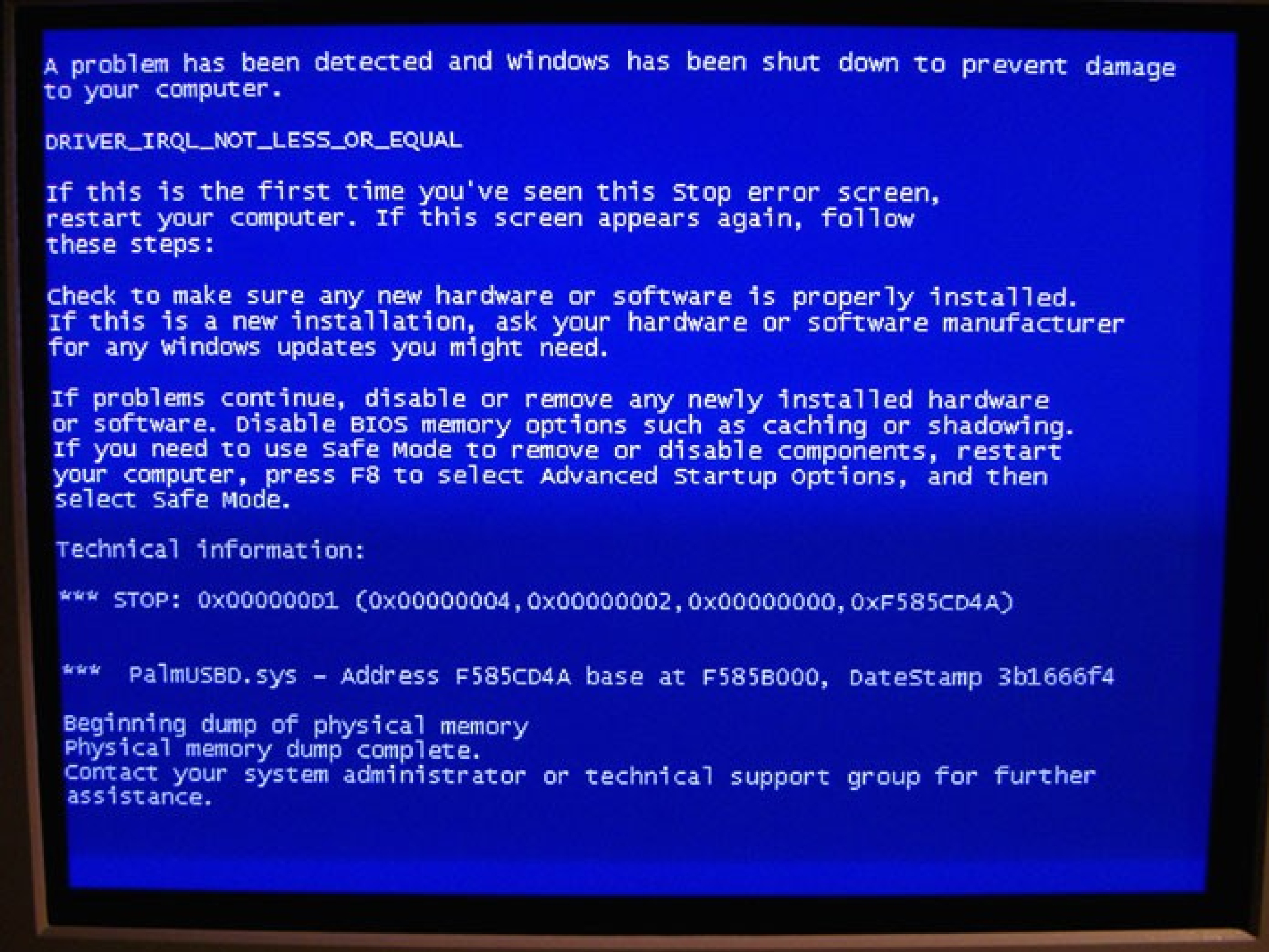
