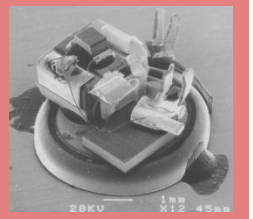
- Megjegyzés: Ezek a protokollok még nem definiáltak!

SMP – Sensor Management Protocol



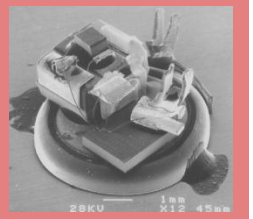
- A hálózat és a szenzorok menedzsmentje fontos feladat.
- Egy alkalmazási réteg-beli menedzsment protokoll feladata az alacsonyabb szintű fizikai és szoftver rétegek átlátszóvá tétele az alkalmazás felé.
- A rendszeradminisztrátorok SMP segítségével tarthatnak interaktív kacsolatot a hálózattal.
- Globális azonosítók hiányában az SMP-nek attribútum- és elhelyezkedés alapú címzést is kell tudnia kezelni.

SMP – Sensor Management Protocol



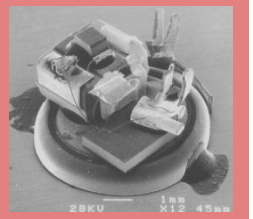
- SMP biztosítja a támogatást az alábbi adminisztratív feladatok ellátására:
  - Adat-aggregáció, attribútum alapú címzés, klaszterképzés szabályainak leírása.
  - Adatcsere helymeghatározó algoritmusokhoz.
  - Szenzorok időszinkronizációja.
  - Szenzorok mobilitásának vezérlése.
  - Szenzorok be- és kikapcsolása (energiamenedzsment).
  - A hálózat konfigurációjának és a szenzorok állapotának lekérdezése, újrakonfigurálás.
  - Hitelsítés, kulcselosztás, biztonsági kérdések.

TADAP – Task Assignment and Data Advertisement



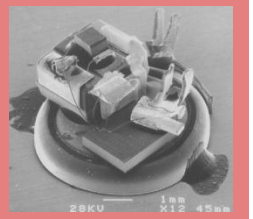
- Fontos feladat a minket érdeklő információk kinyerése a hálózatból.
- Lehetőségek:
  - A felhasználó megmondja a szenzoroknak, vagy a szenzorok egy csoportjának, hogy milyen adatokra van szüksége.
    - Pl. Egy mennyiség értékét, vagy egy jelenség bekövetkeztét
  - A szenzorok hirdetik az elérhető adatokat a felhasználónak, majd a felhasználó a számára érdekes adatokat lekéri.
- Egy alkalmazási réteg-beli feladatkiosztási vagy adathirdetési protokoll nagy segítség lehet az alsóbb rétegek működéséhez (pl. routing).

SQDDP – Sensor Query and Data Dissemination Protocol



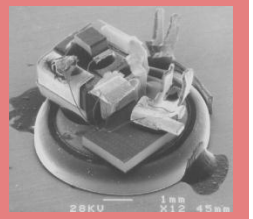
- Az SQDDP egy interfészt nyújt az alkalmazások felé
  - lekérdezések küldéséhez,
  - válaszadáshoz lekérdezésekre,
  - beérkező válaszok összegyűjtéséhez.
- Attribútum- és elhelyezkedés alapú címzést használ
  - Pl: „Hol vannak azok a szenzorok, amelyek 70 foknál magasabb hőmérsékletet érzékelnek?”
  - Pl: „Milyen a hőmérséklet a délnyugati szektorban?”
  - Pl. nem: „Mi a mért érték a #38726-os szenzornál?”
- Az alkalmazástól függően különféle, egyedi SQDDP variánsok hozhatók létre.

