Brief Announcement: On the Feasibility of Local Failover Routing on Directed Graphs

Erik van den Akker[1][0000−0003−4956−255X] and Klaus-Tycho Foerster[1][0000−0003−4635−4480]

TU Dortmund, Germany

Abstract. Local failover mechanisms are used to achieve fast recovery from link failures in communication networks. These mechanisms are typically implemented using static routing tables at the nodes of a network, only relying on failures of outgoing links, as well as the label of the source and target node of a packet (called $s-t$-routing). Static failover $s-t$-routing on undirected graphs has been shown to be able to tolerate at most 2 failures, denoted 2-resilient, with 3-resiliency being impossible without additional rewritable bits in the packet header.

In this work, we investigate local failover routing on directed graphs with n nodes and show lower and upper bounds on the number of bits required. Even 1-resilience cannot be achieved on all topologies without additional bits and we prove that 1-resilience can be obtained with $\lceil\log(n)\rceil$ bits. For $k > 1$ failures, we show that at least $\lceil\log(k+1)\rceil$ bits are necessary, but that $k(\lceil\log(|E|)\rceil)$ bits are sufficient to obtain k-resiliency.

Keywords: routing · directed graphs · networks · failover · connectivity

1 Introduction and Background

Maintaining routing connectivity is crucial in communication networks [2] and hence modern networked systems implement some form of fast failover routing, to quickly react to equipment failures [8] until global routing tables are (relatively slowly [7]) re-computed and re-distributed. However, past research has shown [2] that there is a trade-off w.r.t. coverage (how many failures can be tolerated) and the number of communication rounds needed. Ideally this fast failover routing is implemented without any communication post-failures in the form of static local forwarding rules, to guarantee immediate reaction.

In this setting on undirected graphs $G = (V, E)$, Feigenbaum et al. [5] proved that without source-information (denoted as t-routing) or rewritable bits in the packet header, at most 1 edge failure can be tolerated (denoted 1-resilience). Here Chiesa et al. [3] showed that resilience to 2 edge failures is impossible in general, though in practice a good amount of topologies can still be 2-resilient [1]. When matching on the source as well (denoted as $s-t$-routing), Dai et al. [4] proved that 2-resilience can always be obtained, but that 3-resilience is impossible in general without rewritable bits in the packet header. Moreover, Lakshminarayanan et al. [6] gave a scheme that can survive any k number of arc failures, assuming the

Number of Failures	Lower Bound	Upper Bound
1	1 (Thm.1)	$\lceil \log(n) \rceil$ (Thm.3)
$k > 1$	$\lceil \log(k+1) \rceil$ (Thm.2)	$k(\lceil \log(\lvert E \rvert) \rceil)$ (Thm.4)

Table 1: Overview of the lower and upper bounds for directed failover routing

network stays connected, by carrying information about all encountered failed links in the packet header, requiring $k \cdot \lceil \log |E| \rceil$ rewritable bits to succeed.

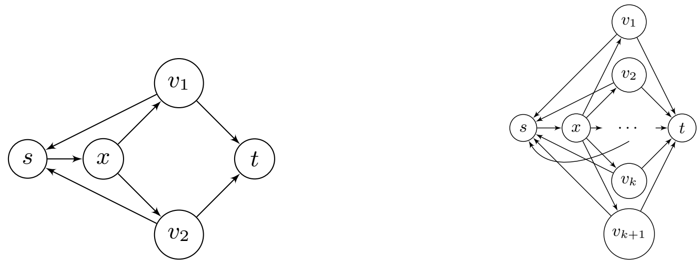
Wada et al. [9] considered a similar problem of fault-tolerant fixed routings on directed graphs, where they studied the size of surviving route graphs. Their model allowed only up to k arc failures for $k+1$-connected graphs, while our model allows an arbitrary amount of arc failures, as long as no deadends are constructed. Also, instead of local forwarding rules for the next hop at any node, their model uses a set of paths between each pair of nodes in the graph to construct the routing.

Contribution. In this work we investigate fast failover $s-t$-routing with static routing tables on *directed graphs*. To the best of our knowledge, there are no known upper or lower bounds in this setting, except for the above mentioned results by Lakshminarayanan et al. [6], which can be applied to directed graphs as well. A summary of our results is given in Table 1.

