17 Fault-tolerant Clock Distribution

Chapter Contents

17.1	Overview	266
17.2	Local Faults and Probabilistic Resilience Guarantees	269
17.3	Gradient TRIX	275

Learning Goals

CL: todo

17.1 Overview

For most of the second part of this book, we have focused on fully connected topologies. This has some justification, as Theorems 9.2 and 14.5 and Corollary 9.3 show that high resilience to (permanent) faults requires high connectivity. However, in many cases it may simply not be practical to have a fully connected system – e.g., having $\frac{n(n-1)}{2}$ links on a chip means that we quickly run out of (physical) space for all these wires! This is only aggravated by the need to avoid correlated faults. In particular, implementing all-to-all communication via some low-degree interconnection network entails that a few faults in all the wrong places may affect communication between all or almost all nodes.

But what if faults are distributed "nicely?" If there is no mastermind orchestrating an attack on the system, it seems rather far-fetched to expect the worst possible distribution of faults. Since we know that we need to avoid correlated faults, our designs should result in an (almost) independent probability of failure for each node in the system. We explored this approach in Chapter 11, where we augmented tree topologies to handle f "local" faults.

Unfortunately, this still does not rid us of one important limitation of tree-like topologies: they perform poorly with respect to the local skew. As we know from Chapter 8, it is possible to achieve skews between adjacent nodes that are logarithmic in the network diameter D. However, in tree-like topologies, this skew is proportional to the depth of the tree structure. In this chapter, we revisit the low-degree setting, focusing on a very simple grid-like topology in which we can simultaneously achieve tolerance to 1 neighbor of each node being faulty, self-stabilization, *and* a strong bound on the local skew.

Like in Chapter 11, we distribute the clock signal in a directed fashion. This greatly simplifies achieving self-stabilization; we can then run a self-stabilizating clock synchronization algorithm to generate the clock signal in a fault-tolerant way by a few fully connected nodes.

In Section 17.2, we revisit the question what good considering local faults does in the context of the grid topology we consider in this chapter. We show that if the occurrence of faults is largely independent between nodes, i.e., they can be considered fault-containment regions (cf. Section 9.2.1), a failure of the system as a whole can be avoided despite way more faulty nodes than the worst-case analysis from Chapter 9 suggests.

Corollary 17.4. *For $f \in \mathbb{N}$, suppose that for each node the probability to be faulty is independently bounded from above by $p = o\left(\frac{1}{n^{-(f+1)}}\right)$ and (in-)degrees are $O(f)$. Then with probability $1 - o(1)$, each node has at most f faulty (in-)neighbors.*

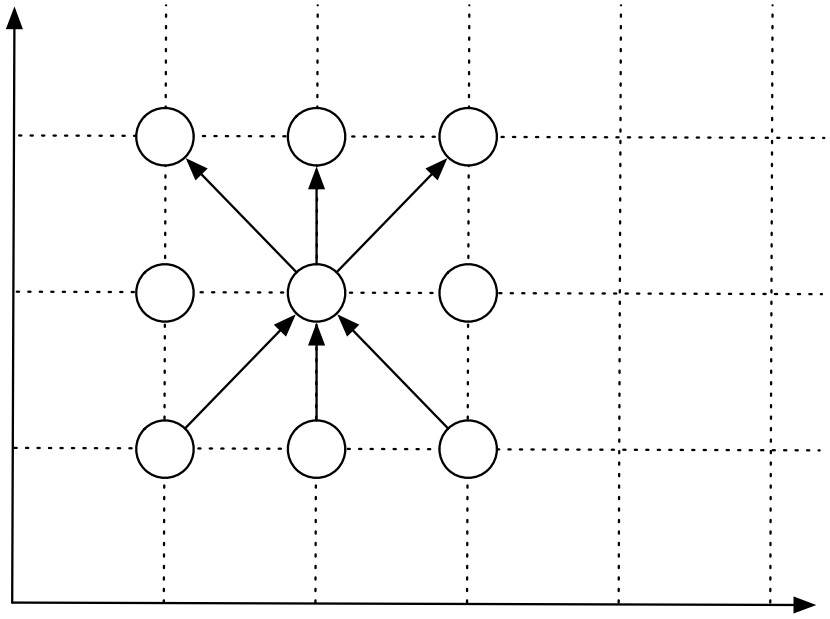

Figure 17.1
Structure of the directed grid used for clock propagation.

