15 Robustness and Resilience

Gunnar W. Klau and René Weiskircher

Intuitively, a complex network is *robust* if it keeps its basic functionality even under failure of some of its components. The study of robustness in networks is important because a thorough understanding of the behavior of certain classes of networks under failures and attacks may help to protect, for instance, communication networks like the Internet against assaults or to exploit weaknesses of metabolic networks in drug design.

Often, we distinguish between random failure and intentional attacks. Examples for random and intentional component failures in real-world complex networks are, for instance, mutations in a cell, pharmaceutical or environmental stress on metabolic networks, router failures in the Internet, or intentional attacks on airline or highway networks. We will see that some networks like the Internet are very robust against random drop-outs of routers but may suffer heavily from targeted attacks against well-chosen central routers.

This chapter is dedicated to network statistics that are of interest with respect to a network's robustness or its resilience against repeated component failure. We give an overview of a variety of statistics and discuss their applicability in practice in terms of usefulness and computational complexity. Often, research on robustness focuses on how these statistics change, by analyzing or measuring the effects if a network undergoes a sequence of component failures. Wherever possible we try to relate the different statistics and discuss their advantages and disadvantages. In many cases, we use examples to illustrate the definitions.

We chose to organize this chapter as follows: We distinguish between worst case, average, and probabilistic statistics. Sections 15.1 and 15.2 cover worst case connectivity and distance measures. Average robustness statistics (Section 15.3) allow a more global perspective on robustness properties whereas probabilistic statistics (Section 15.4) consider the failure probabilities implicitly. While, roughly speaking, the statistics become more and more meaningful the more they are located towards the end of this chapter, they are also more difficult to compute. We conclude this chapter in Section 15.5 with final remarks and list some open problems.

15.1 Worst-Case Connectivity Statistics

This section deals with statistics that answer questions of the form "What is the minimum number of edges or vertices that have to be deleted from the network

U. Brandes and T. Erlebach (Eds.): Network Analysis, LNCS 3418, pp. 417–437, 2005.

such that the resulting network is disconnected and has property P ?". These are worst case statistics because the deletion of an arbitrary set of vertices or edges of the same size may not cause the same effect. So we implicitly assume that the vertex or edge failures are not random but targeted for maximum effect.

15.1.1 Classical Connectivity

Classical connectivity is the basis of many robustness statistics. A network is called *connected*, if there exists a path between every pair of vertices in the network. In many applications, connectedness is a necessary condition for a network to fulfill its purpose. Therefore, one measure of robustness of a network is the number of vertices or edges that have to be removed to make the network unconnected. These are called the *vertex-connectivity* and *edge-connectivity* of the network, respectively. They are treated in depth in Chapter 7. Here we only look at connectivity as a measure of the robustness of a network.

If a network loses its functionality completely as soon as it is not connected anymore, connectivity is indeed a good measure for its robustness. But if we are concerned with the case where the usefulness of a network is not seriously affected by disconnecting a small set of vertices from the network, connectivity is not a meaningful measure. Consider the Internet as an example. A desktop computer is only connected to the net via one link to a provider or server. Cutting this link disconnects the net but has only a negligible influence on the functionality of the whole Internet. Yet the edge-connectivity of the net is only one. Similarly, the failure of a small router will only disconnect a handful of clients from the net but proves that the Internet has vertex connectivity one.

15.1.2 Cohesiveness

The notion of cohesiveness was introduced by Akiyama et al. in [13] and defines for each vertex of the network to what extent it contributes to the connectivity.

Definition 15.1.1. *Let $\kappa(G)$ be the vertex-connectivity of G (see the definition in Section 2.2.4). Let $G - v$ be the network obtained from G by removing vertex v. For any vertex v of G, the* cohesiveness $c(v)$ *is defined as follows:*

$$c(v) = \kappa(G) - \kappa(G - v)$$

Vertex 7 in Figure 15.1(a) has a cohesiveness of -2, because the network has vertex-connectivity 1 if vertex 7 is present and vertex connectivity 3 if we delete it. On the other hand, vertex 6 in Figure 15.1(b) has cohesiveness 1 because if we remove it from the network, the vertex-connectivity drops from 3 to 2.

It follows from the definition that the cohesiveness of a vertex cannot be greater than 1. Intuitively, a vertex with negative cohesiveness is an outlier of the network while a vertex with cohesiveness 1 is central. It can be shown that a network can have at most one vertex with negative cohesiveness and that the neighborhood of this *negative vertex* contains the only set of vertices of

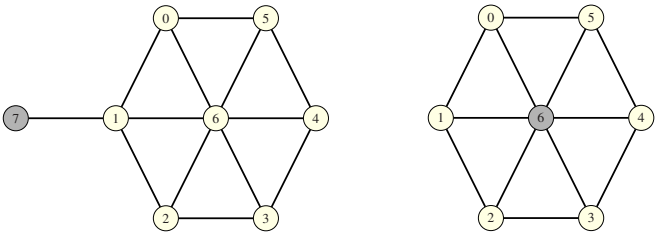


Fig. 15.1. Example graphs for the cohesiveness of a vertex. Vertex 7 in Figure 15.1(a) has cohesiveness -2 and vertex 6 in Figure 15.1(b) cohesiveness 1

size $\kappa(G)$ whose removal disconnects the network. Consider as an example the network shown in Figure 15.1(a), where vertex 7 is the only vertex with negative cohesiveness. The only neighbor of vertex 7 is vertex 1 and this is the only vertex whose deletion splits the network.

Even though a network can have at most one negative vertex, we can compute a set of loosely connected vertices by removing the negative vertex and then looking for the next negative vertex. This algorithm could be used to find loosely connected vertices in a network because a negative vertex is at the periphery of the graph. A drawback of this approach is that this algorithm may stop after a few vertices even for big networks because there are no more vertices with negative cohesiveness.

The cohesiveness of a vertex can be computed using standard connectivity algorithms (see Chapter 7). To compute the cohesiveness of every vertex, the connectivity algorithm has to be called n times where n is the number of vertices in the network.

15.1.3 Minimum m-Degree

The statistics we have mentioned so far make statements about the connectivity of a network. The m-degree was introduced in [65] by Boesch and Thomas. It is concerned with the state of the network after disconnection.