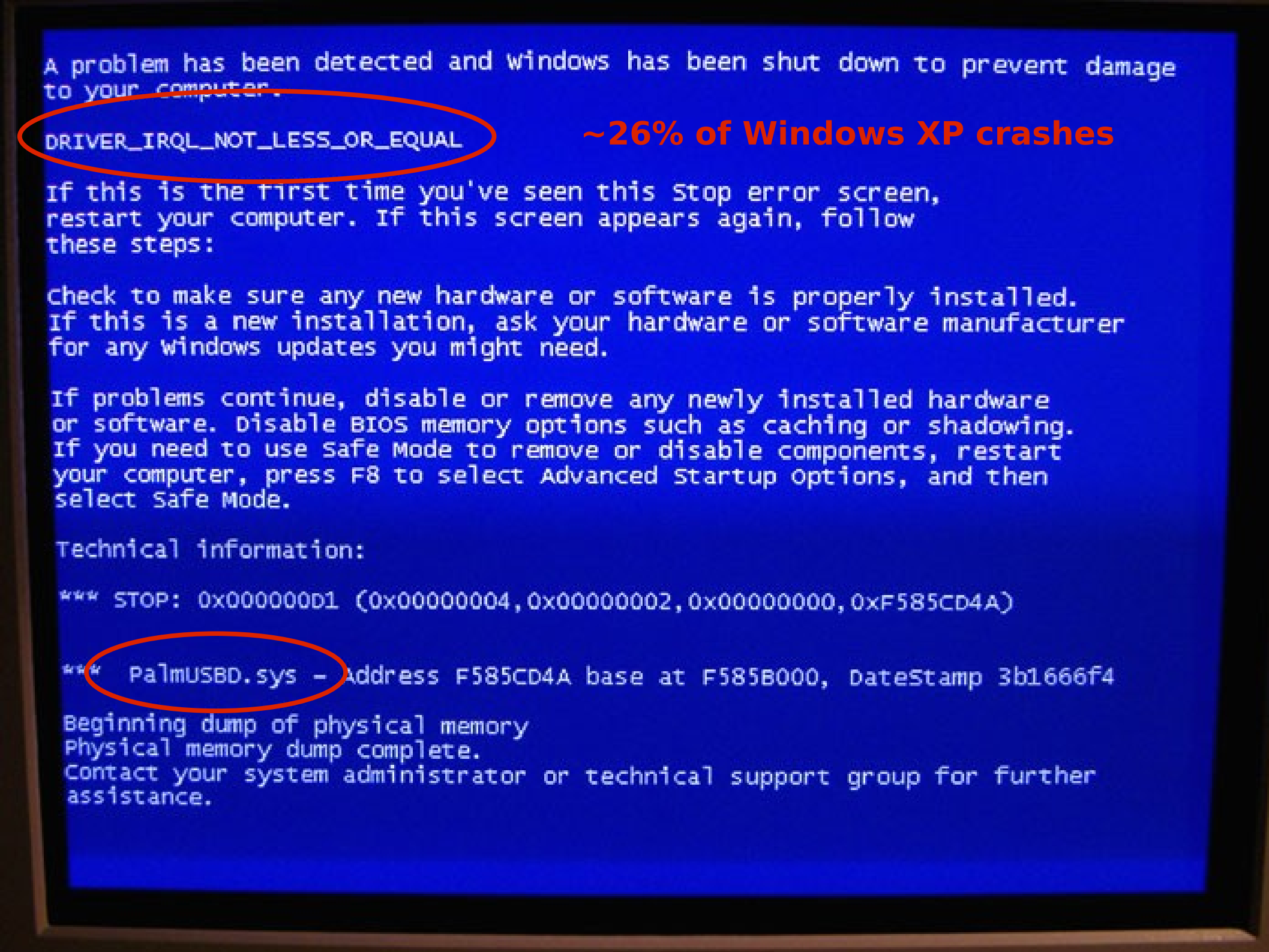
Fault Isolation for Device Drivers