E17.1 The requirement of independence is critical. Find a probabilistic distribution of faults in which each node fails with probability $o\left(\frac{1}{n^{-(f+1)}}\right)$, yet the probability that there are at most f local faults is $o(1)$!

E17.2 Expecting complete independence is unrealistic. However, the statement of the corollary merely asks for probabilities to be "independently bounded." Find a distribution where probabilities are not independent, yet indeed for each node the probability to fail is at most p regardless of which other nodes fail.

Note that Corollary 17.4 is quite resilient to possible dependencies: Even if for each node the probability to fail can vary by a constant factor depending on whether other nodes fail or not, it is sufficient to show for each node that under *some* assigned failure pattern for the other nodes it fails with probability $o\left(\frac{1}{n^{-(f+1)}}\right)$. This is an application of the technique of stochastic dominance, which intuitively means to make use of the fact that additional faults can only "make things worse" and *pretend* that faults are independent, albeit more likely.

In this chapter, we study a specific, very simple directed grid designed for handling 1 fault in each in-neighborhood, see Figure 17.1. It has several

desirable properties: all propagation paths to a node have the same length, all links are local (i.e., short in a physical layout) and there are few edge-crossings (for reasonable layout), and local skew appears to not build up easily. With the most straightforward propagation rule of forwarding a pulse once the pulse signal is received from the second predecessor, most of these properties can be readily shown. If we let nodes "forget" about messages the received too long ago, also self-stabilization is easy to show.

Theorem 17.5. *Suppose that $P_{\min} \geq 2\vartheta(\mathcal{S} + 3Lu) + d$. If nodes on non-input layers follow Algorithm 27, layer $\ell \in [L+1]$ solves pulse synchronization with skew $\mathcal{S}(\ell) = \mathcal{S} + \ell u$, and period bounds $P_{\min}(\ell) = P_{\min} - \ell u$ and $P_{\max}(\ell) = P_{\max} + \ell u$ with stabilization time $\ell(2P_{\min} + \mathcal{S}(L))$.*

Given that the diameter of the grid is larger than ℓ, a *global* skew of $\mathcal{S} + \ell u$ is the best we can hope for.

Remark 17.1.

We used "local skew" here in a slighty informal sense. Of course, the "adjacent" nodes on the same layer would be physically close in a reasonable physical layout of the system, but they are not neighbors in the clock distribution network. The distinction is mostly academic, though: (i) they are only two hops apart in the underlying undirected graph and (ii) the network moving the data used for computation is likely to have links between the regions clocked based on (physically) adjacent nodes.

The time difference between pulses of adjacent nodes in different layers (i.e., a node in layer ℓ and its successors in layer $\ell+1$) is proportional to $\mathcal{S}(\ell+1)+d$. However, assuming that the lower bound on $P_{\min}$ from Theorem 17.5 is smaller than d, we can choose $P_{\min} \approx d$, such that adjacent nodes still produce pulses with skew $O(uL)$ (granted that $\mathcal{S} = O(uL)$).

With respect to the local skew, "appears" was a deliberate choice of wording, because even for a single pulse *without faulty nodes,* the above skew bound is tight not only for the global, but also the *local* skew!

Theorem 17.2. *If the forwarding rule is "wait until received a pulse message from two predecessors," there is a (single pulse) execution for which the skew on layer 0 is $\mathcal{S}$ and on layer ℓ, the skew between adjacent nodes is $\mathcal{S} + u\ell$.*

E17.3 Prove the theorem. Hint: Use delays of $d - u$ for one "half" of the network and delays of d for the other.

In Section 17.3, we seek to remedy this situation by modifying the rules according to which the nodes in this topology forward clock pulses. Making use

of the ideas from Chapter 8, we ensure a gradient property of the propagation algorithm. We make use of a simulation argument in which each communication step (i.e., increase in distance from the source) in the pulse forwarding algorithm takes the role of advancing time by one unit. The uncertainty in link delays thus in part takes the role of the clock drift.

E17.4 Suppose a clock pulse is propagated along a path of n nodes, where as usual the end-to-end delay varies between $d-u$ and d on each hop. If you interpret the sending times of the pulse message at node $i \in [n]$ as a hardware clock reaching value $i(d-u)$, what is its drift?

