Active/Idle Toggling with Low-Power Idle

January 2008

IEEE 802.3az Task Force

Presenter: Robert Hays

Intel Corporation

Contributors: Aviad Wertheimer, Eric Mann

Energy Efficient Ethernet

Supporters

- Ozdal Barkan (Marvell)
- Jim Barnette (Vitesse)
- Hugh Barrass (Cisco)
- Brad Booth (AMCC)
- Joseph Chou (Realtek)
- Dan Dove (HP ProCurve)
- Robert Hays (Intel)
- Adam Healey (LSI)
- Sanjay Kasturia (Teranetics)
- David Koenen (HP)
- David Law (3Com)
- Brian Murray (LSI)
- Gavin Parnaby (Solarflare)
- Wiren Perera (Plato Networks)
- Aviad Wertheimer (Intel)
- Bill Woodruff (Aquantia)
- George Zimmerman (Solarflare)

Agenda

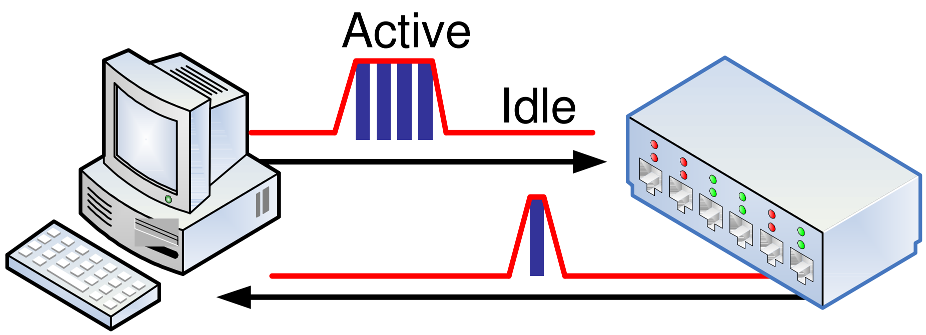
1. Updates from November (see hays_01_1107)
   - Glossary
   - Active/Idle Toggling Concept
   - Low-Power Idle Overview
   - Power Consumption
2. Elaboration on some Elements
   - Asymmetric Operation
   - Supporting Deep Sleep Levels
   - Auto-Negotiation
   - Initiating Transitions
3. Benefits of Active/Idle Toggling
4. Areas for Further Investigation

Glossary

- Electrical Energy Terms:
  - **Operating Power** – (Watts) The rate at which electrical energy is delivered to a circuit or system
  - **Energy Consumption** – (Joules) Aggregate power consumed by a system over a period of time
  - **Energy Efficiency** – (Joules/bit) Energy required to complete a unit of work. E.g. energy required to transmit/receive each bit of data.
  - **Average Power** – (Watts) Energy consumed divided by period of time

- Ethernet Operating States:
  - **Active** – Sending packets. Higher power. Defined today for all PHYs.
  - **Normal Idle (N_IDLE)** – Not sending packets. Same or less power than Active. Defined today as “Idle” for all PHYs.
  - **Low-Power Idle (LP_IDLE)** – Not sending packets. Minimal power. To be defined by IEEE 802.3az.

