Active/Idle Toggling with 0BASE-x for Energy Efficient Ethernet

November 2007

IEEE 802.3az Task Force

Presenter: Robert Hays
Intel Corporation

Contributors: Dave Chalupsky, Eric Mann, James Tsai, Aviad Wertheimer

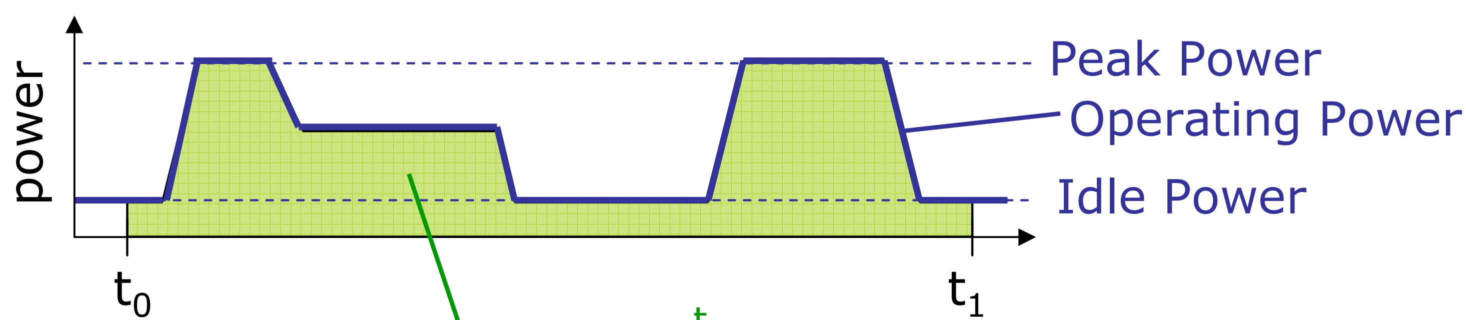
Energy Efficient Ethernet

Agenda

- GbE Controller Power & Energy Consumption
- Active/Idle Toggling with 0BASE-x Proposal
- Average Power Curves
- State Transitions
- Implications to 802.3 Specifications
- PHY Considerations
- Recommendations

Power & Energy Glossary

- **Operating Power** – (Watts) The rate at which electrical energy is delivered to a circuit or system
  - **Peak Power** – (Watts) Maximum operating power of a system
  - **Idle Power** – (Watts) Operating power of a system at rest
- **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 the period measured



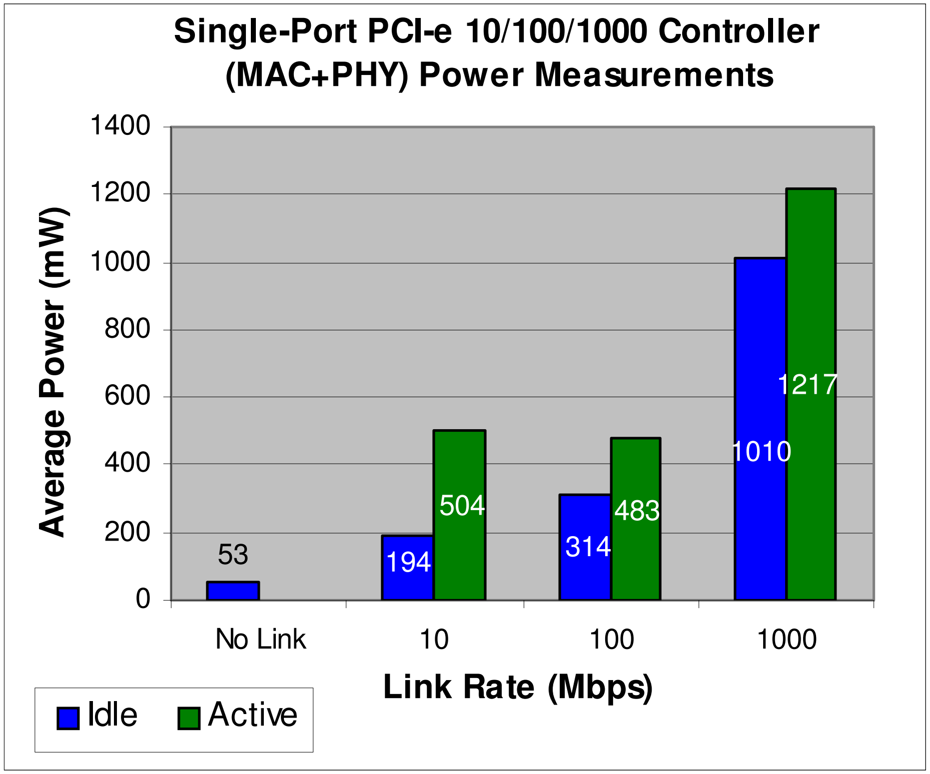
$$\text{Energy Consumption} = \int_{t_0}^{t_1} \text{power}(t)\, dt$$

$$\text{Average Power} = \text{Energy Consumption} / (t_1 - t_0)$$

Energy Efficient Ethernet

intel

GbE Controller Power Measurements



Source: Intel labs

Test Information:

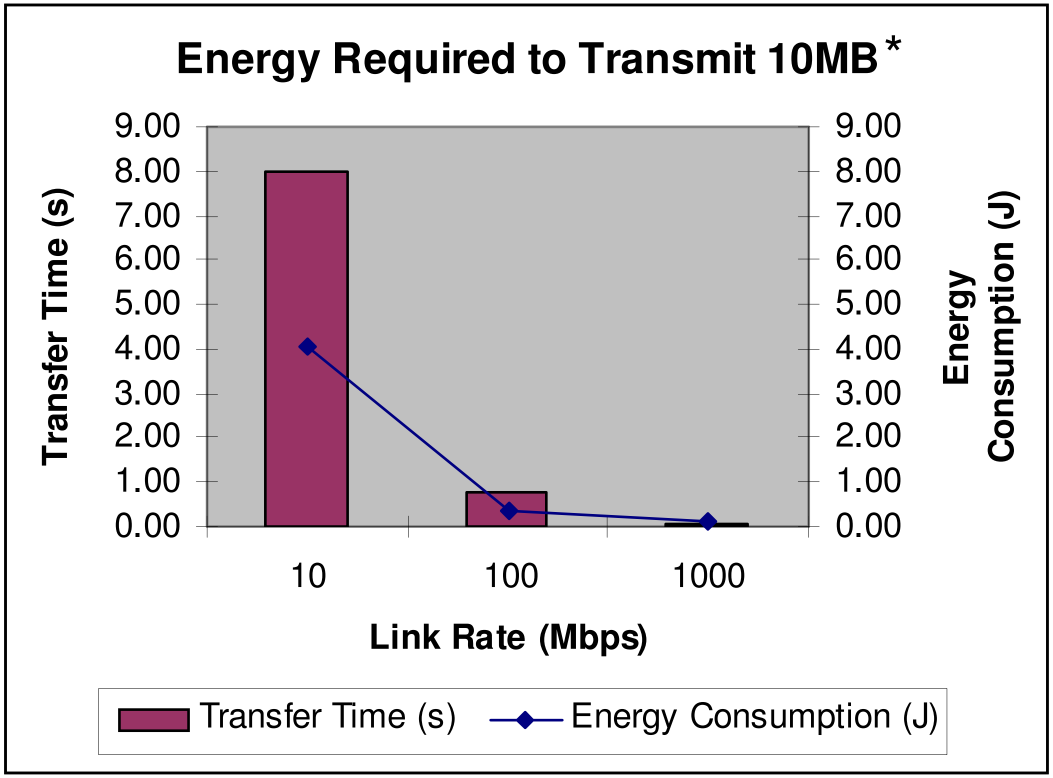
- Intel® 82573L Gigabit Ethernet controller, including:
  - 10/100/1000 PHY, MAC, buffers, PCI-Express x1 host interface
  - Typical client PC NIC device
  - .13µm fab process
- Idle = No traffic, 0 Mbps
- Active = Line-rate bi-directional traffic
- No Link = Cable removed (D0 state)

