Chapter 1 – Introduction to FT Computing/Computer Reliability Engineering

Three dimensions of fault tolerant computer systems:

1. Physical – hardware (h/w), software (s/w), system
2. Time – life of a fault tolerant (FT) system (manufacture, operation, maintenance)
3. Cost – money ($), customer requirements/satisfaction

Definition of Fault Tolerant Computing – the correct execution of a specified algorithm in the presence of defects. This nominally requires a systems approach to FT computing that will encompass numerous disciplines to achieve a desired form of reliability.

Definition of a Fault Tolerant Computer – a computer system that posses the capability to execute a set of programs correctly in the presence of certain specified faults in the system including hardware failures and software errors.

Correct execution of programs
- Programs not halted or modified by faults in the computer
- Results do not contain errors caused by faults

Achievement of Fault Tolerance (methods)
- Hardware replication
- Information Redundancy – error correcting codes
- Software Replication
- Time Redundancy – rollback and recovery
- Operational Discipline – environment, maintenance, man/machine interface, risk analysis

Causes of Faults
- Design Errors
  - Imperfect or incomplete specifications
  - Imperfect implementation of specifications
- Component Failures
- Environmental Impacts

Characterization of Faults
- Duration
  - Permanent
  - Transient
  - Intermittent
- Extent
  - Local
  - Catastrophic (global)
- Models (some examples)
  - Stuck (open/short)
  - Unidirectional
  - Indeterminate
  - Operator Induced/Human Faults

Why Fault Tolerant Computer Systems? (knowing that most computer system implementations are digital verses analog or optical)

1. Typical Requirements for FT Systems
    a. Deep-Space Vehicles (long mission times), Mars Exploration Rovers: Spirit, Opportunity & Curiosity, Hubble Telescope, International Space Station (ISS)
    b. FAA Traffic Control (loss of life, economic impact of long term shutdown)
    c. Aircraft Reliance on Computers
    inherently unstable aircraft, loss of life minimization where acceptable failures rates of 10^{-6} per hour or better, Shuttle, 757/767/777 with Cat 0 landing capability, B-2 stealth bomber, DoD Drones
    d. Reliance on Communications
    Internet , Stock Markets, the Bell Telephone ESS (Electronic Switching System) with its two hours of downtime during its 40-year lifetime

2. System Complexity
    Implications of Moore's Law. Moore's Law deals with the complexity as represented by the number of transistors in a microprocessor/integrated circuit (*the number of transistors on an IC doubles approximately every two years although the period today is more often quoted as 18 months*). The downside of this increasing complexity is that with so many components, the probability of a hardware failure is quite finite.

    For example: Given a pc board with 40 transistors/active devices each with a 1% initial failure rate. The probability that the board is not defective = $(0.99)^{40} = 0.669$ (33% chance that it fails at turn-on)

3. Cost
    A more fault tolerant system (which will cost more than a lesser FT system) can actually reduce the cost of ownership (higher initial investment will save money over the lifetime of the system).

4. Social Economic Considerations
    Quality of life; impact of computers on society – the Information Revolution; flexibility for growth and change (different mission objectives using the same basic hardware); difficult task of managing very complex systems; society's reliance on computers (life/death situations).

SPEED and **MONEY** – probably the two most important aspects of computer systems. In dealing with fault tolerance, money is probably the primary concern.

The economic aspects of fault tolerant computing can be depicted with a simple example of the cost of ownership for two different computer systems, one more reliable than the other.

The cost of ownership or the cost of downtime can be related to maintenance and the time value of money (discount rate).

