

Sisteme Încorporate

Cursul 6

Software Power Management

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București

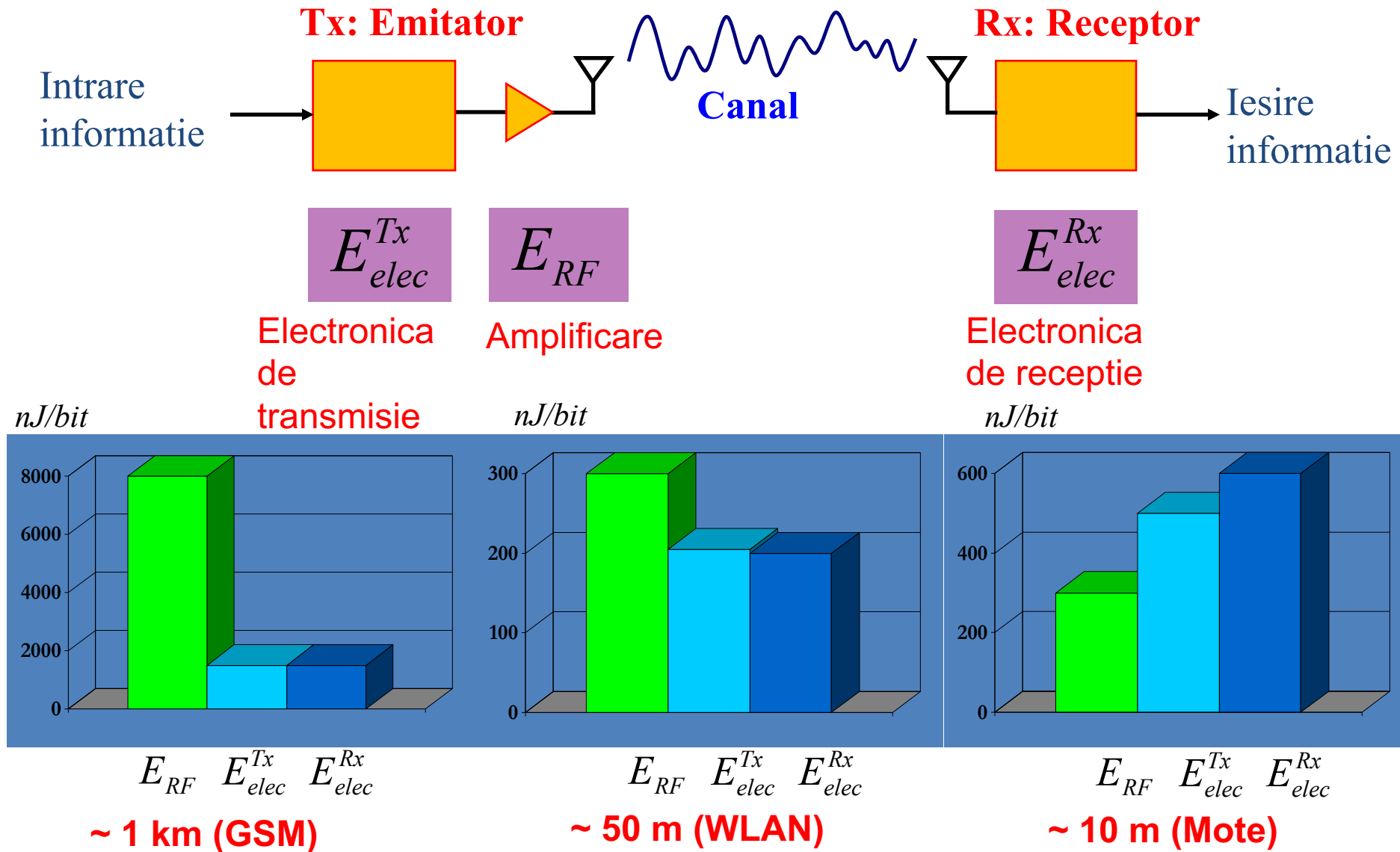
Recapitulare: Power Management

- Unde se poate face PM: H/W, Firmware, OS, Aplicatie, Utilizator
- Hardware si firmware nu pot aprecia starea globala a sistemului
- Utilizatorii nu stiu intotdeauna caracteristicile sistemului pe care lucreaza sau nu pot lua decizii
- Aplicatiile opereaza independent si sistemul de operare nu pune la dispozitia lor suficiente informatii
- PM la nivel de SO pare a fi cea mai buna solutie, dar
 - Trebuie sa puna la dispozitia aplicatiei starea sistemului la momentul respectiv pentru ca ele sa se poata adapta

