

# Distributed Systems

ALI KAMANDI, PH.D.

SCHOOL OF ENGINEERING SCIENCE

COLLEGE OF ENGINEERING

UNIVERSITY OF TEHRAN

KAMANDI@UT.AC.IR

## Process failure models (Benign)

### Fail-stop

در این مدل، از یک لحظه به بعد پروسه ای که به درستی کار می کرده از کار می افتد. سایر پروسه ها می فهمند که این پروسه دچار اشکال شده است. این مدل یک مدل انتزاعی است و نحوه فهم دیگران از خطا می تواند کاملاً متنوع باشد.

### Crash

در این مدل یک پروسه که درست کار می کرده در یک لحظه از کار می افتد. نودهای دیگر از آن مطلع نمی شوند.

### Receive omission:

در این مدل یک پروسه درست، به نحوی دچار خطا می شود که فقط برخی از پیامهایی را که برای آن ارسال شده است را دریافت می کند.

### Send omission:

پروسه فقط بعضی از پیام هایی را که باید ارسال کند، واقعاً ارسال می کند.

### General omission:

پروسه دچار یکی یا هر دو خطای فوق می شود.

## Process failure models (Byzantine)

---

### Byzantine or malicious failure, with authentication

پروسه ممکن است هر رفتار دلخواهی را از خود نشان دهد. اما چنانچه ادعا کند که پیامی را از نود خاصی دریافت کرده است، این ادعا با مکانیزم authentication یا امضا قابل ارزیابی خواهد بود.

### Byzantine or malicious failure

پروسه ممکن است هر رفتار دلخواهی نشان دهد و بر خلاف مدل ادعای دریافت پیام از یک نود سالم، قبل قابل ارزیابی نخواهد بود.

## Stopping Failure

---

**Agreement:** No two processes decide on different values.

**Validity:** If all processes start with the same initial value  $v \in V$ , then  $v$  is the only possible decision value.

**Termination:** All nonfaulty processes eventually decide.

**FloodSet algorithm (informal):**

Each process maintains a variable  $W$  containing a subset of  $V$ . Initially, process  $i$ 's variable  $W$  contains only  $i$ 's initial value. For each of  $f + 1$  rounds, each process broadcasts  $W$ , then adds all the elements of the received sets to  $W$ .

After  $f + 1$  rounds, process  $i$  applies the following decision rule. If  $W$  is a singleton set, then  $i$  decides on the unique element of  $W$ ; otherwise,  $i$  decides on the default value  $v_0$ .

**states<sub>i</sub>:**

$rounds \in \mathbb{N}$ , initially 0

$decision \in V \cup \{unknown\}$ , initially *unknown*

$W \subseteq V$ , initially the singleton set consisting of  $i$ 's initial value

**msgs<sub>i</sub>:**

if  $rounds \leq f$  then send  $W$  to all other processes

**trans<sub>i</sub>:**

$rounds := rounds + 1$

let  $X_j$  be the message from  $j$ , for each  $j$  from which a message arrives

$W := W \cup \bigcup_j X_j$

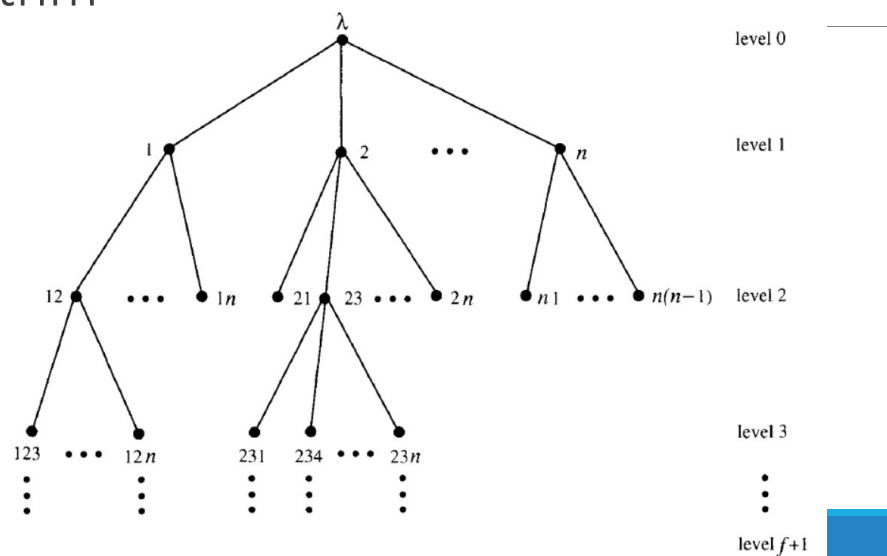
if  $rounds = f + 1$  then

    if  $|W| = 1$  then  $decision := v$ , where  $W = \{v\}$

    else  $decision := v_0$

**Complexity analysis.** *FloodSet* requires exactly  $f + 1$  rounds until all non-faulty processes decide. The total number of messages is  $O((f + 1)n^2)$ . Each message contains a set of at most  $n$  elements (since each element must be the initial value of some process), so the number of bits per message is  $O(nb)$ . Thus, the total number of communication bits is  $O((f + 1)n^3b)$ .