39th International Conference on Dependable Systems and Networks, 30 June 2009, Estoril–Lisbon, Portugal

Jorrit N. Herder
Vrije Universiteit Amsterdam


A problem has been detected and Windows has been shut down to prevent damage
to your computer.
DRIVER_IRQL_NOT_LESS_OR_EQUAL
If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:
Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.
If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.
Technical information:
*** STOP: 0x000000D1 (0x00000004,0x00000002,0x00000000,0xF585CD4A)
*** PalmUSBD.sys - Address F585CD4A base at F585B000, DateStamp 3b1666f4
Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further
assistance.


A problem has been detected and Windows has been shut down to prevent damage
to your computer.
DRIVER_IRQL_NOT_LESS_OR_EQUAL
~26% of Windows XP crashes
If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:
Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.
If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.
Technical information:
*** STOP: 0x000000D1 (0x00000004,0x00000002,0x00000000,0xF585CD4A)
*** PalmUSBD.sys - Address F585CD4A base at F585B000, DateStamp 3b1666f4
Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further
assistance.

Talk outline

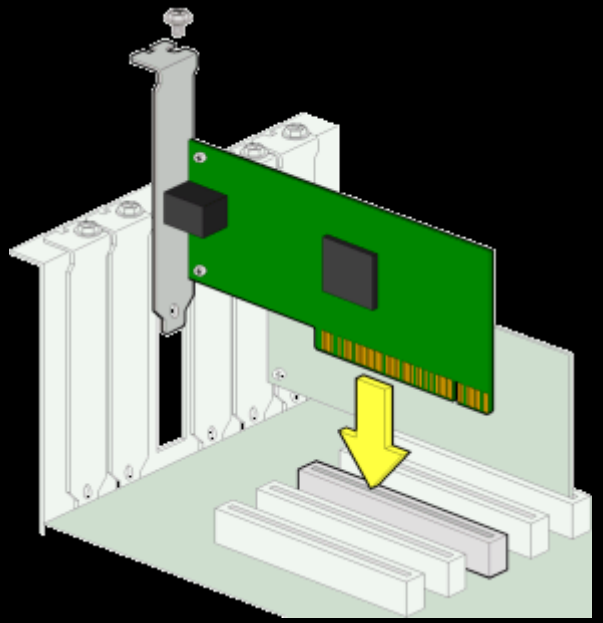
- Driver dependability threats
- MINIX 3 isolation architecture
- MINIX 3 self-repairing properties
- Experimental evaluation
- Summary and conclusion

Talk outline

- Driver dependability threats
- MINIX 3 isolation architecture
- MINIX 3 self-repairing properties
- Experimental evaluation
- Summary and conclusion

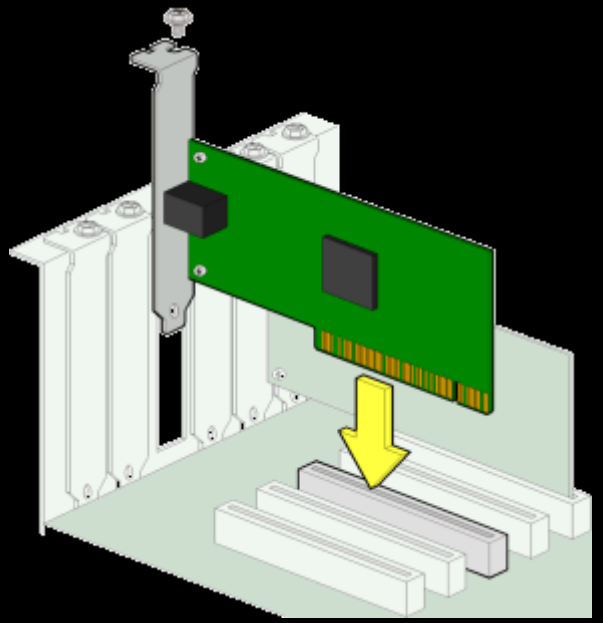
Even if OS were correct ...