Definition 15.1.2. *The* minimum m-degree $\xi(m)$ *of a network is the smallest number of edges that must be removed to disconnect the network into two connected components G_1 and G_2 where G_1 contains exactly m vertices.*

Table 15.1 shows the m-degrees for the network in Figure 15.2.

Let $G = (V, E)$ be a network with $|V| = n$. Boesch and Thomas showed in [65] the following properties of the minimum m-degree:

- $\xi(m) = \xi(n - m)$.
- $\xi(m) \geq m(\delta(G) - m + 1)$ where $\delta(G)$ is the minimum degree of any vertex in G.

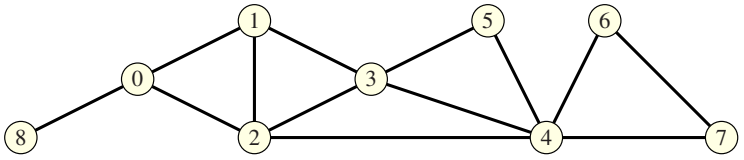


Fig. 15.2. Example network for the minimum m-degree

Table 15.1. The m-degrees for the network in Figure 15.2

1-degree	2-degree	3-degree	4-degree	5-degree
1	2	3	3	3

– Let G be a regular network with degree $r \leq n/2$, $n > 2$ and $m \geq l$. Then

$$r \geq \lfloor \xi(m)/m \rfloor + \lceil \xi(l)/l \rceil \ .$$

There is no asymptotically faster algorithm known for computing the minimum m-degree than trying all sets of vertices of size m and check if the graphs induced by the set and by its complement are connected. If this is the case, we count the number of edges connecting vertices in the set with vertices outside. The minimum over all sets is the m-degree. This results in a running time of $\mathcal{O}(\binom{n}{m}|E|)$.

The main problem of this statistics is that the splitting of the graph has to result in two *connected* components, so it does not express an intuitive concept of robustness. The network in Figure 15.3 has 3-degree 3 while the deletion of the two thick edges is enough to split a component with three vertices from the network.

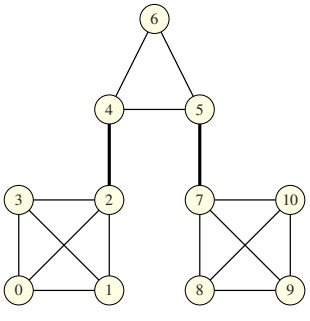


Fig. 15.3. A counter-intuitive example for the m-degree statistics

15.1.4 Toughness

The toughness of a network was introduced by Chvátal [129]. It measures the number of internally connected components that the graph can be broken into by the failure of a certain number of vertices.

Definition 15.1.3. *Let S be a subset of the vertices of G and let $K(G-S)$ be the number of internally connected components that G is split into by the removal of S. The* toughness *of G is defined as follows:*

$$t(G) = \min_{S \subseteq V, K(G-S)>1} \left\{ \frac{|S|}{K(G-S)} \right\}$$

The edge-toughness *of a network is defined analogously for edges.*

Intuitively, the toughness of a network is high if even the removal of a large number of vertices splits the network only into few components. Conversely, if a network can be split into many components by removing a small number of vertices, its toughness is small.

The toughness of a complete network is defined as infinite. The network with the smallest toughness is a star. Removing the central vertex splits the network into components of size one and so the toughness of a star with n vertices is $\frac{1}{n-1}$. Note that the central vertex is also the only one whose removal splits the graph.

It is $\mathcal{NP}$-hard to decide for a general graph if it has toughness at least t [48]. If the network is a tree, the toughness is $\frac{1}{\Delta(G)}$ where $\Delta(G)$ is the maximum degree of any vertex. The toughness of the complete bipartite network $K_{m,n}$ with $m \leq n$ and $n \geq 2$ is $\frac{m}{n}$.

The toughness of a circle is one and it follows that the toughness of a Hamiltonian graph is at least one. In [129], Chvátal also showed a connection between the *independence number* of a network and the toughness. The independence number β_0 is the size of the largest subset S of the vertices with the property that there is no edge in the network connecting two vertices in S. The toughness of G is lower-bounded by $\kappa(G)/\beta_0(G)$ and upper bounded by $(n - \beta_0(G))/\beta_0$.

15.1.5 Conditional Connectivity

Conditional connectivity was introduced by Harary in [276] and is a generalization of the minimum m-degree. The measure is parameterized with a property P that has to hold for all the components created by deleting vertices from the network.

Definition 15.1.4. *The P-connectivity $\kappa(G : P)$ of network G is the smallest number of vertices that have to be deleted from the network such that the remaining network G' has the following properties:*

1. *G' is not connected.*
2. *Every connected component of G' has property P.*

Conditional edge-connectivity is defined analogously for the deletion of edges. Conditional connectivity is potentially very useful in practice because the property P can be chosen according to the characteristics of the task that the network should accomplish. An example could be defining P as: "The component has at most k vertices". The conditional connectivity would then correspond to the

size of the smallest subset of vertices we have to delete to split the network into components of at most k vertices each. Classical connectivity is a special case of conditional connectivity where $P = \emptyset$.

If we define a sequence $S = (P_1, \ldots, P_k)$ of properties according to our application such that P_{i+1} implies P_i for $1 \leq i \leq k-1$, we obtain a vector of conditional connectivity

$$(\kappa(G : P_1), \ldots, \kappa(G : P_k)) \ .$$

If the properties are defined to model increasing degradation of the network with respect to the application, this vector gives upper bounds for the usefulness of the system with respect to the number of failed vertices.

A similar measure is *general connectivity*, also introduced by Harary [277]. If G is a network with property P and Y is a subset of the vertices (edges) of G, then $\kappa(G, Y : P)$ is the smallest set $X \subset Y$ of vertices (edges) in G whose removal results in a network G' that does not have property P. Conditional connectivity is a special case of general connectivity.

The main drawback of these statistics is that there is no efficient algorithm known that computes them for a general graph.

15.2 Worst-Case Distance Statistics

The statistics in this section make statements about the increase of distances in the network caused by the deletion of vertices or edges. These are again worst-case statistics because they give the smallest number of vertices or edges that have to be deleted in order to increase the distances. All the statistics we present in this section are only defined until the network becomes disconnected by the removal of vertices and edges.

