Announcements

Outline

- Every Joule is Previous: The Case for Revisiting Operating System Design for Energy Efficiency

Battery power as a managed resource

- Little change in basic technology
  - store energy using a chemical reaction
- Battery capacity doubles every 10 years
- Energy density/size, safe handling are limiting factor

Energy density of material	***KWH/kg***
Gasoline	14
Lead-Acid	0.04
Li polymer	0.15

Energy conserving opportunities

Energy consumed = time * power

- OS: The OS can reduce energy consumption by reducing the time spent in high energy consumption states
- HW: Hardware can reduce energy consumption by reducing the power consumed by high energy consumption states

Hardware advances

- Lower power CPUs
  - disable idle units in CPU to save power
    - e.g. transmeta crusoe chip
  - lower clock frequency to conserve battery
    - e.g. intel speedstep technology
- Displays
  - active matrix LCD
  - reflective displays (uses ambient light for lighting)
- Memory
  - RAMBUS RDRAM provides various power modes
- Batteries not constant source of power
  - pulsed mode is better

Battery power - disk

- Problem:
  - disk spinup,seek / flash memory erase costs

(power values
for IBM travelstar)

Operation	***Energy Consumption***
Spinup	4.7 W
Read/write	2 W
Idle	0.65 W
Seek	1.8 W
Sleep	0.1 W
Standby	0.25 W

- Predictive spin-ups
- Caching/prefetching

Battery power - memory

- power aware memory hierarchy - e.g. Rambus

(power values for 128 Mb PC800 RDRAM)

Operation	***Energy Consumption***
Burst READ	567 mA
Burst WRITE	575 mA
Attention	148 mA
Standby	101 mA
NAP	4.2 mA
Self refresh (low power)	0.7 mA

- each chip can be put into different power modes
- Initial page placement, power transition, page migration

Battery power - network

- Transmit, listen, idle costs

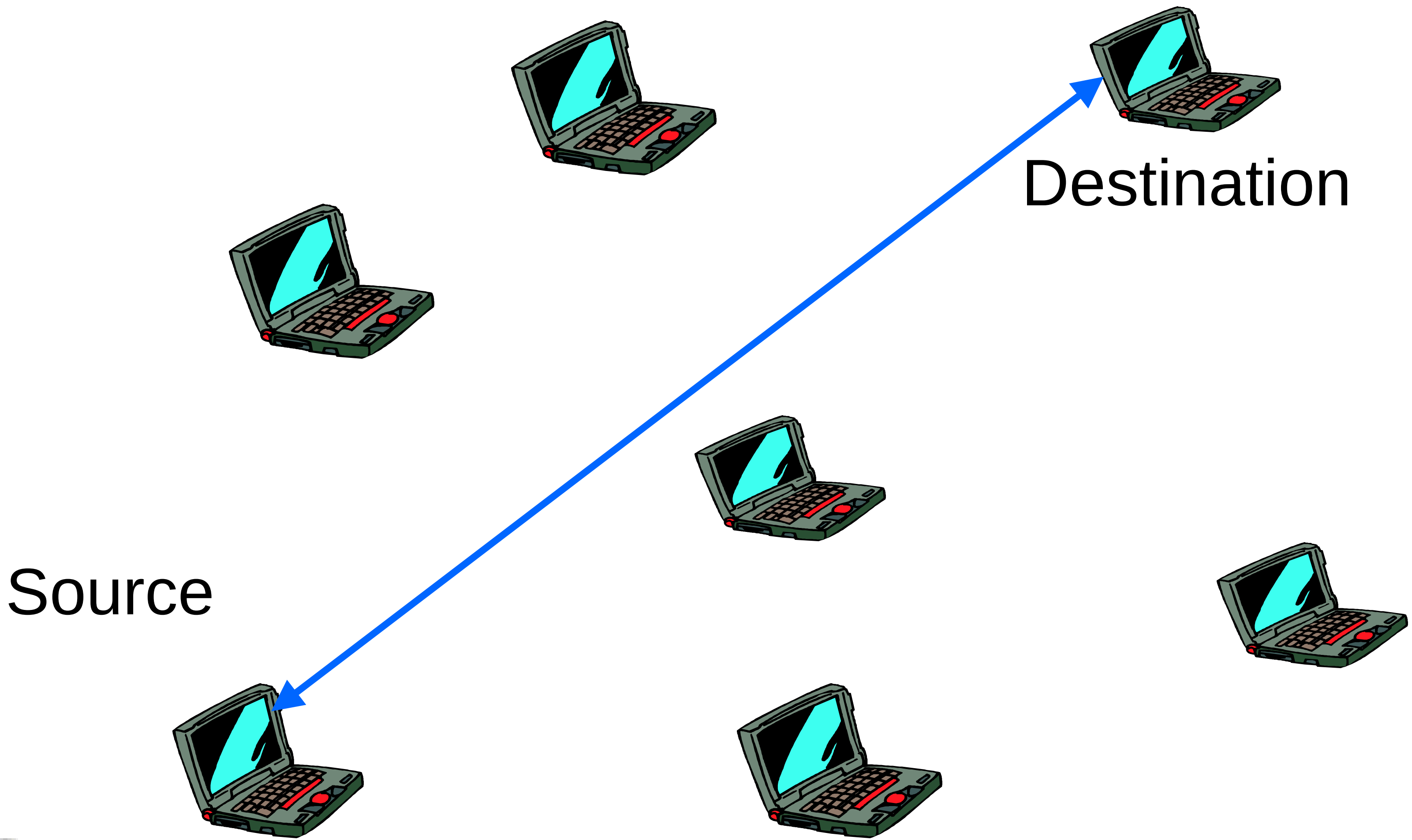
(power values for
Lucent wavelan
2 Mbps)