SINA, SQTL



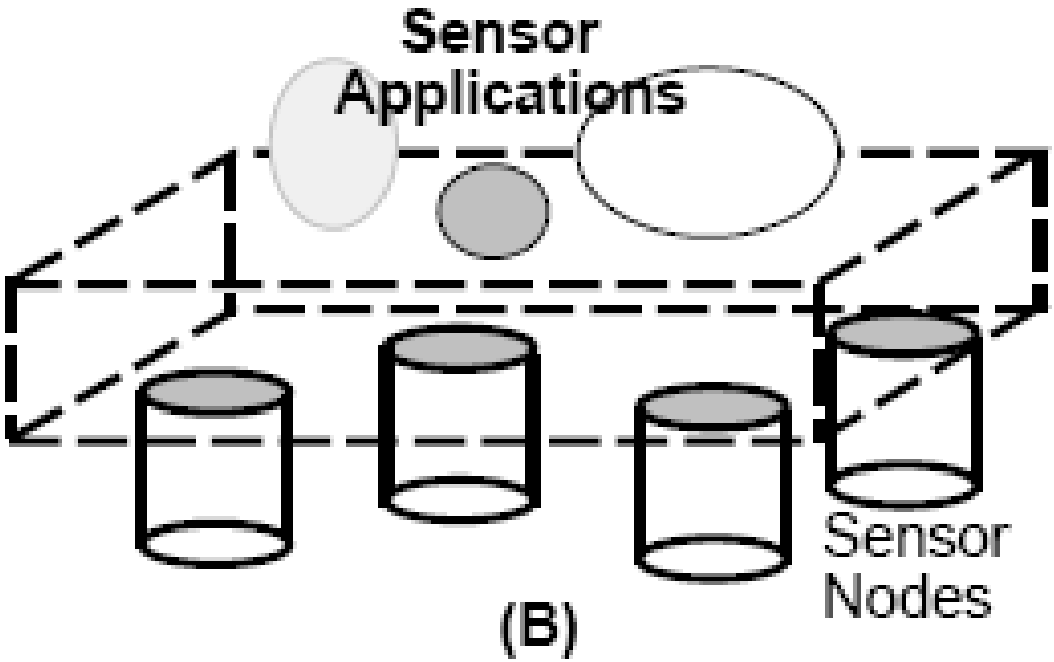
- SINA = Sensor Information Networking Architecture
  - SINA middleware
  - SINA architecture
  - SINA komponensek
  - SINA query
- SQTL = Sensor Query and Tasking Language
  - A SINA architektúra része

SINA és alkalmazások

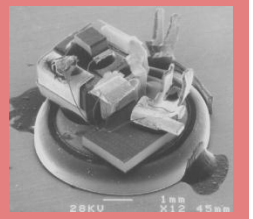


- SINA = Sensor Information Networking Architecture
- Absztrakció: A szenzorhálózatot mint elosztott objektumok gyűjteményét (adatbázis!) tekintjük.