The cost of owning a computer system for n years can be expressed as

$$C = I + \sum_{i=1}^{n} (S_i P_i) / (1 + D)^i$$ (equation really nothing more than the time value of money)

n = system lifetime (assumed operational life, no salvage value at end)
I = Initial Cost of equipment (purchase price)
S_i = cost of one maintenance operation in year i (the cost of each service call)
P_i = the expected number of failures during year i
D = Discount Rate (time value of money for the customer)

Assume that a computer system has a 5-year life, its failure rate is constant over time, a service call costs $300 and the discount rate is 12%. Expressing failures as λ failures per million hours of operation and noting that there are 8760 hours in a year results in

n = 5 year lifetime
S_i = $300 cost of each service call
D = 12% discount Rate
λ = failures/10^6 hours (the failure rate, assumed to be constant)
I = Initial Cost of equipment (purchase price)

$$C = I + 300 \text{ (8760 hours/year) (} \lambda \text{ failures/} 10^6 \text{ hours)} \sum_{i=1}^{5} 1/(1 + 0.12)^i$$

$$\sum_{i=1}^{5 \text{ years}} 1 / (1 + 0.12)^i = 0.892 + 0.797 + 0.712 + 0.636 + 0.567 = 3.605$$

For these nominal assumptions, the Cost of Ownership for 5 years is

$C = I + 9.47335 \lambda$ where λ is in failures per million hours

System # 1 (cheaper, less reliable)

I = $20K λ = 6,000 failures per 10^6 hours

since λ is constant, then MTTF = 1/ λ = 166.67 hours or approximately 262 service calls in a 5 year period (MTTF = mean time to failure; relationship valid only for constant λ)

C = $76,840

System # 2 (expensive but more reliable)

I = $30K λ = 4,000 failures per 10^6 hours

MTTF = 1/ λ = 250 hours or approximately 175 service calls in a 5 year period

C = $67,893

System # 2 is 50% more expensive initially (I)
has a 33% improvement in reliability (MTTF)

The costlier system results in an 11.6% reduction in the cost of ownership over a 5 year period which is a direct result of avoiding the extra service calls for the more reliable System # 2.

Reliability, Availability and Risk

These terms can be viewed as probabilistic or deterministic (an outcome of the laws of nature). We'll concentrate on the probabilistic characterization of these terms.

Reliability is the ability to operate under designated conditions for a period of time. Ability will be designated as a probability or determined deterministically (from the empirical evidence such as failure mechanisms/analysis, testing/inspection, operational performance, etc.)

Availability takes down-time into consideration. It can be viewed as a combination of reliability and maintainability. Or conversely, reliability can be considered as instantaneous availability where no maintenance of repair is performed.

Risk is a more a systematic term – a big picture viewpoint which has a relationship to reliability analysis. Risk in qualitative terms is the potential of loss or injury from exposure to a hazard (danger). More safeguards against exposure to hazards → less risk.
Quantitative risk analysis involves the probability of loss combined with the probability hazard occurrence.

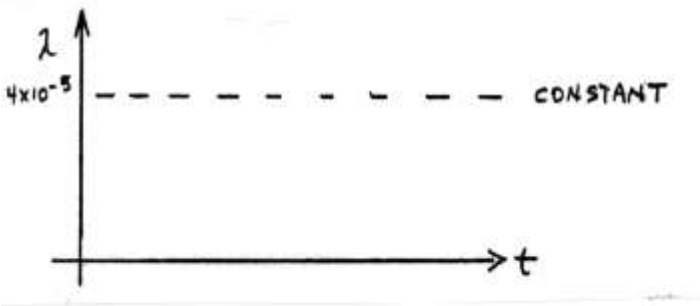
Risk analysis asks the following questions:

1. What can go wrong if exposed to a hazard?
2. How likely is this to happen?
3. If it does, what are the expected consequences?

Example (given without proof at this stage)

Life tests show that a component fails at a constant-failure rate where 100 items are tested for 1,000 hours and 4 of these fail in that period.

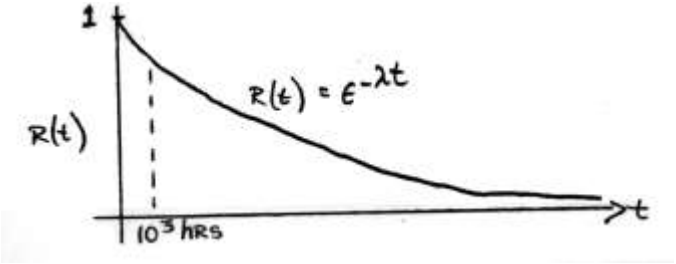
The failure rate λ is 4 failures / (100 items x 1,000 hours) = 4×10^{-5} failures/hour based on the important statement that the failure rate is constant.



