PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 12-01

Bounded Monte Carlo Estimation of Diameter-constrained Network Reliability

Héctor Cancela Franco Robledo Gerardo Rubino
Pablo Sartor

2012

Bounded Monte Carlo estimation of Diameter-constrained network reliability
Cancela, Héctor; Robledo, Franco; Rubino, Gerardo; Sartor, Pablo
ISSN 0797-6410
Reporte Técnico RT 12-01
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay, febrero de 2012

Bounded Monte Carlo Estimation of Diameter-constrained Network Reliability

Héctor Cancela[a], Franco Robledo[a], Gerardo Rubino[b], Pablo Sartor[a],

[a] *Instituto de Computación, Facultad de Ingeniería, Universidad de la República. Julio Herrera y Reissig 565, Montevideo, 11300, Uruguay.*
[b] *INRIA Rennes – Bretagne Atlantique, Campus Beaulieu, 35042 Rennes Cedex, France.*

Abstract

The d-diameter-constrained K-reliability (DCR) problem in networks is an extension of the classical problem of computing the K-reliability (CLR) where the subnetwork resulting from the failure of some edges is operational if and only if all nodes in a set of "terminal nodes" K have pairwise distances not greater than a certain integer d. Computing the CLR is NP-hard which has motivated the development of simulation schemes, among which a family of Monte Carlo sampling plans that make use of upper and lower bounds to reduce the variance attained after drawing a given number of samples. The DCR is receiving increasing attention in contexts like video-conferencing and peer-to-peer networks; since it is an extension of the CLR it is also NP-hard. This paper presents Monte Carlo sampling plans based on bounds adapted to the DCR. These plans are described in detail focusing on their requirements and limitations. Test cases are presented evidencing how the diameter constraint and the terminal nodes set size affect the efficiency as well as the higher performance improvements attained by the best-performing methods in the context of DCR when compared to CLR.

Keywords: Monte Carlo, Rare Events, Variance Reduction, Network Reliability, Diameter Constraints.

1. Introduction

In several contexts a communication network can be modelled as an undirected graph $G = (V, E)$ where the nodes V represent entities to interconnect and the edges E represent the existing links among them. Random failures affecting nodes and edges can be modelled through different probabilistic schemes. The most employed one assumes that each component w has a certain probability p_w of being operational (with probability $1 - p_w$ of being failed) and the network has also one of two possible states (operational or failed) depending on the states of all components. Component's behaviours are also typically assumed to be independent of each other. One possible way to define the network operational state is by specifying a subset of distinguished nodes K (named "terminals") and requiring that they belong to the same connected component after discarding the failing edges (we will work with models where nodes never fail). The network's state is then a binary random variable and the classical K-reliability problem (CLR) deals with computing the probability $R(G, K)$ that the network is operational according to this definition; there is a vast literature on it. Petingi and Rodriguez [1] introduced the problem of computing the d-diameter-constrained K-reliability (DCR) by adding the additional requirement for the network to be operational that the distance among all pairs of nodes of K be at most the integer d after discarding all failing edges. For the DCR we denote the probability that the network is operational as $R(G, K, d)$. It has received increasing attention during recent years because it models situations where limits exist on the acceptable delay times to propagate traffic (like in voice applications over IP networks) or in the amount of hops that packets can undergo (peer-to-peer networks) and previous results for CLR were extended to DCR [2], [3], [4], [5]. The DCR is a superset of CLR since any instance of the latter can be transformed into an instance of the former by adding a constraint $d \geq |E| - 1$. Both problems are known to be NP-hard [6] [7] in the general cases and so exact computation is only feasible for limited-size or particular topologies. This has motivated intense research on developing bounds and simulation techniques to estimate the reliability with precision goals set by each specific application. In [8] and [9] a survey on the CLR is presented as well as in [10] for the DCR. A survey on simulation schemes that have been devised for estimating $R(G, K)$ can be found in [11]. In real applications it is often the case that the edge reliabilities are very high and so sampling a network in the failed state is a rare event. In such a situation it is highly desirable to lower the amount of samples needed to reach certain confidence level/interval goals by using variance reduction techniques. Van Slyke and Frank [12] and Kumamoto, Tanaka and Inoue [13] suggest sampling plans that make use of previous knowledge of the topology of the problem under study to bound the sampling space attaining variance reductions. Fishman includes in [14] those previous works under a general framework of bounded Monte Carlo sampling plans and in [15] compares them to other sampling plans for $K = \{s, t\}$ evidencing the advantages of this approach.

In this work we describe how the mentioned Monte Carlo sampling plans can be adapted to DCR context by focusing on the differences introduced respect to CLR. Section 2 introduces the needed definitions and notation. Section 3 presents the general framework and discusses the consequences of switching to DCR context. Section 4 briefly describes each sampling plan. Section 5 presents numerical test results and compares the performance of the plans among themselves and respect to the CLR. Finally Section 6 summarises conclusions.

2. Definitions and Notation

An instance of the CLR is defined by an undirected graph $G = (V, E)$ with $n = |V|, m = |E|, E = \{e_1 ... e_m\}$, a set of terminal nodes $K \in V$ with $k = |K|$ and given probabilities of operation for each edge. Recall that edges states are assumed to be independent of each other. The addition of a diameter constraint d defines an instance of the DCR which we denote by d-DCR. For each $e \in E$, let X_e be a random binary variable whose value is 1 if e operates and 0 otherwise and let p_e be the reliability of e (the probability that it is operational). We call *network configuration* or *state vector* any m-tuple $X = (X_1 ... X_m) \in \{0, 1\}^m$ encoding the states of all edges. Let $\mathcal{X}$ be the set of the 2^m possible configurations and denote by $\pi(x)$ the probability that the random configuration is x, that is, $\pi(x) = \Pr(X = x)$. The structure function maps each configuration $x \in \mathcal{X}$ into 1 or 0 according to the fact that the network operates or fails when the components are in the states encoded in x. As a consequence, $\Phi(X)$ is the random network's state and $R(G, K, d) = \Pr(\Phi(X) = 1)$.