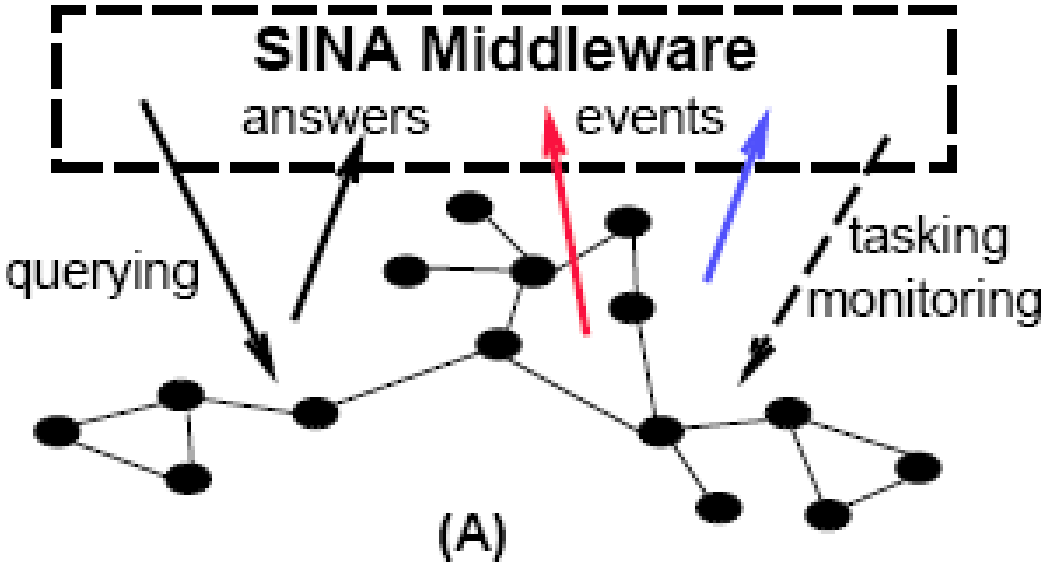
- SINA middleware



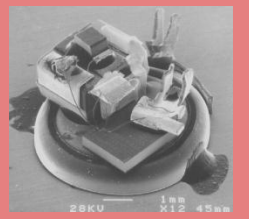
SINA architektúra



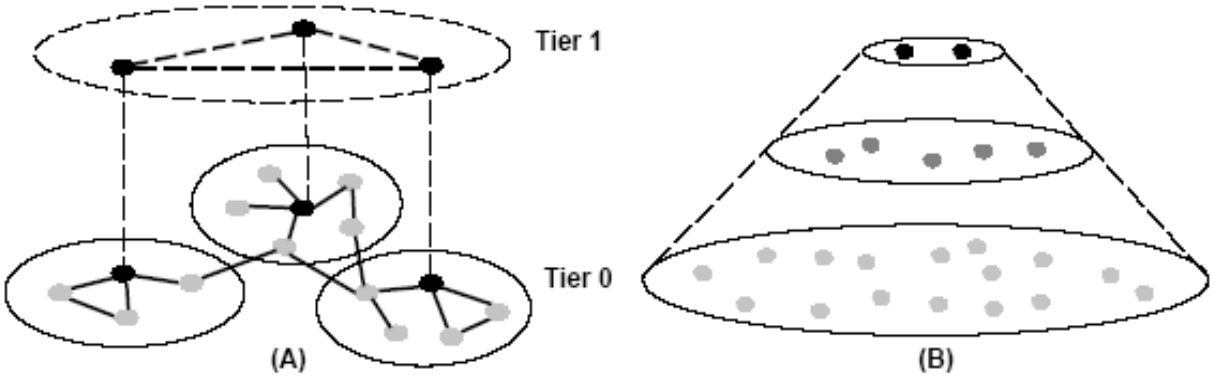
- SINA middleware szerepe: az alkalmazások számára lehetővé teszi
  - lekérdezések (query) indítását,
  - feladatok (task) terjesztését,
  - válaszok és eredmények begyűjtését,
  - hálózat állapotának megfigyelését.



SINA komponensek

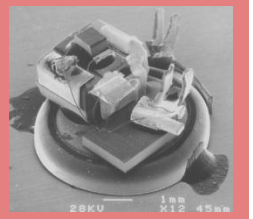


- SINA modulok minden szenzoron futnak.
- SINA architektúra funkcionális komponensei:
  - Hierarchikus klaszterezés a skálázhatósághoz.



  - Attribútum-alapú címzés.
    - Pl: `[type=hőmérséklet, location=É-Ny, hőmérséklet=70]`
  - Helytudatosság.
    - GPS esetleg, más módszerek...

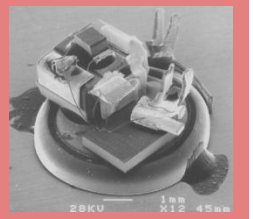
SINA query



Példa: „Hol magasabb a hőmérséklet 70 foknál?”

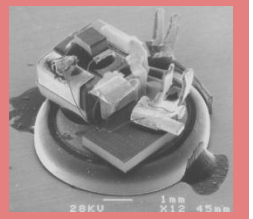
- 1. megoldás:
  - A lekérdezést üzenetszórással eljuttatjuk az összes szenzorhoz.
  - Minden node kiértékeli a kérést, és szükség esetén válaszol.
  - Pontos eredményt kapunk (+)
  - Sok időbe telhet az összes válasz beérkezése (-)
  - Sok energiát fogyaszt.
- 2. megoldás:
  - A klasztervezérlők periódikusan bekérik a mért értékeket, és statisztikákat készítenek belőle (pl. min, max, átlag)
  - A lekérdezést csak a vezérlőknek kell küldeni.
  - Csak a vezérlők értékelik ki és válaszolják meg a kérést.
  - Gyors válaszidő. (+)

SQTL



- SQTL = Sensor Query and Tasking Language
- Az SQTL
  - a SINA architektúra része,
  - programozói interfész az alkalmazás és a SINA middleware között,
  - procedurális szkript-nyelv,
  - felxibilis, kompakt,
  - képes egyszerű deklaratív lekérdezések értelmezésére,
  - képes események kezelésére.
- Pédák SQTL primitívekre:
  - `getTemperature, turnOn, isNeighbor, getPosition, tell, execute`

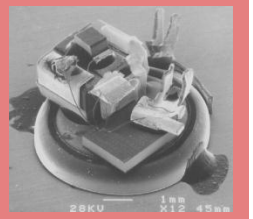
SQTL – Események kezelése



- Háromféle eseménytípus támogatott:
  - ***receive*:** Vett üzenet által kiváltott esemény.
  - ***every***: Periódikusan egy időzítő által kiváltott esemény.
  - ***expire***: Esemény egy időzítő lejártakor.