Observations:

- 100M power is ~40% of 1000M power at 10% performance
- 100M & 1000M Idle power savings (vs. Active) is 17-35%
  - Savings come from PCIe L0s/L1 idle-state usage and turning off some circuits
- 10M Idle power savings is 60%
  - Additional savings from 10BASE-T idle signaling
- "No Link" is the lowest-power mode with the device still "on" (D0 state)
  - Savings come from turning off all unessential digital & analog circuits

Energy Efficient Ethernet

GbE Controller Energy Consumption



Source: Intel labs

Observations:

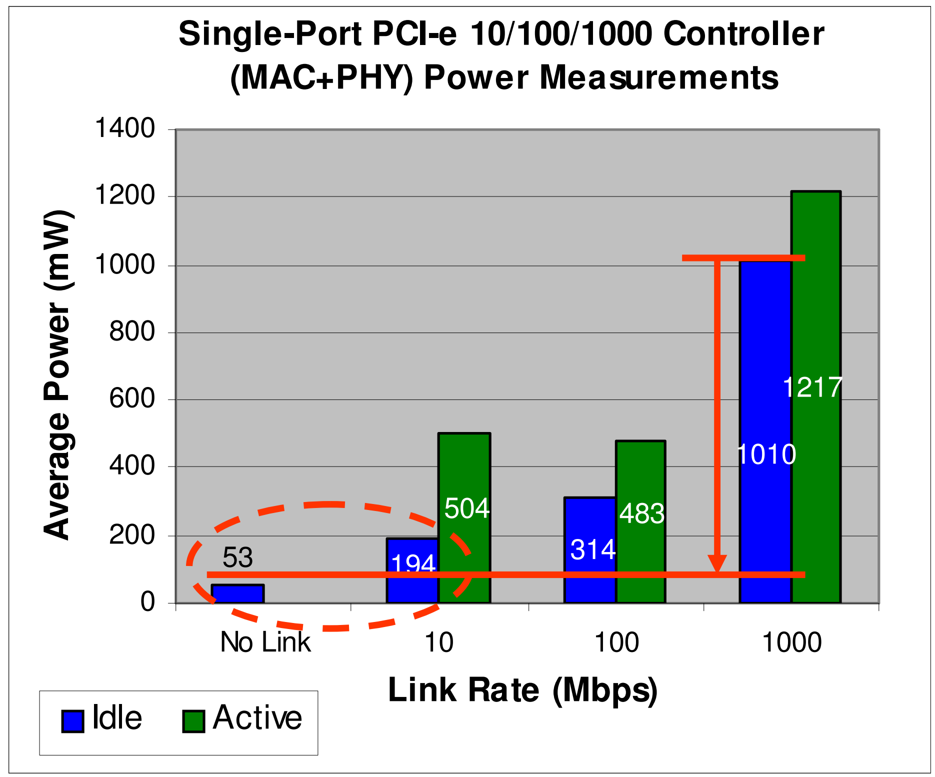
- 1000M transmission is 4x more energy efficient (J/bit) than 100M because it sends the same amount of data in 1/10th the time
- 1000M is 40x more energy efficient than 10M
- An "ideal" EEE solution would transmit data with minimal energy and return to low-power idle between packet bursts
  - An Idle state would need to be defined by 802.3 for higher-speed PHYs

Energy Efficient Ethernet

* Note: Bidirectional traffic. 10MB transferred in each direction.

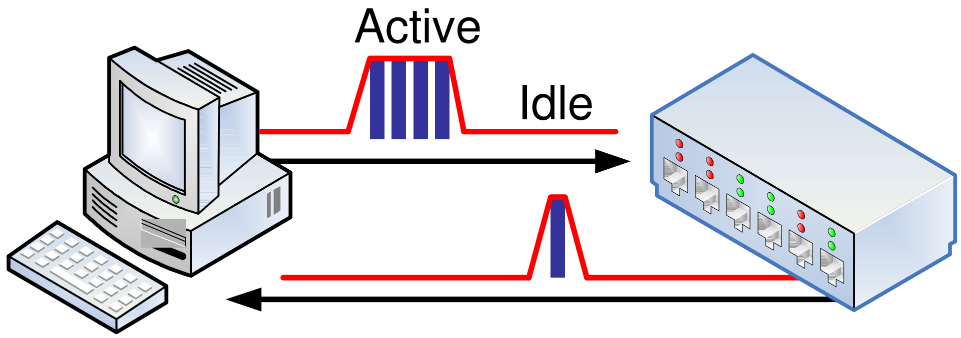
Proposed "0BASE-x" Idle

- 0BASE-x is a "quiet" line idle that consumes minimum power
  - '0' = zero data rate, 'x' = Variable for any supported PHY type
- 0BASE-x idle power expectations for a GbE device:
  - "No Link" ≤ 0BASE-x ≤ 10BASE-T Idle (e.g. 53mW ≤ 0BASE-x ≤ 194mW)
  - Est. it to be closer to "No Link"
- 0BASE-x could take form of:
  - A newly defined idle signal or…
  - Reduced-voltage 10BASE-T idle
- 0BASE-x variations are likely
  - Support for varying PHY types with unique timing requirements
  - Possibly multiple sleep levels with differing resume latencies



