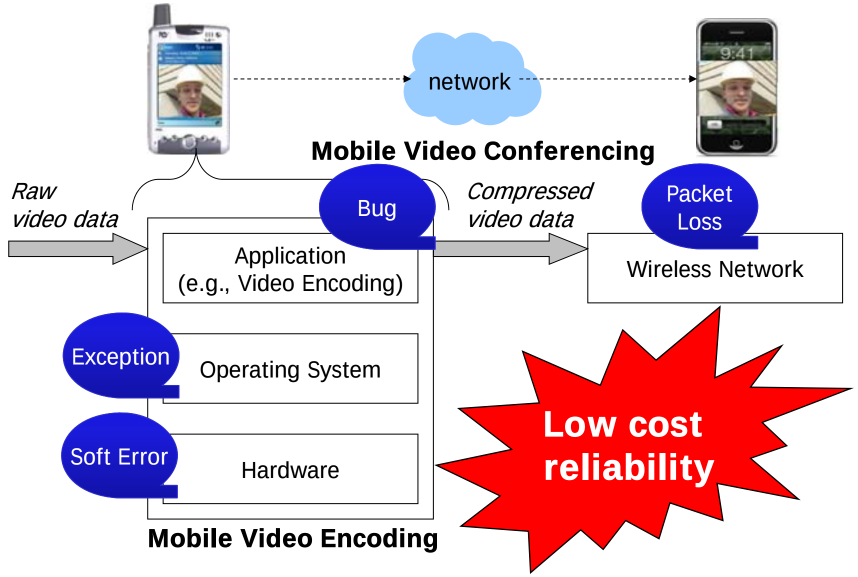
CCPROTECT: Cooperative, Cross-layer Protection for Resource Constrained Mobile Multimedia Systems

Yonsei University
Dept. of Computer Science
Kyoungwoo Lee

Joint Work with Prof. Aviral Shrivastava in ASU, Dr. Minyoung Kim in SRI, and Prof. Nikil Dutt and Prof. Nalini Venkatasubramanian in UCI

Mobile Multimedia System

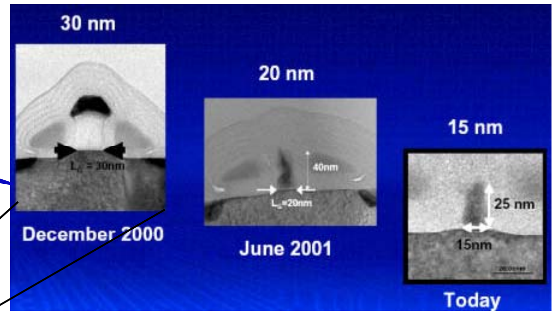


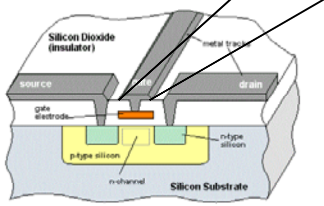
Multimedia Mobile Devices are Popular

Resource-limited mobile devices!
Main problem is to achieve low power with high performance, high QoS, and high reliability

Scaling Drives Technology Advancement

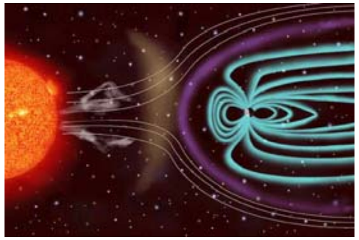
Scaling: The Transistor Gate shrinks in size every year

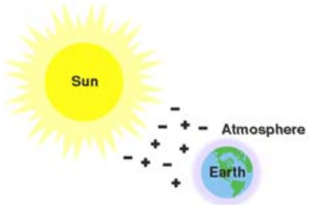




Smaller device dimensions improve on performance and reduce power consumption

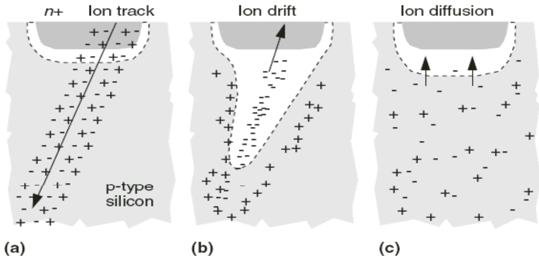
Reliability - a consequence: Soft Errors





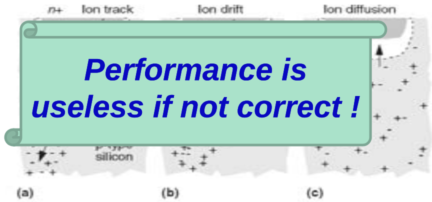
Electrical disturbances can disrupt the operation causing ***Transient Faults***

Soft Error
= Transient Fault
= Bit Flip (memory)



Soft Errors - an Increasing Concern

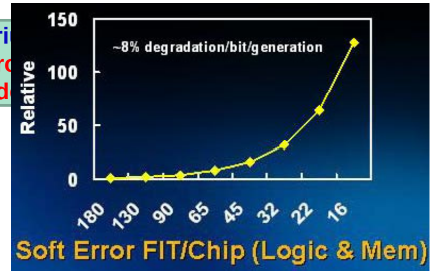
- Charge carrying particles induce Soft Errors
  - Alpha particles
  - Neutrons
    - High energy (100KeV -1GeV)
    - Low energy (10meV – 1eV)



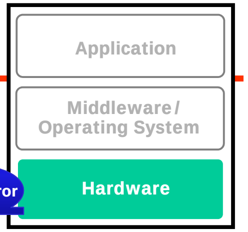
Performance is useless if not correct !

Toyota Pri… as the pro… unintend…

- Soft Error Rate
  - Is now 1 per year
  - Exponentially increases with technology scaling

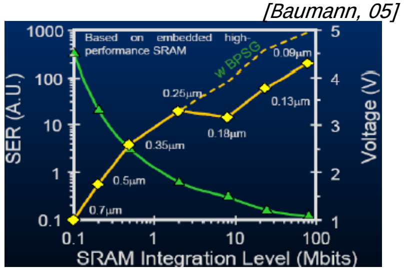


Soft Errors on an Increase



- Soft error rate (SER) increases exponentially as technology scales
- Integration, voltage scaling, altitude, latitude, etc.

[Baumann, 05]



$$SER \propto N_{flux} \times CS \times exp\left\{-\frac{Q_{critical}}{Q_s}\right\}$$

where $Q_{critical} = \text{Capacitance} \times \text{Voltage}$

•N_{flux}: Neutron flux intensity, CS: Area of cross section, Q_S: Charge collection efficiency

5 hours MTTF

1 month MTTF

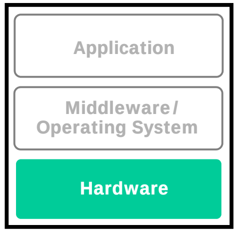
•MTTF: Mean Time To Failure

Soft Error is an Every Second Concern

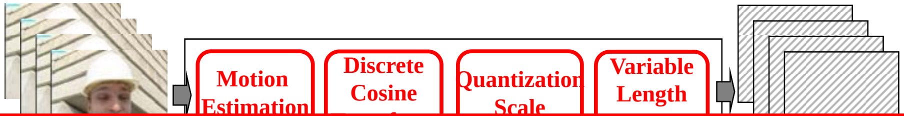
- Soft Error Rate (SER)
  - FIT (Failures in Time) – How many errors in one billion operation hours
    - SER per Mbit @ 0.13 μm = 1,000 FIT ≈ 104 years in MTTF
- Soft error is becoming an every second problem

	SER (FIT)	MTTF	Reason
1 Mbit @ 0.13 μm	1000	**104 years**	
64 MB @ 0.13 μm	**64x8x**1000	**81 days**	High Integration
128 MB @ **65 nm**	**2x1000x**64x8x1000	**1 hour**	Technology scaling and Twice Integration
A system @ 65 nm	**2x**2x1000x64x8x1000	**30 minutes**	Memory takes up 50% of soft errors in a system
A system **with voltage scaling** @ 65 nm	**100x**2x2x1000x64x8x1000	**18 seconds**	Exponential relationship b/w SER & Supply Voltage
A system with voltage scaling **@ flight (35,000 ft)** @ 65 nm	**800x**100x2x2x1000x64x8x1000 FIT	**0.02 seconds**	High Intensity of Neutron Flux at flight (high altitude)

Caches and Video Encoding

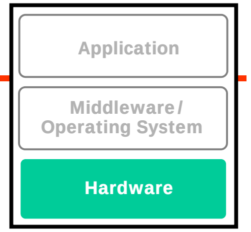


- Soft error rate is proportional to the **time and area** to be exposed *[Cai, 06]*
  - Soft error rate (SER) is measured in FIT (Failures in Time) per unit size
  - SER = 1,000 FIT per Mbit for SRAM
    - The larger memory system, the higher SER
    - The longer the execution, the higher SER
- Caches are most hit due to:
  - Larger portion in processors (more than 50%)
- Video encoding consists of complex algorithms
  - Also, processes the huge amount of video data