- Device drivers OS base functionality
  - provided by untrusted third parties
  - comprise up to 70% of entire OS
  - 3-7x more bugs than other OS code



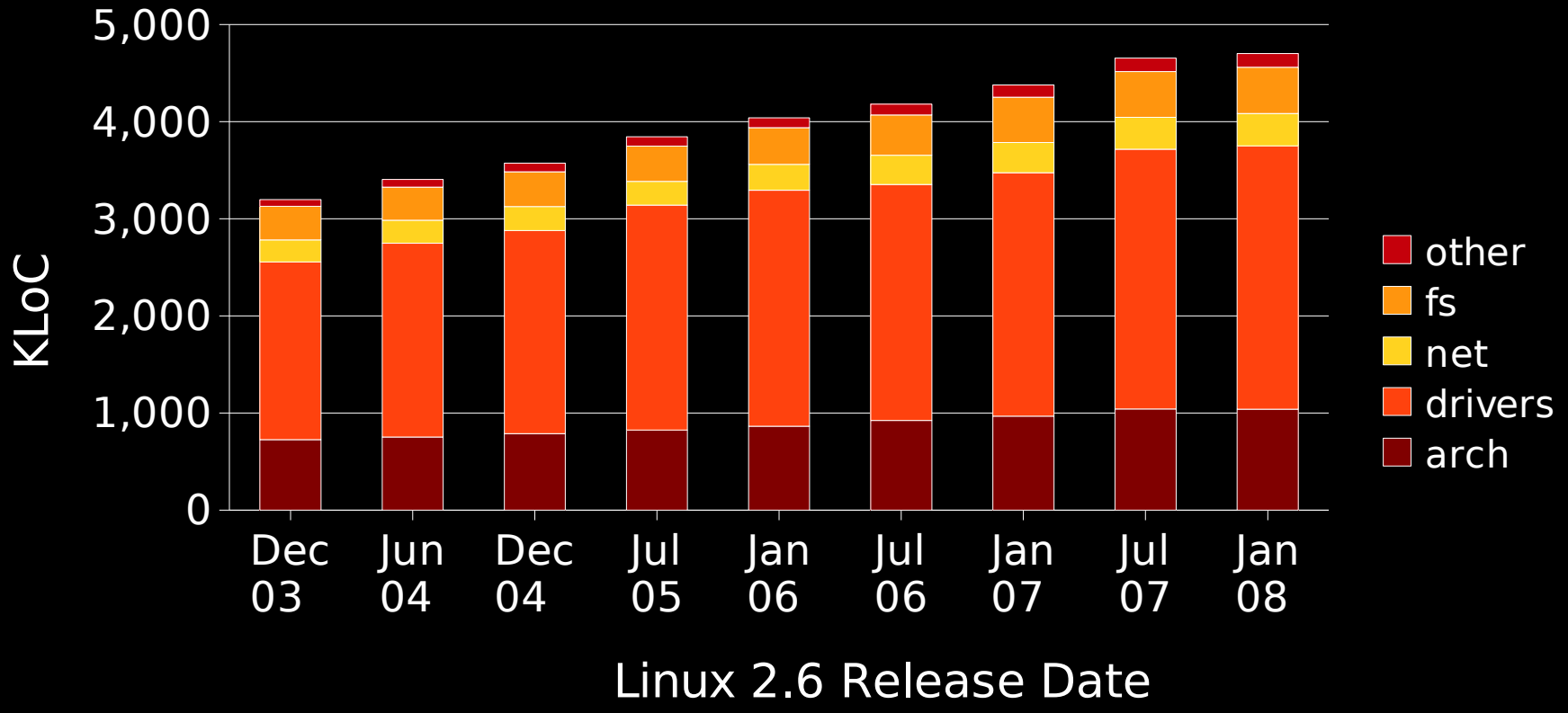
Even if OS were correct ...