Exemplu: Power Management pentru Wireless NIC

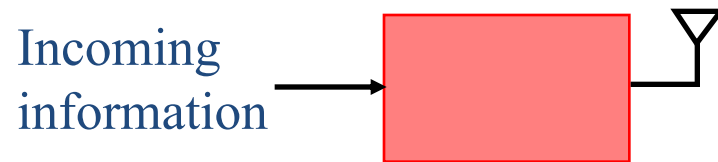
- Moduri de lucru: transmit, receive, idle, sleep, off
 - In general idle mode (nu se transmite si nu se receptioneaza) consuma aproape la fel de mult cat modul receive
 - Puterea de emisie in WLAN este 2x-3x mai mare decat puterea de receptie
 - Diferenta este d.p. cu distanta de emisie (puterea RF domina)
 - Puterea de emisie poate fi variata (dezavantaj: distanta efectiva de emisie)
 - Timpii de tranzitie intre stari sunt semnificativi
 - Transceiver IR: 10 μ s sa intre in sleep mode, si 40 μ s la wake-up
 - Placi “clasice” de retea: timpi de wake-up intre 100ms si 5s

Consumul de energie radio



Consumul de energie la Emițător

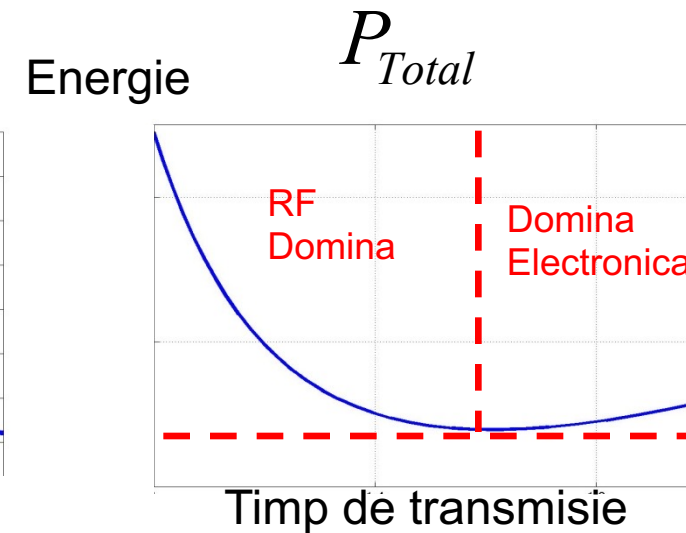
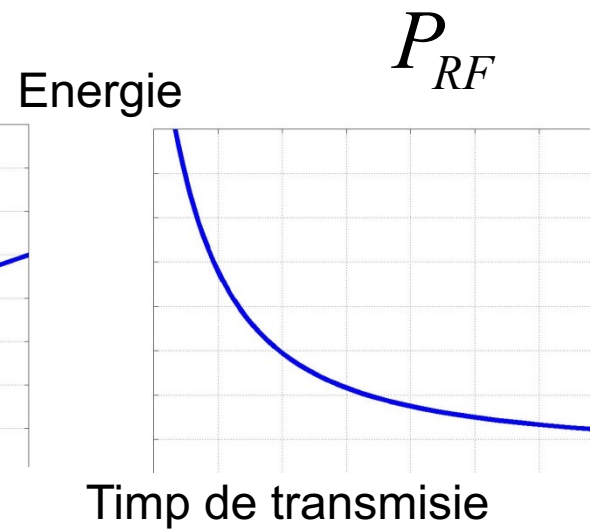
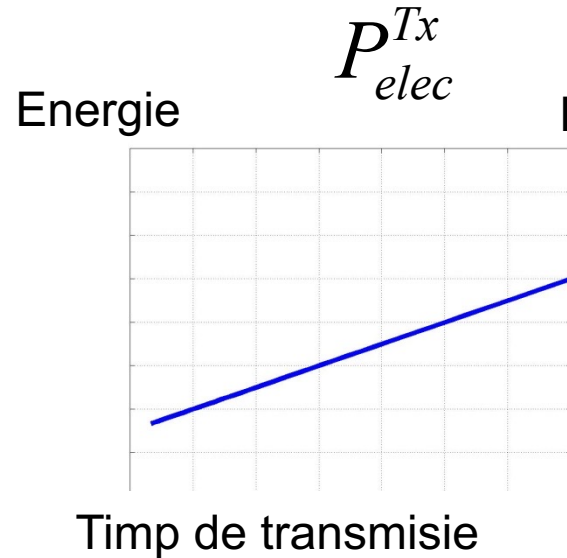
Tx: Emițător