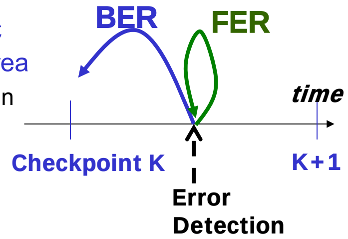


Video encodings are time-intensive and memory-intensive, thus very vulnerable to soft errors

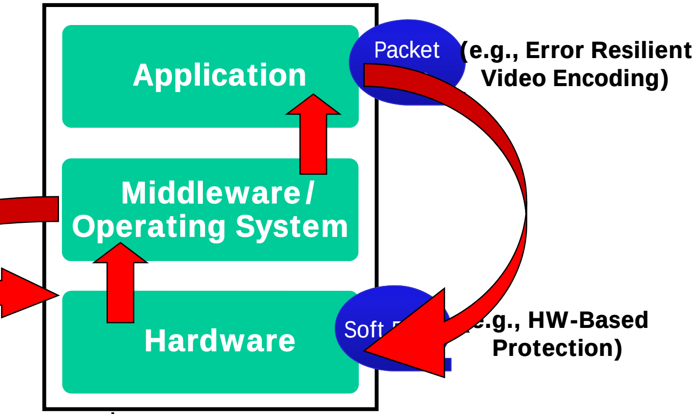
Soft Error Protection Within-HW



- ECC (Error Correction Codes)
  - Forward Error Recovery (**FER**)
  - ECC incurs high overheads in terms of:
    - power (22% [Phelan,03]), performance (95% [Li,05]), and area (25% [Kreuger,08])
- Conventional micro-architectural techniques **within hardware layer** still exploit ECC
- EDC (Error Detection Codes)
  - EDC is much less expensive than ECC in terms of power, performance, and area
    - up to 73% less in power and 47% less in performance than ECC [Li, 04]
  - Need to correct the detected error
    - ➔Checkpoints and Roll backward (**BER** – Backward Error Recovery)
    - Bad for real-time requirement



Cross-Layer Approach?



- Cross-layer approach
  - Integrate and coordinate techniques across system layers in a cooperative manner for system optimization
- Can we **coordinate** within-layer approaches **across layers** to combat errors for minimal cost **reliability**?

Related Cross-Layer Work

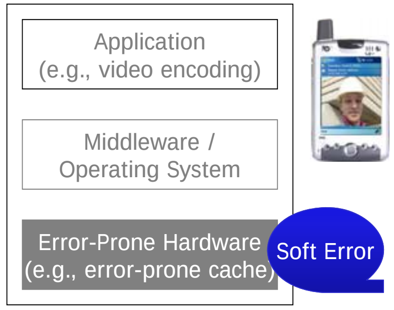
- GRACE project @ UIUC [W. Yuan Ph.D. thesis in '04 and A. F. Harris III, Ph.D. thesis in '06]
  - QoS/Power tradeoffs
    - Primarily OS adaptation for power management in multimedia mobile devices
    - Network adaptation for power management in multimedia communications
- DYNAMO middleware for FORGE project @ UCI [S. Mohapatra Ph.D. thesis in '05 and R. Cornea Ph.D. thesis in '07]
  - QoS/Power tradeoffs for mobile embedded systems
  - Middleware-driven coordination and proxy-based cooperation
    - Content transcoding at the application layer
    - Network traffic shaping at the network layer
    - Backlight (LCD display) setting at the hardware layer
    - NIC shutdown, CPU DVS/DFS at the hardware layer
- xTune framework @ UCI and SRI [M. Kim Ph.D. thesis in '08]
  - QoS/Power/Timeliness adaptation for distributed real-time embedded systems
  - A Formal Methodology for cross-layer tuning and verifiable timeliness of Mobile Embedded Systems
- Our Contribution
  - QoS/Power/**Reliability** system optimization for mobile multimedia embedded systems
  - Use cross-layer approach to provide **reliability** with minimal cost

Outline

- ❑ Motivation and Related Work
- ❑ **Problem Statement**
- ❑ **Our Solution**
  - ➢ **CC-PROTECT – Cooperative Cross-Layer Protection**
  - ➢ **Mitigate the impact of soft errors with minimal cost**
- ❑ Experiments
- ❑ Conclusion

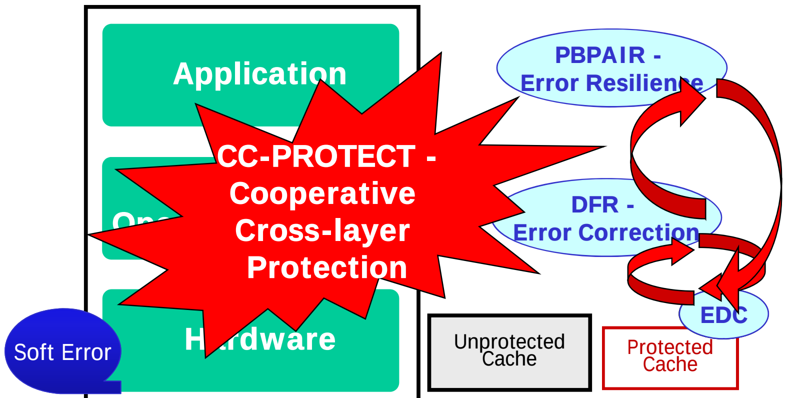
Problem Statement and Our Goals

- ❑ Soft Errors on Caches for Video Encoding
  - ➢ Soft errors are transient faults at hardware layer
  - ➢ SER is becoming a critical concern as technology scales
  - ➢ Caches are most hit
  - ➢ Video encoding is time-intensive and memory-intensive
- ❑ Impact of Soft Errors
  1. Failures
  2. Quality Degradation
- ❑ **Problem**
  - ➢ Develop Cross-Layer approach to mitigate the impact of soft errors
    1. Reducing the failure rate
    2. Minimizing the quality loss
  - ➢ Minimize the cost (power and performance)



Mobile Video Encoding

CC-PROTECT Overview



Previously, Hardware-based Error Protection (ECC, etc.)

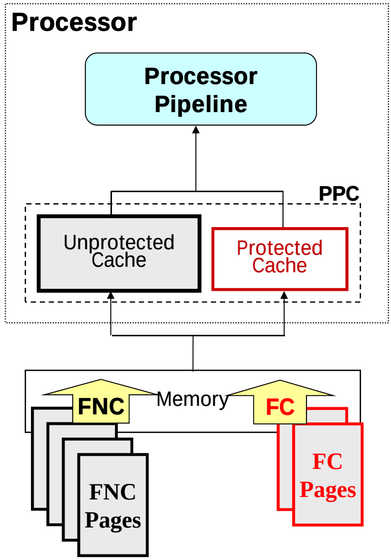
•ECC: Error Correction Codes
•EDC: Error Detection Codes
•DFR: Drop and Forward Recovery
•PBPAIR: Probability-Based Power Aware Intra Refresh

Failure Mitigation

- ❑ Goal 1 – Reduce soft error induced **failures**

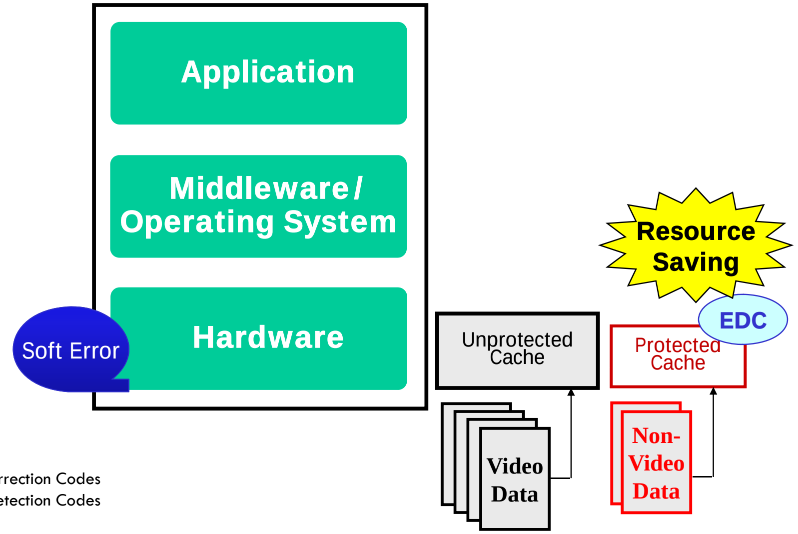
Partial Cross-Layer Protection – PPC

- PPC (Partially Protected Caches) [Lee, 06]:
  - One protected cache
    - ECC, etc.
    - Typically smaller
  - The other unprotected cache
- Compiler
  - Maps failure-critical (FC) data into the protected cache
  - Maps failure-non-critical (FNC) data into the unprotected cache
- Still **incurs overheads** due to high expensive ECC protection
  - 29% energy reduction compared to the protected cache
  - **10%** energy overhead compared to the unprotected cache



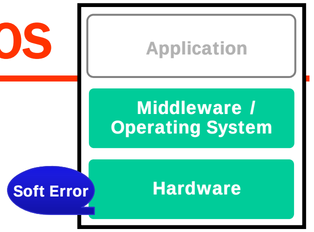
K. Lee, et al., "Mitigating soft error failures for multimedia applications by selective data protection", CASES, Oct 2006.