The reliability function for this type of failure mode (constant failure rate λ) which represents the probability of no failures within a given operational period (1,000 hours in this case) is

$$R(t) = e^{-\lambda t} = e^{-\text{(4 failures / (100 items X 1,000 hrs)) (1,000 hrs)}}$$

$$= e^{-(4 \times 10^{-5} \text{ failures/hr})(1{,}000 \text{ hrs})} = e^{-0.04} = 0.9607$$ (probability of no failures in 1000 hours)



For this failure mode (constant failure rate of $\lambda = 4 \times 10^{-5}$ failures/hr) it is also known that the mean time to failure for the single component is

$$\text{Mean Time To Failure} = \text{MTTF} = 1 / \lambda = 25{,}000 \text{ hours}$$

Even though these parameters are very good (1%), when considering the complexity of using n of these items in a system knowing that all of the items must work in order for the system to work, the reliability of the system R_{sys} becomes

$$R_{sys}(t) = [R(t)]^n = [e^{-\lambda t}]^n = e^{-n\lambda t}$$

So the overall system reliability for 1,000 hours with just 50 of these items would be

$$R_{sys}(t=1000 \text{ hours}) = e^{-n\lambda t} = e^{-50 \times 0.00004 \times 1{,}000} = 0.13$$ or not much of a chance that the system would survive in the first 1,000 hours (a 87% chance of failing in the first 1,000 hours).

Reliability is a figure citing the probability of an object/system working until it fails; that is, the probability of no failures in a given interval. No repair is considered during the 1,000 interval nor are any alternatives to the failure considered once it has failed.

Availability [A(t)] is a measure of performance that does take into account the possibility of repair to a detected failure. It is the probability that a system is operational at a specific and given instant of time. Such activities as preventive maintenance and repair reduce the time that the system is available to the user but hopefully these functions can be performed without serious impact to the system.

A descriptive formula for availability

$$A(t) = \text{Uptime} / (\text{Uptime} + \text{Downtime})$$

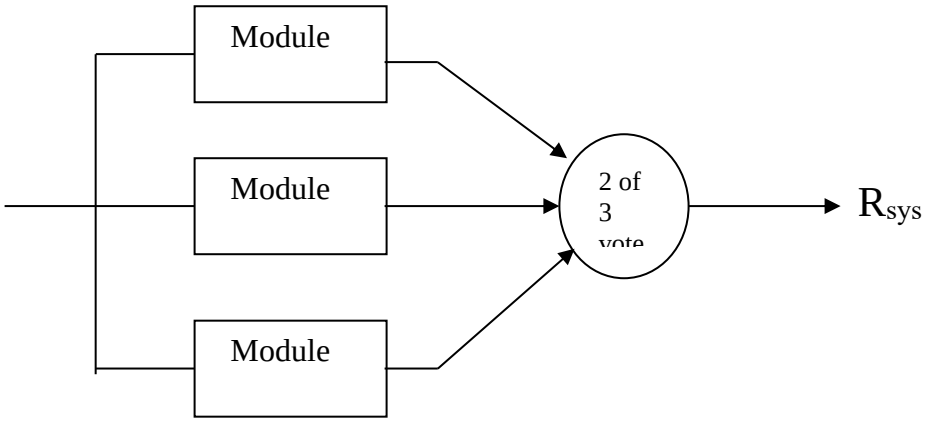
It will be shown in this class that such things as repair can add significantly to the desired operational characteristics of a complex system.

Although Reliability R(t) and Availability A(t) are radically different figures of merit for a reliable system, they are both are based on the same probabilistic measures.

Some Interesting History

The 1st FT digital computer was SAPO, which was built in Prague, Czechoslovakia in 1950 – 1955. It was a 32-bit floating point architecture that was motivated by the very poor component quality and political sensitivity to a project failure.