Active/Idle Toggling Concept



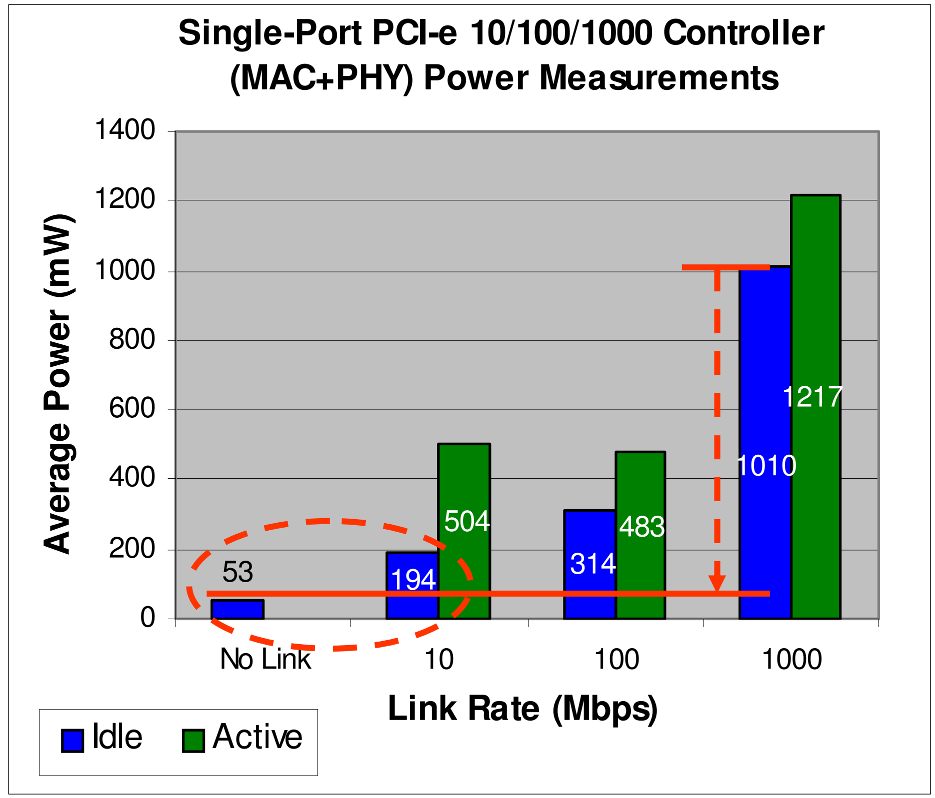
- Principle: Transmit data as fast as possible, return to Low-Power Idle
  - Highest rate provides the most energy-efficient transmission (Joules/bit)
  - LP_IDLE consumes minimal power (Watts)
- Energy savings come from cycling between Active & Low-Power Idle
  - Power is reduced by turning OFF unused circuits during LP_IDLE (e.g. portions of PHY, MAC, interconnects, memory, CPU)
  - Energy consumption scales with bandwidth utilization
- Transmitter initiates LP_IDLE transitions, Receiver acquiescent
  - Control policy is managed by system entity beyond IEEE 802.3 scope

Low-Power Idle Overview

- LP_IDLE is a "quiet" line that consumes minimal power
  - It is used when no data is being transmitted
  - Only essential circuitry (e.g. timing recovery) must remain ON

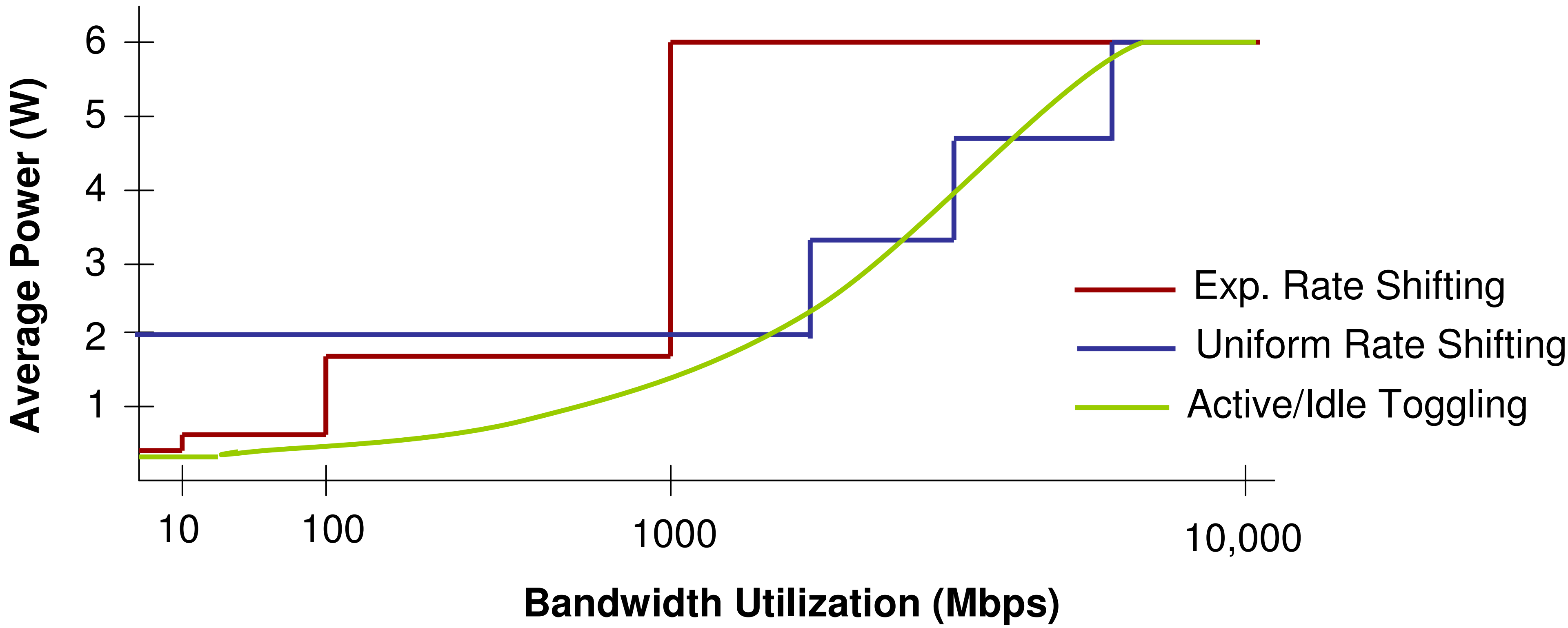
- Rate-specific solutions required:
  - 100BASE-TX (see chou_01_0108)
  - 1000BASE-T (see healey_01_0108)
  - 10GBASE-T (see parnaby_01_0108)
  - 10GBASE-KR
  - 10GBASE-KX4

