Duplex Systems

- Both processors execute the same task
  - If outputs are in agreement - result is assumed to be correct
  - If results are different - we can not identify the failed processor
  - A higher-level software has to decide how failure is to be handled
  - This can be done using one of several methods
Duplex Reliability

♦ Two active identical processors with reliability $R(t)$
♦ Lifetime of duplex - time until both processors fail
♦ $C$ - Coverage Factor - probability that a faulty processor will be correctly diagnosed, identified and disconnected
♦ $R_{\text{duplex}}(t)$ - the reliability of duplex system:

$$R_{\text{duplex}}(t) = R_{\text{comp}}(t) \left[ R^2(t) + 2C R(t)(1 - R(t)) \right]$$

$R_{\text{comp}}(t)$ - reliability of comparator

Duplex - Constant Failure Rates

♦ Each processor has a constant failure rate $\lambda$
♦ Ideal comparator - $R_{\text{comp}}(t)=1$

♦ Duplex reliability -

$$R_{\text{duplex}}(t) = e^{-2\lambda t} + 2Ce^{-\lambda t} \left(1 - e^{-\lambda t}\right)$$

$$\text{MTTF}_{\text{duplex}} = \frac{1}{2\lambda} + \frac{C}{\lambda}$$
Fault Detection: First Method - Acceptance Tests

♦ Acceptance Test - a range check of each processor's output

♦ Example - the pressure in a gas container must be in some known range

♦ We use semantic information of the task to predict which values of output indicate an error

♦ How should the acceptance range be picked?

Acceptance Test - Sensitivity Vs. Specificity

♦ Narrow acceptance range: high probability of identifying an incorrect output, but also a high probability that a correct output will be misidentified as erroneous (false positive)

♦ Wide acceptance range: low probability of both

♦ Sensitivity - the (conditional) probability that the test will recognize an erroneous output as such

♦ Specificity - the (conditional) probability that the output is erroneous if the test identified it as such

♦ Narrow range - high sensitivity but low specificity

♦ Wide range - low sensitivity but high specificity
Second Method – Hardware Testing

- Both processors are subjected to diagnostic tests
- The processor which fails the test is identified as faulty
- Real-life tests are never perfect
- **Test Coverage** - same as **test sensitivity** - the probability that the diagnostic test can identify a faulty processor as such
- **Test Transparency** - the complement of the **test coverage** - the probability that the test passes a faulty processor as good

Third Method – Forward Recovery

- Use a third processor to repeat the computation carried out by the duplex
- If only one of the three processors is faulty, the one that disagrees is the faulty one
- It is possible to use a **combination** of these methods
- **Acceptance test** - quickest to run but often the least sensitive
Pair & Spare System

♦ Avoid disruption of operation upon a mismatch between the two modules in a duplex
♦ Disconnect duplex and transfer task to spare pair
♦ Test offline, and if fault is transient - mark duplex as a good spare

Triplex-Duplex Architecture

♦ Form a triplex out of duplexes
♦ When processors in a duplex disagree, both are switched out
♦ Allows simple identification of faulty processors
♦ Triplex can function even if only one duplex is left - duplex allows fault detection
The Poisson Process - Assumptions

♦ Non-deterministic events of some kind occurring over time with the following probabilistic behavior
♦ For some constant \( \lambda \) and a very short interval of length \( \Delta t \):
   ♦ 1. Probability of one event occurring during \( \Delta t \) is \( \lambda \Delta t \) plus a negligible term
   ♦ 2. Probability of more than one event occurring during \( \Delta t \) is negligible
   ♦ 3. Probability of no events occurring during \( \Delta t \) is \( 1 - \lambda \Delta t \) plus a negligible term

Poisson Process - Derivation

♦ \( N(t) \) - number of events occurring during \([0, t]\)
♦ For a given \( t \), \( N(t) \) is a random variable
♦ \( P_k(t) = \text{Prob}\{N(t)=k\} \) - probability of \( k \) events occurring during a time period of length \( t \) \((k=0, 1, 2, \ldots)\)
♦ Based on the previous assumptions:
   \[
P_k(t + \Delta t) = P_k(t)(1 - \lambda \Delta t) + P_{k-1}(t)\lambda \Delta t
   \]
   (for \( k=1, 2, \ldots \))

| and | \( P_0(t + \Delta t) \approx P_0(t)(1 - \lambda \Delta t) \) |
Poisson Process – Differential Equations