15.2.1 Persistence

The *persistence* of a network is the minimum number of vertices that have to be deleted in order to increase the diameter (the longest distance between a pair of vertices in the network). Again, an analogous notion is defined for the deletion of edges (*edge persistence*). Persistence was introduced by Boesch, Harary and Kabell in [64] where they also present the following properties of the persistence of a network:

- The persistence of a network with diameter $2 \leq d \leq 4$ is equal to the minimum over all pairs of non-adjacent vertices i and j of the maximum number of vertex-disjoint i, j-paths of length no more than d.
- The *edge-persistence* of a network with diameter $d \in \{2, 3\}$ is the minimum over all pairs of vertices i, j of the maximum number of edge-disjoint i, j-paths of length no more than d.

There are many theoretic results on persistence that mainly establish connections between connectivity and persistence, see for example [74, 475]. The *persistence vector* is an extension of the persistence concept. The i-th component of $P(G) = (p_1, \ldots, p_n)$ is the worst-case diameter of G if i vertices are removed. This is the same concept as the vertex-deleted diameter sequence we introduce in Section 15.2.2.

The main drawback of persistence is that there is no efficient algorithm known to compute it.

15.2.2 Incremental Distance and Diameter Sequences

Krishnamoorthy, Thulasiraman, and Swamy have studied the increase of distances in a network caused by the deletion of vertices and edges [371]. They introduce for a network G four sequences A, B, D, and T defined as follows:

Definition 15.2.1. *Let $d(u,v) = d_G(u,v)$ be the distance of the two vertices u and v in G. Let $d(G)$ be the diameter of G. Let l be the vertex connectivity of G and m the edge-connectivity. Then the sequences A, B, D and T are defined as follows:*

$$\begin{aligned}
a_i &= \max_{|V_i|=i}\{d_{G-V_i}(u,v) - d(u,v) \mid u,v \in V - V_i\} && \text{for } 1 \le i \le l-1 \\
b_i &= \max_{|E_i|=i}\{d_{G-E_i}(u,v) - d(u,v)\} && \text{for } 1 \le i \le m-1 \\
d_i &= \max_{|V_i|=i}\{d(G-V_i)\} && \text{for } 1 \le i \le l-1 \\
t_i &= \max_{|E_i|=i}\{d(G-E_i)\} && \text{for } 1 \le i \le m-1 \ .
\end{aligned}$$

Sequence A is called the *vertex-deleted incremental distance sequence*, B the *edge-deleted incremental distance sequence*, D the *vertex-deleted diameter sequence* and T the *edge-deleted diameter sequence*.

Entry i in sequence A is the maximum increase of the distance between a pair of vertices caused by the deletion of i vertices from G. The sequence B contains the maximum increase in distance for the deletion of edges. Entry i in sequence D is the maximum diameter of the graph caused by deleting i vertices, and sequence T is the analogous sequence for the deletion of edges. Table 15.2 contains the four sequences for the network shown in Figure 15.4.

Table 15.2. The vertex- and edge-deletion-sequences for the network of Figure 15.4

A	(1,2)
B	(3,3)
D	(3,4)
T	(4,4)

It is easy to see that the A, B and T sequences are always monotonically nondecreasing. The entries of the A sequence are non-negative and the entries in the B sequence at least 1. If G is complete the four sequences are as follows:

– $A = (0, \ldots, 0)$

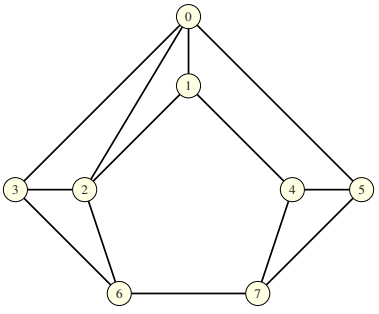


Fig. 15.4. Example graph for incremental distance sequences

- $B = (1, \ldots, 1)$
- $D = (1, \ldots, 1)$
- $T = (2, \ldots, 2)$

Krishnamoorthy, Thulasiraman and Swamy show that the largest increase in the distance between any pair of vertices caused by the deletion of i vertices or edges can always be found among the neighbors of the deleted objects. This speeds up the computation of the sequences significantly and also simplifies the definitions of A and B. These sequences can also be defined as follows (note that $N(V_i)$ is the set of vertices adjacent to vertices in the set V_i and $N(E_i)$ is the set of vertices incident to edges in E_i):

$$a_i = \max_{|V_i|=i} \{d_{G-V_i}(u,v) - d(u,v) \mid u, v \in N(V_i)\} \text{ for } 1 \leq i \leq l-1$$
$$b_i = \max_{|E_i|=i} \{d_{G-E_i}(u,v) - d(u,v) \mid u, v \in N(E_i)\} \text{ for } 1 \leq i \leq m-1$$

The vertex- and edge-deletion sequences are a worst case measure for the increase in distance caused by the failure of vertices or edges and they do not make any statements about the state of the graph after disconnection occurred. So these measures are only suited for applications where distance is crucial and disconnection makes the whole network unusable. Even with the improvement mentioned above, computing the sequences is still only possible for graphs with low connectivity.

15.3 Average Robustness Statistics

The statistics in this section make statements about the average number of vertices or edges that have to fail in order for the network to have a certain property or build an average of local properties in order to cover global aspects of the network.

15.3.1 Mean Connectivity

All of the measures introduced so far are worst-case measures. The *mean connectivity* introduced by Tainiter [538, 539] tries to make statements about the probability that a network is disconnected by the random deletion of edges.

Definition 15.3.1. *Let $G = (V, E)$ be a connected network with n vertices and m edges. Let $S(G)$ be the set of all $m!$ orderings of the edges and $G_0 = (V, \emptyset)$. For each ordering $s \in S(G)$ we define the number $\xi(s)$ as follows: We insert the edges of G into G_0 in the sequence given by s. We define $\xi(s)$ as the index of the edge that transforms the network from disconnected to connected. The mean connectivity of G is then defined as follows:*

$$\mathcal{M}(G) = m - \frac{1}{m!} \sum_{s \in S(G)} \xi(s)$$

Figure 15.5 shows a graph with mean connectivity 3/4. This can be seen as follows: For every edge-sequence where the edge $(2, 3)$ does not come last, we have $\xi(s) = 3$. For all other sequences, we have $\xi(s) = 4$. Since there are six sequences where edge $(2, 3)$ is last and 24 sequences in total, the mean connectivity of the graph is 3/4.