2 Model

We consider a network as a directed, simple graph $G = (V, E)$, in which nodes represent routers and edges represent links, and nodes are labeled with unique ids. For each node we define a set of *local forwarding rules*, having the form $in \times f \times s \times t \times b \rightarrow out, b'$, where on the left side in is the incoming arc the packet came from (or $\perp$ if the packet originates from the current node), f is the set of locally failed outgoing arcs (routers are assumed to keep track of local failures of outgoing links), s is the id of the source, t the id of the target of the packet and b is the bitstring in the packet header. On the right side, out is the outgoing arc chosen under these conditions and b' is the new bitstring written to the packet header. Whenever a packet arrives at a node (router) that is not the target t, it uses the information available locally and in the packet header, finds the corresponding forwarding rule, rewrites the bitstring and sends the packet to the next node. Directed Local Failover Routing can be described as a game between a player 1 and a player 2 (the adversary):

1. Two players are given a network as a directed, simple graph $G = (V, E)$ with $n = |V|$ nodes and a value k.
2. For each node, player 1 defines the set of local forwarding rules, which must be able to route from s to t, as long as these nodes are connected in G.
3. An adversary (player 2) can now remove a subset $F \subseteq E$ in G, containing up to k arcs.



(a) 1 edge failure (b) Generalized for $k > 1$ arc failures

Fig. 1: Lower bound constructions for 1 and k arc failures

4. We now have to check: can our local forwarding rules still route any packet from s to t, as long as any node that can be reached from s still has a connection to t[1]
5. If routing still works: player 1 wins, else the adversary (player 2) wins.

The question is: is there some set of local forwarding rules for G, so that an adversary cannot find a set F of size at most k, that can break the forwarding pattern, or: does player 1 always have a winning strategy for the input (G, k)? If this is the case, we call the forwarding-pattern k-resilient.

3 Lower Bounds

In this section, we cover the lower bounds for the number local failover routing on directed graphs, first in Thm.1 looking at the lower bound with exactly one failed arc, showing that we already need a rewritable bit in the case of a single failure, while in the undirected case for $s-t$-routing, two failures can generally be tolerated. We then show in Thm.2, that for $k \geq 1$ failures, we generally need at least $\lceil \log(k+1) \rceil$ rewritable bits in the packet header.

Theorem 1. ***Lower Bound for a single failure***
$s-t$-routing strategies with local forwarding rules on directed graphs with one failed arc need at least one rewritable bit in the packet header.

Proof. Consider the graph shown in Figure 1a: From s any routing strategy can only send the packet to x, where the routing strategies can only differ in sending the packet to either v_1 or v_2. W.l.o.g we assume v_1 is chosen. The adversary can now remove the arc (v_1, t), which leads the packet to be sent via (v_1, s). From s the only option is to send the packet to x. If there is no rewritable information in the packet header, the packet will now repeatedly follow the loop s, x, v_1, s. $\square$

With one rewritable bit in the packet header, the bit could be flipped when traversing the arc (v_1, s), influencing the decision at x, where then the arc (x, v_2)

[1] Without this assumption, the adversary could easily send the player 1 into deadends.

could be chosen. Since the arc (v_2, t) must be intact for t to be reachable from s, we can reach t using one rewritable bit in the packet header in this case.

Theorem 2. ***Lower Bound for multiple failures***
$s-t$-routing strategies on directed graphs with local forwarding rules with k failed arcs need at least $\lceil \log(k+1) \rceil$ rewritable bits in the packet header.

Proof. Consider a graph as in Figure 1b, with $k+4$ nodes $s, x, t, v_1, ..., v_k, v_{k+1}$. If an adversary is allowed to delete up to k arcs and our strategy has less than $\lceil \log(k+1) \rceil$ rewritable bits in the packet header available, then we can only encode up to k different configurations $c_1, ..., c_k$ in the bitstring of the packet header. The adversary can now remove the arcs (v_j, t) for each arc (x, v_j) that is chosen in any of the k configurations, which will lead to the packets from s not reaching t. □

Note that with $\lceil \log(k+1) \rceil$ rewritable bits, the bitstring has $k+1$ possible configurations, which can then be used to try $k+1$ different arcs (x, v_i), until a node with an arc to t is available (as there are at most k failures).

4 Upper Bounds

In this section, we look at the maximum number of rewritable bits in packet headers needed, first showing in Thm.3, that for a single failure, we need at most $\lceil \log(n) \rceil$ rewritable bits, where n is the number of nodes, which are used to encode the node where the default path failed, implicitly marking the failed arc. Afterwards, we show in Thm.4 that for k failures at most $k(\lceil \log(|E|) \rceil)$ bits are needed, encoding each of the k failed arcs.