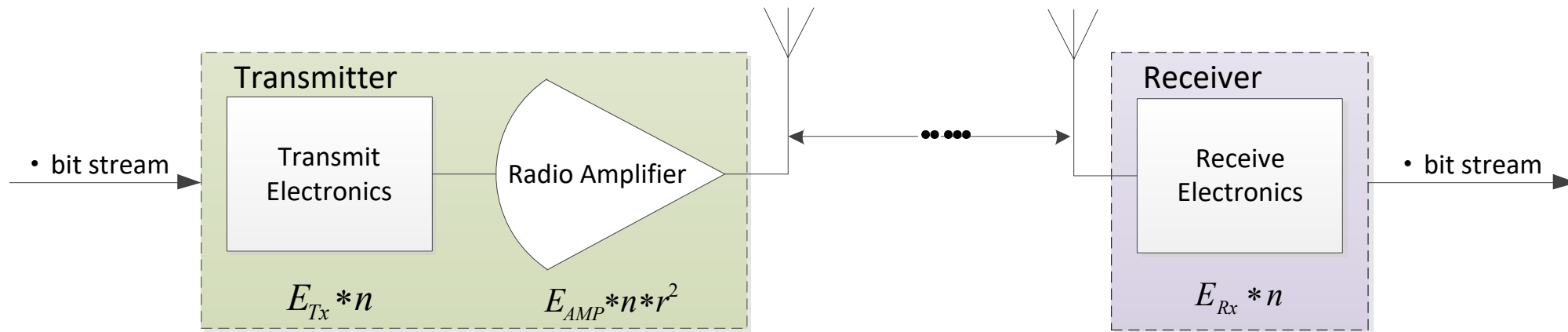
■ Parametru de interes:

- ◆ Consumul de energie pe bit transmis

$$E_{bit} = \frac{P}{T_{bit}}$$



Model Transmisie Radio

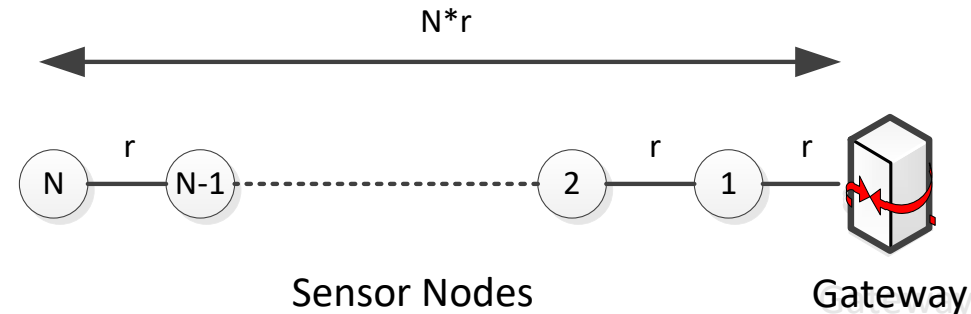
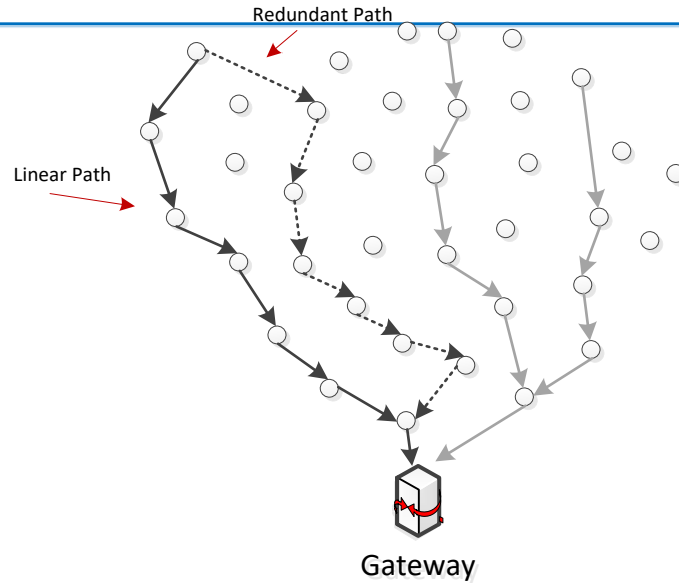


