The Weakest Failure Detector for Solving Consensus

Journal of the ACM, 1996

TUSHAR DEEPAK CHANDRA
IBM T.J. Watson Research Center, Hawthorne, New York

VASSOS HADZILACOS
University of Toronto, Toronto, Ontario, Canada

SAM TOUEG
Cornell University, Ithaca, New York

Problem

- What is weakest failure detector to solve consensus/atomic broadcast in asynchronous system

- Assumptions:
- Crash failures
- Majority of processes are correct
- Message delays are finite

Problem

- Required paper:

Define properties of various failure detectors and propose algorithms to solve Consensus/Atomic Broadcast in asynchronous system assuming reliable broadcast, failures are crash failure and majority of processes are correct.

- Supplemental paper:

Weakest failure detector:

1. Eventually, some correct process always suspect all failed ones
2. Eventually, some correct process always trusted by all correct ones.

Why not

1. Eventually, some correct process always suspect ~~all~~ some failed ones
2. Eventually, some correct process always trusted by ~~all~~ some correct ones.

Failure Detector

- Required paper:
- 8 classes of failure detector -> reduce to 4 class
- Conclusion:
- We can use weak completeness, eventual weak accuracy to solve consensus
  Atomic broadcast in asynchronous environment with n > 2f (crash failure).

Completeness	Accuracy			
	Strong	Weak	Eventual Strong	Eventual Weak
Strong	*Perfect* $\mathcal{P}$	*Strong* $\mathcal{S}$	*Eventually Perfect* $\diamond\mathcal{P}$	*Eventually Strong* $\diamond\mathcal{S}$
Weak	$\mathcal{Q}$	*Weak* $\mathcal{W}$	$\diamond\mathcal{Q}$	*Eventually Weak* $\diamond\mathcal{W}$

FIG. 1. Eight classes of failure detectors defined in terms of accuracy and completeness.

Key techniques

- Proof:

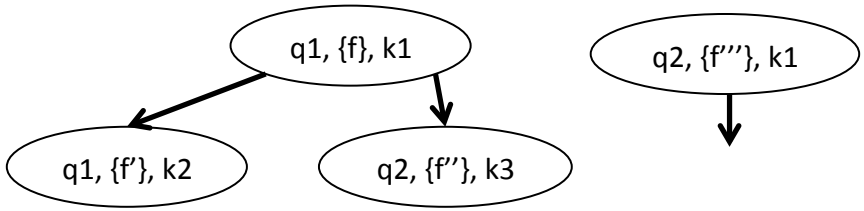
Relax the problem to Eventually all correct process trusts the same process. If any failure detector can solve Consensus is at least as strong as this one, then it is indeed the weakest failure detector.

- Idea:

1. run Consensus n times with each time i; [0, i] propose 1, (i,n] propose 0
2. Exchange failure information and form a DAG

3.1. root of run i is 1-valent, root of i-1 is 0-valent, pi must be correct

3.2. root of run i is {0,1}, the correct process will be found



Key techniques

Distributed algorithm to select a correct process

- Communication:

1. receive Y from others and merge it into its own Y
2. query its own failure detector and add it to Y
3. sends the new Y to all other processes

- Computation:

1. tag the vertex such that its descendants has a correct process decides 0/1.
2. return the correct process described in the previous slide

Key findings

- W is indeed the weakest failure detector.
- However, with completely async system, it is impossible to implement it. ☹

- With some cases of partial synchrony, we can implement "perfect" failure detector!
- One failure detector can solve consensus in multiple partial synchrony models

Take Away

- Weakest failure detector

1. Eventually, some correct processes always suspects all faulty ones

2. Eventually, all correct processes always trusts some correct ones

- Reducibility between failure detectors.
- More practical, Implement failure detector in Partial Synchrony Models.