3. Bounded Monte Carlo simulation

The d-DCR for a graph G with terminal set K can be computed as $R = R(G, K, d) = \sum_{x \in \mathcal{X}} \Phi(x)\pi(x)$. By sampling s configurations $\{X^{(1)} ... X^{(s)}\}$ according to the probability distribution given by $\pi(x)$ (crude Monte Carlo sampling plan) we can build an unbiased estimator $\bar{R}_s = \sum_{i=1...s} \Phi(X^{(i)})/s$ for R with variance $\sigma_s^2 = R(1 - R)/s$. Suppose we partition $\mathcal{X}$ into three pairwise disjoint sets U (set of configurations known to fail i.e. for which $\Phi = 0$), L (set of configurations known to be operational i.e. $\Phi = 1$) and Γ (set of configurations for which we do not know the state of the network), as shown in Figure 1. Then, defining $R_L = \pi(L) = \sum_{x \in L} \pi(x)$ and $R_U = 1 - \pi(U) = 1 - \sum_{x \in U} \pi(x)$ we have that $R_L \leq R \leq R_U$ (R_L and R_U are lower and upper bounds for the DCR) and that

$$\begin{aligned} R &= R_L + \Pr(X \in \Gamma \wedge \Phi(X) = 1) \\ &= R_L + (R_U - R_L) \Pr(\Phi(X) = 1 \,|\, X \in \Gamma). \end{aligned}$$

So, if R_L, R_U are known and if we can sample the components' states from the conditional probabilities $\Pr(X_e = 1 \mid X \in \Gamma)$, one can draw s configurations from Γ according to that conditional probability and build an unbiased estimator $\hat{R}_s = R_L + (R_U - R_L) \sum_{i=1..s} \Phi(X^{(i)})/s$ for R whose variance can be shown to be $(R_U - R)(R - R_L)/s$ and thus unbiasedly estimated as $\hat{\sigma}_s^2 = (R_U - \hat{R}_s)(\hat{R}_s - R_L)/(s - 1)$, achieving a variance reduction against the crude sampling plan if $R_L > 0$ or $R_U < 1$. This is presented in [14] together with bounds on the amount of samples needed to achieve desired confidence level and interval width goals for the absolute and relative errors. The methods here discussed differ in the way they define the sets U and L and the way to implement a configuration sampling coherent with $\Pr(\,. \mid X \in \Gamma)$. Unless otherwise specified we will assume that the edge states are independent. A generic pseudo-code for these simulations is shown in Figure 2. There are three basic steps for which we next discuss the differences introduced when working in DCR context respect to CLR.

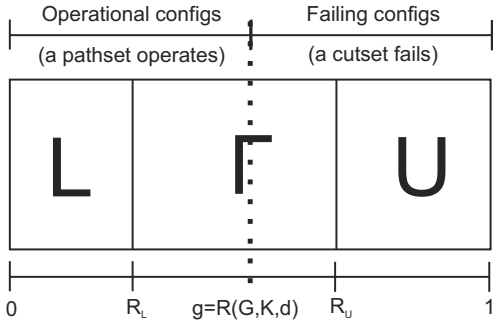

Figure 1: Bounded Monte Carlo sampling

Procedure Simulate(G, K, d, π, s)

1: Precomputing: $Prop \leftarrow$ (properties of the graph, subsets, auxiliary tables, etc.)
2: Bounding: $R_L, R_U \leftarrow$ compute-bounds$(G, K, d, \pi, Prop)$
3: $Z \leftarrow 0$
4: **for all** $iter \in \{1...s\}$ **do**
5: Sampling: $x \leftarrow$ sample-a-configuration$(\pi, Prop)$
6: Checking: if connected(x, K, d) then $Z \leftarrow Z + 1$
7: **end for**
8: $\hat{R} \leftarrow R_L + (R_U - R_L)Z/s$; $\hat{\sigma}^2 \leftarrow ((R_U - R_L)^2(1 - Z/s)Z/s)/(s - 1)$
9: **return** $\hat{R}, \hat{\sigma}^2$

Figure 2: Generic bounded Monte Carlo simulation pseudo-code

3.1. Gathering information about the topology of the network useful for bounding the configurations space

The following discussion applies to the steps "Precomputing" and "Bounding" of Figure 2. The most common topological structures used for computing, bounding and estimating network reliability are the concepts of *pathset* and *cutset*. A pathset is any set of edges such that if they all operate, the network operates, regardless the state of edges not in the pathset. A cutset is any set of edges such that if they all fail, the network fails as well, regardless the state of edges not in the cutset. These definitions are

valid both for CLR and DCR contexts. A pathset is said to fail when at least one of its edges fails (otherwise the pathset is "operational"); whereas a cutset is said to fail when all of its edges fail (otherwise it is "operational"). Both pathsets and cutsets are said to be minimal if there is no edge whose suppression still yields a pathset or cutset, respectively. Some sampling plans discussed here require determining the cardinalities M and N of the smallest cardinality pathsets and cutsets respectively. In the CLR context, determining M takes $O(n^{2/3}m)$ time and determining N takes $O(m)$ time for $k \in \{2, n\}$ and $O(kmn)$ in general. Other plans take advantage of knowing as many edge-disjoint pathsets and cutsets as possible. Building N pathsets and M cutsets (useful for the plans of Subsection 4.3) takes $O(Mm)$ and $O(m)$ times respectively (see [14] for the above results). All these are more complex tasks when working with diameter constraints. Golovach and Thilikos [16] summarise the complexity of the decision problems asking for the existence of edge-disjoint pathsets and cutsets with a given cardinality κ for a given constraint d; these problems are NP-hard when one or both of κ and d are considered as inputs and belong to the FTP complexity class[1] when both are parameters. Note that a certain edge set that is a pathset under diameter d is not necessarily a pathset under diameter $d' < d$; whilst all pathsets under diameter d' are also pathsets under diameter d. The opposite happens regarding cutsets. In other words, for the same topology, as the required diameter decreases there are less pathsets and more cutsets, which shifts the [lower bound, upper bound] intervals computed through pathsets and cutsets suggested in Subsection 4.3. This is reasonable since the DCR decreases monotonously when d decreases (the constraint being more demanding). We will assume that M, N and/or a set of pathsets and cutsets are given *a priori* though finding them can be a difficult task (particularly for the DCR as seen). It is useful to treat these precomputing tasks as problems separated from the simulation itself; once solved for a given topology and set K they can be used multiple times for estimating reliabilities with different values of $(p_e)_{e \in E}$, for instance to perform a sensitivity analysis, or for optimising purposes.

3.2. Sampling configurations

As discussed in Section 4 in this step a configuration must be randomly chosen from the configurations subspace yielded after dropping off the ones for which the network state is known due to the bounding method applied. The selection must be made with probabilities proportional to the ones in the original configurations space. As we will see, regarding the algorithms applied in this step in the sampling plans here discussed, no differences arise when moving from CLR to DCR context.

[1]The class of complexity of the parameterized problems taking time $O(f(p)g(i))$ where f is any function on the parameters p and g is any polynomial on the inputs i

3.3. Determining the network state for a sampled configuration

After sampling a configuration x it must be determined whether it corresponds to an operational or to a failed network state (by computing the number $\Phi(x)$). This can be done in $O(m)$ time for the CLR running a breadth-first-search (BFS) starting at any node of K. Under DCR the time required is in general $O(km)$ (BFS with depth bounded by d starting at each node of K). The values of d and k determine whether this step takes more or less time in DCR than in CLR. With low values of d each BFS might execute faster in DCR than in CLR since it reaches a depth of at most d. On the contrary high values of k will force to run several BFS in DCR instead of one in CLR thus increasing the checking time in DCR. The effect of both parameters is made evident in Section 5, where we show their impact on the relative efficiency (RE) of the methods against crude Monte Carlo attained in CLR and DCR.

