The following discussion is based on a paper by Davis and Wakerly
- Synchronization and Matching in Redundant Systems
- IEEE Trans. on Computers
- Vol. c-27, No 6, June 1978

This is an example of what can happen when one can make assumptions about the capabilities of components of the system.

Main objective:
- this is an old paper, but there are important messages, e.g.:
  - agreement can be “rolled out” in (or supported by) hardware
  - one can manipulate the fault assumptions

Hardware aided solution
- requires $N \geq 2t + 1$ processors + extra hardware
- Synchronizer module

![voter delay d diagram]
- processors with synchronizer modules

**Configuration**

\[ N \geq 2t + 1 = \text{# of lanes} \quad S \geq t + 1 = \text{# of stages} \]
Simplex: Data Transition Error

- Hardware Interstages = Broadcast Repeaters
- Processors vote on multiple copies received
**Davis / Wakerly**

- **Simplex**
  - Case 1: Processor A is faulty (commander is traitor)
    » Interstages may receive different values
    » But: each interstage receives only ONE value
    » Each interstage correctly forwards the values received
    » Each processor receives the SAME three values
    » Majority votes are identical
  - Case 2: An Interstage is faulty (commander is loyal)
    » All interstages receive the same value from Processor A
    » Two correct interstages forward correct value
    » Each processor receives 2 correct values
    » 2-of-3 majority

**Davis / Wakerly**

- **Difference from OM(1) Algorithm**
  - Processor Broadcast => Round 0 (initial broadcast)
  - Interstage Broadcast => Round 1 (rebroadcast)
  - Single-fault lies **either** in processor **or** in interstage, but **not in both**!
    » fault can not cause error in both rounds
    » therefore there is one error free round
    » same effect as discarding data in OM(1) algorithm
    » can thus achieve agreement without discarding data
  - Result: can achieve agreement with 3 processing lanes instead of 4 processors required by OM(1)
  - Disadvantage: requires extra hardware (stages)
**Multiplex Solution**

- Option 1: just replicate Simplex Solution
  - each interstage receives 3 messages and broadcasts 9 messages
  - each processor receives 9 values to vote upon

**Diagram:**

- Input to Proc. A, B, C
- Interstage
- Vote

- Proc. A, B, C receive 9 values

**Option 2: Install voters in interstages**

- each interstage receives 3 messages and broadcasts 3 messages
- each processor receives 3 values to vote upon

**Diagram:**

- Input to Proc. A, B, C
- Vote
- Interstage
- Proc. A, B, C receive 3 values
Multiplex
- Case 1: Processor A is faulty (commander is traitor)
  » Interstages may receive different values
  » Interstage may send different values
  » But: each interstage sends the same value to all processors
  » Each processor receives the SAME set of values
  » Majority votes are identical
- Case 2: An Interstage is faulty (commander is loyal)
  » All interstages receive identical sets of values
  » Two interstages forward correct value to all processors
  » Each processor receives 2 correct values
  » All processors get the same majority

Hardware Requirements
- Number of Lanes (rows) = 3
  » need to get 2-of-3 majority
- Number of Stages (columns) = 2
  » needed to assure one error free round
  » agreement is achieved at output of first non-faulty state.
  » once agreement is achieved, a minority of faulty nodes cannot disrupt it.
Two fault solution

- Proc. A
- Proc. B
- Proc. C
- Proc. D
- Proc. E


- to A
- to B
- to C
- to D
- to E

Davis / Wakerly

Summary

<table>
<thead>
<tr>
<th></th>
<th>Davis / Wakerly</th>
<th>OM(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HW complexity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>messages</strong></td>
<td>$2t^2 + 3t + 1$</td>
<td>$3t + 1$</td>
</tr>
<tr>
<td></td>
<td>$2t^2 + 3t + 1$</td>
<td>$O(N^{t+1})$</td>
</tr>
<tr>
<td><strong>OM(t)</strong></td>
<td>$N \geq 2t + 1$</td>
<td>$N \geq 3t + 1$</td>
</tr>
<tr>
<td></td>
<td>$S = t + 1$</td>
<td>$r \geq t + 1$</td>
</tr>
</tbody>
</table>