Note that $\mathcal{M}(G)$ is not the same as the mean number of edges we have to delete to disconnect G. If we look at all sequences of deleting edges and compute the mean index where the graph becomes disconnected, we obtain the value 7/4 for the graph in Figure 15.5.

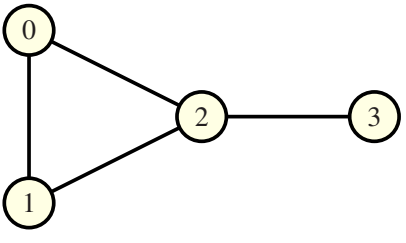


Fig. 15.5. A graph with mean connectivity 3/4

Tainiter has shown the following properties of this measure:

- If $G = (V, E')$ with $E' \subseteq E$ is a connected sub-network of $G = (V, E)$ then $\mathcal{M}(G') \leq \mathcal{M}(G)$
- Let G be a network with n vertices and m edges. We construct a new network G' by adding one new vertex and h edges that connect it to vertices in G. Let $\mathcal{M}(G, k)$ be the number of edge-sequences for G with $\xi(s) = k$. Then the following inequality is satisfied:

$$\mathcal{M}(G') - \mathcal{M}(G) \geq \frac{\mathcal{M}(G) + 1}{m + 1} - \frac{1}{h + 1} \sum_{k=n-1}^{m} \mathcal{M}(G, k) \frac{(h + m - k + 1)!}{(m - k)!(m + h)!}$$

– The following bounds are tight:

$$\lambda(G) - 1 \leq \mathcal{M}(G) \leq m - n + 1$$

where $\lambda(G)$ is the edge-connectivity of G. An example where both bounds are tight is a circle where we have $\lambda(G) = 2$ and $\mathcal{M}(G) = 1$.

If the difference between the mean connectivity and the classical edge-connectivity is large, then there must be connectivity bottlenecks in the network. It follows that the connectivity of the network can be strengthened by inserting only a few edges to bridge the bottleneck. An example would be a complete graph with one 'dangling' vertex connected to the rest of the graph by a single edge. With each edge we add to the dangling vertex, we can increase the connectivity of the graph by one. The principal drawback of the measure is again the fact that there is no efficient algorithm known for computing it. Also, it is useful only in the case of random edge failures.

15.3.2 Average Connected Distance and Fragmentation

In 1999, the article [17] received a lot of attention in the scientific world. Albert, Jeong, and Barabási simulate random vertex failures and intentional attacks at the highest-degree vertices in random and scale-free networks. They measure the effects on two parameters of the network, namely on the *average connected distance* and on the *fragmentation*.

The average connected distance $\bar{d}$ is the average length of the shortest paths between connected pairs of nodes in the network as defined in Section 11.2.[1]

Fragmentation measures the decay of a network in terms of the size of its connected components.

Definition 15.3.2 (Fragmentation). *Let G be a network with k connected components $S_1, \ldots, S_k$. The* fragmentation $\mathrm{frag}(G) = (\mathrm{frag}_1(G), \mathrm{frag}_2(G))$ *is defined by two parameters: The relative size of the largest component*

$$\mathrm{frag}_1 = \frac{\max_{i=1}^{k} |S_k|}{\sum_{i=1}^{k} |S_k|}$$

and the average size of an isolated component

$$\mathrm{frag}_2 = \frac{\sum_{i=1}^{k} |S_k| - \max_{i=1}^{k} |S_k|}{k-1} \ ,$$

where $|S_k|$ denotes the number of vertices in the kth component.

[1] In [17], the authors use the term *interconnectedness* which corresponds to the classical average distance. In their experiments, however, they measure the average connected distance. The classical average distance becomes ∞ as soon as the graph becomes disconnected.

Figure 15.6 shows the effect of vertex failures and attacks on the average connected distance $\bar{d}$ for randomly generated networks whose degree distributions follow a Poisson distribution and a power-law distribution, respectively. The Poisson networks suffer equally from random and targeted failures. Every vertex plays more or less the same role, and deleting one of them affects the average connected distance, on average, only slightly if at all. The scale-free network, in contrast, is very robust to failures in terms of average connected distance. The probability that a high-degree vertex is deleted is quite small and since those vertices are responsible for the short average distance in scale-free networks, the distances almost do not increase at all when deleting vertices randomly. If, however, those vertices are the aim of an attack, the average connected distance increases quickly. Simulations on small fragments of the Internet router graph and the WWW graph show a similar behavior as the random scale-free network, see [17].

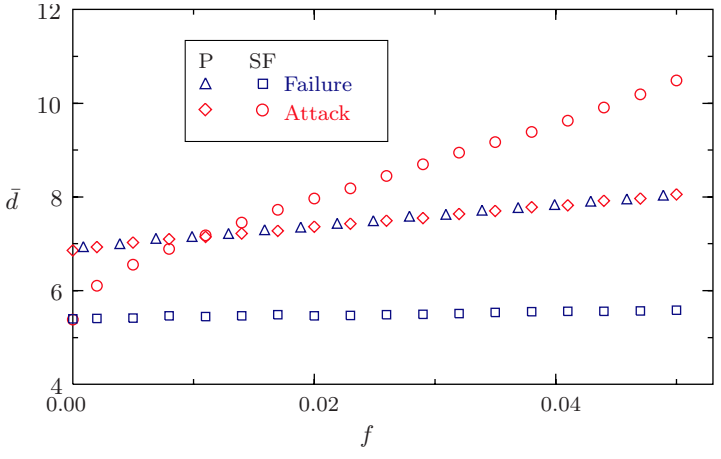


Fig. 15.6. Changes in average connected distance $\bar{d}$ of randomly generated networks ($|V| = 10,000, |E| = 20,000$) with Poisson (P) and scale-free (SF) degree distribution after randomly removing $f|V|$ vertices (source: [17])

The increase in average connected distance alone does not say much about the connectivity status of the network in terms of fragmentation. It is possible to create networks with small average connected distance that consist of many disconnected components (imagine a large number of disconnected triangles: their average connected distance is 1). Therefore, Albert et al. also measure the fragmentation process under failure and attack.