- Az ***upon*** konstruktor segítségével eseménykezelő blokkokat definiálhatunk.

SQTL üzenetek



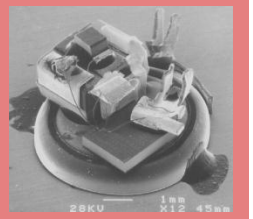
- Egy SQTL üzenet (***message***) egy szkriptet (***script***) tartalmaz.
- Minden node képes értelmezni és végrehajtani egy szkriptet. (SEE – Sensor Execution Environment)
- Hogy egy üzenetet egy adott node-nak (vagy node-ok egy részhalmazának) küldhessünk csak, az üzenetet becsomagoljuk egy ***SQTL Wrapper***-be.
  - Lényegében egy fejrész, amely tartalmazza a feladót, címzettet, az alkalmazást, és esetleges paramétereket.
- Az SQTL Wrapper XML szintaktikával adható meg.

SQTL Wrapper

Table 1. Arguments used by Actions in SQTL wrapper

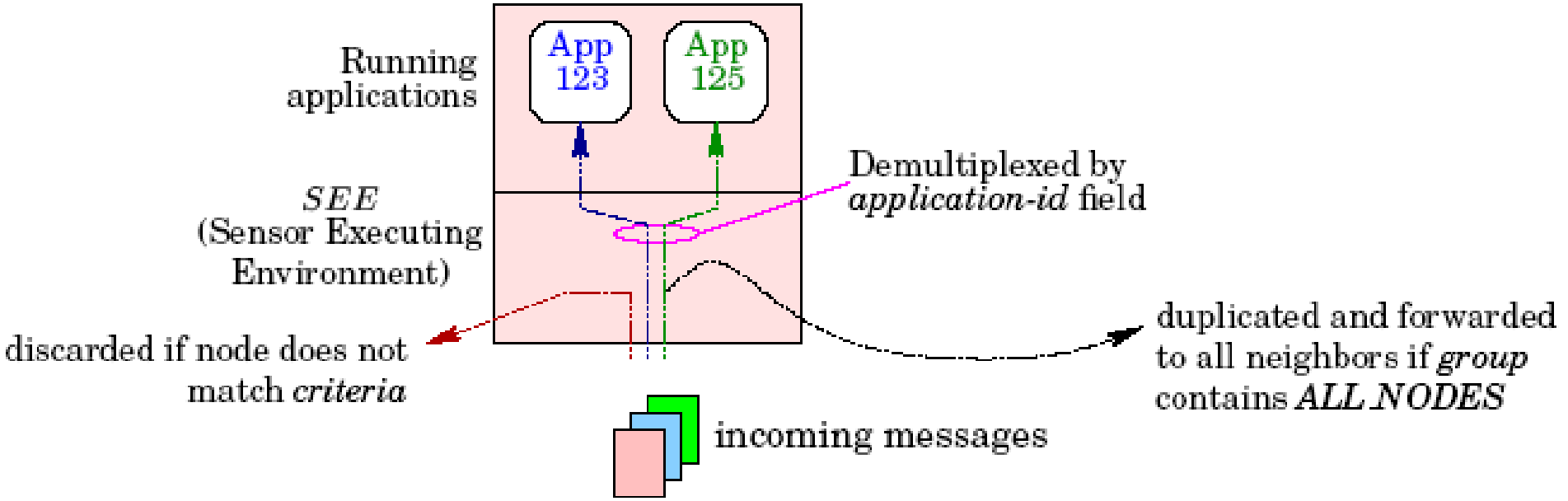
Argument	Meaning
sender	the sender of an SQTL message wrapper
receiver	potential receivers specify by two following subargument
group	subargument of receiver to specify group of receiver, its possible value can be one of ALL_NODES, or NEIGHBORS
criteria	subargument of receiver to specify selection criteria of receivers
application-id	a unique ID for each application in the same sensor network
num-hop	number of hop away from a gateway node
language	specify a language used in content
content	a payload containing a program, a message or return values
with (optional)	tuples of parameters used in the program passed from sender to receiver
parameter	repeatable subargument of with
type	data type of the parameter
name	name of the parameter
value	value of the parameter