PPC with EDC at Hardware

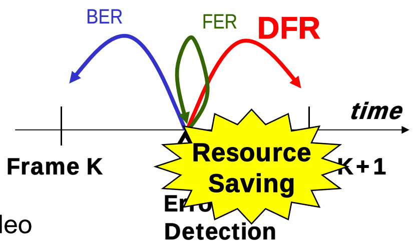


•ECC: Error Correction Codes
•EDC: Error Detection Codes

DFR across HW & MW/OS



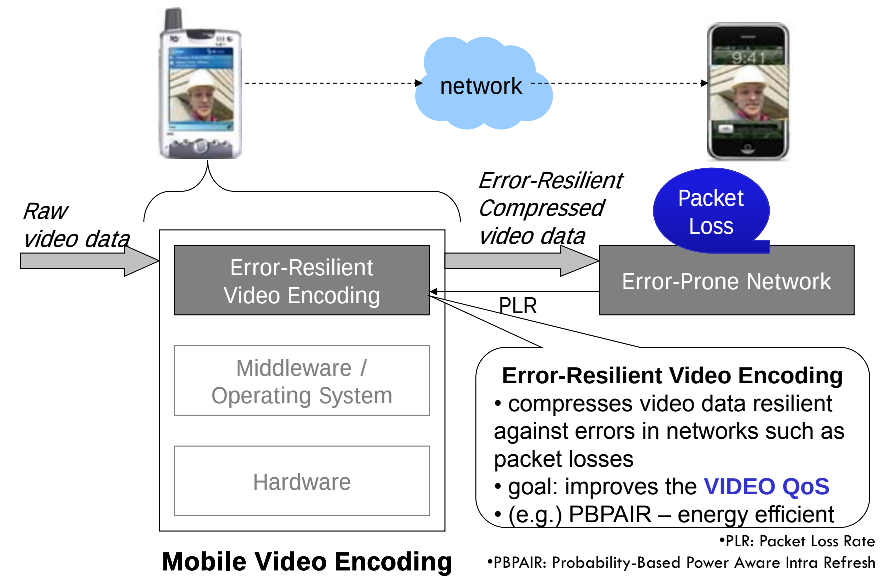
- Drop and Forward Recovery (**DFR**) at video encoding
  - Transform components into the **next** correct state
    - (e.g.) detect an error and move **forward to the next frame encoding**
    - **BER rolls backward**
  - Especially, well-suited for multimedia applications
    - Hardware defects will be managed by DFR (with **timeliness**)
    - **Quality** degradation due to DFR will be minimized by inherent error-tolerance of video data



Mitigation of QoS Degradation

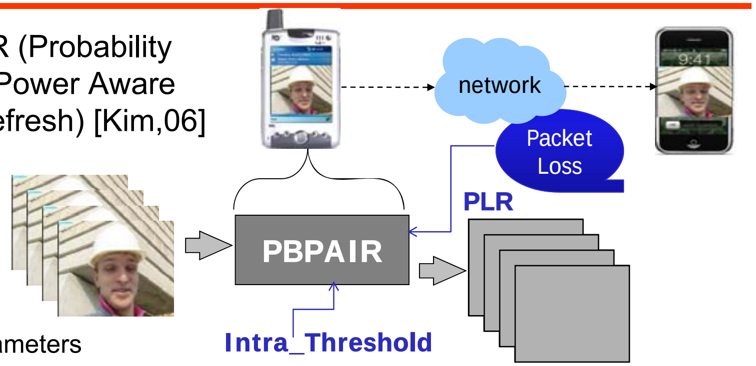
- Goal 2 – Mitigate **quality degradation** due to soft errors and frame drops

Resilience to Network-induced Packet Losses



PBPAIR – Error Resilient Video Encoding

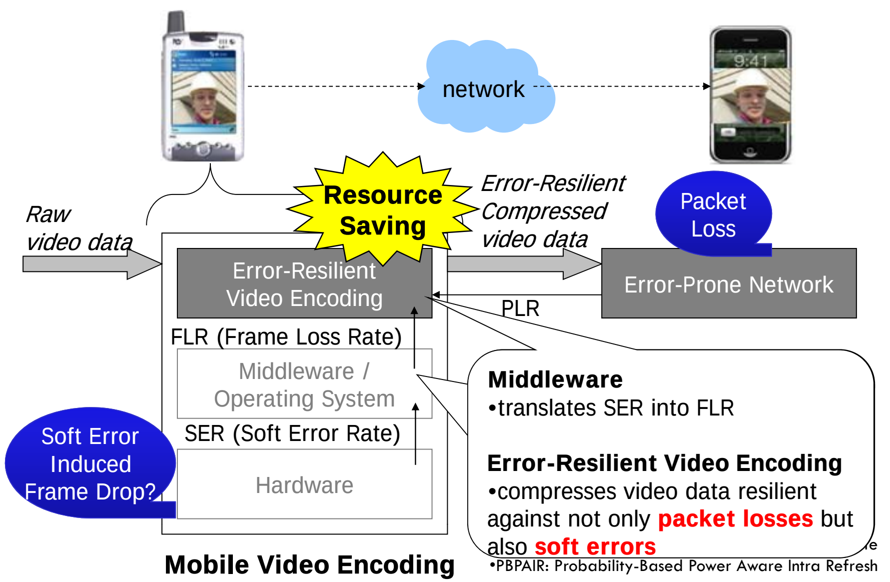
- PBPAIR (Probability Based Power Aware Intra Refresh) [Kim,06]



- Two Parameters
  1) PLR (Packet Loss Rate) – Network Status
     - **The higher PLR, the more intra macro blocks**
  2) Intra_Threshold – User-level Resilience Request
     - **The higher Intra_Threshold, the more intra macro blocks**
- Error resilient and energy efficient video encoding
  - Tradeoffs among energy efficiency, compress efficiency, and QoS
    - **Up to 34% energy reduction compared to previous encodings at 10% PLR**

Minyoung Kim, et al., PBPAIR: An energy-efficient error-resilient encoding using probability based power aware intra refresh", ACM SIGMOBILE MCCR, 2006

Resilience to Soft Error induced Frame Drops

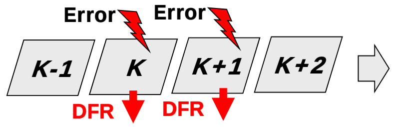


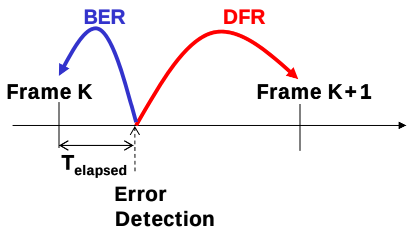
Translation from SER to FLR

- $N_{SE} = S_{cache} \times N_{inst} \times R_{SE}$
  - N_{SE} is the number of soft errors per frame encoding
  - S_{cache} is the size of caches in KB
    - 32 KB unprotected cache and 2 KB protected cache for a PPC in our study
  - N_{inst} is the number of instructions for one frame encoding
    - ACET (Average Case Execution Time) is used in our study
  - R_{SE} is a soft error rate per KB and per instruction
    - 10^{-11} per KB and per instruction is used in our study (accelerated by several orders of magnitude)
- N_{SE} is converted into % value, which is FLR
  - (e.g.) $N_{SE} = 32 \times 10^9 \times 10^{-11} = 0.32$ → FLR = 32%

Adaptive CC-PROTECT

- ❑Naïve DFR
  - ➢Always DFR when an error is detected
  - ➢Significant quality degradation
- ❑Adaptive DFR/BER
  - ➢Slack-Aware DFR/BER
    - ❑Depends on elapsed time
  - ➢Frame-Aware DFR/BER
    - ❑Depends on frame importance
  - ➢QoS-Aware DFR/BER
    - ❑Depends on feedbacked video quality