$$E_{Tx}(n, r) = E_{tc}(n) + E_{amp}(n, r) = n \cdot E_{trans.} + n \cdot \varepsilon_{amp} \cdot r^\gamma$$

$$\varepsilon_{amp} = \frac{\frac{S}{N_r} \cdot NF_{Rx} \cdot N_0 \cdot BW \cdot \left(\frac{4\pi}{\lambda}\right)^\gamma}{G_{ant} \cdot \eta_{amp} \cdot R_{bit}}$$

$$E_{Rx}(n) = E_{rc}(n) = n \cdot E_{recv.}$$

$$PL = P_{Tx[dBm]} - P_{Rx[dBm]} = PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_g$$



$$E_b(N, n, r) = E_{Tx}(n, N \cdot r) = n \cdot E_{trans.} + n \cdot \varepsilon_{amp} \cdot (N \cdot r)^\gamma$$

$$\begin{aligned} E_{MH}(N, n, r) &= N \cdot E_{Tx}(n, r) + (N - 1) \cdot E_{Rx}(n) = N \cdot n \cdot (E_{trans.} + \varepsilon_{amp} \cdot r^\gamma) + (N - 1) \cdot n \cdot E_{recv.} \\ &= n \cdot (N \cdot (E_{trans.} + E_{recv.} + \varepsilon_{amp} \cdot r^\gamma) - E_{recv.}) \end{aligned}$$

$$E_b^{all}(n, r) = \sum_{i=1}^N E_{Tx}(n, i \cdot r) = n \cdot N \cdot E_{trans.} + n \cdot \varepsilon_{amp} \cdot r^\gamma \cdot \sum_{i=1}^N i^\gamma$$

$$\begin{aligned} E_{MH}^{all}(n, r) &= \sum_{i=1}^N E_{MH}(i, n, r) = N \cdot E_{Tx}(n, r) + (N - 1) \cdot E_{Rx}(n) \\ &= \frac{N \cdot (N + 1)}{2} \cdot n \cdot (E_{trans.} + \varepsilon_{amp} \cdot r^\gamma) + \frac{N \cdot (N - 1)}{2} \cdot n \cdot E_{recv.} \end{aligned}$$

- Când e mai bine să fac multi-hop?

$$E_b^{all}(n, r) > E_{MH}^{all}(n, r)$$

- Ipoteză simplificatoare:

$$E_{trans.} = E_{recv.} = E_{circ.}$$

$$\gamma = 2$$

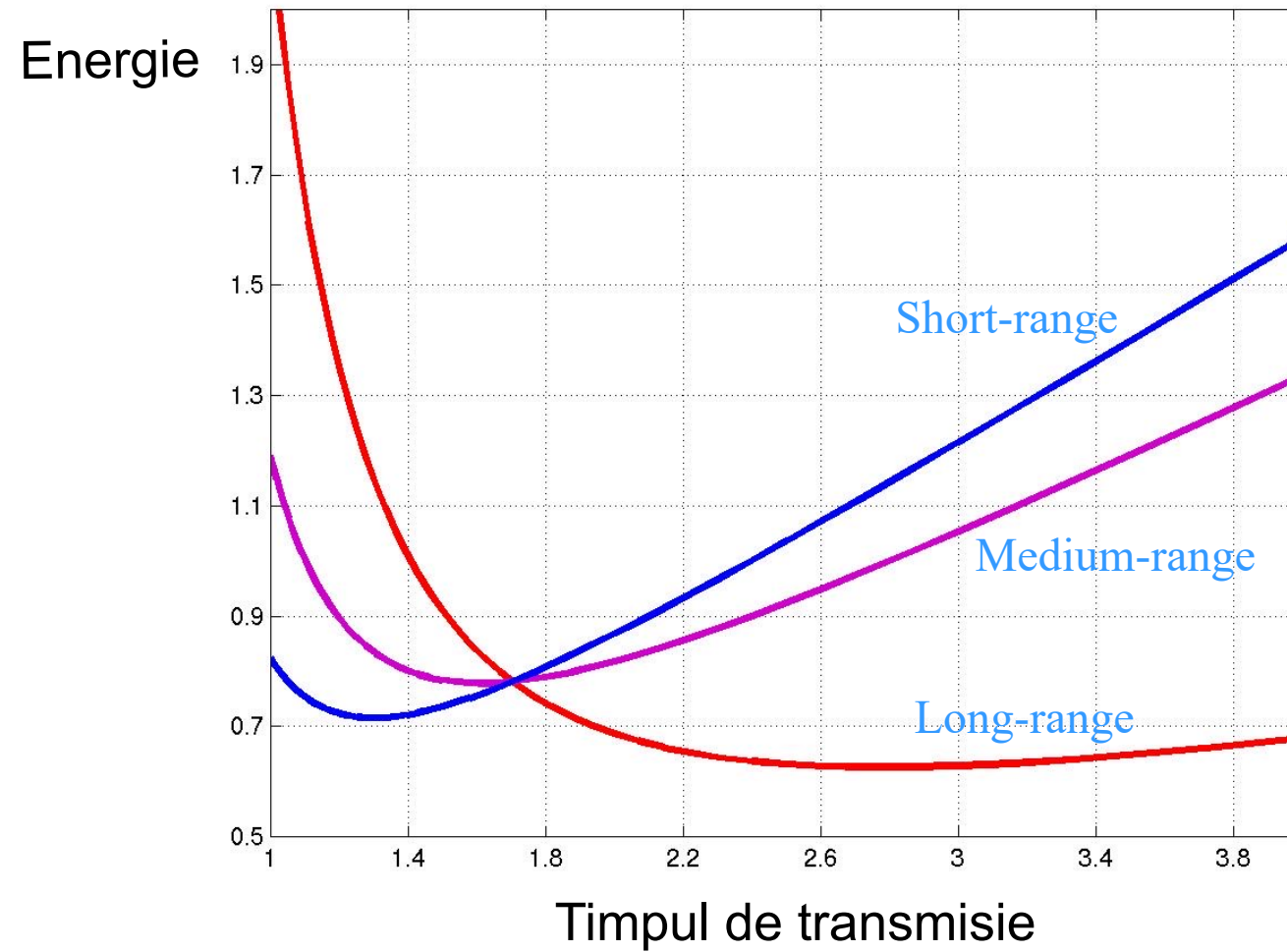
$$nNE_{circ.} + n\varepsilon_{amp}r^2 \sum_{i=1}^N i^2 > \frac{N \cdot (N+1)}{2} n(E_{circ.} + \varepsilon_{amp}r^2) + \frac{N \cdot (N-1)}{2} nE_{circ.}$$

$$\frac{E_{circ.}}{\varepsilon_{amp}} < \frac{N+1}{3} r^2$$

$$\gamma = 4$$

$$\frac{E_{circ.}}{\varepsilon_{amp}} < \frac{(N+1)(6N^2 + 15N + 16)}{30} r^4$$

Efectele Distanței de Transmisie



- Strategiile similare celor pentru hard-disk si CPU
 - Tranzitiile sleep $\leftrightarrow$ wakeup $\ll$ ca la discuri
 - Pot fi implementate la nivel de protocol MAC e.g. 802.11
- Reducerea incarcarii NIC
 - Compresia de date
 - Oprirea emisiei in cazul unei rate de eroare mare
- Reducerea puterii de transmisie
 - Afecteaza direct BER (Bit Error Rate)