SQTL SEE

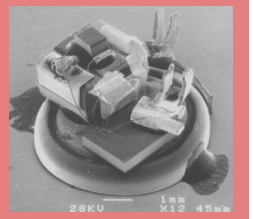


- SEE = Sensor Execution Environment
- Minden node-on fut.
- Veszi és továbbítja az SQTL üzeneteket.
  - A `receiver` mező `criteria` paramétere alapján megvizsgálja, hogy neki szól-e az üzenet.
  - Ha a `group` értéke `ALL_NODES`, üzenetszórással továbbítja az üzenetet, ha pedig `NEIGHBORS`, akkor csak a szomszédainak (címfordítás!).
- Végrehajtja a szükséges műveleteket az üzenetben vett minden akció (***action***) esetén.
- Ha a végrehajtás után eredmények is keletkeznek, egy `tell` üzenetet generál, és visszaküldi azt a lekérdezést küldő node-nak.
- A saját alkalmazások által generált üzeneteket is továbbítja.

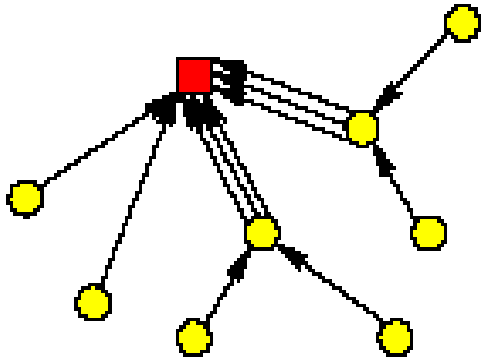
SQTL SEE


Running applications
App 123
App 125
Demultiplexed by *application-id* field
SEE (Sensor Executing Environment)
discarded if node does not match *criteria*
duplicated and forwarded to all neighbors if *group* contains *ALL NODES*
incoming messages

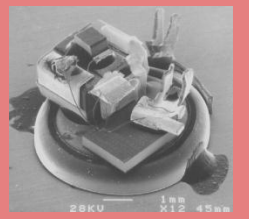
SINA – Információ terjesztése és gyűjtése



- Az alkalmazás szintű kommunikáció logikailag elkülönül az alacsonyabb szintű (fizikai) kommunikációtól.
- Az alkalmazás szintjén nem kell megadni, hogy mi a legalkalmasabb módszer az adatok begyűjtésére.
  - A lekérdezés típusa és a hálózat aktuális állapota alapján a SINA architektúra eldönti, hogy mi a leghatékonyabb módszer.
- Probléma lehet: „Válasz-robbanás”



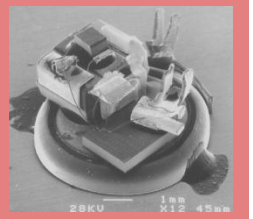
SINA – Információ terjesztése és gyűjtése



- Cél az információgyűjtés során:
  - Az információ minőségének biztosítása.
  - A hálózat erőforrásainak kímélése.

- Megoldások:
  - mintavételezés
  - válasz késleltetés
  - elosztott feldolgozás

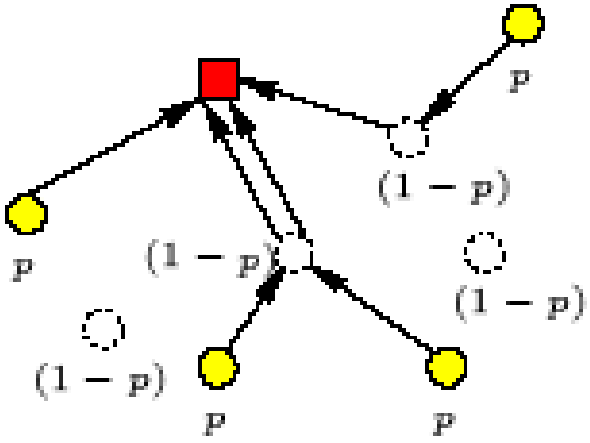
SINA – Információ terjesztése és gyűjtése



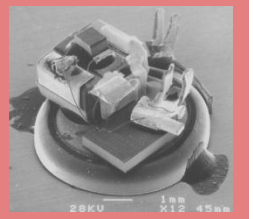
Mintavételezés

- Egyes alkalmazásoknál nem szükséges az összes szenzor részvétele a válaszadásnál.
  - Pl. Átlaghőmérséklet mérése egy területen.
- Minden node eldönti, hogy részt kíván-e venni a válaszadásban.
  - Pl. *p* valószínűséggel válaszol csak.