Figure 15.7 shows the results of the experimental study on fragmentation. The Poisson network shows a threshold-like behavior for $f > f_c \approx 0.28$ when frag_1, the relative size of the largest component, becomes almost zero. Together with the behavior of frag_2, the average size of the disconnected components, that reaches a peak of 2 at this point, this indicates the breakdown scenario as shown

also in Figure 15.8: Removing few vertices disconnects only single vertices. The components become larger as f reaches the percolation threshold f_c. After that, the system falls apart. As in Figure 15.6, the results are the same for random and targeted failures in networks with Poisson degree distribution.

The process looks different for scale-free networks (again, the data for the router and WWW graphs look similar as for the randomly generated scale-free networks). For random deletion of vertices no percolation threshold can be observed: the system shows a behavior known as *graceful degradation*. In case of attacks, we see the same breakdown scenario as for the Poisson network, with an earlier percolation threshold $f_c \approx 0.18$.

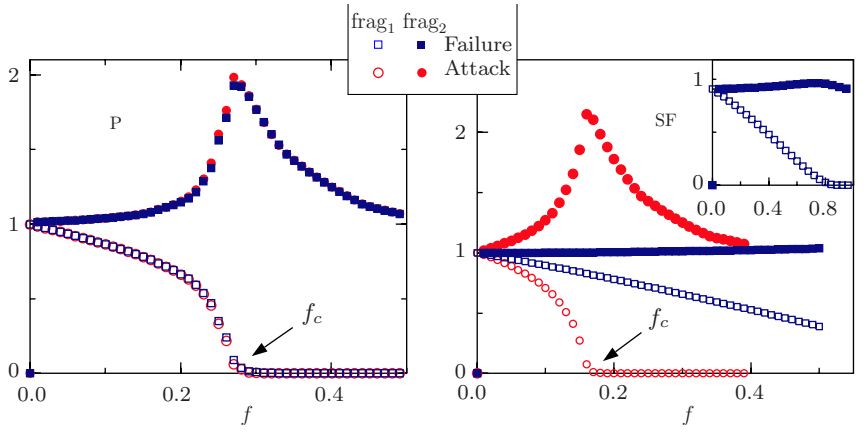


Fig. 15.7. Changes in fragmentation frag $= (\text{frag}_1, \text{frag}_2)$ of random networks (Poisson degree distribution: P, scale-free degree distribution: SF) after randomly removing $f|V|$ vertices. The inset in the upper right corner shows the scenario for the full range of deletions in scale-free networks (source: [17])

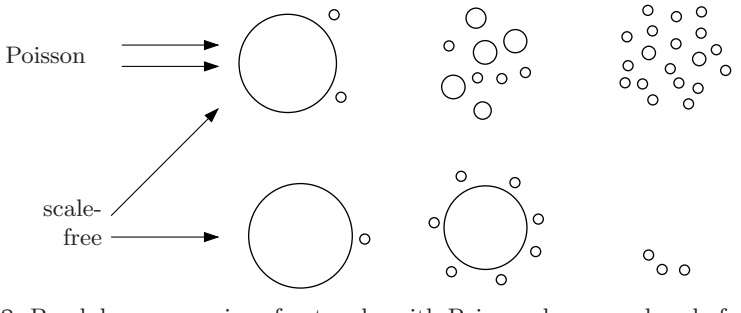


Fig. 15.8. Breakdown scenarios of networks with Poisson degree and scale-free distribution (source: [17])

In summary the experimental study shows that scale-free networks are tolerant against random failures but highly sensitive to targeted attacks. Since the Internet is believed to have a scale-free structure, the findings confirm the vulnerability of this network which is often paraphrased as the 'Achilles heel of the Internet'.

Broder et al. study the structure of the web more thoroughly and come to the conclusion that the web has a 'bow tie structure' as depicted in Figure 4.1 on page 77 in Chapter 3 [102]. Their experimental results on the web graph W reveal that the world wide web is robust against attacks. Deleting all vertices $\{v \in V(W) \mid d^-(v) \geq 5\}$ does not decrease the size of the largest component dramatically, it still contains approximately 30% of the vertices. This apparent contradiction to the results of Albert et al. can be explained by the fact that

$$\frac{|\{v \in V(W) \mid d^-(v) \geq 5\}|}{|V(W)|}$$

is still below the percolation threshold and is thus just another way to look at the same data: while 'deleting all vertices with high degree' sounds drastic this is still a set of small cardinality.

A number of application-oriented papers use the average connected distance and fragmentation as the measures of choice in order to show the robustness properties of the corresponding network. For example, Jeong et al. study the protein interaction network of the yeast proteome (*S. cervisiae*) and show that it is robust against random mutations of proteins but susceptible to the destruction of the highest degree proteins [327]. Using average connected distance and fragmentation to study epidemic propagation networks leads to the advice to take care of the hubs first, when it comes to deciding a vaccination strategy (see, e.g., [469]).

Holme et al. [305] study slightly more complex attacks on networks. Besides attacks on vertices they also consider deleting edges and choose betweenness centrality as an alternative selection criterion for deletion. In addition, they investigate in how far recalculating the selection criteria after each deletion alters the results. They show empirically that attacks based on recalculated values are more effective.

On the theoretical side Cohen et al. [130] and, independently, Callaway et al. [108] study the fragmentation process on scale-free networks analytically. While the first team of authors uses percolation theory, Callaway and his colleagues obtain more general results for arbitrary degree distributions using generating functions (see Section 13.2.2 in Chapter 13). The theoretical analyses confirm the results of the empirical studies and yield the same percolation thresholds as shown in the figures above.