<u>if</u> $QoS_{feedback}$ < $QoS_{requirement}$

BER

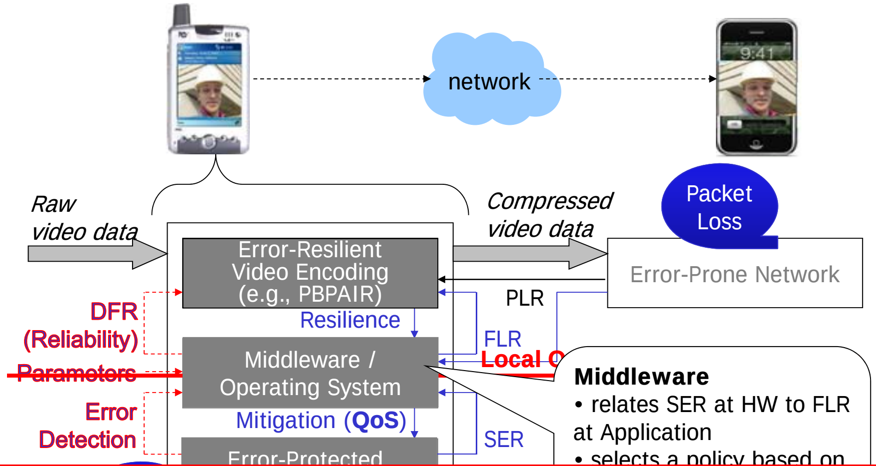
<u>else</u>

DFR

Where $QoS_{feedback}$ is from decoding side

•ACET: Average Case Execution Time

CC-PROTECT – Cross-Layer Protection

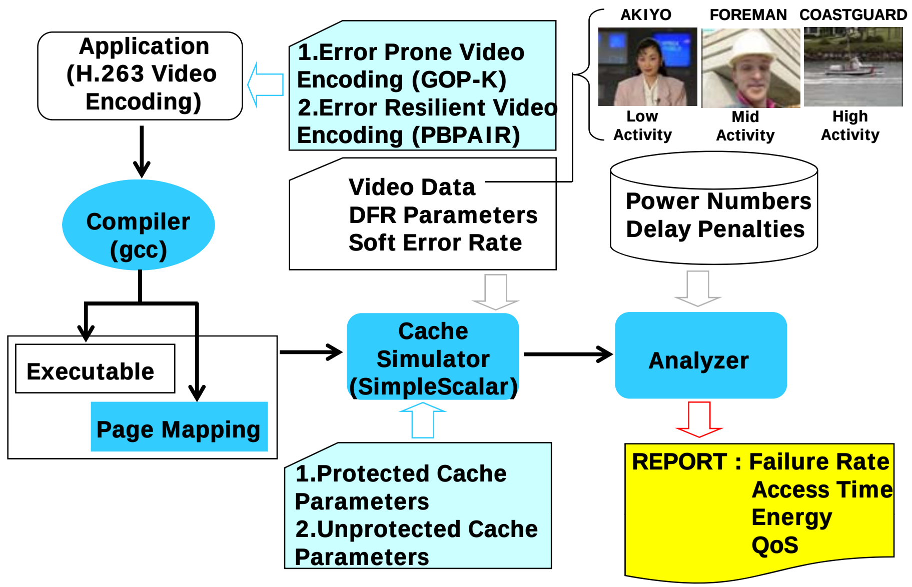


CC-PROTECT
1. achieves system-level optimization
2. extends the applicability of existing schemes

Outline

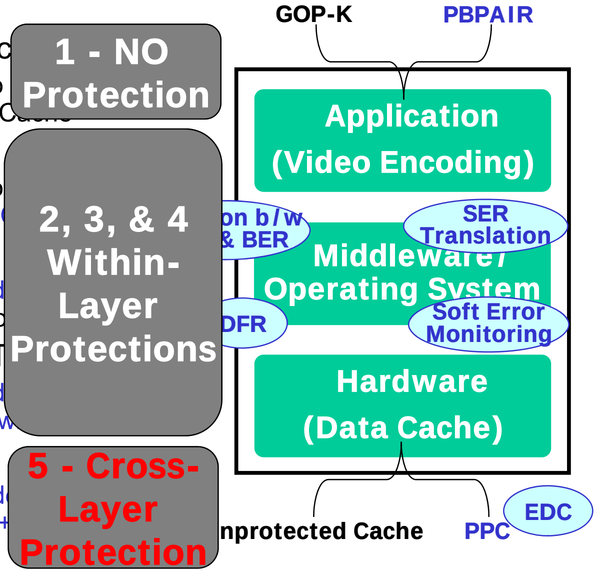
- ❑Motivation and Related Work
- ❑Problem Statement
- ❑Our Solution
- ❑**Experiments**
  - ➢**Experimental Setup and Compositions**
  - ➢**Effectiveness of CC-PROTECT in terms of failure rate, QoS, runtime, and energy consumption**
  - ➢**Effectiveness of Adaptive DFR/BER Schemes**
- ❑Conclusion

Experimental Framework



Compositions

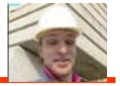
1. BASE – No Protec
   - Error-Prone Video
     K) + Unprotected C
2. HW-PROTECT
   - Error-Prone Video
     K) + PPC with EC
3. APP-PROTECT
   - Error-Resilient Vid
     (PBPAIR) + Unpro
4. MULTI-PROTECT
   - Error-Resilient Vid
     (PBPAIR) + PPC w
5. CC-PROTECT
   - Error-Resilient Vid
     (PBPAIR) + DFR +



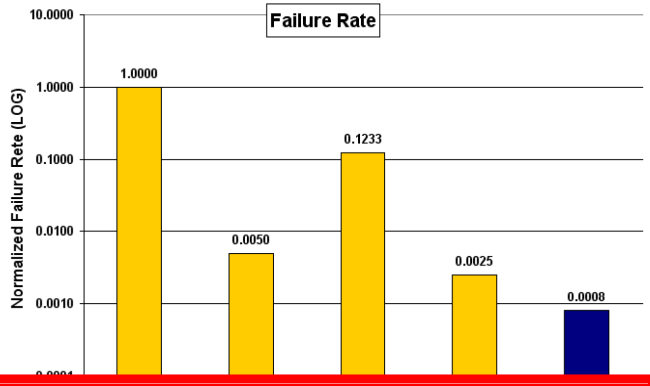
Effectiveness of CC-PROTECT

- First Set of Experiments – Evaluate CC-PROTECT with existing protections in terms of failure rate, video quality, energy consumption, and performance for FOREMAN.QCIF (mid activity)

Failure Rate

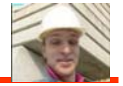


- Failure Rate is the number of failures (e.g., system crash) due to soft errors, out of thousands simulations

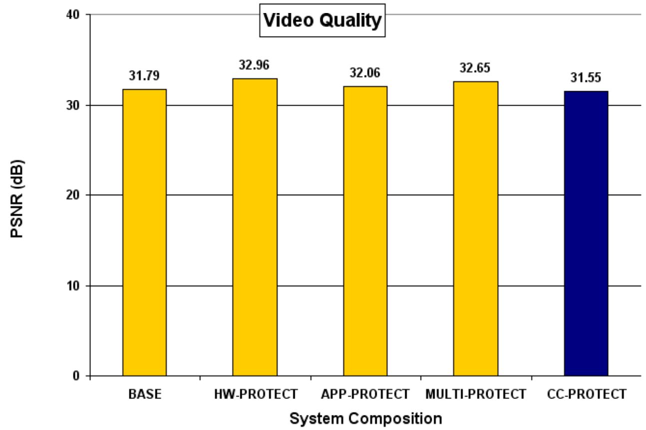


CC-PROTECT reduces the failure rate by more than 1,000 times, as compared to BASE

Video Quality

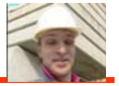


- QoS is the video quality measured in PSNR



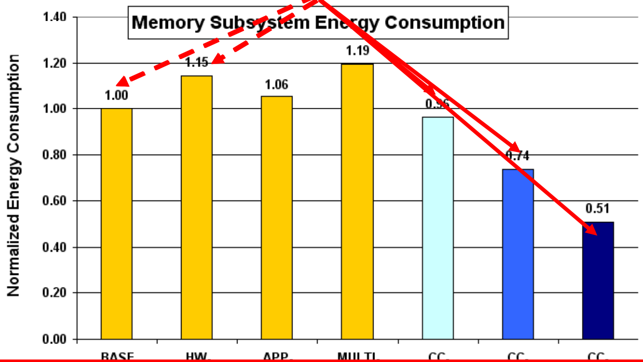
CC-PROTECT demonstrates the video quality close to those of other compositions

Energy Consumption



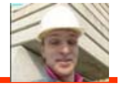
- ❑ Energy consumption includes the energy consumptions of