## Exponential Information Gathering Algorithm



*Round 1:* Process  $i$  broadcasts  $val(\lambda)$  to all processes, including  $i$  itself. Then process  $i$  records the incoming information:

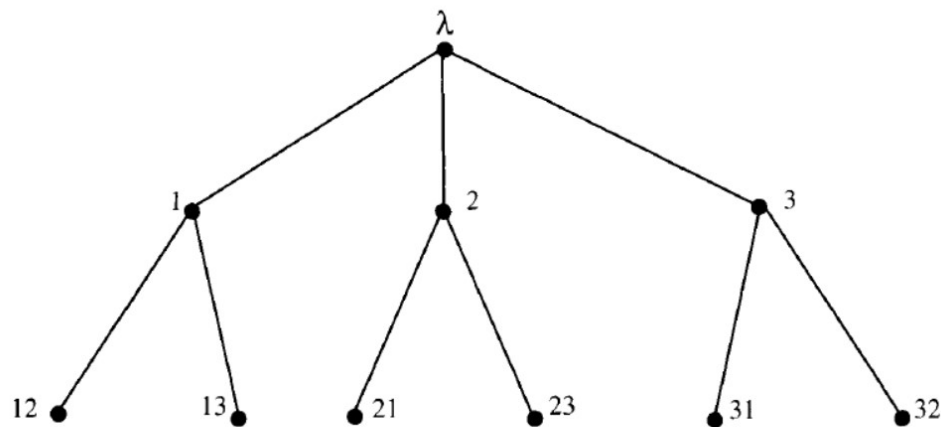
1. If a message with value  $v \in V$  arrives at  $i$  from  $j$ , then  $i$  sets its  $val(j)$  to  $v$ .
2. If no message with a value in  $V$  arrives at  $i$  from  $j$ , then  $i$  sets  $val(j)$  to  $null$ .

*Round  $k$ ,  $2 \leq k \leq f + 1$ :* Process  $i$  broadcasts all pairs  $(x, v)$ , where  $x$  is a level  $k - 1$  label in  $T$  that does not contain index  $i$ ,  $v \in V$ , and  $v = val(x)$ .<sup>1</sup> Then process  $i$  records the incoming information:

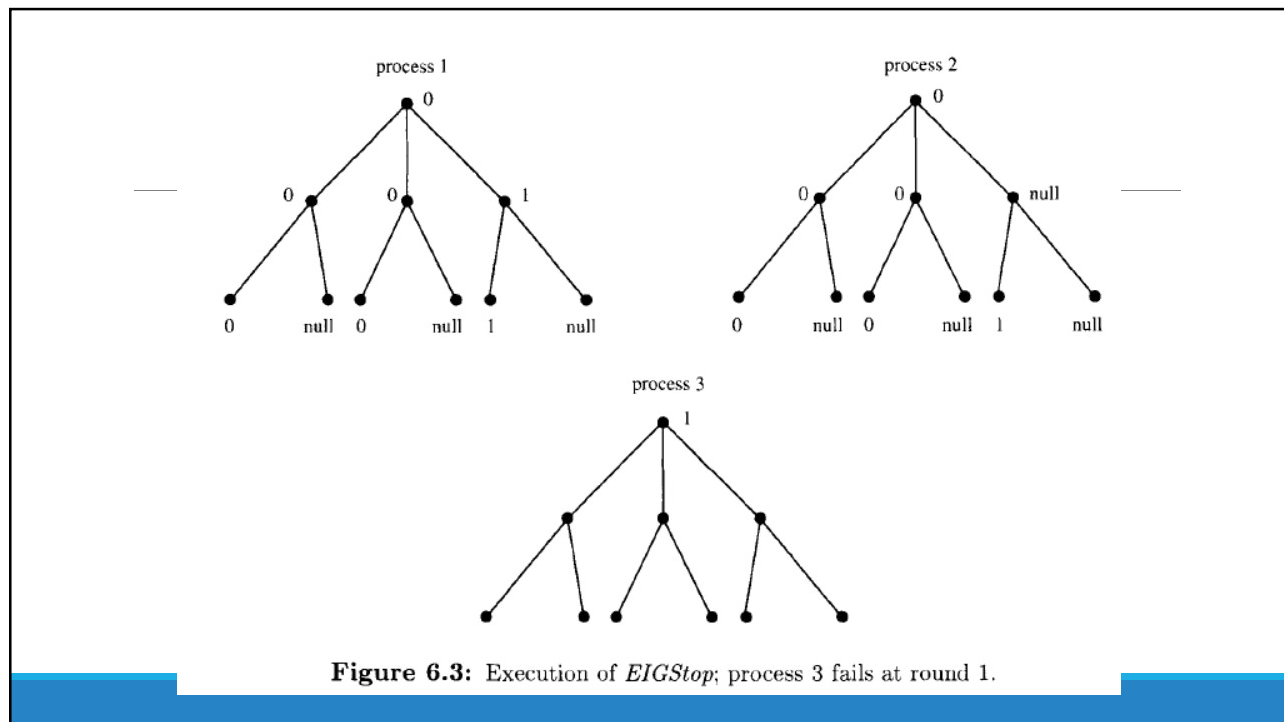
1. If  $xj$  is a level  $k$  node label in  $T$ , where  $x$  is a string of process indices and  $j$  is a single index, and a message saying that  $val(x) = v \in V$  arrives at  $i$  from  $j$ , then  $i$  sets  $val(xj)$  to  $v$ .
2. If  $xj$  is a level  $k$  node label and no message with a value in  $V$  for  $val(x)$  arrives at  $i$  from  $j$ , then  $i$  sets  $val(xj)$  to  $null$ .