It was based on TMR (triple modular redundancy) which is a system that relies on comparison of results or voting



$R_{sys}\ (3, 0) = 3\ R_m^2 - 2\ R_m^3$ where R (3, 0) depicts a redundancy level of 3 (triple) with 0 spares where R_m is the reliability of a single (duplicated) unit

The term fault tolerance is shown in this TMR system since it 'masks' faults by a majority voting scheme (easy to conceptualize, extremely difficult to implement). It does not repair the faults, it tolerates the faults. Note that if the TMR system permanently votes out a module (removes it from the TMR system), then it must revert to a simplex (one) module operation. There is no way to have a majority voting scheme with just two modules.

The basis of the equation $R_{sys}\ (3, 0)$ can be shown as the reliability of all three modules working (R_m^3) plus the reliability of only 2 out of 3 modules working ($R_m^2 - R_m^3$) for which there are three possible combinations of 2-out-of-3 or $3(R_m^2 - R_m^3)$ thus

$$R_{sys}\ (3, 0) = R_m^3 + 3R_m^2 - 3R_m^3 = 3\ R_m^2 - 2\ R_m^3$$

This equation also assumes a perfect voter (no failures) so if we consider the reliability of the one voter which we'll consider to be in series with the three redundant modules R_m as shown above, then

$$R_{sys}\ (3, 0) = R_v\ (3\ R_m^2 - 2\ R_m^3)$$

Some Applications of FT Techniques

Apollo Vehicles (CM, LM, Saturn V computer - LVDC)
Bell Telephone ESS → Communication Networks
Voyager satellite
Kepler Telescope
Mars Rovers
SIFT (software implemented FT)
FTMP (FT multiprocessor)
C.mmp, Cm*, C.vmp – Carnegie Mellon University systems
Commercial Systems – Tandem/Compaq/HP, Stratus, Sun
New York Stock Exchange
India's stock exchange
Personal computers implemented with RAID
Boeing 777, Dreamliner

My involvement as an employee with the MIT Instrumentation Laboratory/Draper Laboratory in FT Systems started with an R&D project for NASA Headquarters executed at Cambridge, Massachusetts and the Johnson Space Center. The project was called AIPS. (This government project was also the genesis of this course at UHCL.)

Advanced Information Processing System (AIPS)

- Develop and demonstrate a FT system that will satisfy a broad spectrum of future NASA missions
- LaRC – advanced aircraft
- JSC – Space Station Freedom, Orbital Transfer Vehicles, Space Shuttle Upgrade/Block II
- Digital System for cost advantages and flexibility
- Design system for growth and change thru system modularity
- Evaluate the system in a flight environment
- Compare the AIPS primarily hardware implementation with software techniques used to achieve fault tolerance
- Incorporate other technology options into system as desired