- Gigabit LP_IDLE power estimate:
  - "No Link" ≤ LP_IDLE ≤ 10Mbps Idle
  - e.g. 53mW ≤ LP_IDLE ≤ 194mW
  - Should be closer to "No Link"



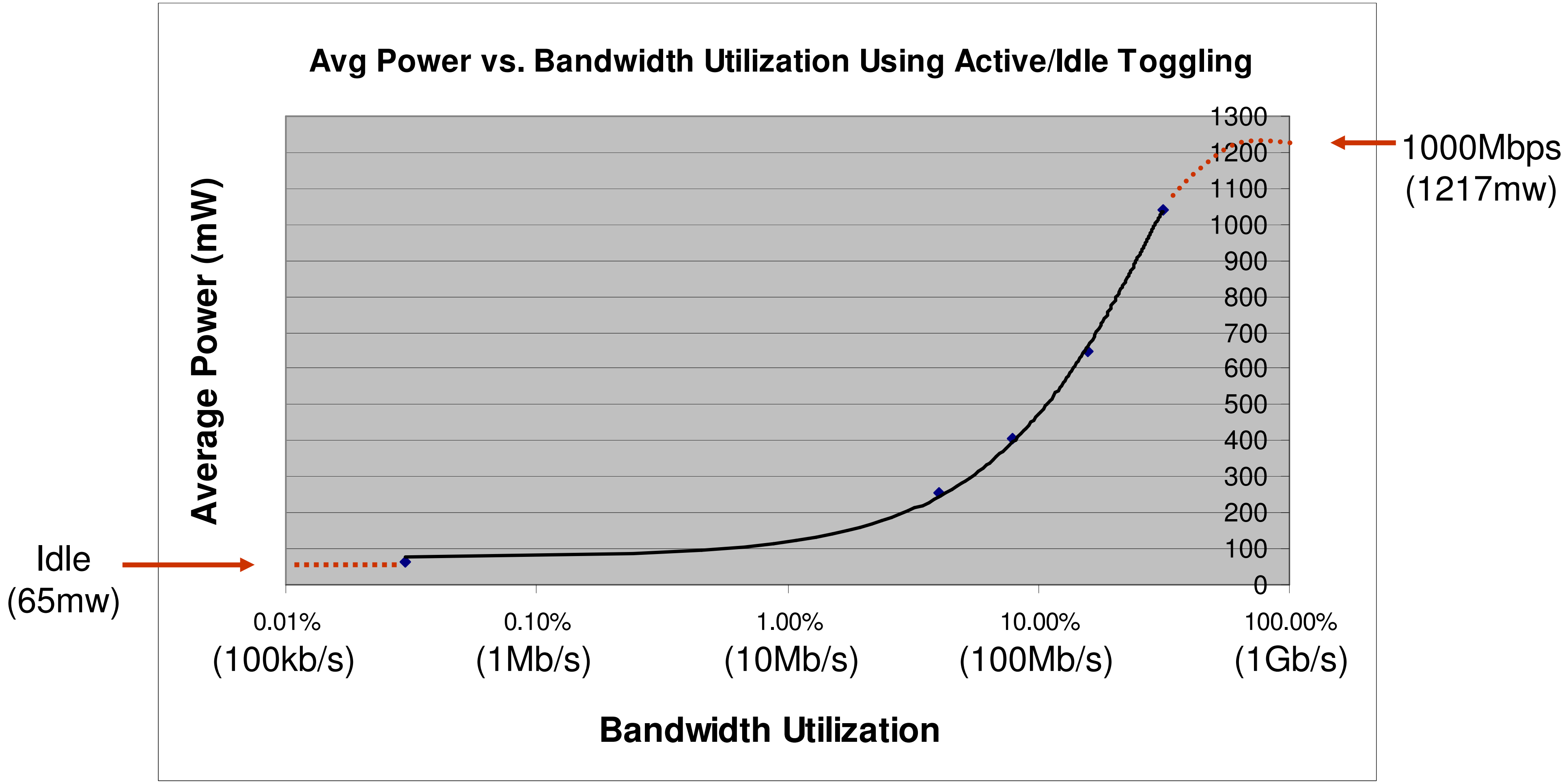
Source: Intel labs. Intel® 82573L Gigabit Ethernet Controller, 0.13µm, "Idle" = no traffic, "Active" = line-rate, bi-directional