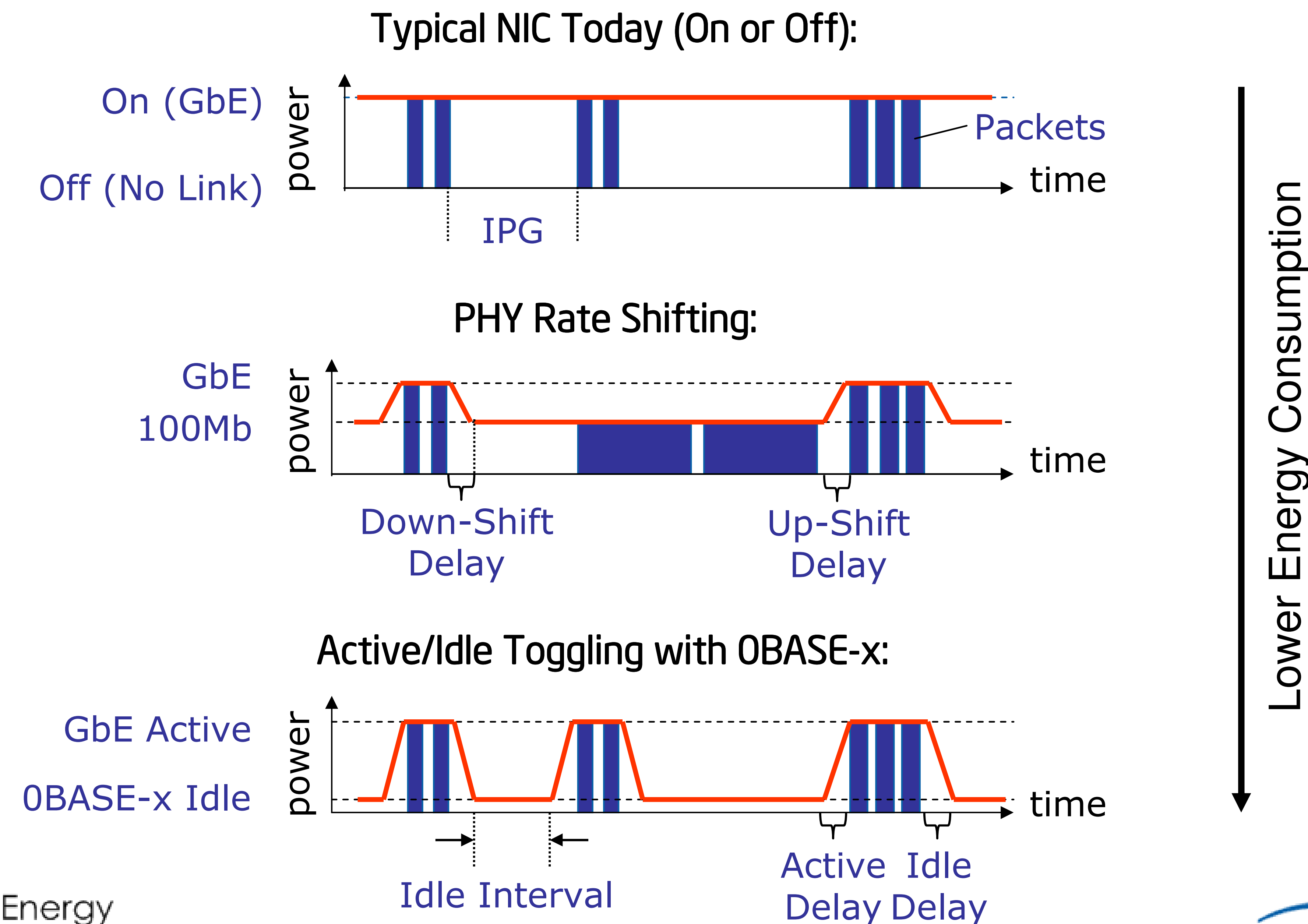
Source: Intel labs

Active/Idle Toggling with 0BASE-x Concept



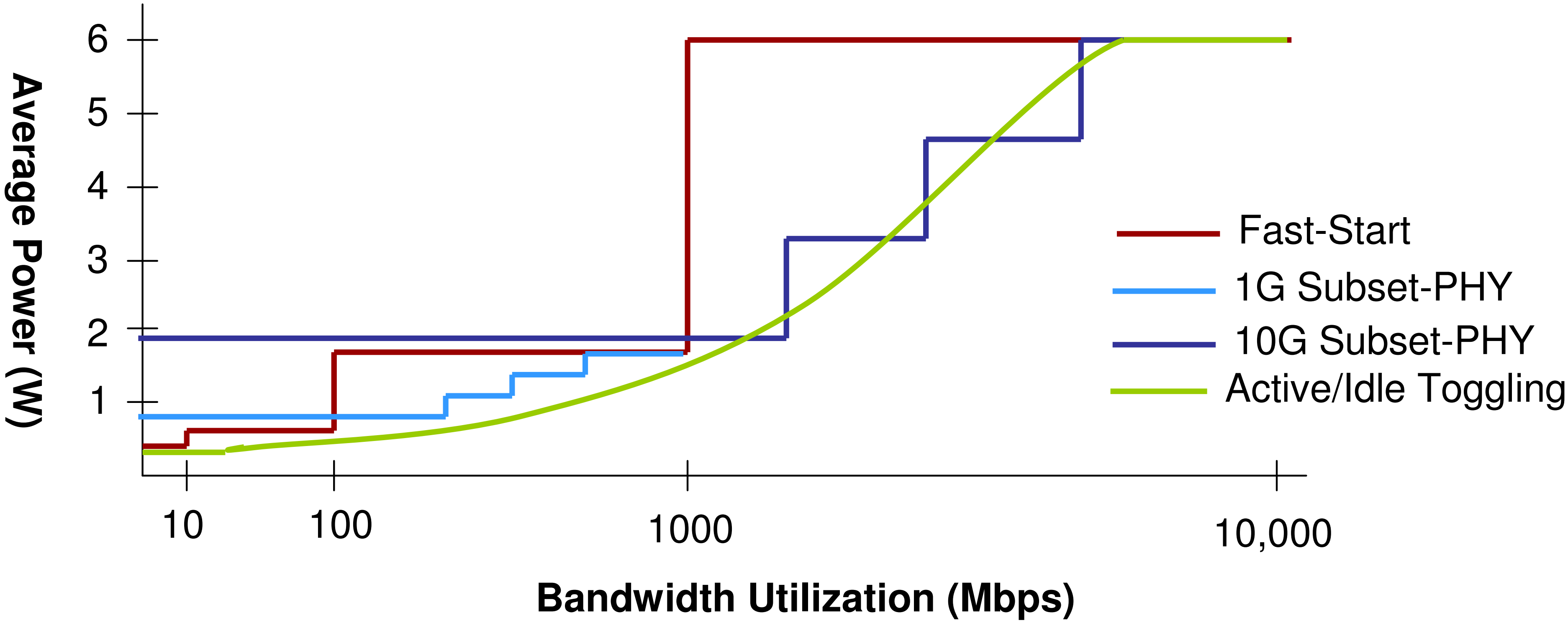
- Principle: Transmit data at fastest rate then return to idle
  - Energy savings come from power cycling between active/idle states
- Active/Idle toggling could be used *instead* of PHY rate shifting
  - Offers the best energy efficiency on links with lower utilization
  - Integrates well with existing PC power management schemes (e.g. ACPI)
  - Clock & power gating (on/off) is easier than rate shifting
- Asymmetrical operation would provide even better energy efficiency
  - Each direction could enter active & idle states independently
  - Most end-node traffic is heavily weighted toward either send or receive
  - Tx & Rx data paths already operate independently above the PHY