- Device drivers OS base functionality
  - provided by untrusted third parties
  - comprise up to 70% of entire OS
  - 3-7x more bugs than other OS code
- Still, drivers run in kernel
  - all powers of the system
  - no proper fault isolation



Bug fixing is infeasible

- Continuously changing configuration
- Maintainability of drivers is very hard



Consequences

- Downtime mainly due to faulty software
  - over 25,000 kernel bugs in Linux/Windows
    - with 5 MLoC kernel and 5 bugs/KLoC
  - not all the code is in use all the time
    - still, *any* kernel bug is potentially fatal

- Windows crash dump analysis confirms:
  - extensions cause 65-83% of all crashes

Talk outline

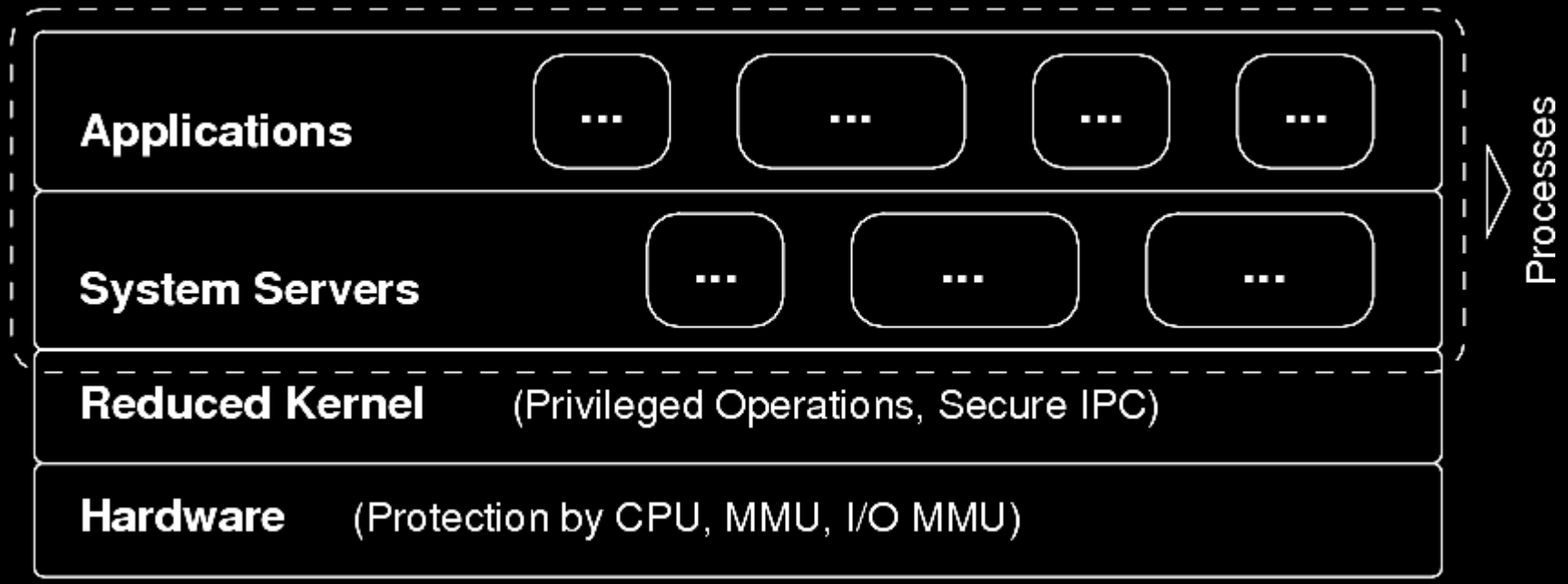
- Driver dependability threats
- MINIX 3 isolation architecture
- MINIX 3 self-repairing properties
- Experimental evaluation
- Summary and conclusion

Isolation architecture

- Goal is to enforce least authority
  - only grant access needed to do job
    - e.g., disk driver can access only disk controller
- This is realized using a combination of
  1) structural constraints
  2) per-driver isolation policies
  3) run-time memory granting