4. Description of the methods

4.1. SIM1: Crude Monte Carlo

As seen above when applying crude Monte Carlo the sets U and L are empty (no information about failing or operational configurations is used when sampling) and thus $R_L = 0$, $R_U = 1$ and $\sigma_s^2 = R(1-R)/s$. The steps "Precomputing" and "Bounding" are trivial (nothing to compute). The step "Sampling" is $O(m)$ since it involves running m Bernoulli trials, one for each edge e with success probability $\Pr(X_e = 1)$ if failures are independent, or taking into account already sampled edges to compute $\Pr(X_e = 1 \mid$ "state of already sampled edges") if there are dependencies. The step "Checking" is identical for all methods and was discussed in Subsection 3.3.

4.2. SIM2: Bounds based on minimal cardinality pathsets and cutsets

Van Slyke and Frank [12] suggest the following way to build U and L. Let M be the cardinality of the pathset with the lowest cardinality (for a given instance G, K, d) and let N be the cardinality of the cutset with the lowest cardinality. It is clear that any configuration with less than M operational edges leads to a failing network. At the same time, any configuration with more than $m - N$ operational edges leads to an operational system. So we can define U as the set of all configurations having more than $m - N$ operational edges and L as the set of all configurations having less than M operational edges. The step "Precomputing" implies determining M and N (refer to Subsection 3.1) which can be more difficult for DCR than CLR; in large instances it can be easier to bound M and N by certain integers $M' \leq M$ and $N' \geq N$ though getting potentially less tight bounds. Determining R_L involves adding the occurrence probabilities of all configurations with less than M operational edges. Note that

when working with uniform probabilities p ($p_e = p$ for all edge e) any configuration with j operational states has an occurrence probability equal to $p^j(1-p)^{m-j}$. Then, defining $F_i(n,p) = \sum_{j=0..i} \binom{n}{j} p^j(1-p)^{n-j}$ we have that $R_L = F_{M-1}(m,p)$ and $R_U = 1 - F_{m-N}(m,p)$ computable in $O(M)$ and $O(m-C+1)$ times respectively. When probabilities are not uniform algorithms known in the literature as of the "k-out-of-n" type must be used [17], [18], [19]. Computing a table with all values for F can be seen as part of "Precomputing". Then, "Sampling" implies that a random configuration must be drawn with at least M but less than $m - N$ operational edges according to their probabilities of occurrence. To do so with uniform probabilities involves two steps. First, one determines the number Q of edges by cutting a table of cumulative $F_i(m,p)$ values through a uniformly distributed random cutpoint. Second, a configuration with such number of operational edges is randomly built by generating an uniformly distributed random integer in $[0, \binom{m}{Q})$ and "translating it" into a unique set of Q operational edges through a bijective function (in [14] the k-canonical representation of integers of Kruskal-Katona is suggested). Both can be executed in $O(m)$ time provided that tables with the values of F and precomputed needed combinations exist. Note that the absence or presence (and value) of a diameter constraint d has no explicit impact on the algorithm used for sampling edges, though it has an indirect impact through the values of M and N, since the minimum-cardinality pathsets and cutsets depend on the constraint d.

4.3. SIM3, SIM4, SIM5: Bounds based on precomputed pathsets and cutsets

Kumamoto, Tanaka and Inoue [13] suggested a method that takes advantage of more information about the topology of the problem instance than just knowing M and C, namely knowing a set of I pathsets $P = P_1...P_I$ and a set of J cutsets $C = C_1...C_J$; this method is presented under a more comprehensive framework in [14]. The set U (as defined in Section 4 together with L) will be exactly the set of all configurations where at least one of $C_1...C_J$ fails (thus making the network fail). The set L will be exactly the set of all configurations where at least one of $P_1...P_I$ is operational (thus guaranteing that the network is operational). So the set Γ from which samples must be drawn contains all configurations for which we do not have evidence of their state when only considering the information given by P and C.

Let us first address the general case where the elements of P are not required to be pairwise disjoint nor are the elements of C and call this plan SIM3. "Precomputing" in this context involves generating the sets P and C which (not having to be edge-disjoint) can be easily done through DFS algorithms for P (both for CLR and DCR) and drop-and-test algorithms for C (observe that we are not constrained to generate all pathsets or cutsets; this is just a matter of trade-off among precomputing effort and bounds tightness). "Bounding" (computing R_L and R_U) can be exponentially complex if the elements in P and/or C are numerous and highly overlap (though trivial if they are disjoint, by factoring probabilities). Regarding "Sampling" let Ω be the set of all edges in $P_1...P_I$ and $C_1...C_J$. Sampling edges in $E \setminus \Omega$ involves just $m - |\Omega|$ Bernoulli trials. Edges in Ω can be sequentially sampled in $O(|\Omega|(\sum_{h=1..I}|P_h| + \sum_{h=1..J}|C_h|)$. In [14] formulae are presented for general pathsets and cutsets (i.e. not necessarily disjoint) that are not valid for non-disjoint pathsets and cutsets. In Appendix A we concisely derive the valid ones both for non-disjoint and disjoint contexts. A variant presented in [14] that we call SIM4 implies precomputing the probability of the occurrence of each of the $2^{|\Omega|}$ possible sub-configurations that exist when only considering the edges of Ω, in $O(2^{|\Omega|})$ time. Then sampling involves just choosing any of them through a random cutpoint access on a table accumulating the precomputed probabilities (thus in $O(m)$ time). This variant is feasible only for limited-sized sets Ω and thus limited-sized problem instances (because the bounds loosen as m grows with a fixed $|\Omega|$). A third variant that we call SIM5 makes use only of pairwise disjoint pathsets $P_1...P_I$ and cutsets $C_1...C_J$. Disjointness allows an easy factorized computation of R_L and R_U in $O(\sum_{h=1..I}|P_h|)$ and $O(\sum_{h=1..J}|C_h|)$ steps respectively, and sampling of all edges in $O(m)$; the algorithm, presented in [14] with mistyped formulae as noted in [20], is included together with its proof in Appendix A. It is important to note that the previous discussions about SIM3, SIM4 and SIM5 are valid for non-uniform probabilities. Again, note that for SIM3, SIM4 and SIM5 the absence or presence (and value) of a diameter constraint d has no explicit impact on the sequential computation of the probabilities for sampling the edges of Ω. There is an indirect impact when building the sets P and C (their elements are pathsets and cutsets for a given value of d). Once in the sampling phase their elements intrinsecally carry the constraint within their topology but the sampling algorithm keeps identical (and independent of d) in CLR and DCR.