Power Management in Transceiverul Radio

- Circuitele digitale pot fi oprite/pornite usor
- Managementul unui transceiver RF nu e facil
- Oprirea unor circuite ale transceiverului (buffere, etaje de amplificare, oscilator) duce la probleme
 - Stabilitatea transmisiei
 - Introduce timpi de asteptare (PLL settle etc.)

802.11 Power Management

NIC poate fi activa tot timpul (CAM)

- ◆ Consum foarte mare (~ 1.5 Watt)
- ◆ Poate sa injumatateasca timpul de viata al unui handheld



Alternativ, poate folosi power-save mode (PSM)

- ◆ Daca nu exista pachete receptionare, interfata intra in sleep
- ◆ Wake-up periodic (beacon de 100 ms)
- ◆ Reduce consumul cu 70-80%

Mecanisme de Suport pentru Power Management

- PM bazat pe hardware si firmware este problematic
 - Nu tine cont de aplicatiile care ruleaza
 - » Ex. Ecranul se stinge in timpul unei prezentari
- Solutia: Incorporeaza informatiile despre aplicatii
 - la decizii de power management bazate pe datele stranse de aplicatie despre utilizator si modul in care acesta foloseste sistemul
- SO este locul potrivit pentru PM, luarea de decizii si coordonarea resurselor
 - Ex.: Arhitectura Microsoft OnNow si extensiile API pentru Windows98, Windows 2000

- Extensia a Windows API care permite aplicatiilor sa
 - la decizii de power management
 - Se adapteaza la un power-state curent
 - Afla daca sistemul ruleaza pe baterie ca sa reduca procesarea
 - Descopera starea hard-discului si amana operatiile I/O de prioritate redusa
- Necesita schimbari in hardware, BIOS, OS si softul de aplicatie
 - Standarde noi de magistrala si device h/w management
 - Interfata standard intre OS si hardware
 - ACPI (Intel & Toshiba)

Advanced Configuration and Power Interface (ACPI)

- Standard deschis pentru device configuration si power management (HP, Intel, Microsoft, Toshiba, Dell 1996)
 - Defineste interfete independente de platforma pentru descoperirea componentelor hardware, configurare, power management si monitorizare.
 - Foloseste tabele de descriptori
 - Continutul unei tabele pentru fiecare device:
 - Cerinte si capabilitati power management
 - Metode pentru a seta si a obtine power-state-ul curent
 - Setari hardware
 - Metode pentru setarea resurselor hardware

- **G0 (S0) Working**
- **G1 Sleeping**
 - **S1:** Toate memoriile cache sunt golite si CPU nu mai executa nici o instructiune. CPU si memoria RAM sunt alimentate; device-urile care nu indica explicit ca trebuie sa ramana operationale sunt oprite.
 - **S2:** CPU este oprit
 - **S3:** Cunoscuta sub numele de *Standby* sau *Sleep*. Mem. RAM este alimentata.
 - **S4:** *Hibernare*. Intregul continut al memoriei volatile este salvat in memoria ne-volatila (hard-disk) si sistemul este oprit.
- **G2 Soft Off.** G2 este aproape acelasi lucru cu G3 *Mechanical Off* dar unele componente raman activate a.i. sistemul sa poata sa fie “trezit” de la tastatura, ceas intern, modem, LAN sau USB.
- **G3 Mechanical Off:** Consumul de energie al sistemului ajunge aproape la zero (de obicei mai ramane alimentat doar ceasul de timp real din bateria proprie.)