Structural constraints

- Multiserver design compartmentalizes OS
- Only microkernel has full CPU privileges
  - manageable due to small size < 5,000 LoC



User-level drivers

- Drivers encapsulated in user processes
  - unprivileged CPU mode
    - cannot change page tables, halt CPU, etc.
  - strict address-space separation
    - memory corruption causes 27% of OS crashes
- Kernel mediates privileged operations
  - e.g., DEVIO kernel call mediates device I/O
  - e.g., SAFECOPY mediates memory copies

Per-driver isolation policies

```
driver rtl8139 {                    # isolation policy
    pci device      10ec/8139;      # RTL8139 PCI card
    ipc             kernel          # Kernel task
                    ...;            # ...
    ipc kernel      DEVIO           # Device I/O
                    SAFECOPY        # Memory copying
                    ...;            # ...
};
```

- Note: MINIX 3 provides only mechanisms

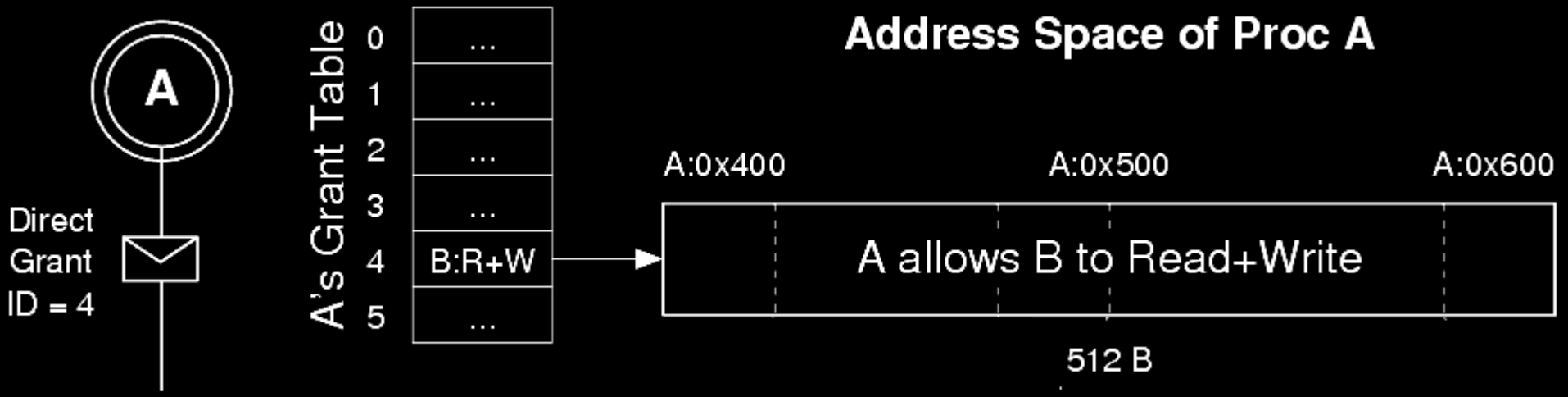
Policy enforcement

- Driver manager installs isolation policy
  1) driver manager forks a new process
  2) OS servers are informed about policy
  3) driver binary can be executed safely
- Policy enforcement done by OS
  - run-time checks for privileged requests

Run-time memory granting

- Address-space separation too strict
  - drivers typically need to exchange data
- Static policies not suitable for memory
  - buffers often are dynamically allocated
- Therefore, run-time memory granting

Memory grants



- Grants are capabilities
  - grant defines precise memory access rights
  - grantor sends grant ID to grantee
  - grant validated by SAFECOPY kernel call

Direct memory access (DMA)

- DMA-capable devices can access memory
- Protection based on IOMMU hardware
  - IOMMU verifies requests from device
    - just like MMU verifies requests from driver
- Trusted driver grants DMA access
  - IOMMU driver programs IOMMU hardware
    - DMA allowed into only driver's address space

Talk outline

- Driver dependability threats
- MINIX 3 isolation architecture
- MINIX 3 self-repairing properties
- Experimental evaluation
- Summary and conclusion

Self-repairing properties

- Isolation prevents fault propagation
  - cannot prevent buggy driver from failing
- Improve availability through recovery
  - driver manager monitor drivers at run-time
  - driver is restarted if a failure is detected
    - often transparent to application and users

How does this work?