4.4. Comparison of sampling plans

Table 1 compares the requirements on uniformity of probabilities, previous-to-simulation knowledge needed and times order for precomputing, bounding and sampling; in the table $\exp(P,C)$ means "potentially exponential in the edges of P and C". As the tests show there are significant differences in time even in cases with identical order.

5. Experimental results

5.1. Test instances

In order to study the performance of the methods presented in the previous sections, we provide computational results obtained over three test cases. Tests I and II are based on the "dodecaedron" network, a common topology in reliability literature, shown in Figure 3. Test I compares

.	SIM1	SIM2	SIM3	SIM4	SIM5
$p_{e\in E}$	Any	Uniform	Any	Any	Any
Pre-knowledge	-	min.sizes M, N	P, C general	P, C general	P, C disjoint
Precomputing	-	m (for table F)	-	$2^{\lvert\Omega\rvert}$	-
Bounding	-	$M + m - N$	exp(P,C)	exp(P,C)	$\sum \lvert P_i\rvert + \sum \lvert C_i\rvert$
Sampling	m	m	$\sum \lvert P_i\rvert + \sum \lvert C_i\rvert$	m	m

Table 1: Comparison of sampling plans for DCR.

the performance of all methods with different d values for a terminal set $K = \{s, t\}$. Test II compares SIM5 and crude Monte Carlo when specifying a larger set K and diameter d. Test III is based on the countrywide transport network topology of ANTEL, the largest telecommunications provider in Uruguay, shown in Figure 4. It also illustrates the effect of specifying different d values as well as two different cases of link reliabilities. In all cases a comparison against CLR is also done. In particular it is interesting to analyse how the RE of the variance reduction methods changes when moving from CLR to DCR context; this is strongly related to the following. Let τ_I^s and τ_{II}^s be respectively the times consumed to generate a sample configuration under two sampling plans I and II (e.g. crude Monte Carlo and a variance reduction plan) and let τ^Φ be the time consumed for evaluating the corresponding Φ. Then the RE among both plans is given by the ratios among the times and the variances obtained σ_I^2 and σ_{II}^2 being $RE = (\tau_I^s + \tau^\Phi)/(\tau_{II}^s + \tau^\Phi) \times \sigma_I^2/\sigma_{II}^2$. We see that a growth in τ^Φ softens the effect of the difference between τ_I^s and τ_{II}^s increasing the impact over RE of the difference between σ_I^2 and σ_{II}^2. All methods were implemented in `C++` and the tests were run on an Intel Core2 Duo T5450 machine with 2 GB of RAM; reported times are expressed in seconds.

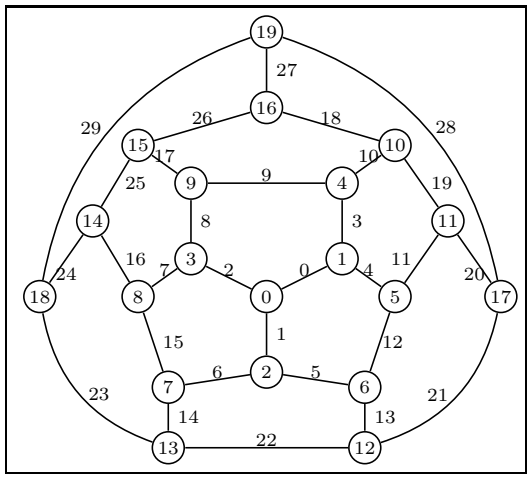

Figure 3: Network for test cases I and II

5.1.1. Test I

In this test we set $K = \{1, 9\}$, $p_e = 0.95$ for all $e \in E$, and we estimated $R(G, K, d)$ running all the discussed methods with a sample size $s = 2^{18}$ and diameter constraints $d = 2, 3, 4, 5, 6, 7$ plus CLR ($d = \infty$). The bounds (M, N) for SIM2 are $(2, 1)$ for $d = 2$, $(2, 2)$ for $d = 3, 4$ and $(2, 3)$ for $d = 5, 6, 7, \infty$. The disjoint sets P contain $\{3, 9\}$ for $d = 2$, $\{3, 9\}, \{8, 2, 0\}$ for $d = 3, 4, 5$ and $\{3, 9\}$, $\{8, 2, 0\}$, $\{17, 26, 18, 19, 11, 4\}$ for $d = 6, 7, \infty$. The disjoint sets C contain $\{3\}, \{9\}$ for $d = 2$, $\{0, 3\}, \{8, 9\}$ for $d = 3, 4$, $\{0, 3, 4\}, \{8, 9, 17\}$ for $d = 5, 6, 7, \infty$. For SIM3 and SIM4 larger (non-disjoint) sets P, C are used for most values of d as follows. P contains $\{3, 9\}, \{9, 10, 19, 11, 4\}, \{8, 2, 0\}$ for $d = 5$, $\{3, 9\}, \{8, 2, 0\}, \{17, 26, 18, 19, 11, 4\}, \{17, 26, 18, 10, 3\}$, $\{8, 7, 15, 6, 1, 0\}$ for $d = 6$ and $\{3, 9\}, \{8, 2, 0\}, \{17, 26, 18, 19, 11, 4\}, \{17, 26, 18, 10, 3\}, \{8, 7, 15, 6, 1, 0\}, \{17, 25, 16, 7, 2, 0\}$ for $d = 7, \infty$. C contains $\{0, 3\}, \{8, 9\}, \{2, 9\}, \{3, 8\}$ for $d = 3, 4$, $\{0, 3, 4\}, \{8, 9, 17\}, \{3, 8, 19\}, \{3, 8, 9\}$ for $d = 5$ and $\{0, 3, 4\}, \{8, 9, 17\}, \{8, 9, 25, 26\}, \{0, 3, 11, 12\}$ for $d = 6, 7, \infty$. Table 2 presents the results. Simulation times are reported as t_{sim}. $RE_{sim(z)}$ stands for the relative efficiency of SIM(z) when compared to SIM1. First, note that the estimated reliability grows when d goes from 2 to 7 and then ∞ as expected due to the progressive loosening of the diameter constraint. Second, observe that the variance is zero for SIM3,4,5 and $d = 2, 3, 4$ due to the fact that the set Γ is empty in those cases (the used pathsets and cutsets easily determined by visual inspection capture all information about R). Third, note that execution times tend to grow when d grows, due to the deeper BFS performed when checking each sample. Fourth, with regard to relative efficiency (RE), SIM2 outperform crude Monte Carlo for all $d \geq 5$ and SIM3, 4, 5 for all d. SIM4 has by far the largest RE ratios though one has to take into account the limitations already discussed in Subsection 4.3 (e.g. in this case building the precomputed tables took a time equivalent to that consumed by the simulation itself and would exponentially grow for larger topologies). SIM3 does not require precomputing the table but it does require a significant effort to find a minimal-size polynomial to compute the probabilities for sampling edges in Ω; and with highly overlapping sets P, C (as is for $d = 5, 6, 7$) the computation time growths outweigh the gains in bound tightness delivering a poorer performance when compared to SIM5. Fifth, in this case ($k = 2$) the time needed for sample checking is equal or lower for DCR than for CLR (one single BFS in both cases, potentially deeper for CLR) and thus the RE in CLR outperform those in DCR; this changes in tests II and III as we see next.