♦ This results in the differential equations:

\[
\frac{dP_k(t)}{dt} = -\lambda P_k(t) + \lambda P_{k-1}(t) \quad \text{and} \quad \frac{dP_0(t)}{dt} = -\lambda P_0(t)
\]

♦ With the initial conditions

\( P_k(0) = 0 \) (for \( k \geq 1 \)) and \( P_0(0) = 1 \)

♦ The solution (for \( k=0,1,2,... \)) is

\( P_k(t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \)

♦ For a given \( t \), \( N(t) \) has a Poisson distribution with the parameter \( \lambda t \)

♦ For all values of \( t \), \( N(t) \) is a Poisson process with rate \( \lambda \)

Poisson Process – Properties

♦ For a Poisson process with rate \( \lambda \):

* Expected number of events in an interval of length \( t \) is \( \lambda t \)

* Length of time between consecutive events has an exponential distribution with parameter \( \lambda \) and mean \( 1/\lambda \)

* Numbers of events in disjoint intervals are statistically independent

♦ Sum of two Poisson processes with parameters \( \lambda_1 \) and \( \lambda_2 \) is a Poisson process with parameter \( \lambda_1 + \lambda_2 \)
Example of a Poisson Process - Duplex with Redundancy

- Two active processors + unlimited number of inactive spares
- Induction process instantaneous, spares always functional
- Each processor has a constant failure rate $\lambda$
- Lifetime of a processor - Exponential distribution with parameter $\lambda$
- Time between two consecutive failures of same logical processor - Exponentially distributed with a parameter $\lambda$
- $N(t)$ - number of failures in one logical processor during $[0,t]$
- $M(t)$ - number of failures in the duplex system during $[0,t]$

Duplex with redundancy - Reliability Calculation

- Duplex has two processors - failure rate is $2\lambda$
- Comparator failure rate - negligible
- Probability of $k$ failures in duplex in $[0,t]$ -
  \[ \text{Prob}(M(t)=k) = e^{-2\lambda t} \left( 2\lambda t \right)^k / k! \]  
  (for $k=0,1,2,...$)
- For the duplex not to fail, each of these failures must be detected and successfully replaced - probability $C$
- For $k$ failures - probability $C^k$

\[ R_{\text{duplex}}(t) = \sum_{k=0}^{\infty} \text{Prob}(k \text{ failures}) C^k = \sum_{k=0}^{\infty} e^{-2\lambda t} (2\lambda t)^k C^k / k! \]
\[ = e^{-2\lambda t} \sum_{k=0}^{\infty} (2\lambda tC)^k / k! = e^{-2\lambda t} \sum_{k=0}^{\infty} \frac{(2\lambda tC)^k}{k!} = e^{-2\lambda t} e^{2\lambda tC} = e^{-2\lambda (1-C)t} \]
Duplex with Redundancy Reliability - Alternative Derivation

- Individual processors fail at rate $\lambda$.
- Rate of failures in the duplex is $2\lambda$.
- Probability $C$ of each failure to be successfully dealt with, and $1-C$ to cause duplex failure.
- Failures that crash the duplex occur with rate $2\lambda(1-C)$.

The reliability of the system is $e^{-2\lambda(1-C)t}$.

More Complex Systems

- NMR systems in which failing processors are identified and replaced from an infinite pool of spares - similar calculation to duplex.
- Finite set of spares - the summation in the reliability derivation is capped at that number of spares, rather than going to infinity.
- Other variations of duplex systems -
  - One processor is active while the second is a standby spare.
  - Processors can be repaired when they become faulty.
- Combinatorial arguments may be insufficient for reliability calculation in more complex systems.
- If failure rates are constant, we can use Markov Models for reliability calculations.
Markov Chains - Introduction

- Markov Models provide a structured approach for the derivation of the reliability of complex systems.
- A Markov Chain is a stochastic process $X(t)$ - an infinite sequence of random variables indexed by time $t$, with a special probabilistic structure.
- For a stochastic process to be a Markov Chain, its future behavior must depend only on its present state, and not on any past state.
- $X(t+s)$ depends on $X(t)$, but given $X(t)$, $X(t+s)$ does not depend on any $X(t')$ for $t' < t$.
- If $X(t)=i$ - the chain is in state $i$ at time $t$.
- We deal only with Markov Chains with continuous time ($0 \leq t \leq \infty$) and discrete state ($X(t)=0,1,2,...$).