Behavioral Comparison



Energy Efficient Ethernet

Conceptual Average Power vs. BW Utilization

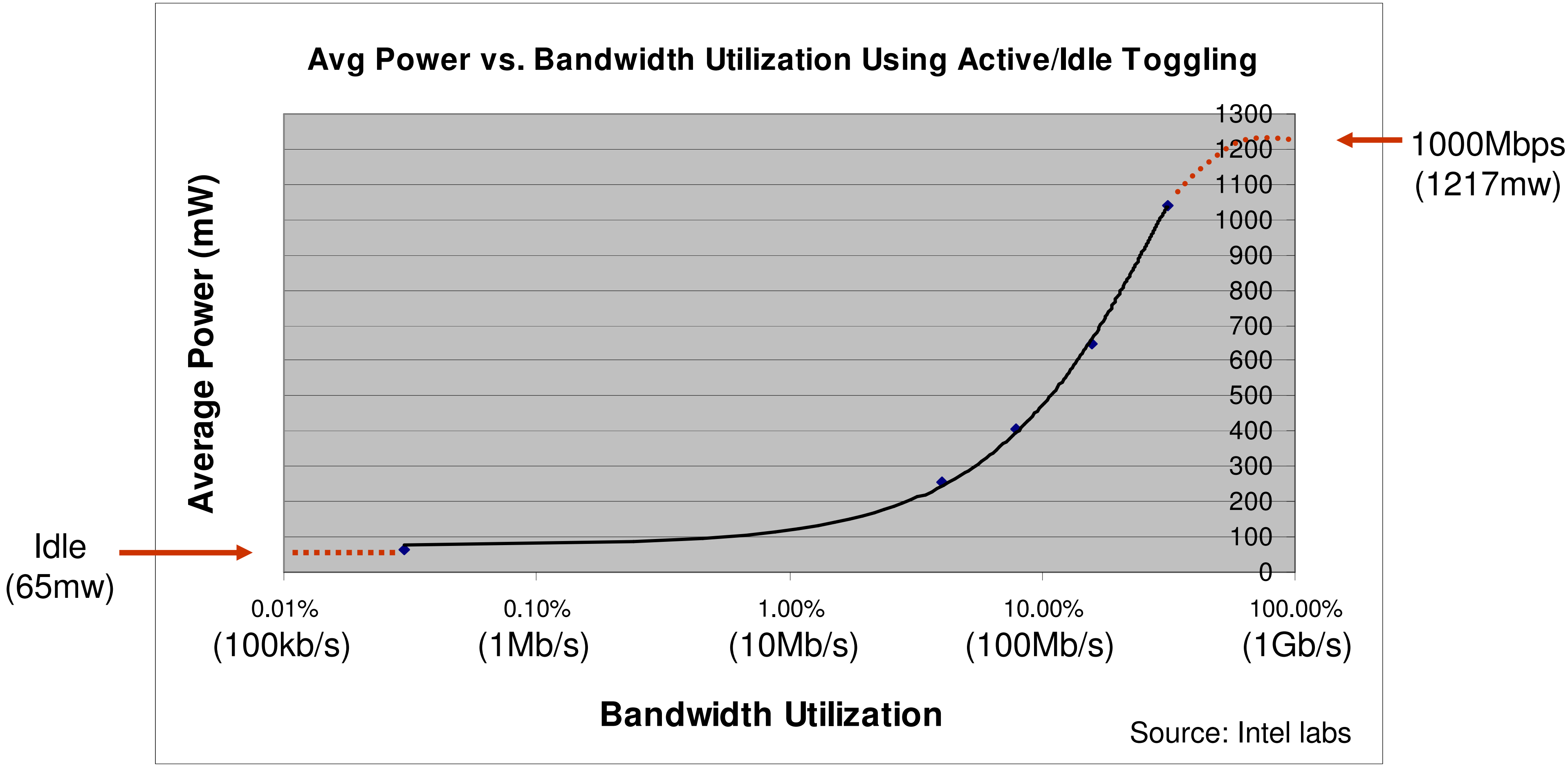


- **Fast-Start** offers course-grain power regulation via 10x PHY choices
- **Subset-PHY** allows finer-grain power steps utilizing new PHY modes
- **Active/Idle Toggling** with 0BASE-x allows smooth power averaging and lowest energy consumption on underutilized connections

Energy Efficient Ethernet

intel

Average Power vs. BW Utilization Simulation



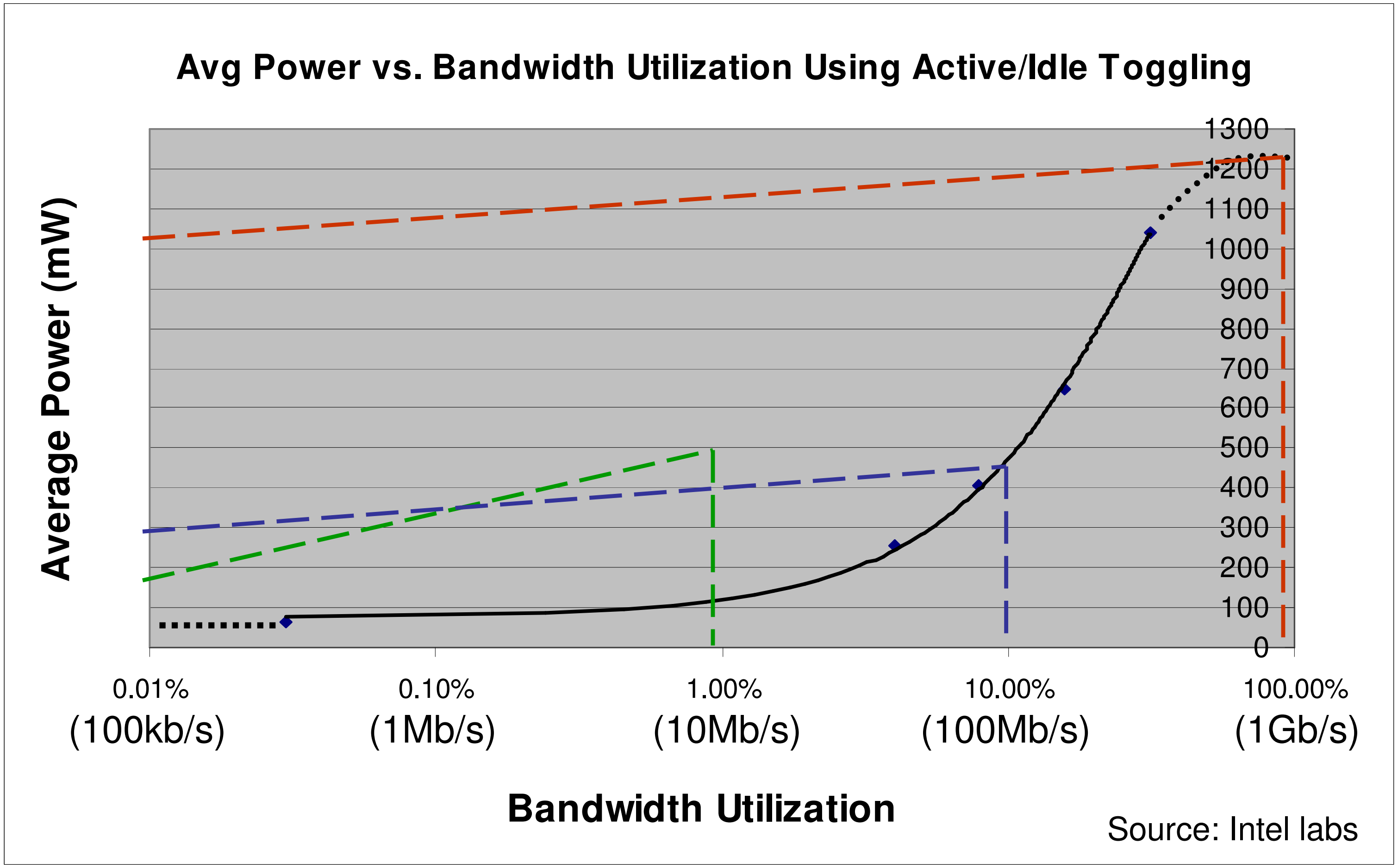
Input Assumptions:

- Traffic Input = Trace_VOIP_*.txt
- 1000Mbps Active Power = 1217mW
- Idle Power = 65mW
- Active (Resume) Delay = 10us
- Idle (Sleep) Wait Period = 10us
- Idle (Sleep) Delay = 1us

Energy Efficient Ethernet

intel

Comparison to Existing GbE Controller Power



Key (Active / Idle Power Assumptions):

Active/Idle Toggling Simulation (1217mW / 65mW)

1000BASE-T Measurements (1217mW / 1010mW)

100BASE-TX Measurements (483mW / 314mW)

10BASE-T Measurements (504mW / 194mW)

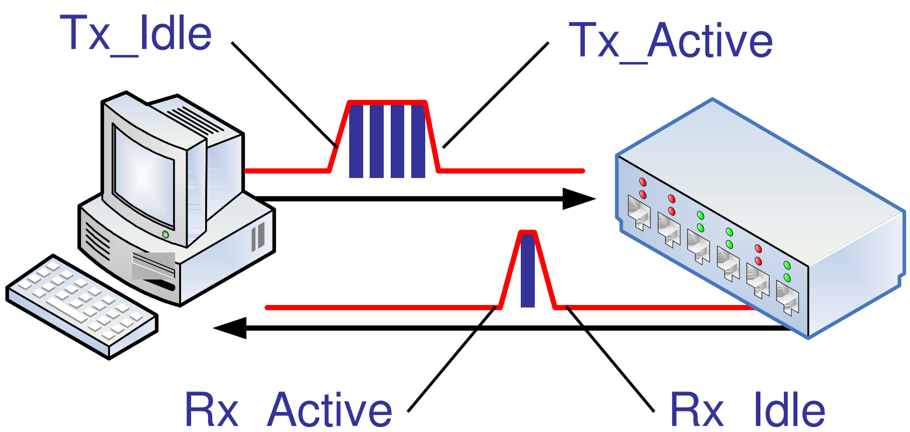
Energy Efficient Ethernet

intel

Simulation Model Contribution

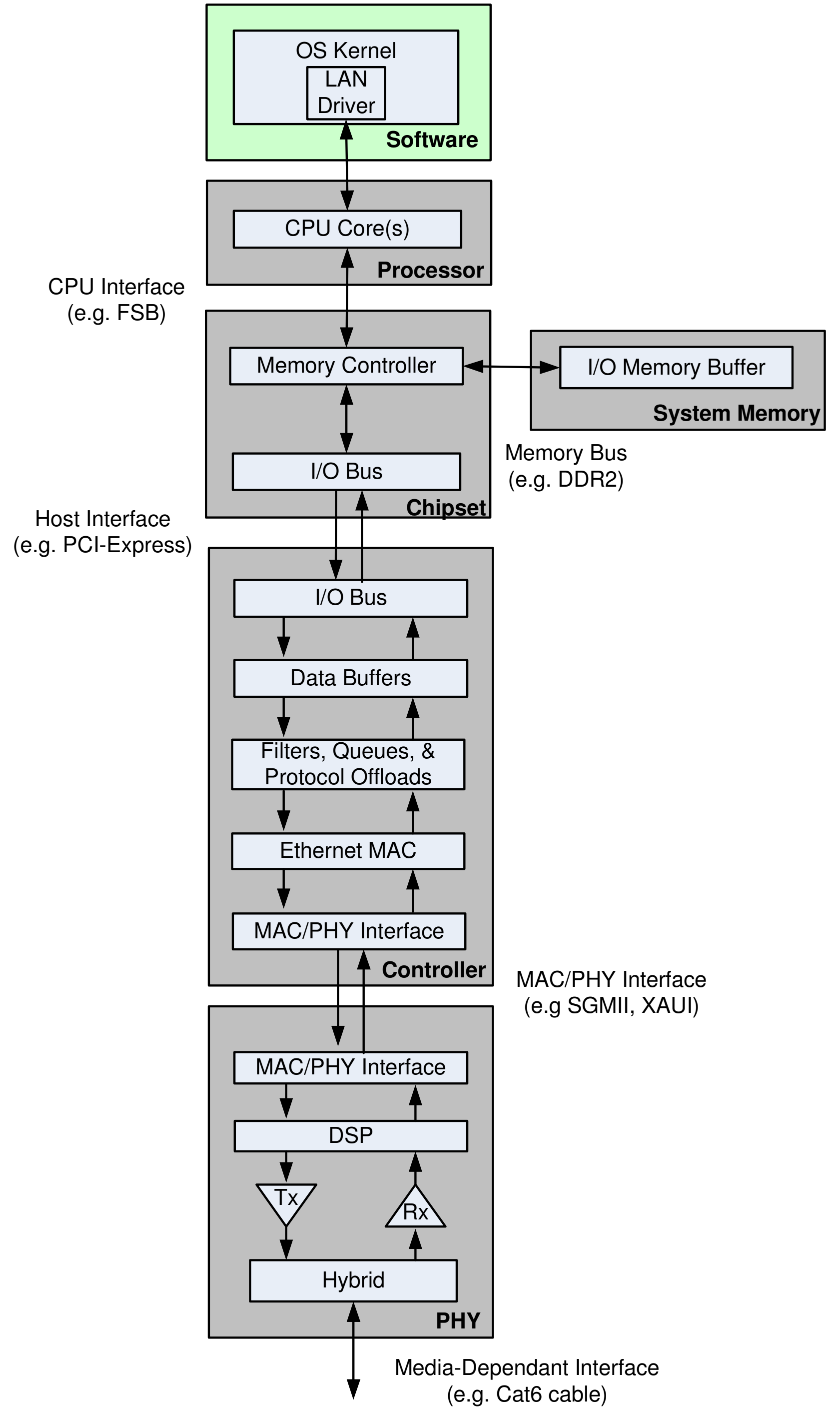
- Intel will contribute the 'Average Power vs. Bandwidth Utilization' simulation model to the IEEE 802.3az Task Force
- The "C" program source code and sample traffic pattern trace files will be posted on the EEE Tools web page:
  - http://grouper.ieee.org/groups/802/3/az/public/tools/index.html

Active/Idle State Transitions



Transition	Description	Transition Initiator
Tx_Active	Transmit data path resumes to Active when the system wants to send data	System Policy Manager (e.g. LAN Driver)
Tx_Idle	Transmit data path goes to Idle when there is no data to send	System Policy Manager (e.g. LAN Driver)
Rx_Active	Receive data path resumes to Active when link partner wants to send data	Link Partner
Rx_Idle	Receive data path goes to Idle when link partner has completed sending data	Link Partner