E17.5 Now suppose the nodes do not immediately forward the pulse, but let T time pass on their hardware clock, which is interpreted as the "path hardware clock" advancing by T as well. What is the drift of the compound clock?

E17.6 If $\vartheta - 1 \ll u/(d-u)$, can you modify how the "path hardware clock" is defined such that it has a better drift bound?

While the gradient clock synchronization algorithm cannot handle faults in general, the effect of a node being faulty for one time step can be kept fairly limited. As in the simulation each faulty node represents only a single time step of a simulated node, this means that the impact of a faulty node can be controlled. The self-stabilization properties of the gradient clock synchronization algorithm then take care of reducing the (possibly) introduced additional skew while propagating the pulse farther.

17.2 Local Faults and Probabilistic Resilience Guarantees

Theorem 17.3. *Suppose that for each node the probability to be faulty is independently bounded from above by p and nodes have (in-)degree at most Δ. Moreover, assume that*

- $p = o\left(\frac{f}{\Delta n^{-(f+1)}}\right)$ *or*
- $p \leq \frac{f+1}{3e\Delta}$ *and* $f \geq \log n$.

Then there are at most f local faults with probability $1 - o(1)$.

Proof. W.l.o.g., we assume that all nodes have in-degree Δ, as for any assignment of faulty nodes adding edges can only increase the number of faulty neighbors a node has. Thus, for a single node, the probability that more than f of its neighbors are faulty equals

$$S := \sum_{f'=f+1}^{\Delta} \binom{\Delta}{f'} p^{f'} (1-p)^{\Delta - f'}.$$

As $p \leq \frac{f}{\Delta}$, for each $f' \in \{f+1, \dots, \Delta - 1\}$ we can bound the ratio of the $(f'+1)$-th and f'-th summand by

$$\frac{p}{1-p} \cdot \frac{\binom{\Delta}{f'+1}}{\binom{\Delta}{f'}} = \frac{p}{1-p} \cdot \frac{\Delta - f'}{f'+1} \leq \frac{f}{\Delta - f} \cdot \frac{\Delta - f'}{f'+1} < \frac{f}{f+1}.$$

Setting $q := \frac{f}{f+1}$, it follows that

$$\begin{aligned}
S \leq \binom{\Delta}{f+1} p^{f+1} (1-p)^{\Delta-f-1} \sum_{i=0}^{\Delta-f-1} q^i &< (f+1)\binom{\Delta}{f+1} p^{f+1} (1-p)^{\Delta-f-1} && \text{geometric sum} \\
&< (f+1)\binom{\Delta}{f+1} p^{f+1} && 1-p < 1 \\
&\leq (f+1)\left(\frac{e\Delta p}{f+1}\right)^{f+1}. && \binom{a}{b} \leq \left(\frac{ea}{b}\right)^b
\end{aligned}$$

If $p = o\left(\frac{f}{\Delta n^{1/(f+1)}}\right)$, we can further bound

$$\begin{aligned}
S < (f+1)\left(\frac{e\Delta p}{f+1}\right)^{f+1} &= o\left(\frac{(f+1)^{1/(f+1)}}{n^{1/(f+1)}}\right)^{f+1} && p = o\left(\frac{f/\Delta}{n^{1/(f+1)}}\right) \\
&= o\left(\frac{1}{n}\right). && a^{1/a} = e^{\frac{\ln a}{a}} < e \text{ for } a > 0
\end{aligned}$$

On the other hand, if $p \leq \frac{f+1}{3e\Delta}$ and $f \geq \log n$, we can bound

$$\begin{aligned}
S < (f+1)\left(\frac{e\Delta p}{f+1}\right)^{f+1} &\leq (f+1)\left(\frac{1}{3}\right)^{f+1} && p \leq \frac{f+1}{3e\Delta} \\
&< (f+1) \cdot \left(\frac{2}{3}\right)^{f+1} \cdot \frac{1}{n} && f \geq \log n \\
&= o\left(\frac{1}{n}\right). && f \geq \log n = \omega(1)
\end{aligned}$$

Hence, either way, $S = o\left(\frac{1}{n}\right)$. Applying the union bound over all nodes, the overall probability of having more than f local faults is thus bounded by $\sum_{i=1}^{n} o\left(\frac{1}{n}\right) = o(1)$. □