Diameter (d)	∞	2	3	4	5	6	7
$\hat{R}$	0.999702	0.902943	0.986343	0.986263	0.997276	0.999355	0.999577
$\hat{\sigma^2}_{sim1}$	1.13E-09	3.34E-07	5.14E-08	5.17E-08	1.04E-08	2.46E-09	1.61E-09
t_{sim1}	2.68	3.29	3.62	3.62	3.64	3.65	3.67
$\hat{\sigma^2}_{sim2}$	2.10E-10	2.54E-07	2.30E-08	2.30E-08	2.04E-09	4.49E-10	3.18E-10
t_{sim2}	9.41	10.12	10.33	10.19	10.36	10.39	10.45
RE_{sim2}	1.54	0.43	0.78	0.80	1.78	1.92	1.78
$\hat{\sigma^2}_{sim3}$	1.71E-13	0.00E+00	0.00E+00	0.00E+00	4.83E-11	2.20E-12	6.96E-13
t_{sim3}	46.74	11.47	17.39	17.26	34.37	51.84	59.92
RE_{sim3}	381.08	∞	∞	∞	22.67	78.71	141.89
$\hat{\sigma^2}_{sim4}$	1.71E-13	0.00E+00	0.00E+00	0.00E+00	4.83E-11	2.20E-12	6.96E-13
t_{sim4}	6.12	2.92	5.21	4.25	6.66	7.80	8.06
RE_{sim4}	2908.92	∞	∞	∞	117.07	523.03	1054.48
$\hat{\sigma^2}_{sim5}$	5.40E-13	0.00E+00	0.00E+00	0.00E+00	1.10E-10	4.65E-12	2.57E-12
t_{sim5}	5.80	2.89	4.56	4.73	7.77	8.57	8.69
RE_{sim5}	971.93	∞	∞	∞	44.06	225.00	265.34
Bounds							
$R_L(sim2)$	0.812179	0.214639	0.553542	0.553542	0.812179	0.812179	0.812179
$R_U(sim2)$	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
$R_L(sim3,4)$	0.998447	0.902500	0.986094	0.986094	0.991612	0.997850	0.998447
$R_U(sim3,4)$	0.999738	0.902500	0.990381	0.990381	0.999513	0.999738	0.999738
$R_L(sim5)$	0.996316	0.902500	0.986094	0.986094	0.986094	0.996316	0.996316
$R_U(sim5)$	0.999750	0.902500	0.995006	0.995006	0.999750	0.999750	0.999750

Table 2: Results for Test I

5.1.2. Test II

In this test we set $K = \{7, 9, 11, 12, 18, 19\}$, $p_e = 0.95$ for all $e \in E$, and we estimated $R(G, K, 9)$ and $R(G, K)$ comparing crude Monte Carlo with SIM5. Two disjoint pathsets are employed, $\{19, 10, 9, 17, 26, 27, 29, 23, 14, 22\}$ and $\{28, 20, 11, 4, 0, 2, 7, 8, 16, 24, 21, 13, 5, 6, 15\}$, as well as five disjoint cutsets, $\{27, 28, 29\}$, $\{8, 9, 17\}$, $\{11, 19, 20\}$, $\{6, 14, 15\}$ and $\{13, 21, 22\}$. Table 3 presents the results. As expected the estimated reliability is slightly higher for CLR than for DCR (operational configurations with diameters higher than nine exist although they represent a small fraction of the total). Due to higher values of k and d than in Test I, the ratio of SIM5 time respect to SIM1 is significantly lower in DCR than that for CLR which yields a higher RE for DCR than for CLR (16.59 vs 12.89). R_L is less close to R than R_U due to the fact that the employed pathsets are less (and larger) than the cutsets; disjointness is often a very tough requirement, particularly for sparse topologies. When relaxed by allowing to share some edges many more pathsets can be taken into account getting tighter lower bounds though at the expense of more costly computation of probabilities when building samples.

5.1.3. Test III

The network for this case has 121 edges and 109 nodes among which 22 are terminal (squares in Figure 4). SIM1 and SIM5 were run with $s = 2^{20}$ samples, uniform probabilities $p = 0.99$ and $p = 0.999$ and diameters $d = 25, 30, 35$ plus ∞ (CLR). The set P consists of a sigle pathset P_1 with the following edges: 18, 21, 22, 19, 20, 11, 12, 49, 50, 51, 52, 53, 54, 43, 42, 41, 40, 39, 38, 55, 56, 66, 65, 64, 63, 62, 61, 60, 59, 58, 75, 76, 96, 102, 95, 94, 105, 106, 113, 85 and 84. The set C consists of fifteen disjoint cutsets, easily generated by grouping the edges incident to fifteen terminal nodes, namely nodes 59, 0, 67, 52, 46, 7, 97, 3, 91, 35, 105, 31, 50, 33 and 12. The results are shown in Table 4. First observe that for this topology and set K the constraint $d = 25$ is very demanding (e.g. the distance between terminals 52 and 105 is 24) hence the reliability drop obtained for $d = 25$ when compared to the other values of d. This happens particularly for $p = 0.99$ because with lower edge reliabilities many failed configurations are sampled when working with d values close to 24. Second, observe that the RE (SIM5 vs SIM1) is higher for $d = 25, 30, 35$ than for CLR. One reason is that the time spent in checking d-connectedness is much higher than that spent for simple connectedness (as a consequence of high values of k and d) which dilutes the lower sampling time implied by a simpler computation of samples for SIM1. Another reason (particularly visible for $d = 25$ but also for $d = 30$) is that execution times are indeed lower for SIM5 than for SIM1. This is due to the fact that forcing P_1 to fail (in order to sample within Γ) often leads to sampling failed configurations, quickly identified as such when being checked, thus cutting down average iteration times in SIM5.