At the end of  $f + 1$  rounds, process  $i$  applies a decision rule. Namely, let  $W$  be the set of non- $null$  vals that decorate nodes of  $i$ 's tree. If  $W$  is a singleton set, then  $i$  decides on the unique element of  $W$ ; otherwise,  $i$  decides on  $v_0$ .

As an example of how the *EIGStop* algorithm executes, consider the case of three processes ( $n = 3$ ), one of which may be faulty ( $f = 1$ ). Then the protocol executes for 2 rounds, and the tree has 3 levels. The structure of the *EIG* tree  $T_{3,1}$  is as in Figure 6.2.



**Figure 6.2:** Structure of *EIG* tree  $T_{3,1}$ .




---

**Complexity analysis.** The number of rounds is  $f + 1$ , and the number of messages sent is  $O((f + 1)n^2)$ . (This counts each combined message sent by any process to any other at any round as a single message.) The number of *bits* communicated is exponential in the number of failures:  $O(n^{f+1}b)$ .

## Byzantine Failure

---

**Agreement:** No two nonfaulty processes decide on different values.

**Validity:** If all nonfaulty processes start with the same initial value  $v \in V$ , then  $v$  is the only possible decision value for a nonfaulty process.

**Termination:** The termination condition is the same.

---

فرضیات: گراف کامل با  $N$  گره

مقادیر مورد توافق می تواند باینری یا چند تایی باشد.

$$N > 3f$$

برای  $f$  گره خطا دار،  $2f + 1$  گره لازم است.

*triple-modular redundancy,*

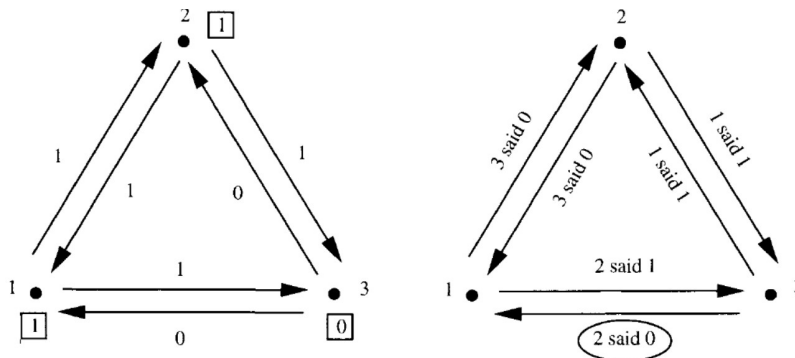
یک وظیفه به سه نفر سپرده می شود و اکثریت به عنوان نتیجه در نظر گرفته می شود.

شکلهای خطادار همیشه غلط نمیکن! ممکن هست ۹۹٪ راست و تنها  
در ۱٪ مواقع غلط بکن.