Conceptual Average Power vs. BW Utilization



- **Exponential Rate Shifting** offers power steps at 1/10th, 1/100th, 1/1000th rates for savings during periods of low-utilization (<10%)
- **Uniform Rate Shifting** offers power steps on 1/4th rate increments for savings during periods of medium to high utilization (25%-75%)
- **Active/Idle Toggling** with Low-Power Idle allows smooth power averaging across a broad range of bandwidth utilization (<80%?)

Simulated Active/Idle Toggling Avg. Power

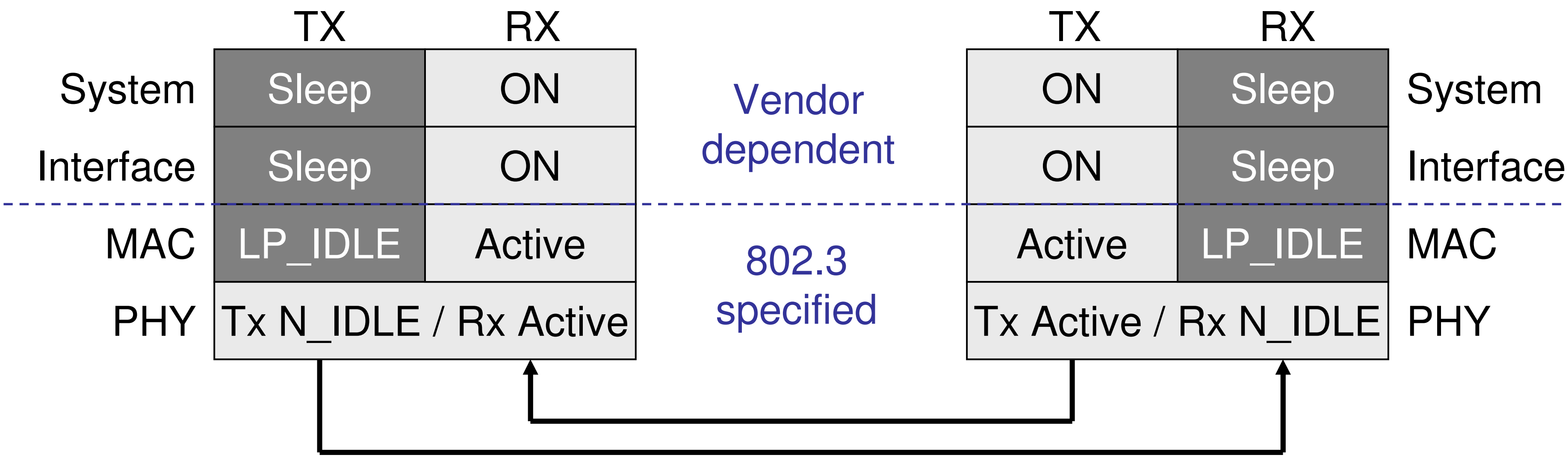


Source: Intel labs. Simulation program source code and sample traffic pattern trace files posted on the EEE Tools web page: http://grouper.ieee.org/groups/802/3/az/public/tools/index.html

