Engineering Faculty Document No. 16-06
November 9, 2006

TO: The Faculty of the College of Engineering
FROM: The Faculty of the School of Electrical and Computer Engineering
RE: ECE 572 Changes in Course Description, Content, and Prerequisites

The faculty of the School of Electrical and Computer Engineering has approved the following changes in ECE 572. This action is now submitted to the Engineering Faculty with a recommendation for approval.

From: **ECE 572 – Fault-Tolerant Computer Systems**

Sem. 2, Class 3, cr. 3
Prerequisite: ECE 302 and 565 or ECE 302, 365, and consent of instructor

An introduction to methodologies for specifying, modeling and designing fault-tolerant systems supported by case studies and real systems, a term project and relevant papers. Topics include fault classification, measurement and evaluation, techniques for fault detection and recovery, combinatorial and Markov modeling techniques.

To: **ECE 572 – Fault-Tolerant Computer Systems**

Sem. 2, Class 3, cr. 3
Prerequisite: ECE 302 and 368.

An introduction to the hardware and software methodologies for specifying, modeling, and designing fault-tolerant systems supported by case studies of real systems. The material presents a broad spectrum of hardware and software error detection and recovery techniques that can be used to build reliable networked systems. The lectures discuss how the hardware and software techniques interplay, what techniques can be provided in COTS hardware, what can be embedded into operating system and network communication layers, and what can be provided via a distributed software layer and in the application itself.

Reason: The course description and prerequisites have been changed to reflect the updated content of the course.

.

Mark Smith, Head
School of Electrical & Computer Engineering

ECE 572 Fault Tolerant Computer Systems

Course Outline

Saurabh Bagchi
Electrical and Computer Engineering Department, Purdue University
1285 EE Building, West Lafayette, IN 47907.
Email: sbagchi@purdue.edu

Text Book

D. P. Siewiorek and R. S. Swarz, Reliable Computer Systems - Design and Evaluation, 3rd edition, 1999, A.K. Peters, Limited.

Reference

D. K. Pradhan, ed., Fault Tolerant Computer System Design, 1st edition, 1996, Prentice-Hall. K. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2nd edition, 2001, John Wiley & Sons.

Prerequisites

ECE 302 and ECE 368. Equivalent courses may be used in satisfying the prerequisites with the consent of the instructor.

Description

An introduction to the hardware and software methodologies for specifying, modeling and designing fault-tolerant systems supported by case studies of real systems. The material presents a

broad spectrum of hardware and software error detection and recovery techniques that can be used to build reliable networked systems. The lectures discuss how the hardware and software techniques interplay, what techniques can be provided in COTS hardware, what can be embedded into operating system and network communication layers, and what can be provided via a distributed software layer and in the application itself.

Course Outcomes

A student who successfully fulfills the course requirements will have demonstrated:

i. an ability to evaluate the dependability of a system. [1,2,4;a,b,e]
ii. an ability to analyze a system for performance-dependability tradeoffs. [1,4;a,b,c,e,k]

iii. an ability to select the appropriate detection techniques (hardware and software) for a given environment. [1,4;a,c,e,k]

iv. an ability to select the appropriate recovery techniques (hardware and software) for a given environment. [1,4;a,c,e,k]

v. an ability to select the appropriate points in an end-to-end system to embed fault-tolerant techniques. [1,4;a,c,e,k]

Student assessment of the course outcomes will be in the form of a midterm exam, a final exam, and the grading of a design and implementation project. Each student working in a group of two will choose a project from a list. Each project will focus on one aspect of fault-tolerant system design and will test the ability to design, model or implement, execute experiments and perform evaluation.

Class Outline

TOPICS	**NUMBER OF LECTURES**
Introduction: Motivation, System view of high availability design, Two commercial examples (Stratus and Chameleon)	2
• Probability review, distributions	2
Hardware redundancy: Basic approaches, Static & Dynamic, Voting, Fault tolerant interconnection networks • Application: FTMP	3
Fault tolerant VLSI architectures & Design for testability	2
Error detection techniques: Watchdog processors, Heartbeats, Consistency and capability checking, Data audits, Assertions, Control-flow checking • Application: DHCP	5
Error control coding	2
Software fault tolerance: Process pairs, Robust data structures, N version programming, Recovery blocks, Replica consistency & reintegration, Multithreaded programs • Application: VAX	5
Network fault tolerance: Reliable communication protocols, Agreement protocols, Database commit protocols • Application: Distributed SQL server	6

Practical steps in design of high availability networked systems • Application: Web services, High-available clusters (a la Wolfpack)	2
Checkpointing & Recovery • Application: Microcheckpointing, IRIX checkpoints	5
Experimental Evaluation: Modeling and simulation based, Fault injection based • Application: NFTAPE fault injector	3
Modeling for performance, dependability and performability: dependability-specific methods (fault trees, reliability block diagrams), queues, stochastic Petri nets and stochastic activity networks • Application: UltraSAN	2
Practical Systems for Fault Tolerance: Putting it all together • Application: Ad-hoc wireless network • Application: NASA Remote Exploration & Experimentation System	3
Project Presentations	3
TOTAL	**45**