Markov Chain - Probabilistic Interpretation

- $\text{Prob}\{X(t+s)=j \mid X(t)=i, X(t')=k\} = \text{Prob}\{X(t+s)=j \mid X(t)=i\} \quad (t < t')$.
- Once the chain moves into state $i$, it stays there for a length of time which has an exponential distribution with parameter $\lambda_i$ - it has a constant rate $\lambda_i$ of leaving state $i$.
- The probability that when leaving state $i$ the chain will move to state $j$ (with $j \neq i$) is $P_{ij}$.
- Transition rate from state $i$ to state $j$ is $\lambda_{ij} = P_{ij} \lambda_i$.

$$\sum_{j \neq i} P_{ij} = 1 \quad \sum_{j \neq i} \lambda_{ij} = \lambda_i$$
State Probabilities

- \( P_i(t) \) - probability that the process is in state \( i \) at time \( t \), given it started at state \( i_0 \) at time 0

- For given time instant \( t \), state \( i \) and a very small interval of time \( \Delta t \), the chain can be in state \( i \) at time \( t+\Delta t \) in one of the following cases:
  - It was in state \( i \) at time \( t \) and has not moved during the interval \( \Delta t \) - probability \( \approx P_i(t)(1-\lambda_i \Delta t) \)
  - It was at some state \( j \) at time \( t \) (\( j \neq i \)) and moved from \( j \) to \( i \) during \( \Delta t \): probability \( \approx P_j(t)\lambda_{ji} \Delta t \)
  - Probability of more than one transition is negligible if \( \Delta t \) is small enough

- These assumptions result in
  \[
  P_i(t + \Delta t) \approx P_i(t)(1-\lambda_i \Delta t) + \sum_{j \neq i} P_j(t)\lambda_{ji} \Delta t
  \]

Differential Equations for State Probabilities \( P_i(t) \)

- \[
  \frac{dP_i(t)}{dt} = -\lambda_i P_i(t) + \sum_{j \neq i} \lambda_{ji} P_j(t)
  \]
  Since \( \sum_{j \neq i} \lambda_{ij} = \lambda_i \)

- \[
  \frac{dP_i(t)}{dt} = -\sum_{j \neq i} \lambda_{ij} P_i(t) + \sum_{j \neq i} \lambda_{ji} P_j(t)
  \]

- This (for \( i = 0,1,2,\ldots \)) can now be solved, using the initial conditions
  \( P_{i_0}(0) = 1 \) and \( P_i(0) = 0 \) for \( i \neq i_0 \)
Markov Chain for a Duplex with a Standby

- **Example:** One active processor and a one standby spare - connected when the active unit fails
- **Constant failure rate** $\lambda$ of an active processor
- **C- coverage factor** - probability that a failure of the active processor is correctly detected and the spare processor is successfully connected
- **The Markov chain** -

\[ \begin{aligned}
    &\text{2} \quad \text{Both good} \quad \lambda e \\
    &\text{1} \quad \text{One failed} \quad \lambda \\
    &\text{0} \quad \text{Failed System} \quad \lambda(1 - e)
\end{aligned} \]

Differential Equations for Duplex with Standby

\[ \begin{aligned}
    dP_1(t)/dt &= -P_1(t) \sum_{j \neq i} \lambda_{ij} + \sum_{j \neq i} \lambda_{ji} P_j(t) \\
    dP_2(t)/dt &= -\lambda P_2(t) \\
    dP_1(t)/dt &= \lambda CP_2(t) - \lambda P_1(t) \\
    dP_0(t)/dt &= \lambda(1 - C)P_2(t) + \lambda P_1(t)
\end{aligned} \]