Corollary 17.4. *For $f \in \mathbb{N}$, suppose that for each node the probability to be faulty is independently bounded from above by $p = o\left(\frac{1}{n^{-(f+1)}}\right)$ and (in-)degrees are $O(f)$. Then with probability $1 - o(1)$, each node has at most f faulty (in-)neighbors.*

Proof. Because $\Delta = O(f)$, we have that $p = o\left(\frac{1}{n^{-(f+1)}}\right) = o\left(\frac{f}{\Delta n^{-(f+1)}}\right)$. Thus the claim is immediate from Theorem 17.3. □

17.2.1 Using the Grid Naively

In the following, assume that we have solved pulse synchronization for layer 0, i.e., layer 0 is synchronized and can serve as an "input layer" providing the grid with pulses of skew $\mathcal{S}$ and desirable period bounds $P_{\min}$ and $P_{\max}$. The total number of layers is $L + 1 \in \mathbb{N}_{>0}$, i.e., there are L non-input layers.

A simple approach then is to have each node in the grid wait until it received pulse messages from two of its predecessors and then propagate the pulse. Nodes can safely clear the local memory indicating that a pulse was received from a predecessor after some time, as well as that the pulse has been generated. Assuming that pulses are far enough apart, this readies them in time for the next pulse.

Algorithm 27 Naive forwarding algorithm

```
1: while true do
2:     wait until received ⟨pulse⟩ from two distinct predecessors within ϑ(S + Lu) local time
3:     locally generate pulse and send ⟨pulse⟩ to successors
4:     wait for ϑ(S + Lu) local time
5: end while
```

Since at most one predecessor is faulty, this means that the node forwards the pulse within the time interval spanned by the arrival times of pulse messages from its predecessors. And since each node locally clears its memory after each pulse, all but possibly the first (correct) pulse from the input layer are propagated correctly. The approach is thus trivially self-stabilizing. Unfortunately, the skew bound is also trivial: the worst-case skew is $u\ell + \mathcal{S}$ after ℓ layers, where $\mathcal{S}$ is the skew guarantee provided by the input layer 0—both for the global *and* local skew.

Theorem 17.5. *Suppose that $P_{\min} \geq 2\vartheta(\mathcal{S} + 3Lu) + d$. If nodes on non-input layers follow Algorithm 27, layer $\ell \in [L+1]$ solves pulse synchronization with skew $\mathcal{S}(\ell) = \mathcal{S} + \ell u$, and period bounds $P_{\min}(\ell) = P_{\min} - \ell u$ and $P_{\max}(\ell) = P_{\max} + \ell u$ with stabilization time $\ell(2P_{\min} + \mathcal{S}(L))$.*

Proof. We prove the claim by induction on the layer index. It trivially holds for layer 0 by assumption.

For the induction step, assume that at time t_ℓ, layer $\ell \in [L]$ has stabilized and consider layer $\ell+1$. We distinguish two cases. If no correct node in layer ℓ sends a pulse message during $[t_\ell, t_\ell+P_{\min}]$, no node on layer $\ell+1$ receives such a message during $(t_\ell+d, t_\ell+P_{\min}]$. Hence, during $(t_\ell+d+\vartheta(\mathcal{S}+Lu), t_\ell+P_{\min})$, no correct node on layer $\ell+1$ generates a pulse. It follows that by time

$$t_\ell + d + 2\vartheta(\mathcal{S}+Lu) < t_\ell + P_{\min},$$

each correct node on layer $\ell+1$ is executing the first wait instruction of the loop with no pulse message from a correct predecessor stored and no such message sent within the last $\mathcal{S}(\ell) < P_{\min}(\ell)$ time. Let us call a time satisfying these conditions *quiet at layer* $\ell+1$.