15.3.3 Balanced-Cut Resilience

Among other statistics, Tangmunarunkit et al. use a new measure of robustness to link failures in their experimental study [541]. The aim of their experiments is

to evaluate generators that supposedly simulate the Internet topology. Besides *expansion* and *distortion* (see Chapter 11), the authors measure the similarity of generated and real networks with respect to the size of a balanced cut through the network. In terms of the new statistics, a network is resilient to component failure if the average size of a balanced cut within an h-neighborhood around each vertex is large. We give a more formal definition:

Definition 15.3.3 (Balanced-cut resilience). *Let $G = (V, E)$ be a network with n vertices, and let the capacity of each edge in G be equal to one. The* minimum balanced cut *of G is the capacity of a minimum cut such that the two resulting vertex sets contain approximately the same number, namely $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$, of vertices. The balanced-cut resilience $R(N(v,h))$ is the average size of a minimum balanced cut within the h-neighborhood $\mathrm{Neigh}_h(v)$ around each vertex v, that is,*

$$R(N(v,h)) = \frac{1}{n} \left(\sum_{v \in V} \textit{min. balanced cut in } \mathrm{Neigh}_h(v) \right) \quad .$$

The h-neighborhood of a vertex v contains all vertices with distance less than or equal to h from v, see also the definition on page 296 in Chapter 11. The balanced-cut resilience is a function of the number of nodes $N(v,h)$ in the h-neighborhood of a vertex v, not the radius h itself, to factor out the fact that networks with high expansion have more nodes in neighborhoods of the same radius. Clearly, we have $R(h) = 1$ for paths and trees. The resilience of random graphs in the Erdős-Rényi model with average degree k is proportional to kn, whereas it is proportional to n for complete graphs, see [541]. For regular grid graphs, the balanced-cut resilience grows with $\sqrt{n}$. See Figure 15.9 for an illustrative example.

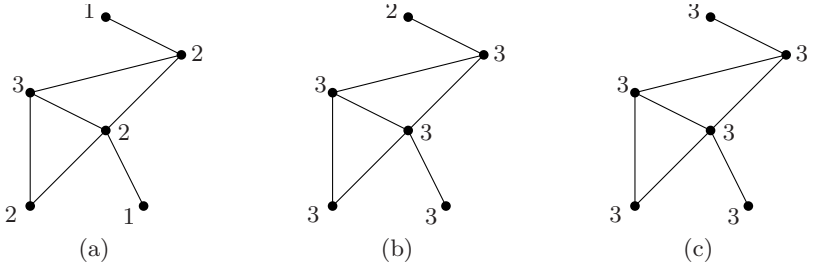


Fig. 15.9. Balanced-cut resilience for an example graph. Balanced cut shown for each vertex for (a) 1-neighborhoods, (b) 2-neighborhoods, and (c) 3-neighborhoods

Computing a minimum balanced cut is $\mathcal{NP}$-hard [240] and thus the drawback of this statistics is certainly its computational complexity which makes it impractical for large networks. There are, however, a number of heuristics that yield reasonably good values so that the balanced-cut resilience can at least be

estimated. Karypis and Kumar [348], for instance, propose a multilevel partitioning heuristics that runs in time $\mathcal{O}(m)$ where m is the number of edges in the network.

15.3.4 Effective Diameter

Palmer et al. introduce in [462] the *effective eccentricity* and the *effective diameter* as measures of resilience against vertex and edge failures. These statistics are based on the hop-plot and we recall their definitions (see also Sections 11.2.4 and 11.2.3 on neighborhoods and eccentricity in Chapter 11):

Definition 15.3.4 (Effective eccentricity, effective diameter). *The* effective eccentricity $\varepsilon_{\text{eff}}(v, r)$, $0 \leq r \leq 1$, *of a vertex v is the smallest h such that the number of vertices $N(v, h)$ within a h-neighborhood of v is at least r times the total number of vertices, that is,*

$$\varepsilon_{\text{eff}}(v, r) = \min\{h \in \mathbb{N} \mid N(v, h) \geq rn\} \ .$$

The effective diameter $\text{diam}_{\text{eff}}(r)$ *of a network is the smallest h such that the number of pairs within a h-neighborhood is at least r times the total number of reachable pairs:*

$$\text{diam}_{\text{eff}}(r) = \min\{h \in \mathbb{N} \mid P(h) \geq rP(\infty)\} \ ,$$

where P denotes the number of pairs within a certain neighborhood (hop-plot), that is,

$$P(h) := \left|\left\{(u, v) \in V^2 \mid d(u, v) \leq h\right\}\right| = \sum_{v \in V} N(v, h) \ ,$$

see also Chapter 11. In the case that this distribution follows the power law $P(h) = (n + 2m)h^{\mathcal{H}}$, the value $\mathcal{H}$ is also referred to as the hop-plot exponent.

The authors perform experiments on the network of approximately 285,000 routers in the Internet to investigate in how far and under which circumstances the effective diameter of the router network changes. The experiments consist of deleting either edges or vertices of the network and recomputing the effective diameter diam_{eff} after each deletion, using a value of 0.9 for the parameter r. Since an exact calculation of this statistics would take days, they exploit the approximate neighborhood function described in Section 11.2.6 of Chapter 11. Using these estimated values leads to a speed-up factor of 400.

Figures 15.10 and 15.11 show the effect of link and router failures on the Internet graph. Confirming previous studies, the plots show that the Internet is very robust against random failures but highly sensitive to failure of high degree vertices. Also, deleting vertices with low effective eccentricity first rapidly decreases the connectivity.

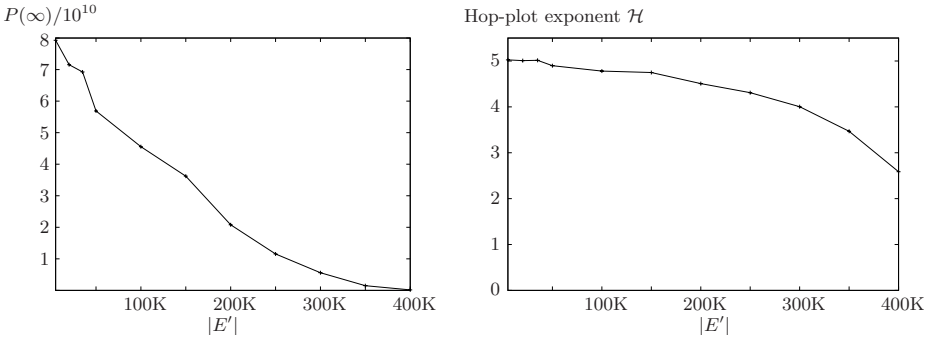


Fig. 15.10. Effect of edge deletions (link failures) on the network of 285,000 routers (source: [462]). The set E' denotes the deleted edges