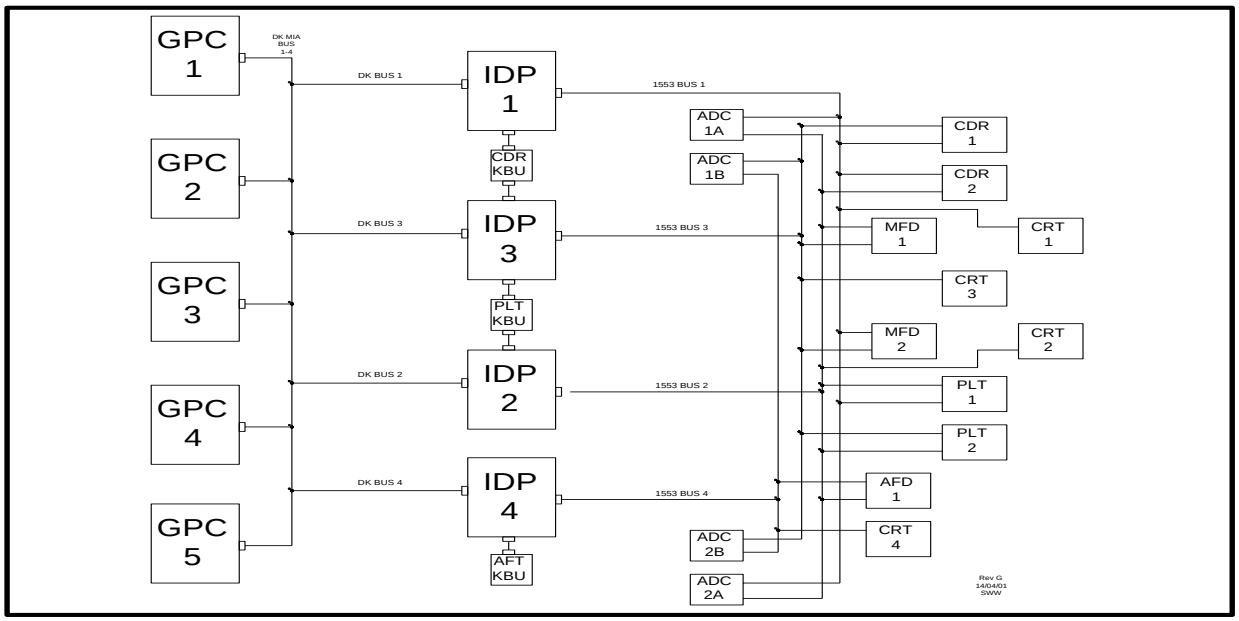
This eventually led to reliable computer systems for the Shuttle and X-38/ISS Crew Return Vehicle

The Shuttle's redundant (but not formally fault-tolerant) computer system can be explained by looking at a proposed upgrade to the computer system.

The Shuttle Cockpit Avionics Upgrade (CAU) – desired primarily for crew safety (loss of vehicle) and to reduce the crew's workload (more pertinent/graphical display of critical data).

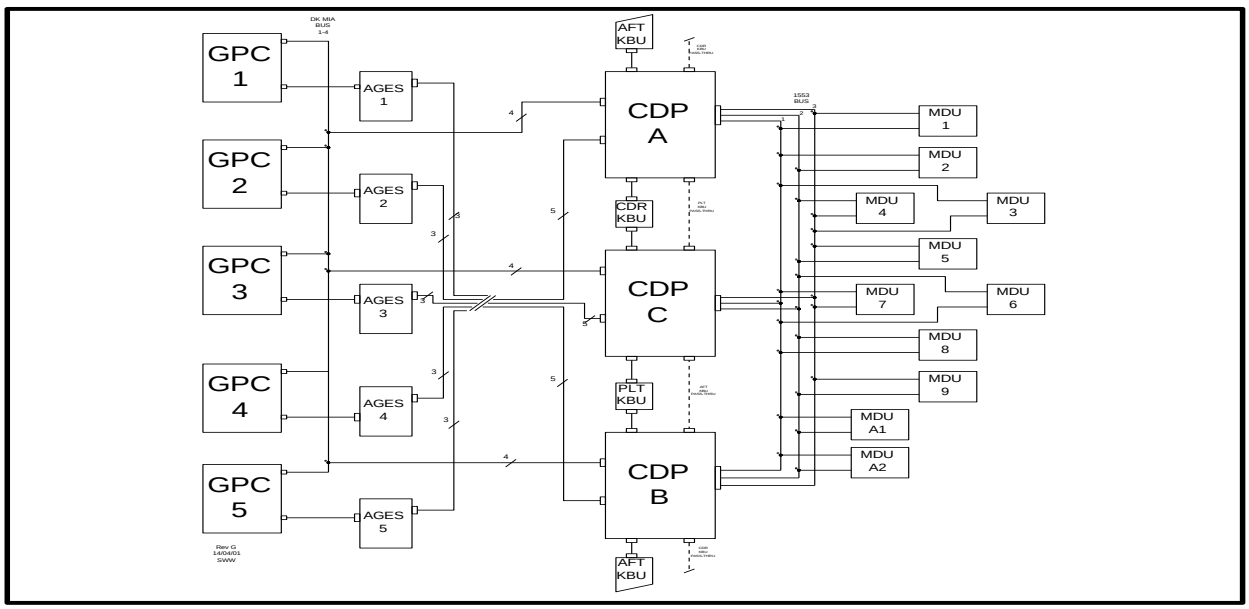
Project executed through the CDR (Critical Design Review) phase and then cancelled when it was decided to terminate the Shuttle Program with the last Shuttle flight in 2010 which completed the major construction phase of the International Space Station (ISS). The overall computer system concept of complementing the existing PFS (Primary Flight System) with CDPs (Command & Data Processors) was demonstrated by USA (United Space Alliance) in 2003 at JSC.