.	$\hat{R}$	$\hat{\sigma^2}$	t_{sim}	R_L	R_U	RE
sim1 (CLR)	0.999138	3.49E-09	4.63	0.000000	1.000000	
sim5 (CLR)	0.999130	2.00E-10	6.26	0.784639	0.999375	12.89
sim1 (d=9)	0.999081	3.72E-09	19.44	0.000000	1.000000	
sim5 (d=9)	0.999105	2.05E-10	21.26	0.784639	0.999375	16.59

Table 3: Results for Test II

.	$\hat{R}$	$\hat{\sigma^2}_{sim1}$	$\hat{\sigma^2}_{sim5}$	t_{sim1}	t_{sim5}	RE
p=0.99						
CLR	0.990734	8.65E-09	2.90E-09	95.77	108.94	2.62
d=35	0.984802	1.41E-08	4.67E-09	1596.03	1583.41	3.05
d=30	0.953564	4.21E-08	1.29E-08	1547.34	1469.22	3.44
d=25	0.788949	1.59E-07	2.55E-08	1335.24	817.93	10.17
p=0.999						
CLR	0.999916	7.64E-11	3.20E-12	95.49	108.64	21.00
d=35	0.999866	1.27E-10	5.13E-12	1628.81	1641.85	24.62
d=30	0.998572	1.38E-09	5.28E-11	1613.50	1598.75	26.46
d=25	0.977157	2.13E-08	3.78E-10	1590.04	945.27	94.84

Table 4: Results for Test III

6. Conclusions

We have shown how bounded Monte Carlo sampling plans can be applied to estimate the DCR. We briefly presented methods which make use of previously-known topological information and discussed the consequences of working in the DCR context. Then we illustrated them with three tests based on a case usually employed in the literature and a real network topology. These tests show that the methods based on precomputing pathsets and cutsets have the best relative efficiencies when compared to crude Monte Carlo. When the pathsets and cutsets involve a limited amount of edges precomputing tables (SIM4) can be the best option (Test II) particularly if a large number of samples are to be generated due to high significance goals for the estimators. Otherwise, disjoints pathsets and cutsets (SIM5) are recommended due to low sampling times, although it is often the case that allowing a small number of overlapping edges significantly tightens the bounds (leading to variance reductions) without adding major complexity to the sampling computations; further research has to be conducted regarding this trade-off. We also have discussed and illustrated the important fact that two features present in most real cases (large sets K and diameters d) yield higher relative efficiencies for DCR than for CLR (Tests II and III) which makes the research on high variance reduction methods for DCR even more appealing than for CLR.

Appendix A. Sequential sampling for pathset/cutset based bounding

Let P, C and Ω be defined as in Subsection 4.3. Let the edges of Ω be arbitrarily ordered with $h = |\Omega|$, $(z_1, ..., z_h) \in \{0,1\}^h$ an h-tuple encoding the sampled states for the edges of Ω and $\rho_1, ..., \rho_h$ their (unconditioned) probabilities of operation. Suppose that the sampling procedure has achieved a certain point where $r < h$ edges from Ω have been sampled. Then, in order to sample a value for z_{r+1} (corresponding to an edge that we will denote e) we need to compute the probability p that the $(r+1)$-th edge of Ω is operational provided that the configuration under construction belongs to Γ (we denote that event as E_Γ) and the first r edges have already received certain sampled values (we denote that event as E_r). In the context of this sampling method, belonging to Γ means that the event E_{ncf} ="no cutset of C fails" must be met while at the same time the event E_{omp} ="one or more paths of P are operational" must not. It is easy to see that $E_{omp} \to E_{ncf}$. For convenience we will denote the probability of event A conditioned to event B both as $\Pr(A|B)$ and $\Pr|_B(A)$. Using this notation,

$$p = \Pr(z_{r+1} = 1 \mid E_\Gamma \wedge E_r) = \frac{\Pr(z_{r+1} = 1 \wedge E_\Gamma \wedge E_r)}{\Pr(E_\Gamma \wedge E_r)} = \frac{\Pr|_{E_r}(z_{r+1} = 1 \wedge E_\Gamma)\Pr(E_r)}{Pr|_{E_r}(E_\Gamma)\Pr(E_r)} = \Pr|_{E_r}(z_{r+1} = 1)\frac{\Pr|_{E_r}(E_\Gamma \mid z_{r+1} = 1)}{\Pr|_{E_r}(E_\Gamma)}$$

Since the failures are independent we have that $\Pr|_{E_r}(z_{r+1} = 1) = \Pr(z_{r+1} = 1) = \rho_{r+1}$; and since $E_{omp} \to E_{ncf}$ then

$$p = \rho_{r+1}\frac{\Pr|_{E_r}(E_{ncf} \mid z_{r+1}=1) - \Pr|_{E_r}(E_{omp} \mid z_{r+1}=1)}{\Pr|_{E_r}(E_{ncf}) - \Pr|_{E_r}(E_{omp})} = \tag{A.1}$$

$$\rho_{r+1}\frac{\Pr(E_{ncf} \mid E_r \wedge z_{r+1}=1) - \Pr(E_{omp} \mid E_r \wedge z_{r+1}=1)}{\Pr(E_{ncf} \wedge E_r) - \Pr(E_{omp} \wedge E_r)} \tag{A.2}$$

Let $\omega(\rho_1,...,\rho_n)$ and $\lambda(\rho_1,...,\rho_n)$ be the polynomials that respectively compute $p(E_{ncf})$ and $p(E_{omp})$. Then Eq. A.1 translates into

$$p=\rho_{r+1}\frac{\omega(z_1,...,z_r,1,\rho_{r+2},...,\rho_h)-\lambda(z_1,...,z_r,1,\rho_{r+2},...,\rho_h)}{\omega(z_1,...,z_r,\rho_{r+1},...,\rho_h)-\lambda(z_1,...,z_r,\rho_{r+1},...,\rho_h)} \tag{A.3}$$

Finding the polynomials ω and λ can be a quite complex task if the pathsets and cutsets are numerous and highly overlap. Only if they are pairwise disjoint they can be easily computed as

$$\omega(\rho_1,...,\rho_h) = \prod_{j\in\{1..J\}}\left(1-\prod_{i\in C_j}\rho_{\Omega(i)}\right) \tag{A.4}$$

$$\lambda(\rho_1,...,\rho_h) = 1-\prod_{j\in\{1..I\}}\left(1-\prod_{i\in P_j}\rho_{\Omega(i)}\right) \tag{A.5}$$

where $\Omega(i)$ denotes the index of edge i in the ordering used for the edges of Ω. In this case the probability for sampling each edge of Ω can be computed in $O(|\Omega|)$ time as next section shows.