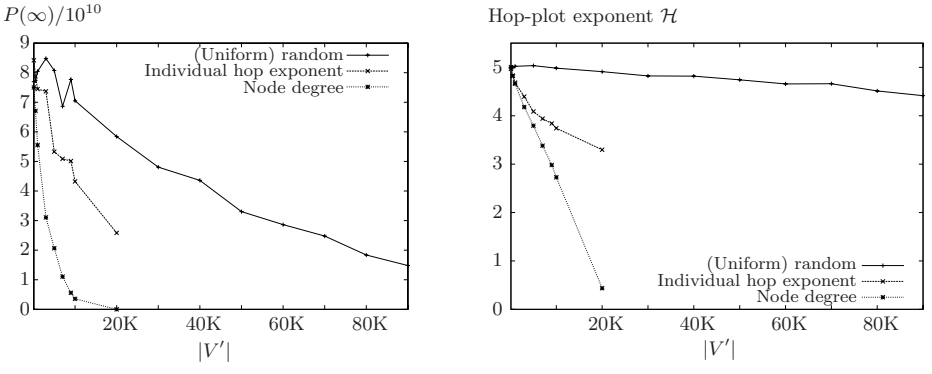


Fig. 15.11. Effect of vertex deletions (router failures) on the network of 285,000 routers (source: [462]). The set V' denotes the deleted edges

15.4 Probabilistic Robustness Statistics

This section describes robustness statistics that explicitly consider the failure probabilities of network components and are thus more appropriate to describe untargeted component failure. We present two different approaches to determine the probability of network disconnection given the failure probability: the *reliability polynomial* and *probabilistic resilience*.

We chose not to cover purely theoretical approaches such as the symbolic approach to robustness by Flajolet et al. [214], in which the authors define a measure of robustness by determining the expected number of edge-disjoint paths to get from a start vertex s to a target vertex t in a graph.

15.4.1 Reliability Polynomial

The *reliability polynomial* was already used in 1977 by Boorstyn and Frank [75].

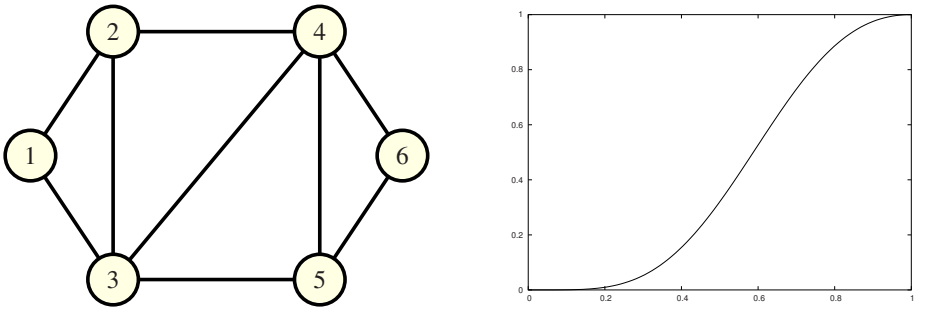
Definition 15.4.1. *Let G be a connected network with n vertices and m edges. We assume that the edges of G fail independently with probability $1-p$ where $0 \leq p \leq 1$. The reliability polynomial $R(G,p)$ is the probability that G is connected.*

Obvious properties of the reliability polynomial $R(G,p)$ are:

1. $R(G,0) = 0$, $R(G,1) = 1$.
2. $p_1 < p_2$ implies $R(G,p_1) < R(G,p_2)$.
3. Let G be a connected graph and G_{-e} be the graph obtained from G by removing e. Let G_e be the graph obtained from G by contracting e. Then the following equality holds:
$$R(G,p) = (1-p)R(G_{-e},p) + pR(G_e,p) \ .$$
4. If G is a tree with m edges, than we have $R(G,p) = p^m$.

In his doctoral thesis [497], Rosenthal showed that it is $\mathcal{NP}$-hard to decide for a given edge failure probability if the probability that the network is connected is at least a certain value q. The same is true if we are given a failure probability for vertices and edges. In [480], Pönitz and Tittmann have shown that the problem can be solved in time $\mathcal{O}((2n+m)B(k))$ for graphs with pathwidth k where $B(k)$ is the Bell number of k. The bell number of k is the number of ways the set of natural numbers from 1 to k can be partitioned into nonempty subsets. It follows that the problem is polynomially solvable for graphs with bounded pathwidth. Figure 15.12 shows a graph with pathwidth two from [480] together with a plot of its reliability polynomial. The polynomial has the following formula:

$$R(G,p) = 55p^5 - 155p^6 + 169p^7 - 84p^8 + 16p^9$$



(a) Graph with pathwidth two (b) The reliability polynomial

Fig. 15.12. A graph and a plot of its reliability polynomial

There is no polynomial time algorithm known to compute the reliability polynomial for general graphs.

15.4.2 Probabilistic Resilience

In contrast to the deterministic probability measures presented in Section 15.1 on worst-case connectivity statistics, Najjar and Gaudiot study a probabilistic variant of connectivity [438]. The authors consider a class of regular networks and examine the probability of disconnection through random vertex failures.

They define the *disconnection probability* of a network G as

$$P(G, i) = \Pr[G \text{ disconnected exactly after } i\text{th failure}]$$

Motivated by the architectures of large-scale computer clusters the authors study a family $\mathcal{F}$ of k-regular graphs that includes, for example, tori and hypercubes. They show that for networks in $\mathcal{F}$ the disconnection probability $P(G, i)$ can be approximated by the term

$$P_1(G, i) = \Pr[G \text{ disconnected exactly after } i\text{th failure and one component contains exactly one vertex}] \ ,$$

that is, the disconnection probability can be estimated by the probability of disconnecting only one vertex from the network. For networks in the family $\mathcal{F}$, $P_1(G, i)$ and thus an estimation of $P(G, i)$ can be derived analytically.

The function $P(G, i)$ is a bell-shaped curve whose height increases with n, the number of vertices in the network, whereas the x-coordinate of the maximum depends on k, the degree of the vertices (see Figure 15.13). The larger the connectivity of a regular network in terms of k the more failures are needed until disconnection occurs. The authors confirm their theoretical predictions by running Monte-Carlo experiments on a large number of graphs from $\mathcal{F}$.