- <u>Továbbfejlesztés</u>: Ha a szenzorok elhelyezkedése egyenetlen, a sűrűbb területekről kisebb valószínűséggel válaszolnak. (APR – Adaptive Probability Response)



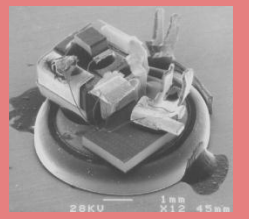
SINA – Információ terjesztése és gyűjtése



Válasz késleltetése

- Egyes alkalmazásoknál szükséges lehet az összes információ begyűjtése.
- Megoldás lehet a robbanás elkerülésére, ha a node-ok véletlen ideig késleltetik a válaszadást.
  - Így nem egyidőben, hanem időben elosztva jelentkezik a terhelés.
- Hátrány: A késleltetés növekszik.

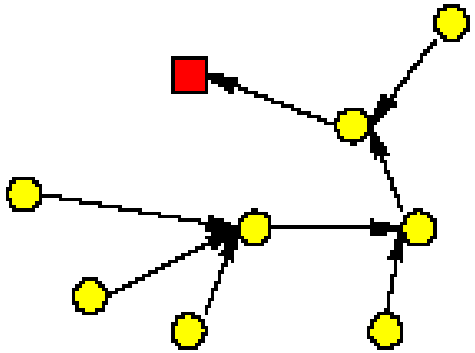
SINA – Információ terjesztése és gyűjtése



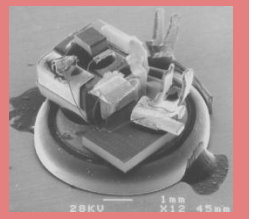
Elosztott feldolgozás

- A node-ok a hozzájuk beérkező információt feldolgozzák, majd a tömörített információt továbbítják csak.

- Az elő-feldolgozáshoz szükséges információt és algoritmust egy SQTL szkriptben kapják meg.

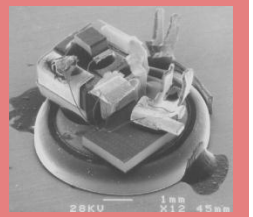


SINA – Példa alkalmazás



- A szenzorhálózat diagnosztikája.
  - Pl1: A szenzorok állapotának lekérdezése, és a hibás szenzorok kiszűrése.
  - Pl2: Elemek energiatartalmának meghatározása.

SINA – Példa alkalmazás



- Megoldás-1: Az összes válasz begyűjtése helyett mintavételezéssel és késleltetett válaszadással.

Algorithm 1 Centralized operation with sampling and self-orchestrated operation

Centralized_Diagnose(*replyProb*, *kh*)

rebroadcast this message to all neighbors;
$prevNode \leftarrow$ message sender node;
if $uniform\_random(0,1) < replyProb$ **then**
  $numHops \leftarrow$ number of hops this message traversed;
  $delay \leftarrow kh \times [numHops^2 - (2 \times numHops - 1) \times uniform\_random(0,1)]$;
  wait for $delay$;
  read power level and position, then send them back via $prevNode$;
end if
Upon receiving a return message, relay it back to $prevNode$;

SINA – Példa alkalmazás

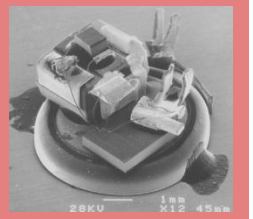
- Megoldás-2: Adaptív mintavételezéssel.

Algorithm 2 Adaptive probabilistic response operation

Apr_Diagnose(*ENRC*)

rebroadcast this message to all neighbors;
prevNode ← message sender node;
if this is a cluster head **then**
  $prob \leftarrow \frac{ENRC}{\#\text{ of children}}$;
  construct a script requesting all cluster members to return value with probability *prob*;
end if
Upon receiving a return message, relay it back to *prevNode*;

SINA – Példa alkalmazás



- Megoldás-3: Elosztott feldolgozással.

Algorithm 3 Diffused computation operation

Diffused_Diagnose(*timeout*)

$confirmCount \leftarrow 0$;
$prevNode \leftarrow$ message sender node;
send a **confirm** to $prevNode$;
rebroadcast this message to all neighbors;
set timer for *timeout* period;
while not timeout **do**
  if receive a message of type **confirm** **then**
    $confirmCount \leftarrow confirmCount + 1$;
  end if
end while
$answerList \leftarrow \{getPowerLevel()\}$;
while $confirmCount \neq 0$ **do**
  if receive a message of type **return** **then**
    insert the returned value into $answerList$;
    $confirmCount \leftarrow confirmCount - 1$;
  end if
end while
return $answerList$ back to $prevNode$;