Operation	***Energy Consumption***
Transmit	330 mA
Receive	280 mA
Doze	9 mA

- Power consumed when device is active
- Receiver does not know when sender has data to be sent - continous wait is expensive

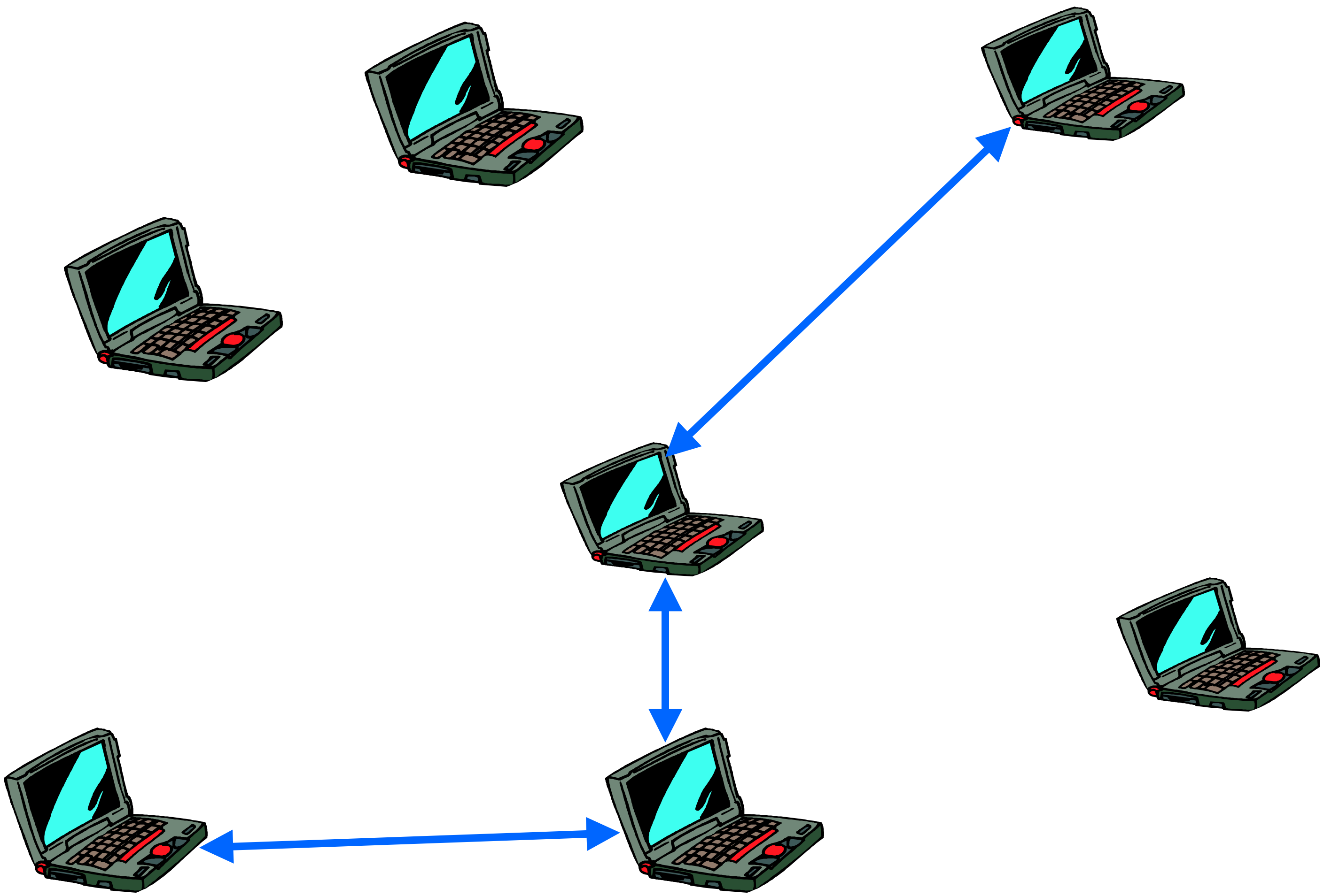
Ad-hoc networking

Cooperative routing in a mobile, highly variant scenario



Power Aware Ad-hoc networking

Power consumption exponential with distance



Battery power – application level

- Application aware adaptation to manage high power energy states
  - Managing the power states in a palm
  - Managing battery in a digital camera using image transcoding
  - Application level adaptation to manage energy
  - Client/server computation split

Hypothesis: OS should manage power

- Typical Operating Systems are designed to hide latency (caching), fairly share resources (CPU, memory, network, etc.)
- What about power? If two processes are runnable and one is expected to consume more power, which do you run?
- Do you run cleanup daemon when the battery power is low?

Discussion