EDC + DFR + PBPAIR(CC-PROTECT) impact
56% Reduction compared to HW-PROTECT
49% Reduction compared to BASE

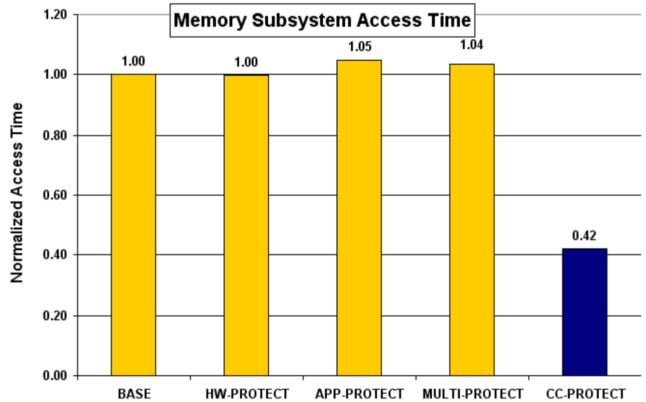


CC-PROTECT reduces the energy consumption of memory subsystem by 49%, compared to BASE

Performance

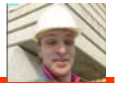


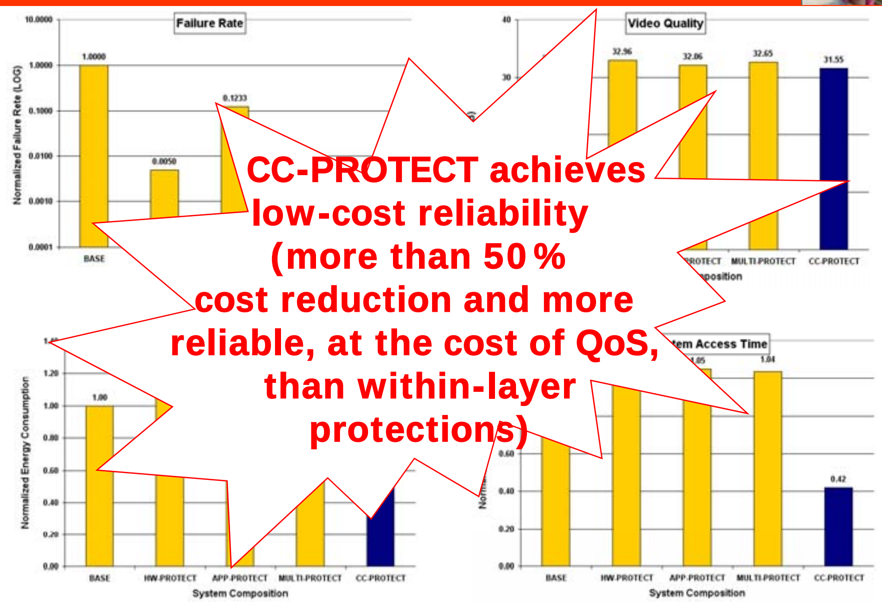
- ❑ Performance is estimated in access time to memory subsystem (caches, bus, and memory)



CC-PROTECT reduces the memory access time by 58%, compared to BASE

Effectiveness of CC-PROTECT



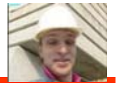


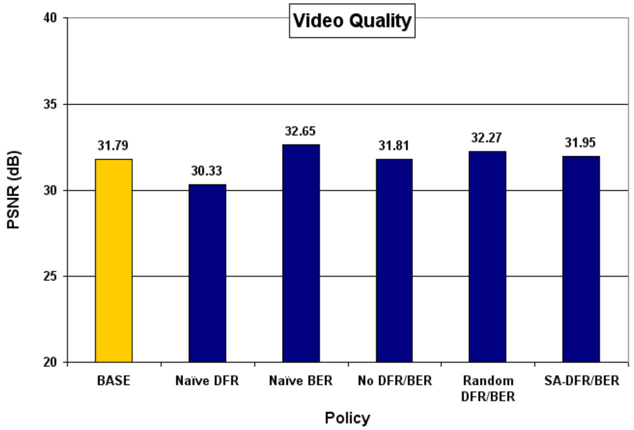
Effectiveness of Adaptive CC-PROTECT

- ❑ Second Set of Experiments – Evaluate adaptive CC-PROTECT schemes (SA-DFR/BER, FA-DFR/BER, and QA-DFR/BER) to naïve schemes (Naïve DFR and Naïve BER) in terms of video quality and energy consumption with FOREMAN.QCIF (mid activity)
  - ➢ For failure rate and performance, please refer to our paper

- ❑ **SA-DFR/BER** – 60% ACET (Average Case Execution Time) is the threshold value
  - ➢ 60% is the least threshold value, causing better QoS than BASE
- ❑ **FA-DFR/BER** – 2nd Frame must be protected
  - ➢ Losing 2nd frame affects the QoS most
- ❑ **QA-DFR/BER** – 31.79 dB is the threshold value to select DFR or BER
  - ➢ 31.79 dB is the PSNR value in case of BASE for FOREMAN

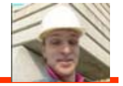
QoS

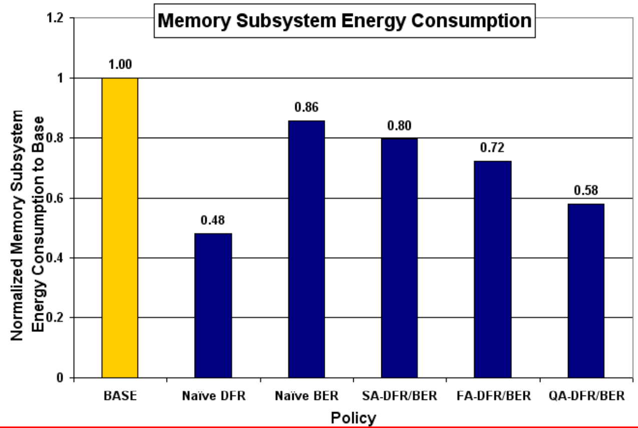




Adaptive CC-PROTECT improves the video quality, as compared to Naïve DFR

Energy Consumption





Adaptive CC-PROTECT balances energy consumption between Naïve DFR and Naïve BER, and QA-DFR/BER is the best in terms of energy

Conclusion

- Soft error is a critical design concern for mobile multimedia embedded systems
  - Previously proposed protection techniques within layers are expensive for resource-constrained mobile devices
- Propose **CC-PROTECT** approach, which **cooperates existing schemes across layers** to mitigate the impact of soft errors on the failure rate and video quality in mobile video encoding systems
  - PPC (Partially Protected Caches) with EDC (Error Detection Codes) at hardware layer
  - DFR (Drop and Forward Recovery) at middleware
  - PBPAIR (Probability-Based Power Aware Intra Refresh) at application layer
- Demonstrate the effectiveness of low-cost (about 50%) reliability (1,000x) at the minimal cost of QoS (less than 1%)
- Future work includes:
  - Expand CC-PROTECT for various errors
  - Robust software techniques for unreliable hardware in a cross-layer manner
  - Design space exploration techniques

Thanks!

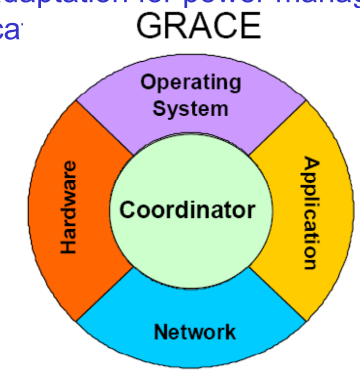
Any Questions?

klee@cs.yonsei.ac.kr

Backup Slides

Related Cross-Layer Work – GRACE

- ❑ GRACE project @ UIUC
  - ➢ Primarily OS adaptation for power management in multimedia mobile devices
  - ➢ Network adaptation for power management in multimedia communica



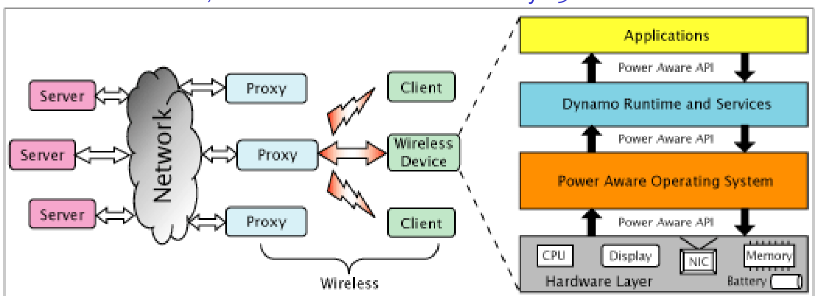
[GRACE, 05]