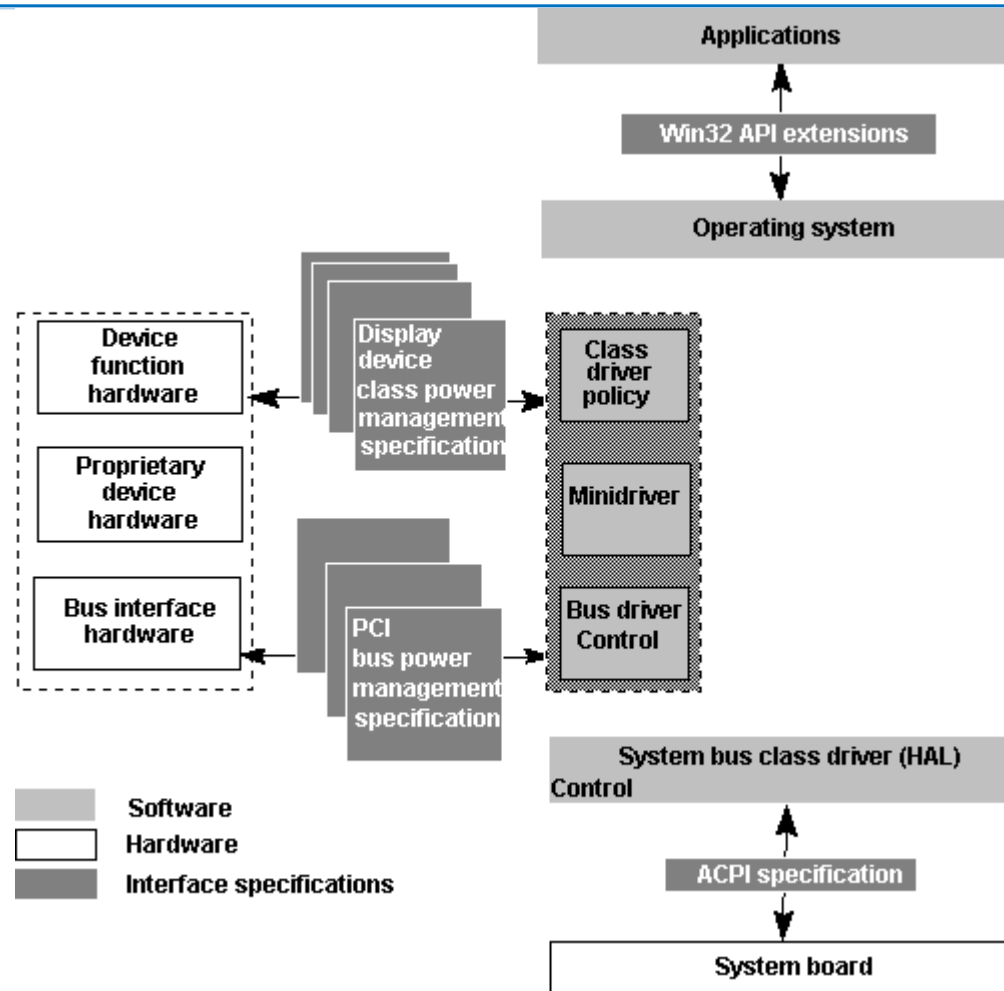
- **Device states**

- D0 *Fully-On* – starea normala de operare.
- D1 si D2 sunt stari intermediare care variaza in functie de device.
- D3 *Off* – dispozitivul nu raspunde la comenzi pe bus.