Appendix A.1. Sequential sampling for SIM5

Let $\lambda_j = \prod_{i\in P_j} p_i$ for $j = 1,...,I$ be the probability that pathset P_j is operational and $\omega_j = \prod_{i\in C_j} q_i$ for $j = 1,...,J$ the probability that cutset C_j fails (with $q_i = 1-p_i$). Then let us denote $\lambda = 1-\prod_{j=1}^{I}(1-\lambda_j)$ the probability of the event E_{omp} and $\omega = \prod_{j=1}^{J}(1-\omega_j)$ the probability of the event E_{ncf}. For convenience let us also define $\lambda_0 = \omega_0 = 0$ and the following pointers j_i (pathset to which edge $\Omega(i)$ belongs) and k_i (cutset to which edge $\Omega(i)$ belongs):

$$j_i = \begin{cases} j & \text{if } e_{\Omega(i)} \in P_j \quad j=1...I \\ 0 & \text{if } e_{\Omega(i)} \notin \bigcup_{k=1...I} P_k, \end{cases}$$

$$k_i = \begin{cases} j & \text{if } e_{\Omega(i)} \in C_j \quad j=1...J \\ 0 & \text{if } e_{\Omega(i)} \notin \bigcup_{k=1...J} C_k. \end{cases}$$

From Eq. A.1 the failure probability to sample the first edge of Ω will be

$$q = \frac{\Pr(z_1=0\wedge E_{ncf}) - \Pr(z_1=0\wedge E_{omp})}{\omega-\lambda} \tag{A.6}$$

The event $z_1 = 0 \wedge E_{ncf}$ will be met whenever the edge $e_{\Omega(1)}$ fails and all cutsets C (among which one could eventually include this edge) have at least one operating edge, so

$$\Pr(z_1=0\wedge E_{ncf}) = q_{\Omega(1)}\prod_{j=1,j\neq k_1}^{J}(1-\omega_j)(1-\prod_{i\in C_{k_1}\setminus\{h\}} q_{\Omega(1)})$$

and thanks to the definition of ω_0 this yields the compact form

$$\Pr(z_1=0\wedge E_{ncf}) = q_{\Omega(1)}\frac{\omega}{1-\omega_{k_1}}\left(1-\frac{\omega_{k_1}}{q_{\Omega(1)}}\right) \tag{A.7}$$

At the same time, the event $z_1 = 0 \wedge E_{omp}$ will be met whenever the edge $e_{\Omega(1)}$ fails and not every pathset in P fails (note that an eventual pathset including this edge will be failed), so

$$\Pr(z_1=0\wedge E_{omp}) = q_{\Omega(1)}\left(1-\prod_{j=1,j\neq j_1}^{I}(1-\lambda_j)\right)$$

that, thanks to the convenient definition of λ_0 yields

$$\Pr(z_1=0\wedge E_{omp}) = q_{\Omega(1)}\frac{\lambda-\lambda_{j_1}}{1-\lambda_{j_1}} \tag{A.8}$$

Once the probability of failure for edge $e_{\Omega(1)}$ has been computed through Eq. A.7, A.8 A.6, z_1 is sampled and its resulting state must be taken into account when computing the probability of failure for sampling the remaining edges of Ω. First of all, the probability ω_{k_1} that the cutset that includes $e_{\Omega(1)}$ fails (provided it exists) will be zero if the edge operates but exclude the factor $(1-p_{\Omega(1)})$ from its definition if it fails. Similarly, the probability λ_{j_1} that the pathset that includes that edge operates (provided it exists) will be zero if it fails but exclude the factor $p_{\Omega(1)}$ from its definition if it operates, so before sampling next edges in Ω both must be adjusted the following way (apostrophes mean new values to take before addressing next edge):

$$\omega'_{k_1} = \frac{(1-z_1)\omega_{k_1}}{q_h} \qquad \lambda'_{j_1} = \frac{\lambda_{j_1} z_1}{p_{\Omega(1)}}$$

Finally ω and λ must also be updated to reflect the changes in ω_{k_1} and λ_{j_1}:

$$\omega' = \frac{\omega}{1-\omega_{k_1}}(1-\omega'_{k_1}) \qquad \lambda' = 1-\frac{1-\lambda}{1-\lambda_{j_1}}(1-\lambda'_{j_1})$$

By repeating the preceding steps for $z_2,...,z_h$ the remaining edges of Ω get sampled. This is summarized in the pseudo-code of Figure A.5.

Procedure SampleOmega$(E, \Omega, p_i, I, J, \lambda, \omega, \{\lambda_i\}, \{\omega_i\}, \{j_i, k_i\})$
Returns: Set of edges in Ω sampled as operating

Sample the state of each edge e in $E \setminus \Omega$ as Binomial(p_e)
$S \leftarrow \emptyset$
for all $h = 1, ..., |\Omega|$ **do**
  $\alpha \leftarrow \frac{\omega}{1-\omega_{k_h}}(1 - \frac{\omega_{k_h}}{q_{\Omega(h)}})$
  $\beta \leftarrow \frac{\lambda - \lambda_{j_h}}{1-\lambda_{j_h}}$
  $q \leftarrow q_{\Omega(h)} \frac{\alpha - \beta}{\omega - \lambda}$
  Sample z_h as Binomial$(1-q)$
  if $z_h = 1$ **then**
    $S \leftarrow S \cup \{e_{\Omega(h)}\}$
  end if
  $\omega'_{k_h} \leftarrow \frac{(1-x_h)\omega_{k_h}}{q_{\Omega(h)}}$
  $\lambda'_{j_h} \leftarrow \frac{\lambda_{j_h} x_h}{p_{\Omega(h)}}$
  $\omega' \leftarrow \frac{\omega}{1-\omega_{k_h}}(1 - \omega'_{k_h})$
  $\lambda' \leftarrow 1 - \frac{1-\lambda}{1-\lambda_{j_h}}(1 - \lambda'_{j_h})$
  $\omega_{k_h} \leftarrow \omega'_{k_h}; \lambda_{j_h} \leftarrow \lambda'_{j_h}; \omega \leftarrow \omega'; \lambda \leftarrow \lambda'$
end for
return S

Figure A.5: Pseudo-code for sampling edges in SIM5

References

[1] L. Petingi, J. Rodriguez, Reliability of networks with delay constraints, in: Congressus Numerantium, Vol. 152, 2001, pp. 117–123.

[2] H. Cancela, L. Petingi, Diameter constrained network reliability: exact evaluation by factorization and bounds, in: ICIL'2001 (International Conference on Industrial Logistics), 2001, pp. 359–366.

[3] H. Cancela, L. Petingi, On the characterization of the domination of a diameter-constrained network reliability model, Discrete Appl. Math. 154 (2006) 1885–1896. doi:10.1016/j.dam.2006.03.029.
URL http://dl.acm.org/citation.cfm?id=1163910.1163924

[4] H. Cancela, L. Petingi, Properties of a generalized source-to-all-terminal network reliability model with diameter constraints, Omega 35 (6) (2007) 659 – 670, special Issue on Telecommunications Applications.

[5] H. Cancela, M. El Khadiri, L. Petingi, Polynomial-time topological reductions that preserve the diameter constrained reliability of a communication network, IEEE Transactions on Reliability (accepted, to appear 2012).