W. Yuan and K. Nahrstedt, "Practical voltage scaling for mobile multimedia devices", ACM international conference on Multimedia, 2004.
D. G. Sachs, et al., "GRACE: A cross-layer adaptation framework for saving energy", IEEE Computer, special issue on Power-Aware Computing, Dec 2003

Related Cross-Layer Work – Dynamo

- ❑ DYNAMO – Proxy-based middleware-driven cross-layer approach for QoS/Energy Tradeoffs
  - ➢ Content transcoding at application layer
  - ➢ Network traffic shaping at network layer
  - ➢ Backlight (LCD display) setting at hardware layer
  - ➢ NIC shutdown, CPU DVS/DFS at hardware layer

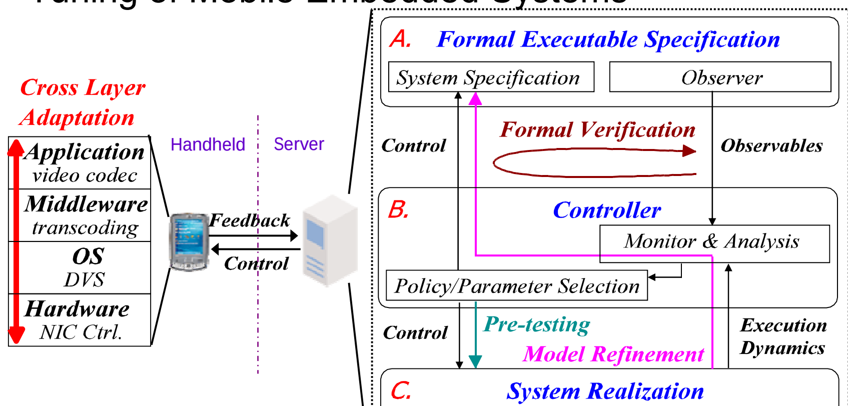
Middleware Coordination



Shivajit Mohapatra, "DYNAMO: Power aware middleware for distributed mobile computing", Ph.D. Thesis, University of California, Irvine, 2005
Radu Cornea, "Content annotation for power and quality trade-offs in mobile multimedia systems", Ph.D. Thesis, University of California, Irvine, 2007
Shivajit Mohapatra, et al., "DYNAMO: A cross-layer framework for end-to-end QoS and energy optimization in mobile handheld devices", IEEE JSAC, May 2007
Radu Cornea, et al., "Software annotations for power optimization on mobile devices", DATE, 2006
Shivajit Mohapatra, et al., "Integrated power management for video streaming to mobile handheld devices", ACM Multimedia, Nov2003

Related Cross-Layer Work – xTune

- ❑ xTune – A Formal Methodology for Cross-layer Tuning of Mobile Embedded Systems



- **Informed selection from formal model and analysis**
- **Enhanced by integrating it with observations of system**

➔Adaptive reasoning and proactive control

Soft Errors on an Increase

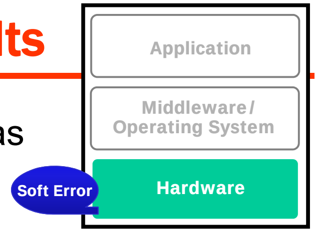
$$SER \propto N_{flux} \times CS \times exp\{-\frac{Q_{critical}}{Q_s}\}$$

$$\text{where } Q_{critical} = C \times V$$

- Increase exponentially due to technology scaling
  - 0.18 µm
    - 1,000 FIT per Mbit of SRAM
  - 0.13 µm
    - 10,000 to 100,000 FIT per Mbit of SRAM
- Voltage Scaling
  - Voltage scaling increases SER significantly

Soft Error is a main design concern!

Temporary Hardware Faults



- Temporary hardware faults such as transient faults (=soft errors) or intermittent faults cause failures
  - System crash, infinite loops, segmentation faults, etc.
- Causes of transient faults or soft errors
  - Environmental causes – Natural or man-made external radiation such as alpha particle, proton, and neutron
  - Technology factors – Technology scaling, increase of transistor densities, lower operating voltages, etc.
  - Marginal design parameters – Timing problems due to races, hazards, and skew
  - Signal integrity problems – Crosstalk, ground bounce, etc.

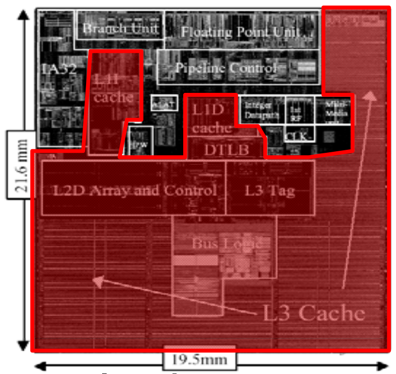
Soft Error is an Every Second Concern

- Soft Error Rate (SER)
  - FIT (Failures in Time) – How many errors in one billion operation hours
    - SER per Mbit @ 0.13 µm = 1,000 FIT ≈ 104 years in MTTF
- Soft error is becoming an every second problem
  - SER for 64 MB @ 0.13 µm = **64x8**x1,000 FIT ≈ 81 days in MTTF
  - SER for 128 MB @ 0.65 nm = **2x1,000**x64x8x1,000 FIT ≈ 1 hour in MTTF
  - SER for a system @ 0.65 nm = **2**x2x1,000x64x8x1,000 FIT ≈ 30 minutes in MTTF
  - SER with voltage scaling for a system @ 0.65 nm = **100**x2x2x1,000x64x8x1,000 FIT ≈ 20 seconds in MTTF
  - SER with voltage scaling for a system @ flight (35,000 feet) @ 0.65 nm = **800**x100x2x2x1,000x64x8x1,000 FIT ≈ 0.02 seconds in MTTF

Actel, "Neutrons from above – Soft Error Rates", Actel tech. rep., 2002
Robert Baumann, "Soft errors in advanced computer systems", IEEE Design and Test of Computers, 2005
Gorden E. Moore, "Cramming more components onto integrated circuits", Electronics, 1965
S. Mitra, et al., "Robust system design with built-in soft-error resilience", IEEE Computer 2005
P. Hazucha et al., "Impact of CMOS technology scaling on the atmospheric neutron soft error rate", IEEE Trans. on Nuclear Science, 2000
Ritesh Mastipuram and Edwin C. Wee, "Soft errors' impact on system reliability", http://www.edn.com/article/CA454636, 2004

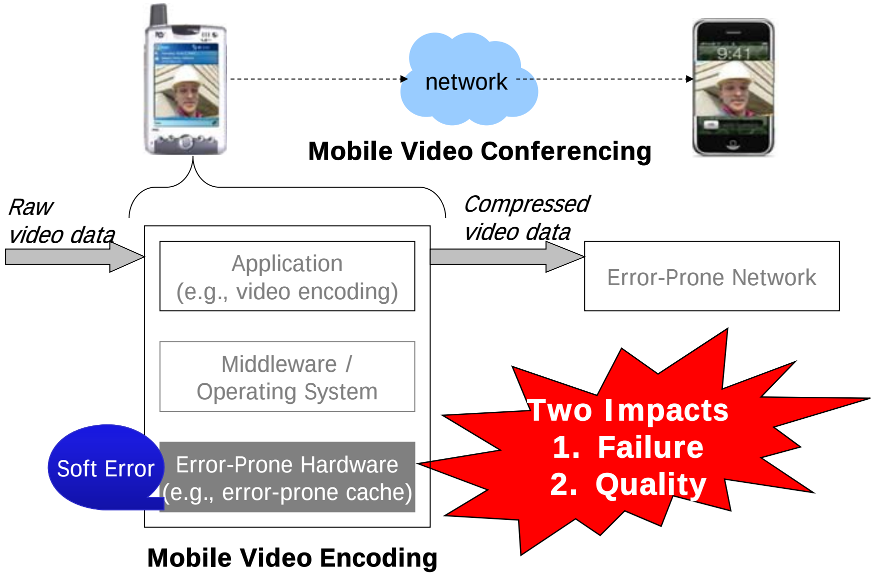
Most Vulnerable Caches

- Caches are most hit due to:
  - Larger portion in processors (more than 50%)
  - No masking effect (e.g., no logical masking)



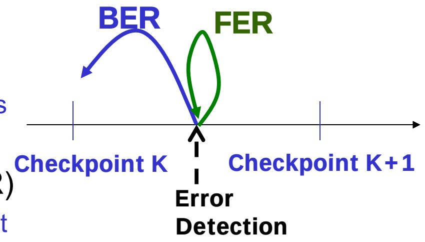
Intel Itanium II Processor

Problem Statement and Our Goals

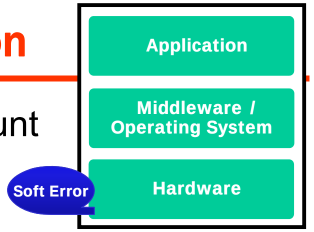


FER and BER

- ❑ Forward Error Recovery (FER)
  - ➢ Transform components into any correct state
  - ➢ ECC
  - ➢ Overkill for multimedia applications
- ❑ Backward Error Recovery (BER)
  - ➢ Roll back into the **previous** correct state
  - ➢ EDC + Checkpoint and Roll backward
  - ➢ Bad for the real-time requirement