- **Initial conditions:**
  - $P_2(0) = 1$, $P_1(0) = P_0(0) = 0$
Reliability of Duplex with Standby

♦ Solution of differential equations:

\[ P_2(t) = e^{-\lambda t} \]
\[ P_1(t) = C\lambda t \cdot e^{-\lambda t} \]
\[ P_0(t) = 1 - P_2(t) - P_1(t) \]

\[ R_{\text{system}}(t) = 1 - P_0(t) = P_2(t) + P_1(t) = e^{-\lambda t} + C\lambda t \cdot e^{-\lambda t} \]

♦ Exercise - derive this expression using combinatorial arguments

Markov Chain for a Duplex with Repair

♦ Two active processors: each with failure rate \( \lambda \) and repair rate \( \mu \) (repair time is exponential with parameter \( \mu \))

♦ The Markov model

\[
\frac{dP_i(t)}{dt} = -P_i(t)\sum_{j \neq i} \lambda_{ij} + \sum_{j \neq i} \lambda_{ji} P_j(t)
\]

♦ The differential equations -

\[
\frac{dP_2(t)}{dt} = -2\lambda P_2(t) + \mu P_1(t)
\]
\[
\frac{dP_1(t)}{dt} = 2\lambda P_2(t) + 2\mu P_0(t) - (\lambda + \mu) P_1(t)
\]
\[
\frac{dP_0(t)}{dt} = \lambda P_1(t) - 2\mu P_0(t)
\]

♦ Initial conditions -

\( P_2(0) = 1, P_1(0) = P_0(0) = 0 \)
Duplex with Repair - State Probabilities

- The solution to the differential equations -

\[ P_2(t) = \mu^2/(\lambda + \mu)^2 + 2\lambda\mu/(\lambda + \mu)^2 e^{-(\lambda+\mu)t} + \lambda^2/(\lambda + \mu)^2 e^{-2(\lambda+\mu)t} \]

\[ P_1(t) = 2\lambda\mu/(\lambda + \mu)^2 + 2\lambda(\lambda - \mu)/(\lambda + \mu)^2 e^{-(\lambda+\mu)t} - 2\lambda^2/(\lambda + \mu)^2 e^{-2(\lambda+\mu)t} \]

\[ P_0(t) = 1 - P_2(t) - P_1(t) \]

Availability vs. Reliability

- In systems without repair, mainly the reliability measure is of significance
- With repair - availability is more meaningful than reliability

- Point Availability - \( Ap(t) \)
  \[ = \text{Prob(The system is operational at time } t) = 1 - P_0(t) \]
- Reliability - \( R(t) = \text{Prob(The system is operational during [0,t])} \) - can be calculated by removing the transition from state 0 to state 1, solving the resulting new differential equations - \( R(t) = 1 - P_0(t) \)
Long-Run Availability

♦ We calculate $A$ - the long-run availability - the proportion of time in the long run that the system is operational

♦ We first calculate the steady-state probabilities - $P_2(\infty)$, $P_1(\infty)$, and $P_0(\infty)$ (or $P_2,P_1,P_0$)

♦ These steady-state probabilities can be calculated in one of the two methods:

* letting $t$ approach $\infty$ in $P_i(t)$
* setting $dP_i(t)/dt=0$ ($i=0,1,2$) and solving the linear equations for $P_i$, using the relationship $P_2+P_1+P_0=1$

$$A=1-P_0$$

Duplex with Repair - Long-Run Availability

♦ Steady state probabilities -

$$P_2 = \frac{\mu^2}{(\lambda + \mu)^2}$$

$$P_1 = \frac{2\lambda \mu}{(\lambda + \mu)^2}$$

$$P_0 = \frac{\lambda^2}{(\lambda + \mu)^2}$$

♦ Long-run availability -

$A = P_2 + P_1 = 1 - P_0$

$$= \frac{(\mu^2 + 2\lambda \mu)}{(\lambda + \mu)^2} = 1 - \frac{\lambda^2}{(\lambda + \mu)^2}$$