## حالت ۱

۱ و ۲ سالم و ۳ خطادار

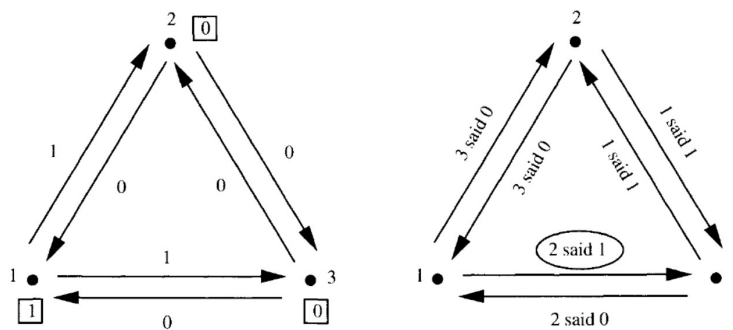
شرط اعتبار: باید روی ۱ توافق شود.



## حالت ۲

۲ و ۳ سالم و ۱ خطادار

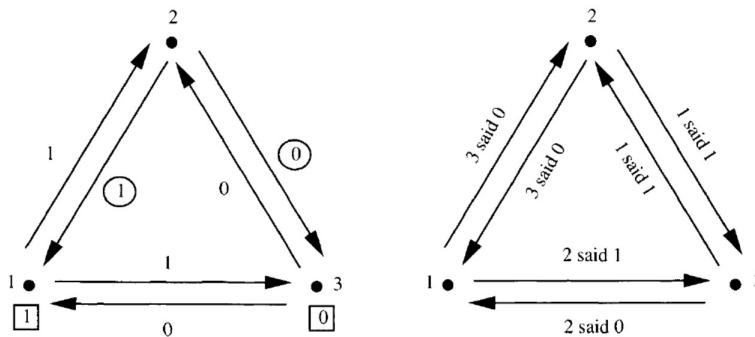
شرط اعتبار: باید روی ۰ توافق شود.





## حالت ۳

۱ و ۳ سالم و با ۱ و ۰ شروع می کنند و ۲ خطادار  
شرط اعتبار: باید روی ۰ توافق شود.



## جمع بندی

$$\alpha_3 \sim^1 \alpha_1$$

$\alpha_3$  and  $\alpha_1$  are indistinguishable to process 1

$$\alpha_3 \sim^3 \alpha_2$$

۱ در حالت ۱ تصمیم ۱ می گیرد پس باید در اجرای ۳ هم تصمیم ۱ بگیرد.  
۳ در حالت ۲ تصمیم ۰ می گیرد پس باید در حالت ۳ هم تصمیم ۰ بگیرد.  
در حالت ۳، ۱ تصمیم ۱ و ۳ تصمیم ۰ می گیرد که با شرط توافق در تناقض است.  
برای سه گره و یک گره خطادار، راه حل وجود ندارد. برای اثبات باید تعداد مراحل بیشتر و حالت های کامل تری در نظر گرفته شود.  
گره ۱ در حالت ۳ می داند یکی از گره ها خطا دارد، اما نمی داند کدام؟

## EIGByz Algorithm

---

$N > 3f$

$n = 7$  and  $f = 2$

هر نود، مقدار خود را در  $f + 1$  مرحله ارسال می کند (مشابه EIGStop)، به استثنای اینکه اگر پیامی دریافت کند که معیوب باشد، دور انداخته می شود (همانند وقتی که پیامی دریافت نکرده باشد رفتار می شود).

پس از مرحله  $f + 1$  تصمیم گیری صورت می پذیرد.

## نحوه تصمیم گیری

---

همه مقادیر null با مقدار پیش فرض  $v_0$  جایگزین می شوند.

برای تصمیم گیری نهایی، هر نود به هر یک از گره ها یک مقدار جدید تخصیص می دهد.

این کار از برگ ها آغاز می شود.

برای برگ ها:  $newval(x) := val(x)$ .

برای گره های غیر برگ: مقدار جدید بر اساس اکثریت مطلق مقادیر فرزندان تعیین می شود. چنانچه اکثریت وجود نداشت، مقدار پیش فرض در نظر گرفته می شود.

تصمیم نهایی نود  $i$  برابر است با:

$newval(i)$

---

**Complexity analysis.** The number of rounds is  $f + 1$ , and the number of messages sent is  $O((f + 1)n^2)$ . (This counts each combined message sent by any process to any other at any round as a single message.) The number of *bits* communicated is exponential in the number of failures:  $O(n^{f+1}b)$ .

---

منبع:

فصل 6 از کتاب Nancy Lynch