The second case is that there is a correct node in layer ℓ sending a pulse message during $[t_\ell, t_\ell+P_{\min}]$. By the induction hypothesis, the corresponding pulse has skew $\mathcal{S}(\ell) \le \mathcal{S}(L)$. Hence, if $t \le t_\ell + P_{\min} + \mathcal{S}(L)$ is the latest pulse message of this pulse send by a correct node from layer ℓ, then no correct node in layer ℓ sends a pulse message during $(t, t+P_{\min}(\ell))$. Reasoning as for the first case, we get that there is a time that is quiet at layer $\ell+1$ no later than

$$t_\ell + d + 2\vartheta(\mathcal{S}+Lu) \le t_\ell + P_{\min} - Lu \le t_\ell - P_{\min}(\ell).$$

Thus, using that $P_{\min}(\ell) \le P_{\min}$ and $t_\ell \le \ell(2P_{\min}+\mathcal{S}(L))$ by the induction hypothesis, in both cases there is a time $t_{\ell+1} \in [t_\ell+\mathcal{S}(\ell), (\ell+1)(2P_{\min}+\mathcal{S}(L))$ that is quiet at layer $\ell+1$. We claim that layer $\ell+1$ has stabilized by time $t_{\ell+1}$, which we show by induction on the pulses generated by nodes on layer $\ell+1$ from time $t_{\ell+1}$ on. Denote by $p_{v,i}$ the i-th pulse generated by node $v \in V_g$ on layer ℓ at or after time $t_{\ell+1}$. Since no pulse messages were sent by correct nodes on layer ℓ during $[t_{\ell+1}-\mathcal{S}(\ell), t_{\ell+1}]$, we have that $p_{v,1} \ge t_{\ell+1}$. Hence, each correct node will receive the pulse messages sent by its correct predecessors within a time window of length $\mathcal{S}(\ell)+u = \mathcal{S}(\ell+1)$ starting after $t_{\ell+1}$. As they execute the first waiting statement at this time (no pulse can be triggered by a single faulty predecessor at a quiet time), they will generate a pulse during this time window, showing the skew bound for this pulse. Note that $\vartheta(\mathcal{S}(\ell+1)) < P_{\min}(\ell) - d$ time after this first pulse on layer $\ell+1$ (after time $t_{\ell+1}$), i.e., before pulse messages for the next pulse from correct nodes in layer ℓ arrive, layer $\ell+1$ is quiet again.

Now we proceed by induction on the pulse index i, where the induction hypothesis includes that layer $\ell+1$ is quiet before pulse messages for pulse $i+1$ arrive. We just handled the base case of $i=1$. The same arguments show the skew bound for pulse $i+1$ and that layer $\ell+1$ is quiet again before the next pulse arrives, assuming that the claim holds for pulse i. To show the period

bounds, we apply the period bounds for layer ℓ and note that they deteriorate by u due to message delays varying between $d-u$ and d. □

Interestingly, while the naive approach performs poorly in the worst-case, local skew appears to be extremely small under independently random link delays. Here are some results from computer experiments, where for simplicity link delays are either 0 or 1 with (independent) probability of $1/2$ each, and there is no skew in the first layer.

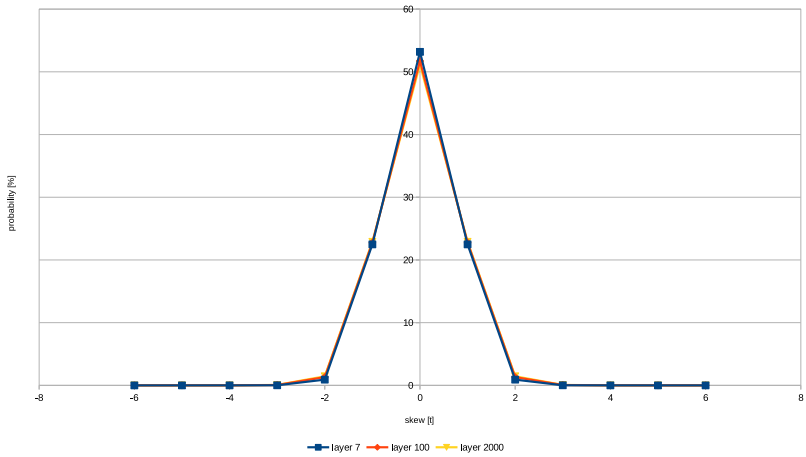

Figure 17.2
Distributions of skews between neighbors in layers 7, 100, and 2000 for 0-1-delays chosen by fair independent coin flips. This suggests a distribution with extremly small standard deviation, where the number of layers has very limited influence.