Shuttle Computer Configuration (4 redundant set computers, 1 backup computer)

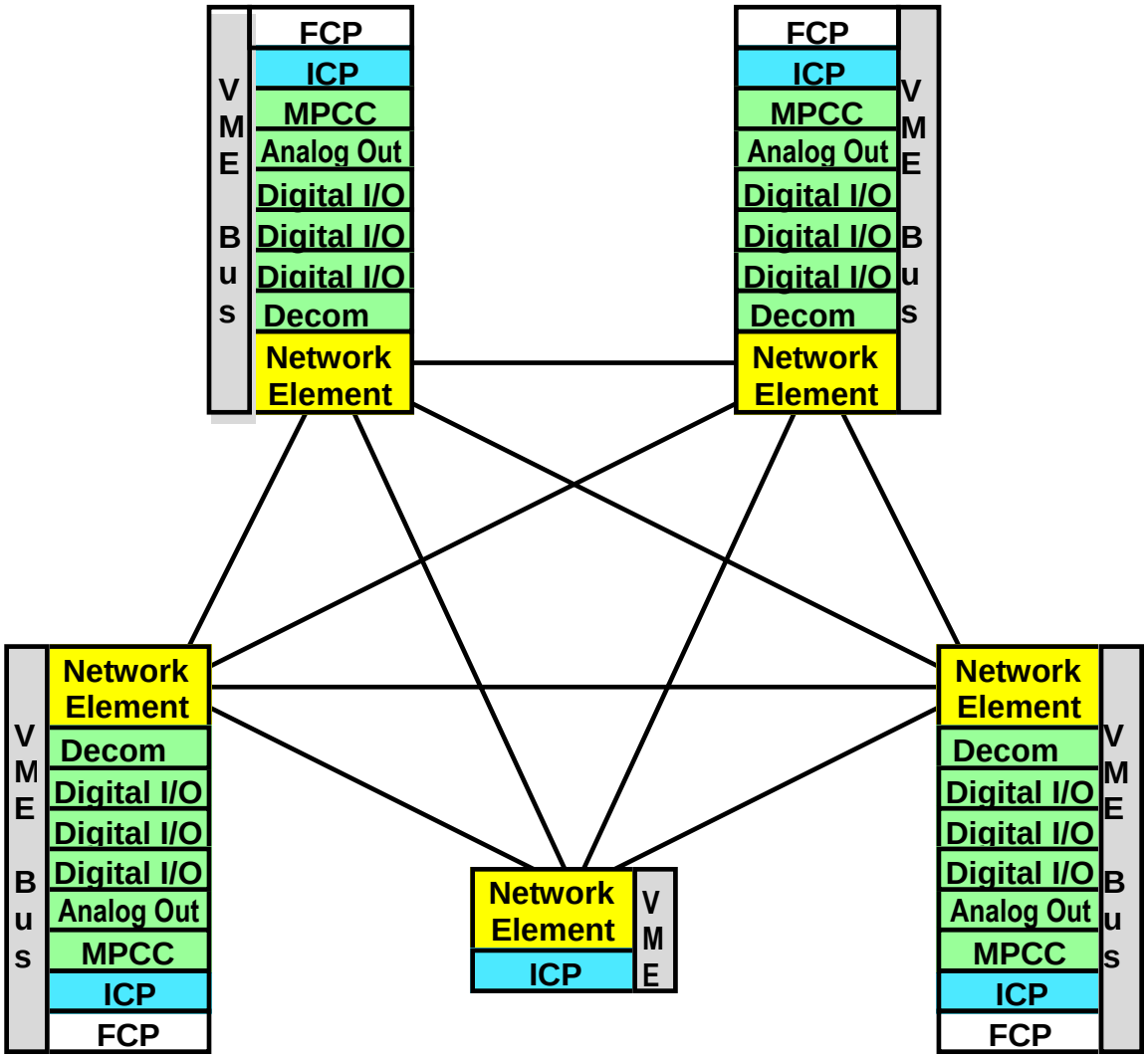


Cockpit Avionics Upgrade (CAU)
proposed upgrade to the Shuttle's computer system

Rev F Architecture



X-38 Vehicle Computer – built for NASA JSC utilizing a fault-tolerant parallel processor (FTPP) configuration with 5 Network Elements



- Each channel forms a fault containment region
- Input data distributed to each channel for data congruency (same data at same time)
- Redundant processing channels execute same instruction sequence on congruent data at the same time
- Results are voted and output for execution
- Errors are detected; failed items are removed and/or reset (brought back into set, a repair feature)
- Processing elements configures in groups to obtain balance of throughput and redundancy
- Multiple simplex groups provide high throughput of parallel processing
- Redundant groups (triplex or quadruplex) provide fault-tolerance (mixed levels of redundancy)
- Processing elements: Flight Critical Processors (FCP) and Instrumentation Control Processors (ICP)
- I/O devices can be hosted by a processing element
- Five fault-containment regions (FCRs)
  4 Flight Critical Processors (FCP) with a fifth unit made up of 1 Network Element (NE)
- One Network Element (NE) per each Fault Containment Region (FCR)
- Nine Processing Elements (FCP + ICP) configured in 6 processing groups
- System can accommodate 2 arbitrary non-simultaneous faults
- Software implements fault recovery/repair during non-critical periods

Delta pins down glitch cause

By Ashley Halsey III
WASHINGTON POST

Delta Air Lines said Tuesday that an internal problem, not the loss of power from a local utility, was to blame for the disruption that caused hundreds of flight cancellations and delayed tens of thousands of travelers Monday.

Delta initially pointed to a loss of electricity from Georgia Power, which serves its Atlanta hub, when its worldwide computer network crashed at 2:30 a.m. Monday. Georgia Power questioned that premise, saying that no other customers in the area of Delta's headquarters had lost power.

