

User-Level Access to System Area Networks

Lambert Schaelicke



The OS Bottleneck

Role of OS

- Atomic device access
- Address translation & protection check
- Interrupt handling

Hardware Support

- Conditional Store Buffer
- Device TLB
- User-level notifications



Conditional Store Buffer

Pack arguments into a single bus transaction.

- Detect sequence of: store – store – store – ... – flush
- Reset buffer when sequence is interrupted
- **User-level control over store combining**
- **Lightweight, non-blocking synchronization**



I/O Device TLB

Translate addresses at the I/O device.

- Use OS page tables
- TLB with programmable tablewalk engine
- **User-level DMA with arbitrary application buffers**
- **No page pinning or OS protection check needed**



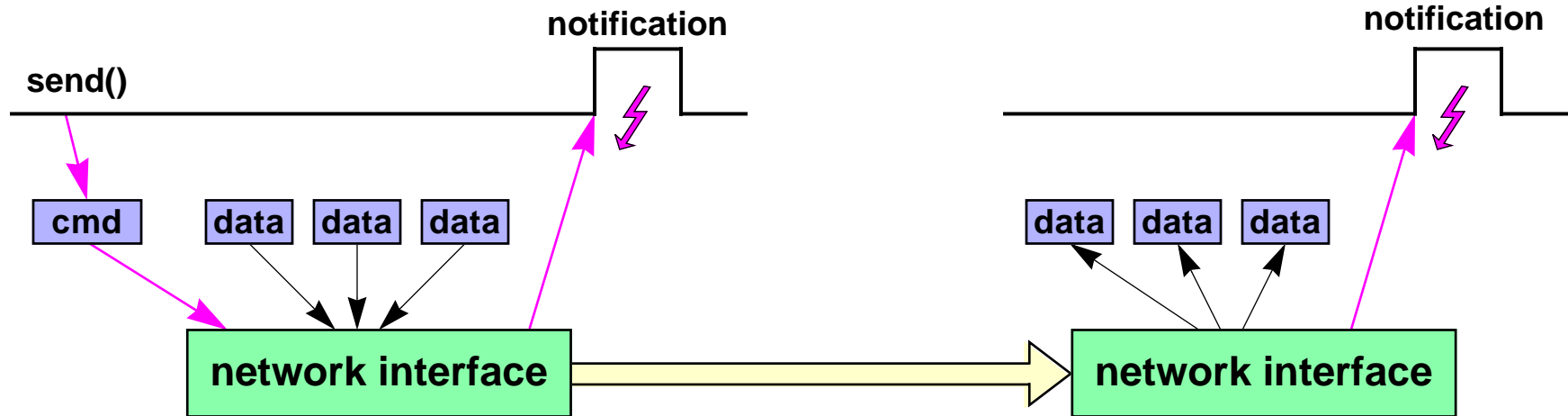
Low-overhead Interrupts

Provide detailed information to identify interrupt cause.

- Interrupt transaction contains handler address w/ arguments
- Hardware or OS jumps to handler in user space
- **Low-overhead notification**
- **Fast dispatch to user-level notification threads**



Programming Interface



- Non-blocking send request supplies handler address
- Handler synchronizes upon transfer completion
- Existing models implemented in libraries



Summary

- **Inexpensive hardware features for**
 - Atomicity
 - Address translation
 - Application notification
- **Provide unrestricted user-level device access**
- Work in progress – many open questions – no results yet
- <http://www.cs.utah.edu/~lambert>



Device TLB

Inexpensive TLB

- Operates only at bus speed
- No single-cycle access needed

Programmable table walk engine

- Small and simple instruction set
- Code downloaded by OS at boot-time



User-level Notifications

- Deliver only notifications for successful transfers
- Does not require application response
- CPU jumps to address provided by interrupt
Problem: can't use normal interrupt hardware to save state
- Fast OS handler saves state and calls user routine
Problem: user-routine may not return (e.g., longjmp)



Existing Solutions

- User-level DMA with shadow addresses
not necessarily atomic, does not allow extra arguments
- No DMA (Memory Channel) – low performance for bulk messages
- Pre-arranged buffer space – restrict programming model
- User-TLB – places burden upon programmer