Theorem 3. ***Upper Bound for a single failure***
There exists a routing strategy for $s-t$ routing with local forwarding rules on directed graphs with n nodes and one failed arc, that needs at most $\lceil \log(n) \rceil$ rewritable bits in the packet header.

Proof. A forwarding pattern can define a default path from s to t, using local forwarding rules. In an attempt to break the routing pattern, the adversary must remove an arc f from the default path. However, the forwarding pattern can write the id of the node where the failed arc was encountered to the bitstring b with $\lceil \log(n) \rceil$ bits. The bitstring is initially set to the id of s, meaning no failure has occured yet, and will be set to the id of t, if the failure already occurs at s. Since every node that is reachable from s still must have a path to t in $G - \{f\}$, there must still be some path from any reachable node to t, that does not use the arc f. We can then encode this path with local forwarding rules, which are used when b is set to the id of the node from the default path. Since only one arc can fail in this case, this implicitly means, that the arc f failed. □

Theorem 4. ***Upper Bound for multiple failures***
There exists a routing strategy for $s-t$ routing with local forwarding rules on directed graphs with n nodes and k failed arcs, that needs at most $k(\lceil \log(|E|) \rceil)$ rewritable bits in the packet header.

Our proof of Theorem 4 is based on the idea of Lakshminarayanan et al. [6], who also encoded all so far encountered arc failures as packet header bits.

Proof. We can encode fallback paths using information about all failed arcs we have already encountered, by encoding fallback paths for each node under each possible failure set up to size k. We can use the bitstring of size $k(\lceil \log(|E|) \rceil)$, to encode all k failed arcs. Whenever we encounter an error, we add the newly found failure to the bitstring, which can be initially set to all zeroes and try the fallback path defined for the currently found set of failures. If we have not reached t yet when finding the last failed arc, we can reach t using the fallback path for the complete failure set. □

5 Conclusion

We investigated upper and lower bounds on the number of bits required for local failover routing on directed graphs, showing that even for $s-t$-routing, resilience to one (k) failure(s) is not possible with 0 ($\lceil \log(k) \rceil$) bits, while in the undirected case two failures can always be tolerated without the use of any additional bits. We also showed that one (k) failure(s) can always be tolerated with logarithmic multiplicative overhead. Besides closing the gaps between upper and lower bounds in general, it would also be interesting to investigate the resilience and number of bits required for restricted graph classes, like planar graphs, since even for $k = 2$ failures, the graph with 6 nodes used in our lower bound construction (Figure 1b) is not planar.

References

1. van den Akker, E., Foerster, K.: Short paper: Towards 2-resilient local failover in destination-based routing. In: ALGOCLOUD (2024)
2. Chiesa, M., Kamisinski, A., Rak, J., Rétvári, G., Schmid, S.: A survey of fast-recovery mechanisms in packet-switched networks. IEEE Commun. Surv. Tutorials **23**(2), 1253–1301 (2021)
3. Chiesa, M., Nikolaevskiy, I., Mitrovic, S., Gurtov, A.V., Madry, A., Schapira, M., Shenker, S.: On the resiliency of static forwarding tables. IEEE/ACM Trans. Netw. **25**(2), 1133–1146 (2017)
4. Dai, W., Foerster, K., Schmid, S.: A tight characterization of fast failover routing: Resiliency to two link failures is possible. In: SPAA. pp. 153–163. ACM (2023)
5. Feigenbaum, J., Godfrey, B., Panda, A., Schapira, M., Shenker, S., Singla, A.: Ba: On the resilience of routing tables. In: PODC. pp. 237–238. ACM (2012)
6. Lakshminarayanan, K., Caesar, M., Rangan, M., Anderson, T., Shenker, S., Stoica, I.: Achieving convergence-free routing using failure-carrying packets. In: SIGCOMM. pp. 241–252. ACM (2007)
7. Liu, J., Panda, A., Singla, A., Godfrey, B., Schapira, M., Shenker, S.: Ensuring connectivity via data plane mechanisms. In: NSDI. pp. 113–126. USENIX (2013)
8. Rak, J., Hutchison, D. (eds.): Guide to Disaster-Resilient Communication Networks. Computer Communications and Networks, Springer (2020)
9. Wada, K., Kawaguchi, K.: Efficient fault-tolerant fixed routings on $(k+1)$-connected digraphs. Discret. Appl. Math. **37/38**, 539–552 (1992)