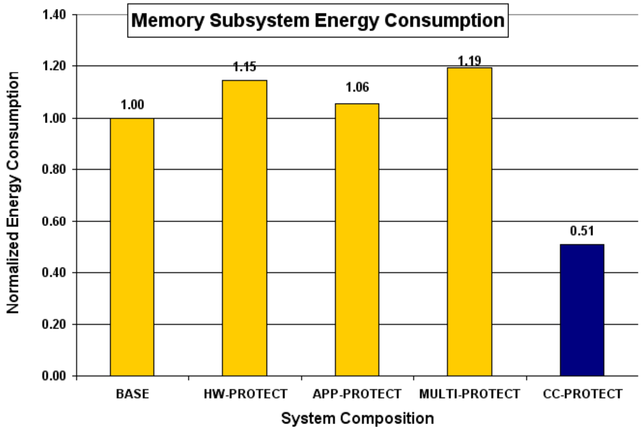
Error-Resilience at Application



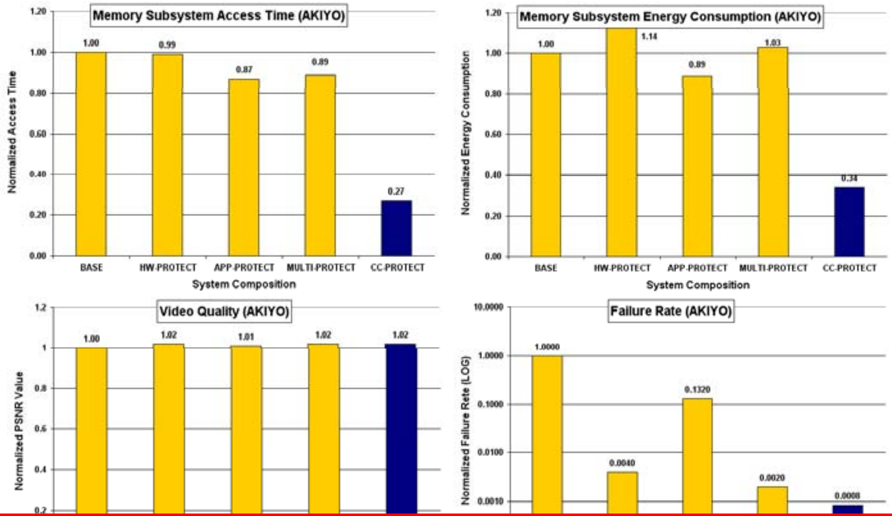
- ❑ PBPAIR [Kim, 06] takes into account packet loss rate to determine the error resilience level
  - ➢ <original PBPAIR>
    - ❑ Error Rate = Packet Loss Rate
- ❑ EE-PBPAIR [Lee, 08] has a mechanism to adjust packet loss rate
  - ➢ EE-PBPAIR at application encodes the video data resilient against not only **packet losses** but also **soft errors**
  - ➢ <EE-PBPAIR in CC-PROTECT>
    - ❑ Error Rate = PLR + FLR (Frame Loss Rate)
    - ❑ SER (Soft Error Rate) at Hardware is translated into FLR (Frame Loss Rate) at Middleware

Preliminary and Extra Experimental Results

Energy Consumption

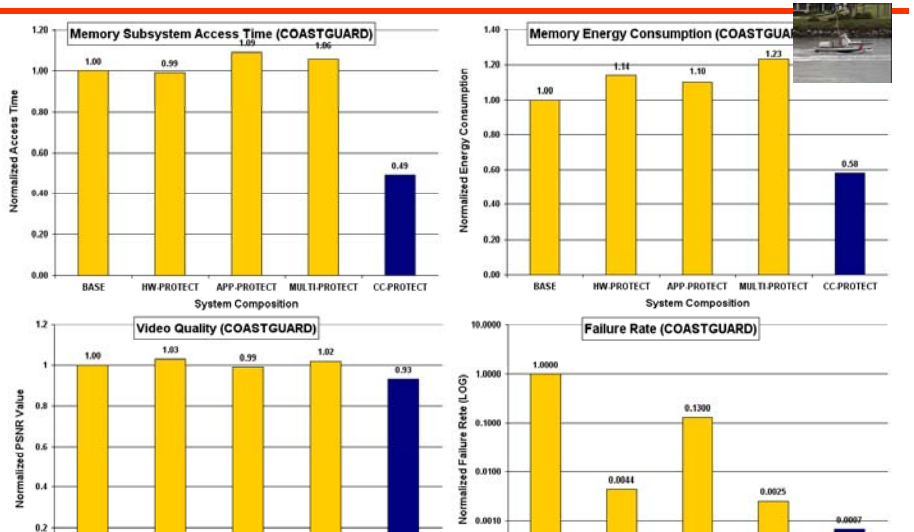


CC-PROTECT for AKIYO (low activity)



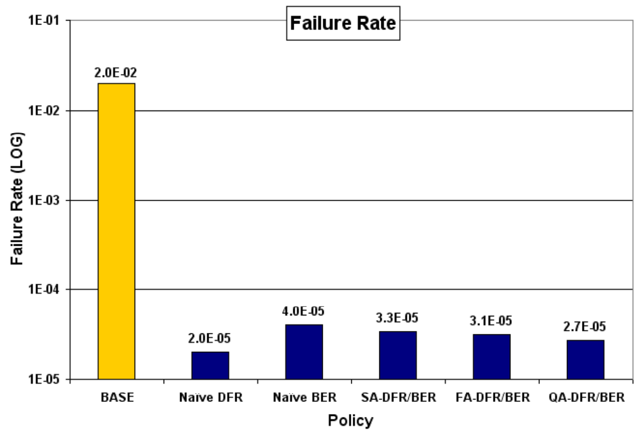
CC-PROTECT obtains better results with low activity of video streams

CC-PROTECT for COASTGUARD (high activity)



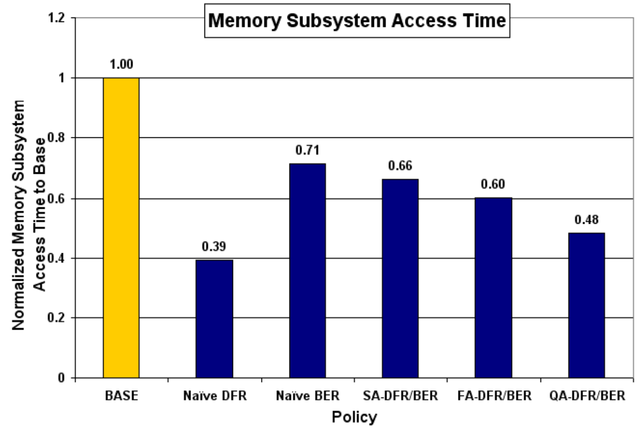
CC-PROTECT obtains effective results with various video streams

Failure Rate



Adaptive CC-PROTECT obtains the worse failure rate than Naïve DFR, still better than BASE

Performance

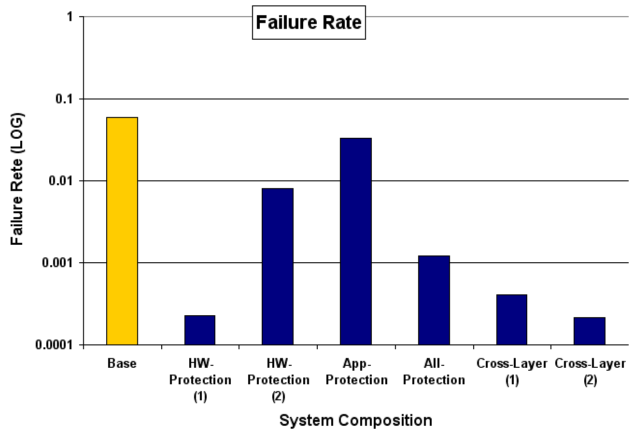


Adaptive CC-PROTECT balances between Naïve DFR and Naïve BER

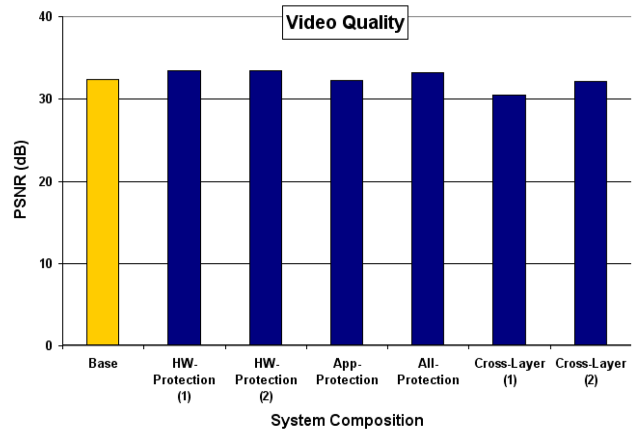
Compositions in the following slides

- Base
  - GOP + Unprotected Cache
- HW-Protection 1
  - GOP + Protected Cache with ECC
- HW-Protection 2
  - GOP + Protected Cache with EDC + BER (checkpoint and roll-backward)
- App-Protection
  - PBPAIR + Unprotected Cache
- All-Protection
  - PBPAIR + Protected Cache with ECC
- Cross-Layer Protection 1
  - GOP + PPC with EDC + DFR (drop and forward recovery)
- Cross-Layer Protection 2
  - PBPAIR + PPC with EDC + DFR (drop and forward recovery)