Remark 17.6.
We do not understand why the grid appears to work so extraordinarily well with randomized link delays and naive forwarding.
Do the skews just grow very slowly, or converge to a limit distribution?
Is the grid even reducing larger skews effectively?
For a practical realization, one has to further adapt the topology. Concretely, the clock source should not be a (wide) intial layer, but rather a few nodes. This suggests a layout with layers being nested circles, adding a constant number of nodes per layer. These and other concerns will affect a final design, requiring further studies.

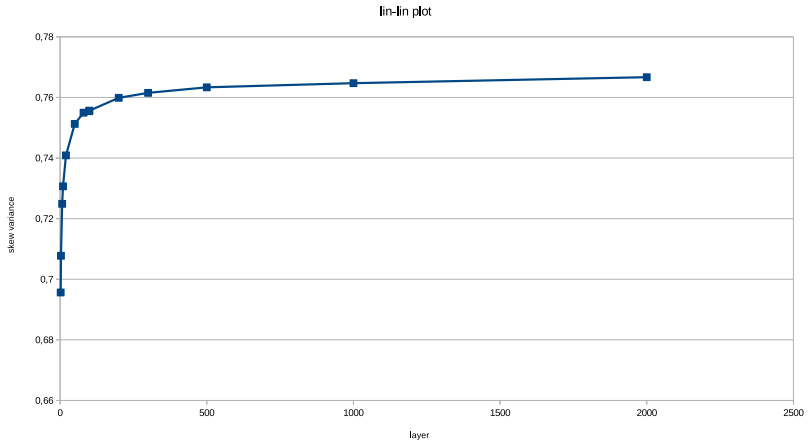

Figure 17.3
Plot of the variance of the above distribution as function of the number of layers. Does it grow very slowly, but is unbounded, or does it converge to a fixed value corresponding to a limit distribution?

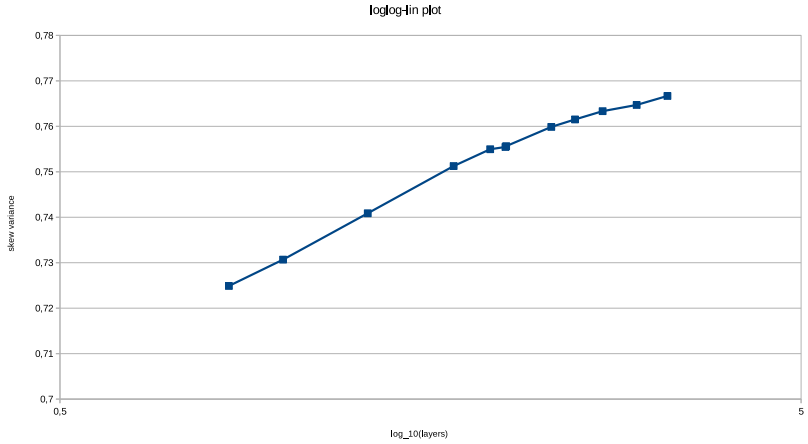

Figure 17.4
The same plot, but with a doubly logarithmic x-scale. One may suspect that the growth is bounded by $\log \log L$, where L is the number of layers. However, the considered range of values is rather small for this (especially given the very small change of the actual y-values), and the number of experiments may be insufficient for a reliable assessment.

17.3 Gradient TRIX

Remark 17.7. *This section is in need of being written. We hope to give a preview of its contents in form of the corresponding paper draft.*

Bibliography

[1] Hopkins, A. L., T. B. Smith, and J. H. Lala. 1978. Ftmp – a highly reliable fault-tolerant multiprocess for aircraft. *Proceedings of the IEEE* 66 (10): 1221–1239. doi:10.1109/PROC.1978.11113.

[2] Kopetz, H. 2003. Fault containment and error detection in the time-triggered architecture. In *The sixth international symposium on autonomous decentralized systems, 2003. isads 2003.*, 139–146. doi:10.1109/ISADS.2003.1193942.

[3] Pease, M., R. Shostak, and L. Lamport. 1980. Reaching agreement in the presence of faults. *J. ACM* 27 (2): 228–234. doi:10.1145/322186.322188. http://doi.acm.org/10.1145/322186.322188.

[4] Srikanth, T. K., and Sam Toueg. 1987. Optimal clock synchronization. *J. ACM* 34 (3): 626–645. doi:10.1145/28869.28876. https://doi.org/10.1145/28869.28876.