- Many faults tend to go away after restart
  - For example:
    - transient hardware faults
    - race condition due to timing issues
    - aging bugs due to memory leaks
- Details of recovery described elsewhere
  - “Failure Resilience for Device Drivers,” Proc. 37th DSN, pp. 41-50, June 2007

Talk outline

- Driver dependability threats
- MINIX 3 isolation architecture
- MINIX 3 self-repairing properties
- Experimental evaluation
- Summary and conclusion

Dependability testing

- Goal: “Show that errors occurring in an isolated device driver cannot propagate and damage the rest of the OS.”

- Method: “Use software-implemented fault injection (SWIFI) to induce driver failures, and observe how the OS is affected.”

SWIFI setup

- Faults representative for common errors
  - bad pointers, infinite loops, etc.
- Inject fault into text segment at run-time
  - based on variant of UNIX process tracing
- Workload may cause fault activation
  - triggered faults may cause driver failures

Observed robustness

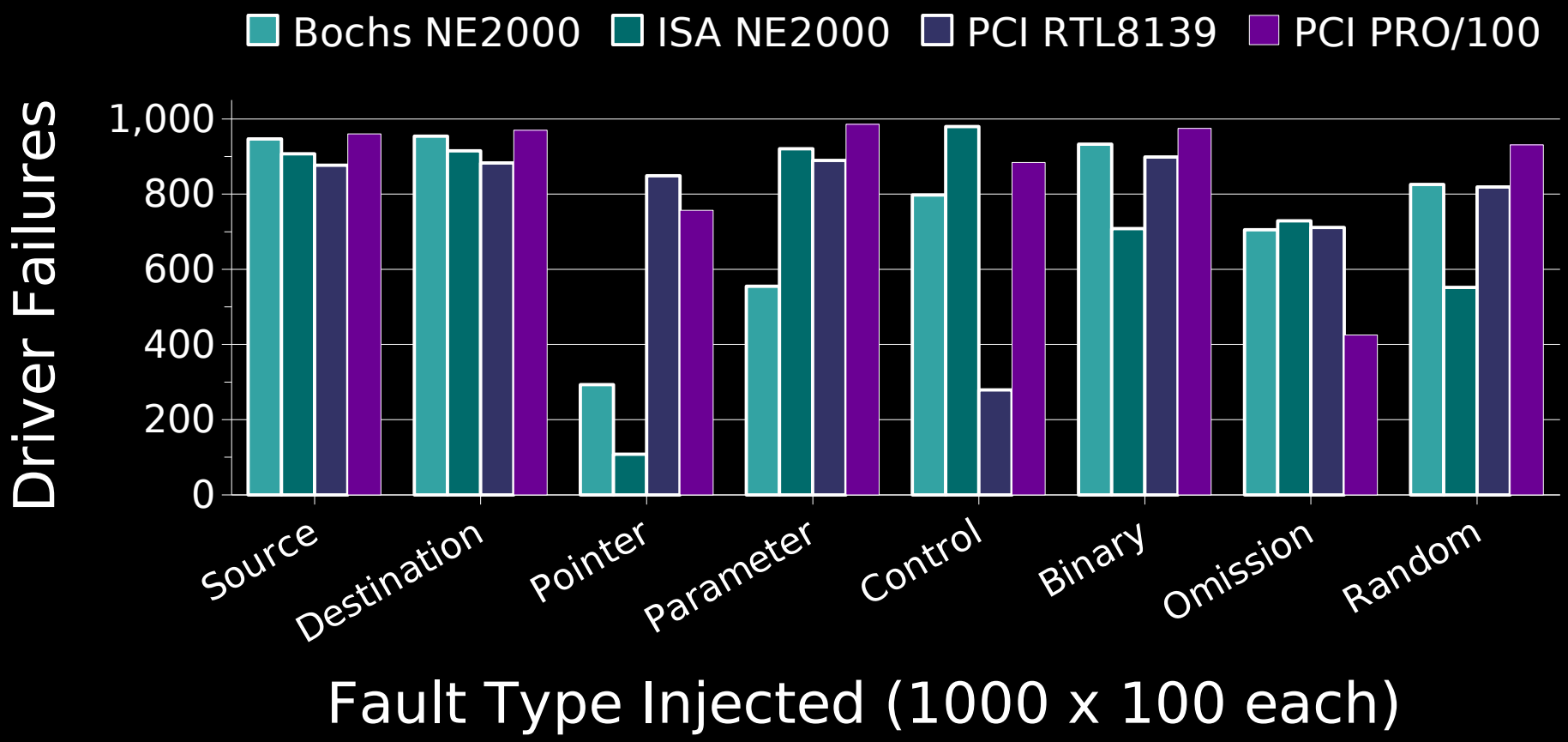
- One experiment injected 3,200,000 faults
  - 4 network driver configurations
    - ISA and PCI bus, programmed I/O and DMA
  - 8 fault types * 1000 trials * 100 faults/trial
    - induced driver failure with high probability