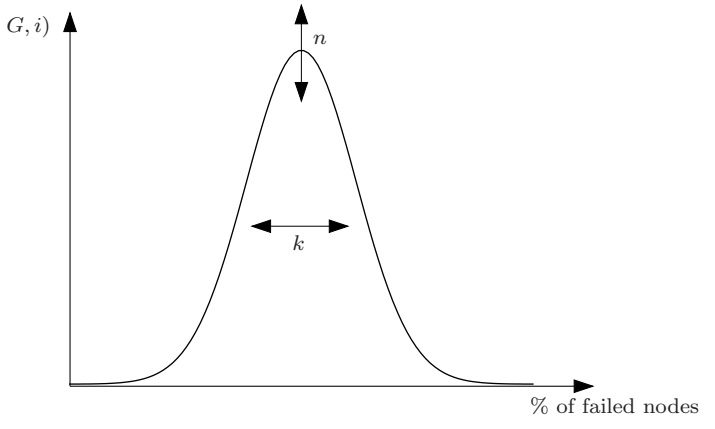


Fig. 15.13. The probability $P(G, i)$ for members of $\mathcal{F}$. The number of vertices in the network, n, determines the height of the curve. Their vertex degree, k, determines the offset on the abscissa

The concept of disconnection probability enables us to define a probabilistic version of connectivity: *probabilistic resilience*. Intuitively, a resilient network should sustain a large number of vertex failures until it becomes disconnected.

Definition 15.4.2 (Probabilistic resilience). *Let G be a network with n vertices. The* probabilistic resilience[2] $\mathrm{res}_{prob}(G,p)$ *is the largest number of vertex failures such that G is still connected with probability $1-p$, that is,*

$$\mathrm{res}_{prob}(G,p) = \max\{I \mid \sum_{i=1}^{I} P(G,i) \leq p\} \ .$$

The relative probabilistic resilience *relates* $\mathrm{res}_{prob}(G,p)$ *to the size of G:*

$$\overline{\mathrm{res}}_{prob}(G,p) = \frac{\mathrm{res}_{prob}(G,p)}{n} \ .$$

Clearly, this probabilistic measure is related to classical connectivity, and the identity $\mathrm{res}_{\mathrm{prob}}(G,0) = \kappa(G) - 1$ holds.

Analyzing $P(G,i)$ for regular graphs shows that the probabilistic resilience $\mathrm{res}_{\mathrm{prob}}(G,p)$ grows with the size of G. The relative probabilistic resilience $\overline{\mathrm{res}}_{\mathrm{prob}}(G,p)$, however, decreases with the size if the degree of the network remains constant. Therefore, the relative resilience increases for hypercubes and decreases for tori with increasing network size.

It is quite difficult to compute the probabilistic resilience for more complicated families of networks than $\mathcal{F}$. Even in this case, $P(G,i)$ can only be estimated. Nevertheless, the probabilistic variant of connectedness seems well-suited to describe system degradation under random component failure. Due to its analytical complexity, however, it will most likely be used only in empirical evaluations.

15.5 Chapter Notes

Many different statistics have been studied in order to describe how networks change under component failures or intentional attacks. In this chapter we have given an overview of analyses and experimental results that aim at describing robustness and resilience properties of complex networks.

We first looked at worst case connectivity statistics that implicitly assume optimal attacks. Apart from classical connectivity, we also considered cohesiveness, the minimum m-degree, toughness and conditional connectivity. Only the first two measures can be computed in polynomial time. For a fixed parameter m, the minimum m-degree is also computable in polynomial time. Toughness is known to be $\mathcal{NP}$-hard and the complexity of conditional connectivity depends on the chosen property.

In an application, the function of a network might not only depend on its connectivity, but also on the length of the shortest paths. In Section 15.2, we

[2] In the original paper [438], Najjar and Gaudiot use the term *network resilience*.

looked at two worst case distance statistics, namely the persistence and incremental distance sequences. The second concept is more general than the first but for neither of them a polynomial time algorithm is known.

The main drawback of all the worst case statistics is that they make no statements about the results of random edge- or vertex-failures. Therefore, we looked at average robustness statistics in Section 15.3. The two statistics in this section for which no polynomial algorithm is known (mean connectivity and balanced-cut resilience) make statements about the network when edges fail while the two other statistics (average distance/fragmentation and effective diameter) only characterize the current state of a network. Hence, they are useful to measure robustness properties of a network only if they are repeatedly evaluated after successive edge deletions—either in an experiment or analytically.

In Section 15.4, we presented two statistics that give the probability that the network under consideration is still connected after the random failure of edges or vertices. The reliability polynomial gives the probability that the graph is connected given a failure probability for the edges while the probabilistic resilience for a network and a number i is the probability that the network disconnects after exactly i failures. There is no polynomial time algorithm known to compute any of these two statistics for general graphs.

The ideal statistics for describing the robustness of a complex network depend on the application and the type of the failures that are expected. If a network ceases to be useful after it is disconnected, statistics that describe the connectivity of the graph are best suited. If distances between vertices must be small, diameter-based statistics are preferable.

For random failures, the average and probabilistic statistics are the most promising while the effects of deliberate attacks are best captured by worst case statistics. So the ideal measure for deliberate attacks seems to be generalized connectivity but this has the drawback that it is hard to compute. A probabilistic version of generalized connectivity would be ideal for random failures.

In practice, an experimental approach to robustness seems to be most useful. The simultaneous observation of changes in average connected distance and fragmentation is suitable in many cases. One of the central results regarding robustness is certainly that scale-free networks are on the one hand tolerant against random failure but on the other hand exposed to intentional attacks.

Robustness is already a very complex topic but there are still many features of real-world networks that we have not touched in this chapter. Examples include the bandwidth of edges or the importance of vertices in an application as well as routing protocols and delay on edges.

Another interesting area are networks where the failures of elements are not independent of each other. In power networks for example, the failure of a power line puts more stress on other lines and thus makes their failure more likely, which might cause a domino effect.

At the moment, there are no deterministic polynomial algorithms that can answer meaningful questions about the robustness of complex real-world net-

works. If there are no major theoretic breakthroughs the most useful tools in this field will be simulations and heuristics.

Acknowledgments. The authors thank the editors, the co-authors of this book, and the anonymous referee for valuable comments.