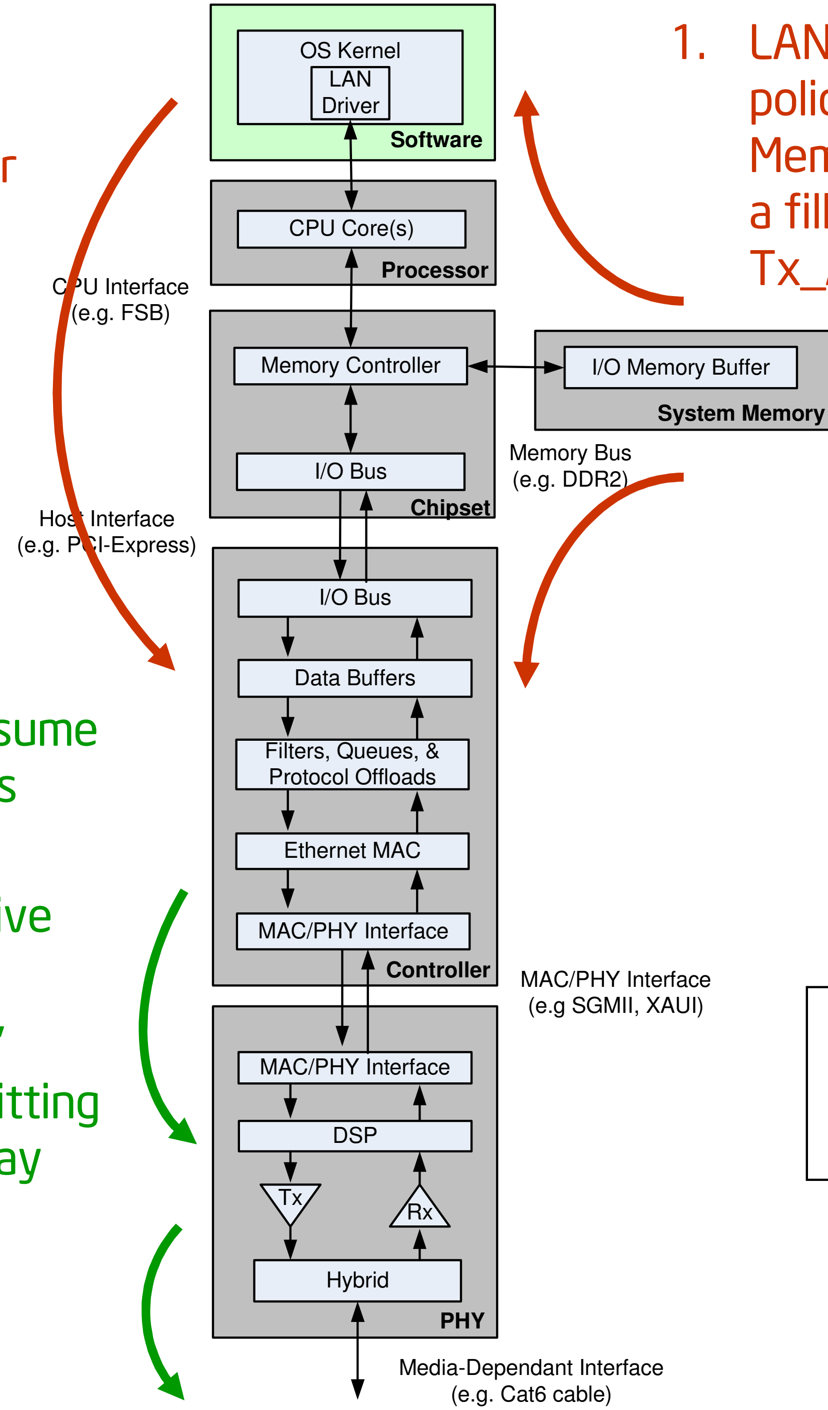
Computer Networking System Block Diagram



Energy Efficient Ethernet

Tx_Active Transition

1. LAN Driver manages Tx toggling policy. Example: It monitors I/O Memory Buffer for data to reach a fill threshold, then initiates Tx_Active
2. LAN Driver tells Controller when to enter Tx_Active
3. Controller enters Tx_Active and begins buffering data
4. Controller tells PHY to resume Tx_Active and then sends data to PHY
5. Local PHY enters Tx_Active
6. Local PHY sends it's link partner a "start-transmit" frame and begins transmitting data after a specified delay