[6] M. Ball, Computing network reliability, Operations Research 27 (4) (1979) 823–838.

[7] H. Cancela, L. Petingi, Reliability of communication networks with delay constraints: computational complexity and complete topologies, International Journal of Mathematics and Mathematical Sciences 2004 (2004) 1551–1562.

[8] C. J. Colbourn, The Combinatorics of Network Reliability, Oxford University Press, Inc., New York, NY, USA, 1987.

[9] G. Rubino, Network reliability evaluation. In State-of-the-art in performance modeling and simulation, Gordon and Breach Books, 1996, Ch. 11.

[10] L. Petingi, A diameter-constrained network reliability model to determine the probability that a communication network meets delay constraints, WTOC 7 (2008) 574–583.
URL http://dl.acm.org/citation.cfm?id=1456091.1456096

[11] G. Rubino, B. Tuffin, Rare event simulation using Monte Carlo methods, Wiley, 2009.
URL http://books.google.com/books?id=zLE4zb3N6l0C

[12] R. Van Slyke, H. Frank, Network reliability analysis: Part i, Networks 1 (3) (1971) 279–290. doi:10.1002/net.3230010307.
URL http://dx.doi.org/10.1002/net.3230010307

[13] H. Kumamoto, K. Tanaka, K. Inoue, Efficient evaluation of system reliability by Monte Carlo method, IEEE Transactions on Reliability R-26 (5) (1977) 311 –315. doi:10.1109/TR.1977.5220181.

[14] G. S. Fishman, A Monte Carlo sampling plan for estimating network reliability, Operations Research 34 (4) (1986) 581–594.

[15] G. S. Fishman, A comparison of four Monte Carlo methods for estimating the probability of s-t connectedness, IEEE Transactions on Reliability 35 (2) (1986) 145 –155. doi:10.1109/TR.1986.4335388.

[16] P. A. Golovach, D. M. Thilikos, Paths of bounded length and their cuts: Parameterized complexity and algorithms., in: IWPEC'09, 2009, pp. 210–221.

[17] R. Barlow, K. Heidtmann, Computing k-out-of-n system reliability, IEEE Transactions on Reliability 33 (1984) 58 – 61.

[18] L. Belfore, An o(n.(log[2](n))[2]) algorithm for computing the reliability of k-out-of-n:g and k-to-l-out-of-n:g systems, IEEE Transactions on Reliability 44 (1) (1995) 132 – 136.

[19] G. Arulmozhi, Direct method for reliability computation of k-out-of-n: G systems, Applied Mathematics and Computation 143 (2-3) (2003) 421 – 429.

[20] E. Manzi, M. Labbe, G. Latouche, F. Maffioli, Fishman's sampling plan for computing network reliability, IEEE Transactions on Reliability 50 (1) (2001) 41 –46. doi:10.1109/24.935016.

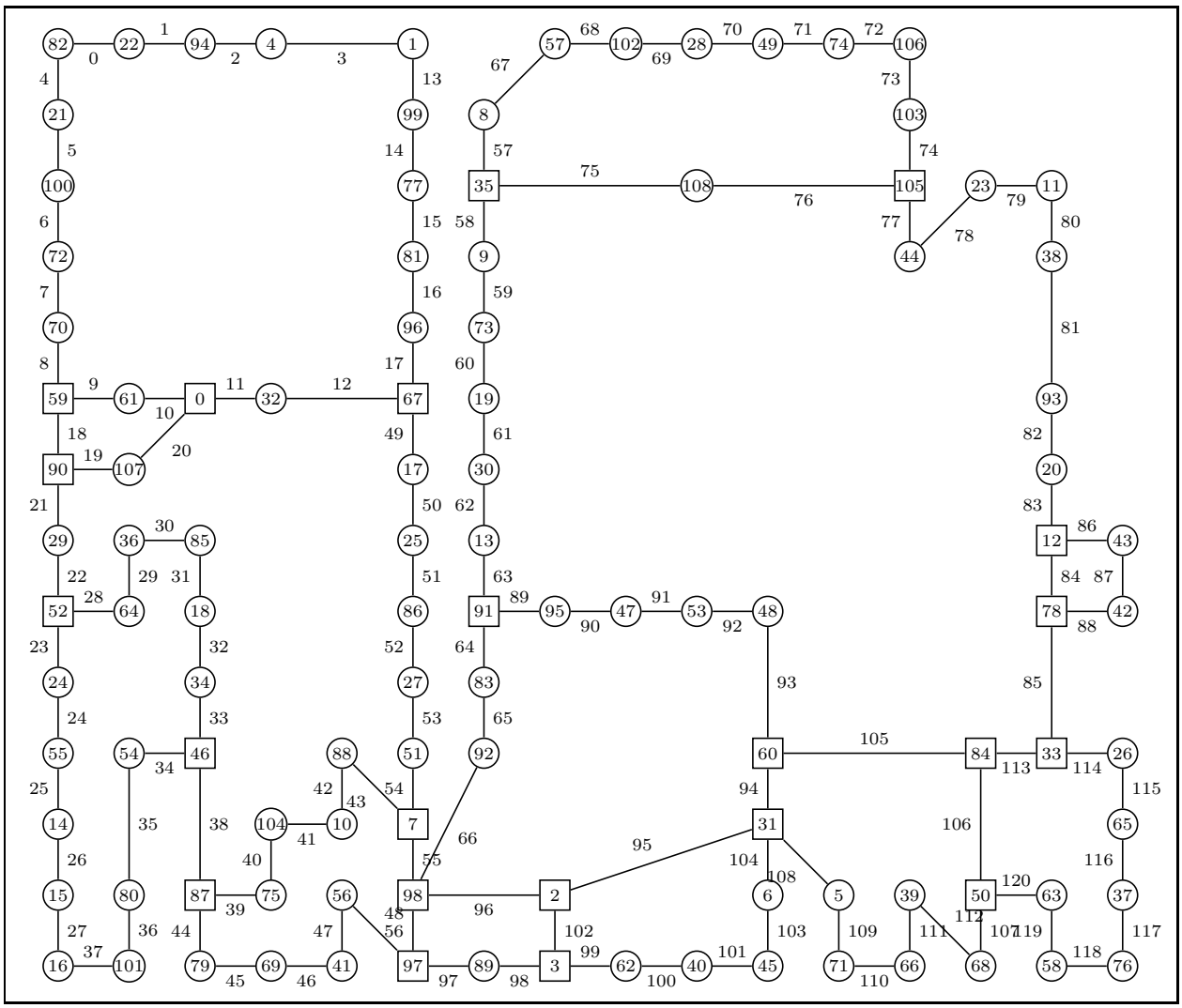

Figure 4: Network topology for Test III