"It has nothing to do with Georgia Power," Delta spokeswoman Sarah Lora said after the airline further investigated the outage, which resulted in the cancellation of 300 additional flights Tuesday.

What happened, in fact, was that the Delta computers that control everything from reservations and boarding passes to crew and gate assignments toppled like a row of dominoes when one thing went wrong early Monday.

A power control mod-

Delta continues on B7

Delta restoring its schedule

Delta from page B1

ule malfunctioned, causing a surge that cut off power to the airline's main computer network. When that happens, the system is designed to switch in the blink of an eye to backup computer systems. On Monday, however, some of the backups did not kick in.

"When this happened, critical systems and network equipment didn't switch over to backups," Delta Chief Operating Officer Gil West said in a statement. "Other systems did. And now we're seeing instability in these systems."

West said getting both the computer systems and planes and air crew back into service was complicating Delta's operations for a second day Tuesday.

"We're seeing slowness in a system that airport customer service agents use to process check-ins, conduct boarding and dispatch aircraft," West said. "Delta agents today are using the original interface we designed for this system while we continue with our resetting efforts."

Delta spokeswoman Susan Hayes elaborated: "We are actually fully operational, it's just that we're not able to use that newer interface."

The meltdown at Delta, which has seven daily departures from Houston's Hobby Airport and 27 from Bush Intercontinental Airport, was at least the third occasion in little more than a year when computer malfunctions have caused flight cancellations. Southwest Airlines passengers were delayed last month by computer problems, and United Airlines experienced similar woes last summer.