- **Processor states**

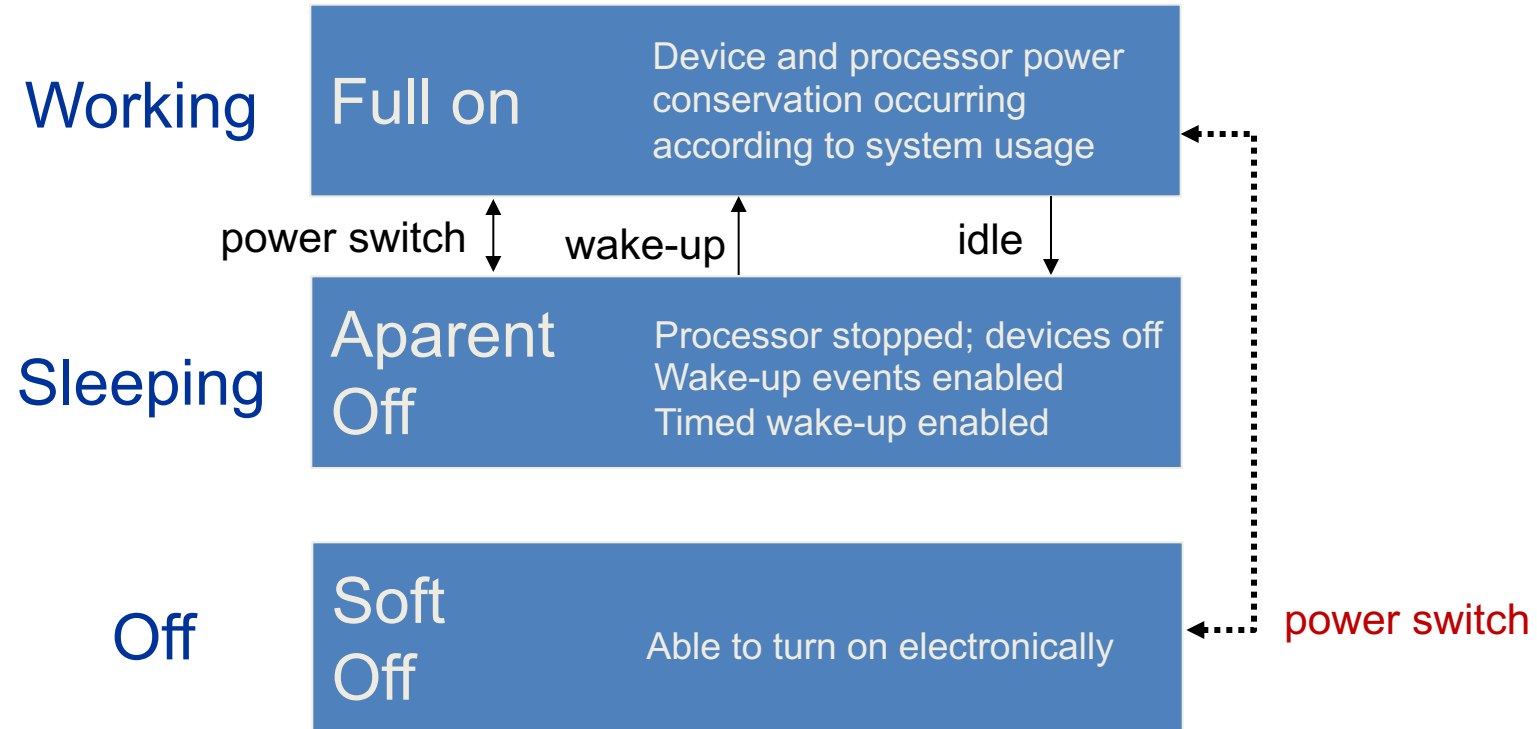
- C0 – starea normala de operare.
- C1 (*Halt*) este o stare in care procesorul nu executa instructiuni dar poate sa se intoarca la C0 aproape instantaneu
- C2 (*Stop-Clock*) procesorul nu primeste semnal de ceas (revenirea la C0 dureaza mai mult).
- C3 (*Sleep*) procesorul nu mentine coerenta memoriei cache. Unele procesoare au variatii ale acestei stari (Deep Sleep, Deeper Sleep, etc.) care difera prin timpul de wake-up.

Componentele OnNow



<http://www.microsoft.com/whdc/archive/OnNowApp.msp>

- Punctul de vedere al utilizatorului: sistemul este pornit/oprit
- In realitate sistemul trece prin mai multe stari in functie de politica sistemului de operare
- Stari globale
 - working: aplicatiile se executa normal
 - sleep: aplicatiile sunt oprite
 - SO monitorizeaza actiunile utilizatorului si aplicatiile pentru a determina intrarea-iesirea din aceasta stare
 - off: sistemul este oprit in totalitate



Stari Globale OnNow

<http://www.microsoft.com/whdc/archive/OnNowApp.msp>

Concluzie: Compromis între consum și performanță

