Consensus and Agreement

Y. V. Lokeswari

Reference: Ajay Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems

Agreement

- Co-ordination require the **processes** to **exchange information** to negotiate with one another and eventually reach a common understanding or **agreement**, before taking application-specific actions.
- A classical example is that of the **commit decision** in database systems, processes collectively decide whether to **commit** or **abort** a transaction.
- **Assumptions**
  - **Failure models**
  - **Synchronous / Asynchronous Communication**
  - **Network Connectivity**
  - **Sender Identification**
  - **Channel Reliability**
  - **Authentication vs Non-Authenticated messages**
  - **Agreement Variable**

Agreement

- **Failure models:** Among the n processes in the system, at most f processes can be faulty.
- The various failure models – **fail-stop, send omission and receive omission and Byzantine failures**.
- It may send a message to only a **subset** of the **destination set** before crashing.
- In **Byzantine failure** model, a process may **behave arbitrarily.**
- **Synchronous/asynchronous communication:** If a Failure-prone process chooses to send a message to process Pi but fails, then Pi cannot detect the **non-arrival** of the message in an asynchronous system because this scenario is indistinguishable from the scenario in which the **message takes a very long time in transit**.

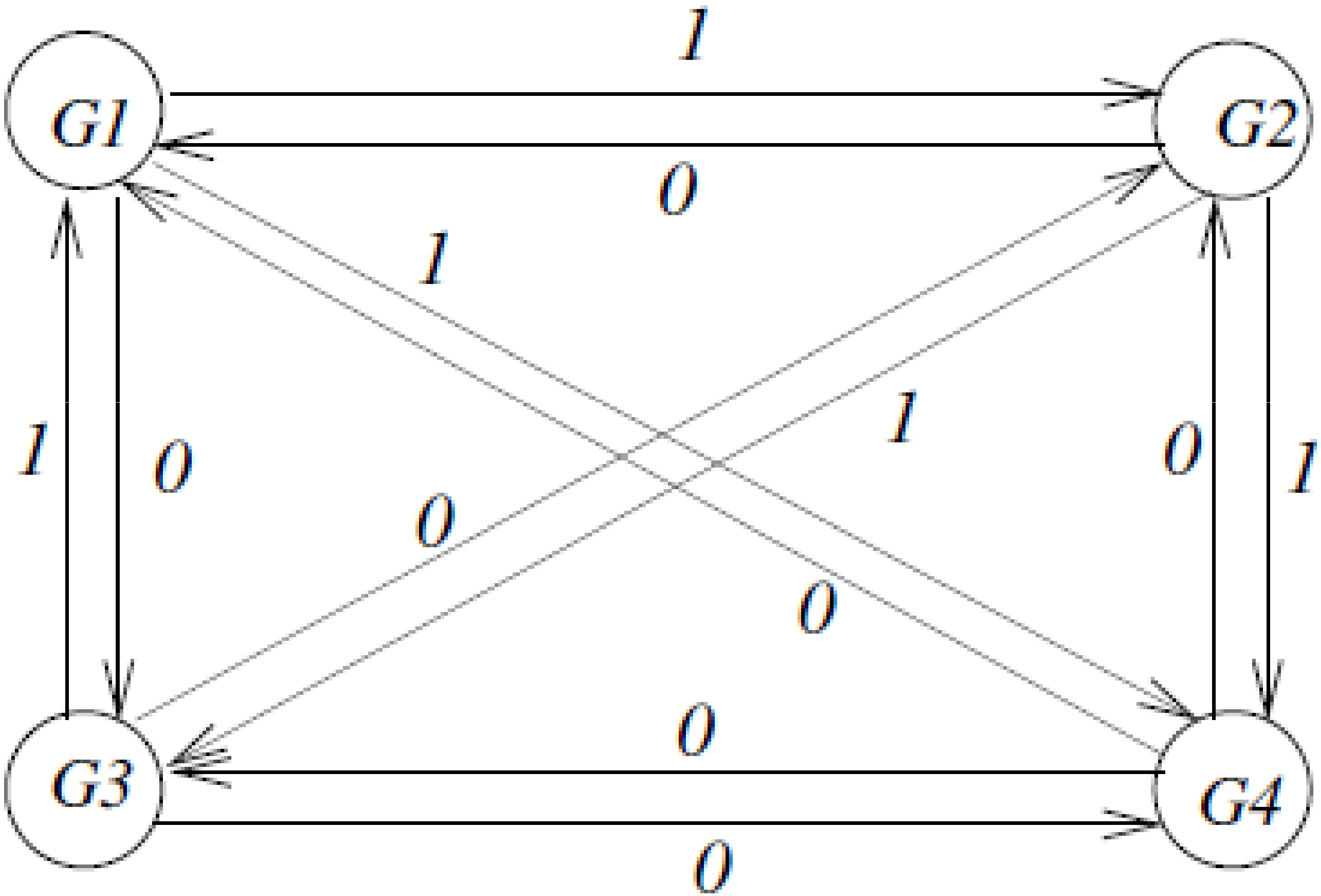
Agreement

- **<u>Network connectivity</u>:** The system has full **logical connectivity**, i.e., each process can communicate with any other by direct message passing.
- **<u>Sender identification</u>:** A process that receives a message always knows the **identity** of the **sender** process.
- **<u>Channel reliability</u>:** The channels are **reliable**, only the **processes** may **fail**.
  - **With unauthenticated messages**, when a faulty process relays a message to other processes, it can **forge** and claim that it was received from another processor or **tamper** the **contents** of the message.
  - Using **authentication** via techniques such as **digital signatures**, it is easier to solve the agreement problem.

Agreement

- **Network connectivity:** The system has full **logical connectivity**, i.e., each process can communicate with any other by direct message passing.
- **Sender identification:** A process that receives a message always knows the **identity** of the **sender** process.
- **Channel reliability:** The channels are **reliable,** only the **processes** may **fail**.
  - **With unauthenticated messages**, when a faulty process relays a message to other processes, it can **forge** and claim that it was received from another processor or **tamper** the **contents** of the message.
  - Using **authentication** via techniques such as **digital signatures**, it is easier to solve the agreement problem.
- **Agreement variable** The agreement variable may be **Boolean** or **multi-valued** and **need not** be an **integer**.

Byzantine General sending Confusing Messages



Problem Specification

Byzantine Agreement (single source has an initial value)

- **Agreement:** All non-faulty processes must agree on the same value.