Aviation analysts on Monday said such problems often are a result of the multiple mergers in the past 15 years, causing airlines to rely on a patchwork of computer networks. Hayes, however, said that Delta's merger with Northwest Airlines, finalized in 2010, did not result in a hybrid system.

"The passenger service system that we're currently using is original to Delta," she said.

Part of the problem that caused Delta cancellations and delays Tuesday was akin to what happens when airports are closed after a massive snow storm or hurricane. Planes that would have reached certain destinations had things gone according to plan Monday would have been in position to fly from those airports early Tuesday. But many of those planes were out of place for the flights they were intended to make Tuesday morning. Flight crews also were in the wrong places.

"Flight crews — pilots and flight attendants — carry out their responsibilities in a rotation, a schedule of flights and hotel reservations, that is usually three or four days," West said. "As cancellations occur, rotations become invalid. Multiplied across tens of thousands of pilots and flight attendants and thousands of scheduled flights, rebuilding rotations is a time-consuming process."

Senators request details on airlines' computer systems

By Curtis Tate
MCCLATCHEY NEWS SERVICE

WASHINGTON — A week after a computer failure caused a worldwide service meltdown at Delta Air Lines, two senators have asked all domestic carriers to explain how resilient their information technology systems are.

Sens. Richard Blumenthal of Connecticut and Ed Markey of Massachusetts, Democrats on the Senate Committee on Science, Commerce and Transportation, wrote to 13 airlines to express concern that there aren't enough backups in place to prevent service disruptions like the ones that paralyzed Delta last week and Southwest Airlines last month.

The senators also questioned the airlines on their vulnerability to cyberattacks.

Delta canceled more than 2,100 flights last week, more than it had in the first seven months of the year.

Experts questioned last week why the airlines failed to put adequate backup systems in place. Industry consolidation has turned airline computer systems into a complex jumble.

While the U.S. Department of Transportation and the Department of Homeland Security regulate aviation safety and security, they have very little oversight of airline service.

The senators noted that just four carriers — Delta, American, United and Southwest — now control 85 percent of domestic air travel, and a disruption experienced by just one can wreak havoc across the entire aviation network.

"In light of these recent technology issues," they wrote, "we encourage you to ensure that your IT systems have the appropriate safeguards and backups in place to withstand power outages, technological glitches, cyberattacks and other hazards that can adversely affect IT systems."

Blumenthal and Markey also demanded that the airlines offer better rebooking options or compensation to inconvenienced travelers.

Delta did offer passengers delayed more than three hours last week a $200 travel voucher.

TRADING HALT

Nasdaq pins blame on a surge of data

ASSOCIATED PRESS

The Nasdaq OMX Group on Thursday attributed last week's three-hour trading halt a surge of data that overwhelmed its server, in the stock market operator's most detailed accounting yet of the market outage.

In a statement, the company highlighted more than 20 attempts by Arca, one of the exchanges run by NYSE Euronext, to connect and then disconnect to the system that provides prices for recent trades in Nasdaq stocks. Those were accompanied by what Nasdaq described as a stream of quotes for inaccurate symbols from Arca, which Nasdaq's system was forced to reject.

The two incidents together inundated Nasdaq's system with more than twice the data that it was designed to handle.

A flaw in Nasdaq's own server then emerged that essentially led to the failure of the backup system to kick in, forcing to shut down the system. At 11:14 a.m. Central time, the exchange sent a notice to traders notifying them of the complete market halt.

"They obviously had issues, and it caused an event," Robert Greifeld, Nasdaq's chief executive, said in an interview Thursday, referring to the NYSE exchange. "We obviously had issues, we should be able to handle that. We were supposed to be able to fail over, and we did not."

He added that Nasdaq was not blaming Arca for the outage. But he said Nasdaq was accepting responsibility for its share of problems while also pointing to what he described as broader issues affecting the stock market industry. A NYSE Euronext spokesman declined to comment.