Input Assumptions:

- Traffic Input = Trace_VOIP_*.txt
- 1000Mbps Active Power = 1217mW
- LP_IDLE Power = 65mW
- LP_IDLE Initiation Wait = 10μs
- LP_IDLE Transition Latency = 1μs
- Active Resume Latency = 10μs

Asymmetric Operation



- Asymmetric operation would further improve energy efficiency
  - Independent Tx & Rx transitions into LP_IDLE
  - End-node traffic is typically weighted toward either send or receive

- Asymmetric toggling is valuable at MAC-layer and above
  - Tx & Rx data paths already operate independently above the PHY
  - Transition initiation would need to occur between MACs
  - PHYs would only enter LP_IDLE if both Rx & Tx are in N_IDLE

Supporting Deeper Sleep Levels

	Active PC Example	Quick-Resume PC Example (~10µs)	Longer-Resume PC Example (~100µs)	
Memory	M0 Active	M0 Active	M1 Standby (100µs)	Vendor dependent
PCIe	L0 Active	L0s Standby (3µs)	L1 Standby (6µs)	
MAC	Active	LP_IDLE (1µs)	LP_IDLE (1µs)	802.3 specified
PHY	Active	LP_IDLE (10 µs)	LP_IDLE (10µs)	

- Variable resume latencies allow performance vs. power optimization

Higher Performance ← → Lower Power

Quicker Resume — Slower Resume

- Resume predictability allows more intelligent power management
  - Greater power savings doesn't come from just longer LP_IDLE duration, it comes from being able to <u>safely turn OFF/ON more circuitry</u>
  - Two ways to provide predictability:
    1. Rx tells Tx how long to wait before sending data (via negotiated resume latency)
    2. Tx tells Rx how long it will be in LP_IDLE (via notification of sleep duration)

Auto-Negotiation

- Negotiate EEE capabilities during Auto-negotiation:
  1. EEE support for each speed
     a. 10G
     b. 1G full-duplex
     c. 100M full-duplex
  2. LP_IDLE Resume Latency values
     a. Maximum T_RESUME (may be specified by 802.3az)
     b. Minimum T_RESUME (may be specified by 802.3az)
     c. Desired T_RESUME
  3. Possibly… LP_IDLE Duration parameters:
     a. Maximum T_LP_IDLE (PHY or system limitation)
     b. Minimum T_LP_IDLE (for effective power saving)

- Updates (e.g. T_RESUME changes) could be negotiated via MAC control frames or other means

Initiating Transitions

- Transition control policy is managed by a system entity beyond IEEE 802.3 scope
- Transition initiated by Tx (data source), Rx acquiescent
  - 2-way negotiation or Acks are unnecessary
- Example transition to/from LP_IDLE:
  1. When no data to transmit, Tx signals entry into LP_IDLE
  2. Rx detects entry into LP_IDLE and may reduce it’s power
  3. PHYs may periodically wake for Link Training
     - Training may only be necessary for some PHYs, e.g. 10GBASE-T
  4. When data to transmit, Tx PHY enters N_IDLE and MAC waits negotiated T_RESUME before beginning data transmission

Benefits of Active/Idle Toggling for EEE

- Reduced power during low utilization
- Energy consumption scales with bandwidth utilization
- Minimal impact to performance
- Turning circuits ON/OFF is easier than rate shifting
- Integrates well with PC & server power management
- Simple, one-way transition initiation
- May allow Asymmetric operation to save additional energy

Areas for Further Investigation

- Low-Power Idle state for each PHY type
- Negotiating resume latencies and/or LP_IDLE durations
- Transition signaling scheme
- MAC-PHY sync control
- Asymmetric operation

Thank You!

- Questions?