- **Validity:** If the source process is non-faulty, then the agreed upon value by all the non-faulty processes must be the same as the initial value of the source.
- **Termination:** Each non-faulty process must eventually decide on a value.

Consensus Problem (all processes have an initial value)

- **Agreement:** All non-faulty processes must agree on the same (single) value.
- **Validity:** If all the non-faulty processes have the same initial value, then the agreed upon value by all the non-faulty processes must be that same value.
- **Termination:** Each non-faulty process must eventually decide on a value.

Interactive Consistency (all processes have an initial value)

- **Agreement:** All non-faulty processes must agree on the same array of values $A[v_1 \dots v_n]$.
- **Validity:** If process i is non-faulty and its initial value is v_i, then all non-faulty processes agree on v_i as the ith element of the array A. If process j is faulty, then the non-faulty processes can agree on any value for $A[j]$.
- **Termination:** Each non-faulty process must eventually decide on the array A.

Overall Results

Failure mode	Synchronous system (message-passing and shared memory)	Asynchronous system (message-passing and shared memory)
No failure	agreement attainable; common knowledge also attainable	agreement attainable; concurrent common knowledge attainable
Crash failure	agreement attainable $f < n$ processes $\Omega(f+1)$ rounds	agreement not attainable
Byzantine failure	agreement attainable $f \leq \lfloor (n-1)/3 \rfloor$ Byzantine processes $\Omega(f+1)$ rounds	agreement not attainable

Table: Overview of results on agreement. f denotes number of failure-prone processes. n is the total number of processes.

In a failure-free system, consensus can be attained in a straightforward manner

Agreement in a failure-free system (synchronous or asynchronous)

- In a **failure-free** system, **consensus** can be **reached** by **collecting information** from the **different processes**, arriving at a "**decision**," and **distributing** this **decision** in the **system.**
- A distributed mechanism would have each process **broadcast** its **values** to others, and each **process computes** the **same function** on the **values received**.
- Examples being the **majority, max, and min functions**.
- Distribute the decision may be based on the **token circulation** on a **logical ring**, or the **three-phase tree-based broadcast, convergecast–broadcast**, or **direct communication** with all nodes.

Agreement in a failure-free system (synchronous or asynchronous)

- In a **synchronous system**, this can be done simply in a **constant number of rounds**.
- **common knowledge** of the decision value can be obtained using an **additional round.**
- In an **asynchronous system**, consensus can similarly be reached in a **constant number of message hops** .
- **concurrent common knowledge** of the **consensus** value can also be attained, **using** the **algorithms**.
- **Reaching agreement** is **straightforward** in a **failure-free system**.
- **Focus** on **failure-prone** systems.

Summary

- Agreement.
- Problem Definition:
  - Byzantine Problem.
  - Consensus.
  - Interactive Consistency.
- Agreement in a failure-free system.

Thank You