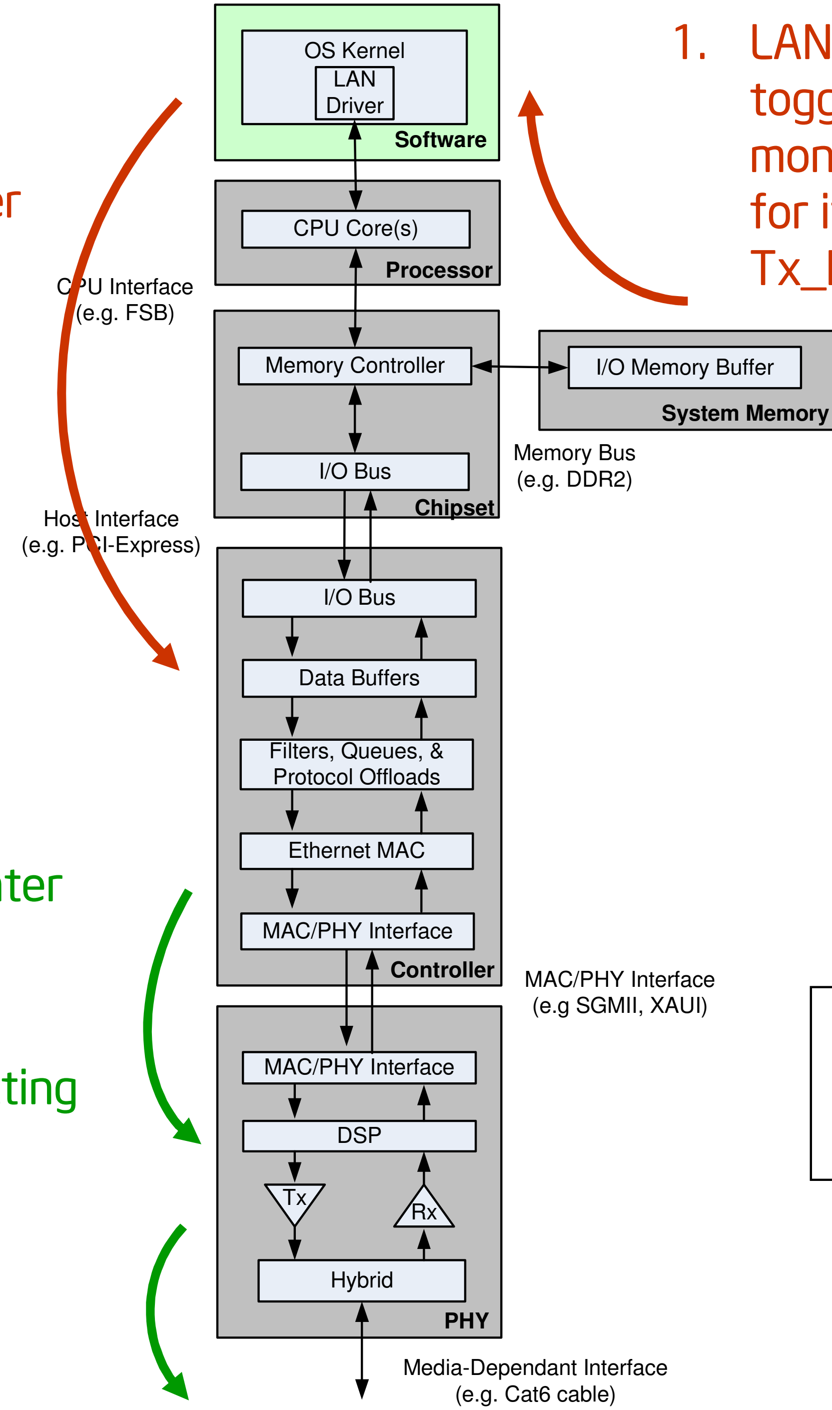
Key:
Vendor Dependent
802.3az Specified

Energy Efficient Ethernet

intel

Tx_Idle Transition

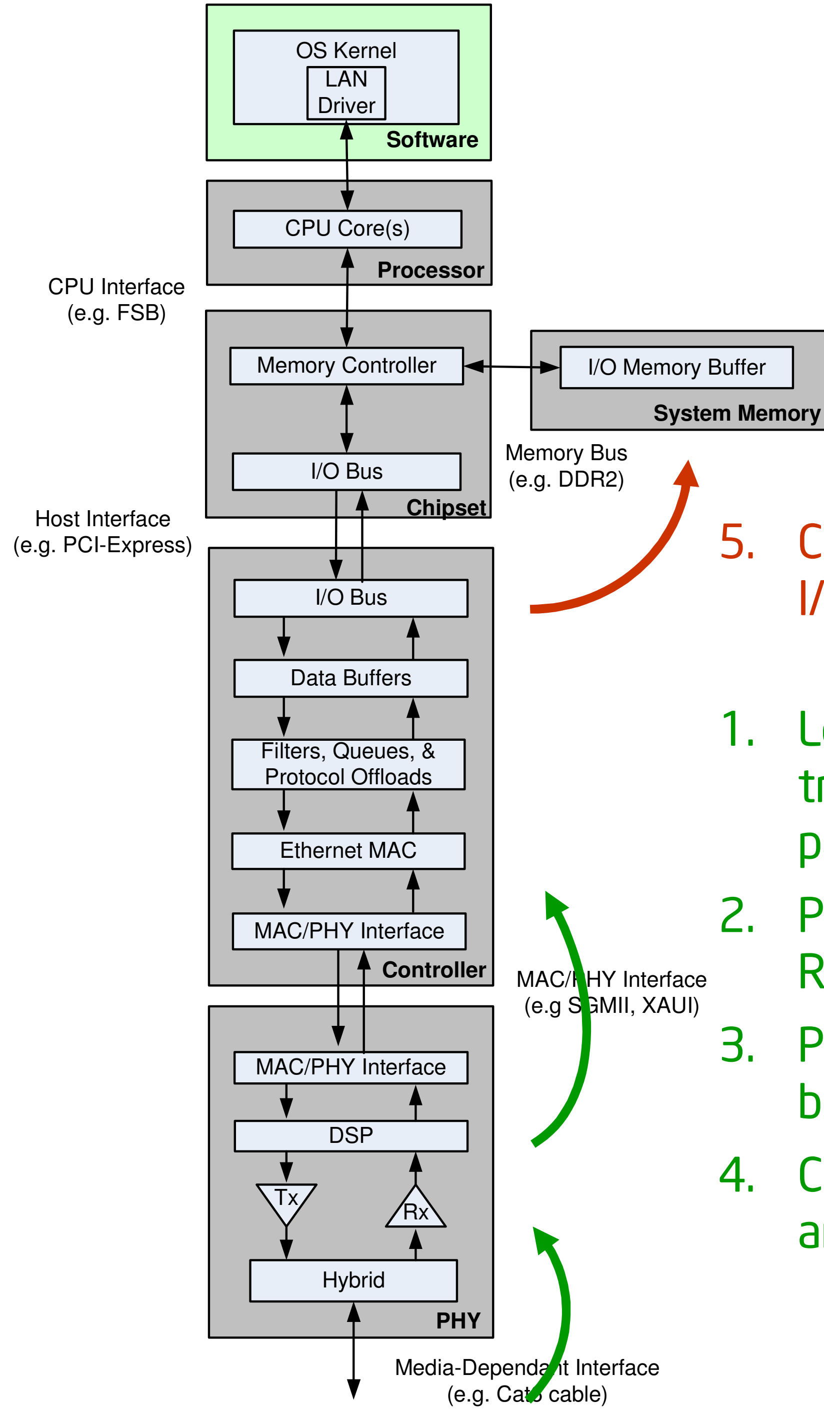
1. LAN Driver manages the Tx toggling policy. Example: It monitors I/O Memory Buffer for it to empty, then initiates Tx_Idle
2. LAN Driver tells Controller when to enter Tx_Idle
3. Controller tells PHY to enter Tx_Idle
4. Controller enters Tx_Idle
5. Local PHY stops transmitting and enters Tx_Idle



Key:
Vendor Dependent
802.3az Specified

Energy Efficient Ethernet

Rx_Active Transition



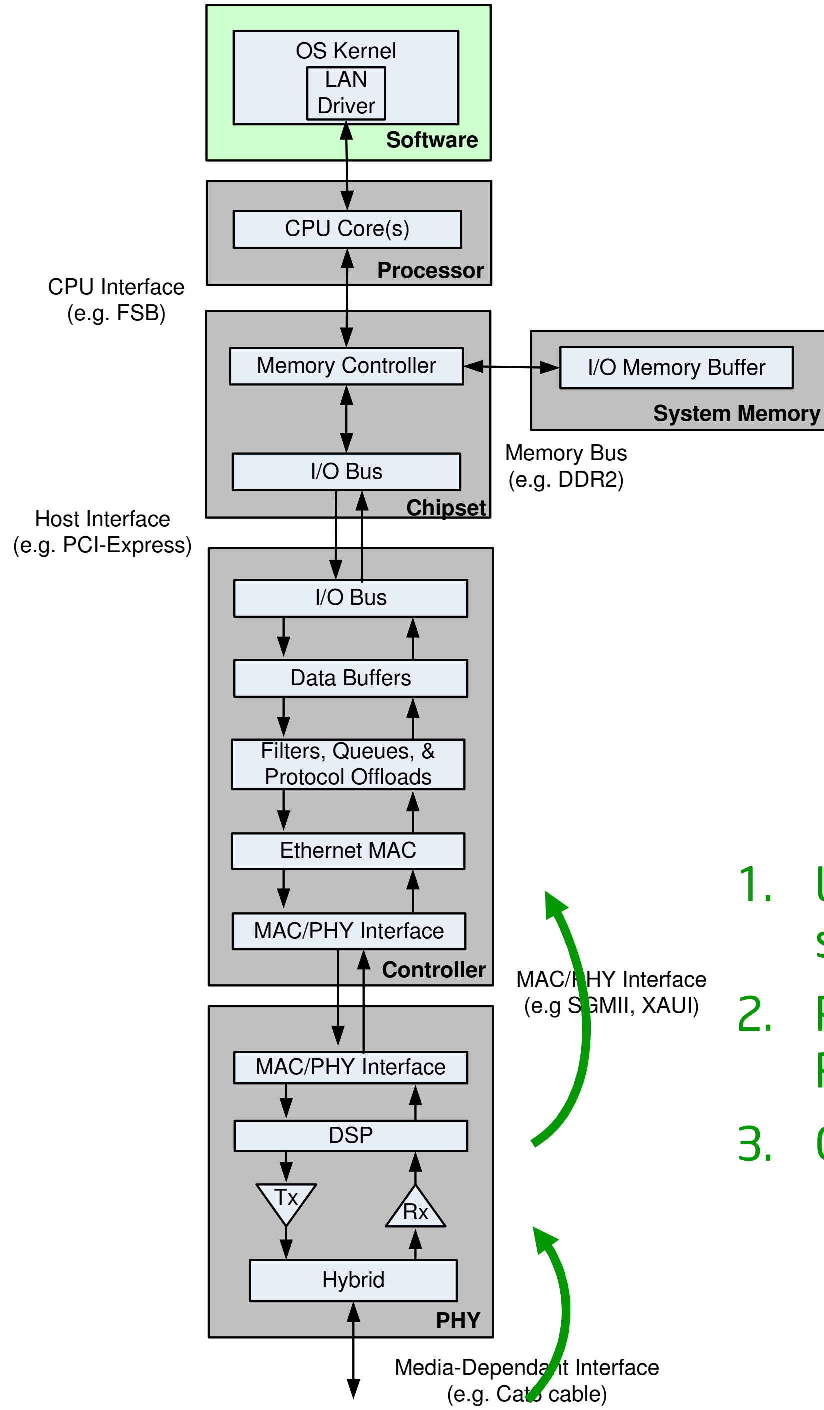
5. Controller places data into I/O Memory Buffer