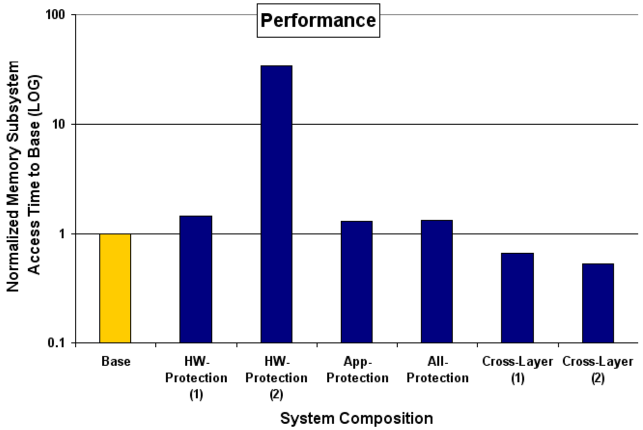
Failure Rate



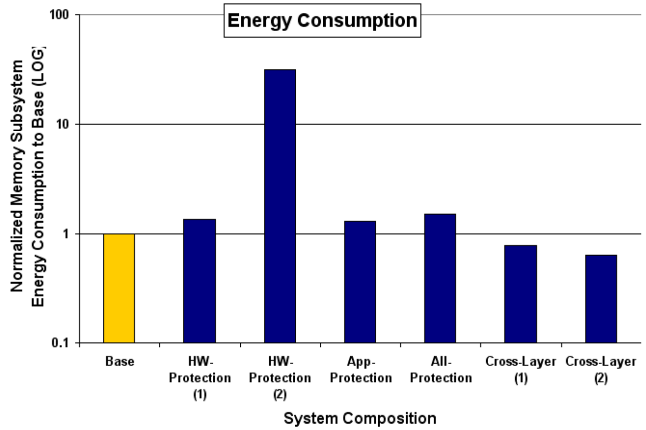
Video Quality



Performance

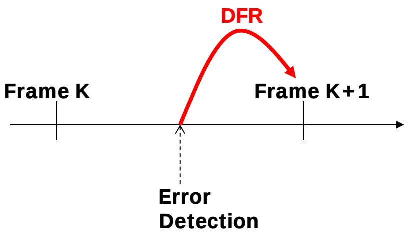


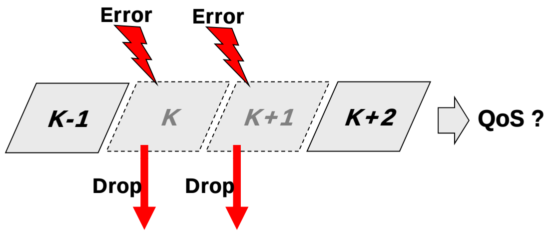
Energy Consumption



Naïve DFR

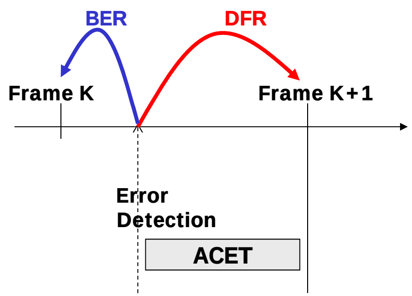
- **Naïve DFR**
- Strategy – Any soft error results in DFR
- Pros – High Energy Saving and High Reliability
- Cons – QoS degradation
  - e.g.) Consecutive frames dropped

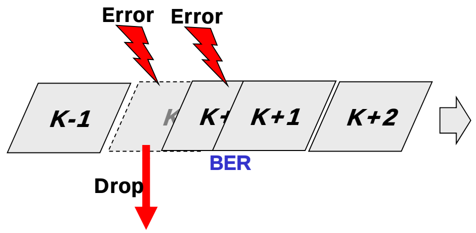




Slack-Aware Adaptive DFR/BER

- **SA-DFR/BER**
- Strategy – Enough slack time can help improve the QoS by retrying it
- Pros – QoS Improvement
- Cons – Increasing Energy Consumption

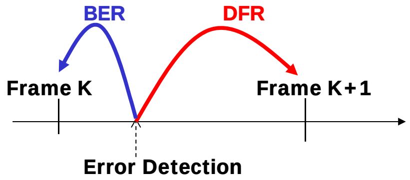


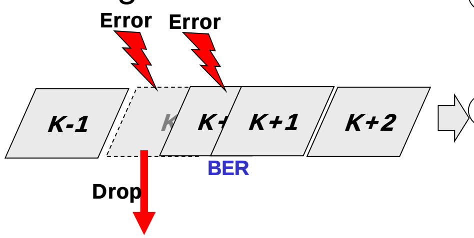


if $T_{elapsed} < T_{threshold}$
go back to Frame K
else
drop and move forward to Frame K+1
where $T_{threshold}$ is C% of ACET

Frame-Aware Adaptive DFR/BER

- **FA-DFR/BER**
- Strategy – Important frame with perspective of QoS should not be dropped
- Pros – QoS Improvement
- Cons – Increasing Energy Consumption and need to change the encoder





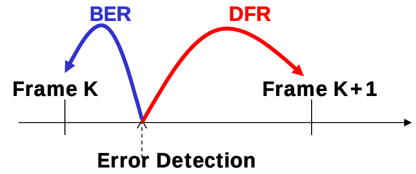
A
if F_K == $F_{I\text{-}frame}$
go back to Frame K
else
drop and move forward to F_{K+1}

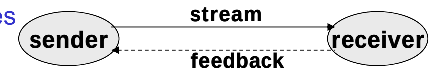
B
if F_{K-1} *(previous frame)* was dropped
go back to Frame K
else
drop and move forward to F_{K+1}

C
if $Diff_{K-1 \text{ and } K}$ > $Diff_{threshold}$
go back to Frame K
else
drop and move forward to F_{K+1}

QoS-Aware Adaptive DFR/BER

- **QA-DFR/BER**
- Strategy – QoS/Delay feedback from receiver helps adjust DFR policies.
  - (e.g.) QoS degradation makes BER work
  - (e.g.) QoS degradation can increase the time threshold, increasing the chance to retry it
  - (e.g.) if delay matters, apply DFR aggressively
- Pros – QoS is managed by user-end
- Cons – it may call BER always





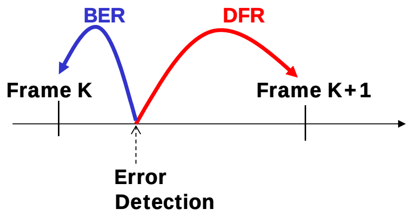
Low quality-feedback increases error-resilience aggressively or decreases DFR by adjusting threshold values

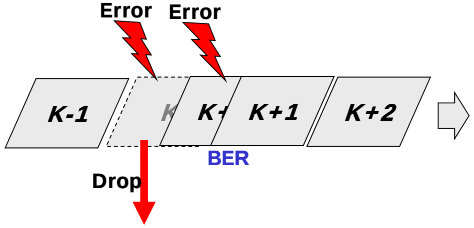
$T_{threshold}$ is increasing by quality-feedback → BER will be applied more often

$T_{threshold}$ is decreasing by delay-feedback → DFR will be applied more often

Randomly Adaptive DFR/BER

- **Random DFR/BER**
- Strategy – select DFR or BER based on pseudo random generation with Probability
- Pros – new knob to adjust DFR policy
- Cons – no intelligence





if $P_{pseudo\text{-}random}$ > $P_{threshold}$
go back to Frame K
else
drop and move forward to Frame K+1

where $P_{threshold}$ is weight of DFR and $P_{pseudo\text{-}random}$ is one number b/w 0 to 100 in pseudo-random

Results for DFR + BER