Observed robustness

- One experiment injected 3,200,000 faults
  - 4 network driver configurations
    - ISA and PCI bus, programmed I/O and DMA
  - 8 fault types * 1000 trials * 100 faults/trial
    - induced driver failure with high probability
- Results indicate success!
  - SWIFI caused 24,883 detectable errors
    - driver failed, but the OS was never affected
  - transparent recovery in all 24,883 cases

Distribution of fatal errors

- See paper for explanation of fault types



Unauthorized accesses

- RTL8139 displayed 5887 fatal errors
  - detected and repaired by driver manager
    - CPU & MMU exceptions
    - exit due to internal panic
    - missing driver heartbeat
- 3 orders of magnitude more other errors
  - access attempt denied and logged by kernel
    - unauthorized device I/O (1,754,886 x)
    - unauthorized kernel call (322,005 x)
    - unauthorized IPC request (66,375 x)

Hardware limitations

- Sometimes hardware could not be reset
  - device lacked master-reset command
    - <0.1% of all NE2000 ISA driver crashes
- Sometimes the entire system froze
  - misbehaving device caused PCI bus hang
    - had to give up on RTL8029 PCI card

Hardware limitations

- Sometimes hardware could not be reset
  - device lacked master-reset command
    - <0.1% of all NE2000 ISA driver crashes
- Sometimes the entire system froze
  - misbehaving device caused PCI bus hang
    - had to give up on RTL8029 PCI card

- Note: not a shortcoming of our design

Talk outline

- Driver dependability threats
- MINIX 3 isolation architecture
- MINIX 3 self-repairing properties
- Experimental evaluation
- Summary and conclusion

Conclusion (1/2)

- Drivers threaten OS dependability
- Fault isolation prevents global failure
  - structural constraints
  - per-driver isolation policy
  - run-time memory granting
- Failure resilience repairs local damage
  - monitor driver failures at run-time
  - automated, transparent recovery

Conclusion (2/2)

- Fault-injection testing proves viability
  - first-ever to inject millions of faults
    - needed to find the nasty bugs as well
- Observed few hardware limitations
  - not dramatic, but not fixable in software
    - hardware dependability must also improve
- Demonstrated effectiveness of design
  - achieved 100% transparent recovery
    - for 3,200,000 faults injected into 4 drivers

Thank you!

- Co-authors
  - Herbert Bos
  - Ben Gras
  - Philip Homburg
  - Andrew S. Tanenbaum
- Try it yourself!
  - download MINIX 3
    - www.minix3.org