1. Local PHY receives “start-transmit” frame from it’s link partner
2. PHY tells controller to enter Rx_Active
3. PHY enters Rx_Active and begins receiving data
4. Controller enters Rx_Active and receives data

Key:
Vendor Dependent
802.3az Specified

Energy Efficient Ethernet

Rx_Idle Transition



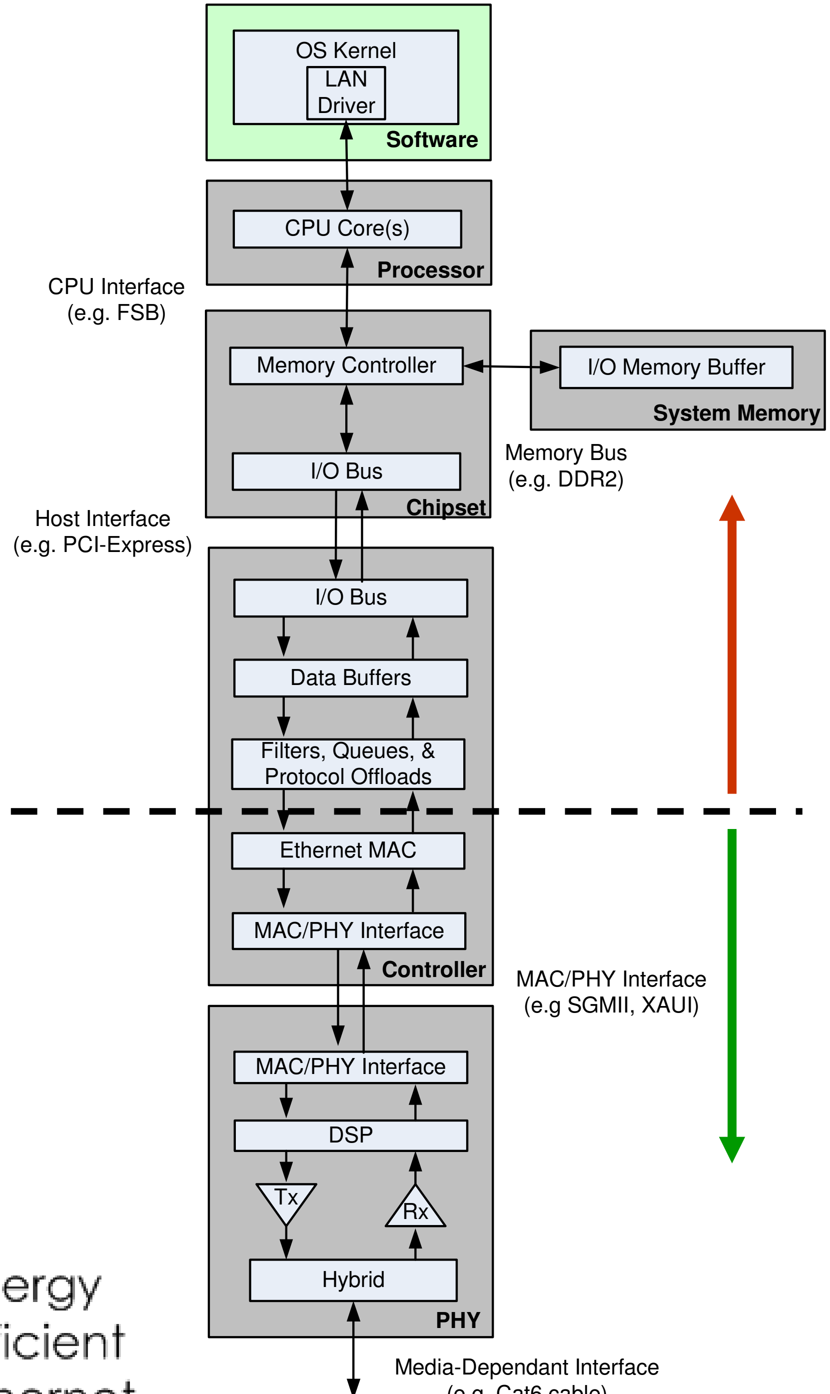
1. Local PHY detects Idle signal and enters Rx_Idle
2. PHY tells controller to enter Rx_Idle
3. Controller enters Rx_Idle

Key:
Vendor Dependent
802.3az Specified

Energy Efficient Ethernet

intel

Implications to IEEE 802.3 Specifications



Vendor dependent (unspecified):

- Tx state transition control & policy
- System interface & data movement

802.3 specifications:

- 0BASE-x signaling for each PHY type
- Active/idle transition signals
- Transition timing requirements
- MAC/PHY interface control
- Control/status registers for active & idle states
- Auto-negotiation capability registers
- Clause 30 MIB implications
- Error detection & recovery

Energy Efficient Ethernet

0BASE-x PHY Considerations*

1. Idle power consumption
   - How low can Idle get for each PHY type?
2. Transition delays
   - How quickly can the PHY resume active operation?
3. Timing recovery
   - How to maintain link status and clock sync?
4. Asymmetric operation
   - How to support transmission one way while the other is idle?
5. Implementation cost & complexity
   - How would 0BASE-x compare to Fast-Start or Subset-PHY?

*Note: These considerations are addressed in zimmerman_01_1107.pdf

Energy Efficient Ethernet

intel

Recommendations to 802.3az Task Force

- Define a 0BASE-x Idle state for each supported PHY type in the IEEE 802.3az Objectives

- Consider Active/Idle Toggling with 0BASE-x as an alternative to PHY Rate Shifting for Energy Efficient Ethernet