A Queueing Approach to Model Network Flow Dynamics

T. Kleiberg and P. Van Mieghem *

Delft University of Technology
August 31, 2006

Abstract

The traffic dynamics in networks are investigated at the flow level. The relation between network and traffic characteristics is modeled using a queueing approach. When we view the network as a regular M/M/1/K queue, we find a linear relation between the "network capacity" and the number of links, $K = \beta\binom{N}{2}$. This relation facilitates to explain several aspects of the dynamic behavior inside the network, such as loading and rejection rate. Additionally, we study the number of flows that can be allocated in a network before rejection occurs. We have found a good approximation of the residual capacity of the network subject to the allocation of flows and compare the network with the classical random graph regarding the rejection rate, connectivity and degree distribution.

1 Introduction

Voice over IP, Online gaming and video streaming are a grasp of the numerous emerging applications that prelude a new generation of services on the Internet. Characteristic to these services are the time-critical nature and high susceptibility to the network performance, e.g. delay and packet loss. Currently, network providers meet the stringent requirements on end-to-end communications by over provisioning the core network, because it is the simplest and most effective solution at hand (the utilization of backbone links rarely reaches 60 percent). However, when Fibre-To-The-Home will be a fact, the classical telecom network control and management will re-enter because the core will use a same technology as in the access networks. The wide deployment of bulky services that we envisage, will place high capacity demands on the core and access networks. Over-provisioning may not suffice anymore and network providers may be forced to use alternate measures to meet the performance requirements of user-demanded Quality-of-Service (QoS) [5]. Providing QoS requires a good understanding of the (dynamic) network behavior and performance.

The primary tools for evaluating network performance are modeling and simulation. Measurements on real networks and test-beds are in most situations not possible or feasible due to the required network size and/or issues related to ownership, authority or secrecy. The difficulties in network modeling stem from the heterogeneous and dynamic nature of real networks. Floyd and Paxson [3] describe the Internet as "An Immense Moving Target" that is constantly changing. A universal model

*Faculty of Electrical Engineering, Mathematics and Computer Science, P.O Box 5031, 2600 GA Delft, The Netherlands; *email*:T.Kleiberg, P.VanMieghem@ewi.tudelft.nl.

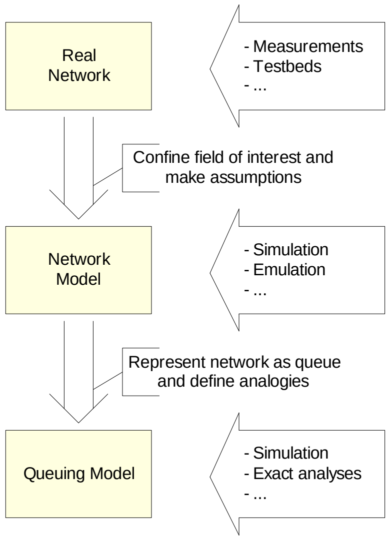


Figure 1: Performance evaluation at different levels (real network, network model and queueing model). The white arrows indicate the main techniques that are commonly available at a specific level.

that adequately describes all network characteristics is not available. Instead, researchers must craft a model based on the specific problem that captures all relevant behavior but that ignores redundant aspects. A complementary problem is that network performance is often evaluated according so-called "emergent properties". Emergent properties are system global features and capabilities which are not specified by network design and are difficult or impossible to predict from knowledge of its constituents. Examples of such properties are the hopcount and the betweenness. Computation of such measures is often possible for networks in equilibrium, but little is known about these measures when the load in the network varies. The multitude in variables and their cross-correlations strongly complicate stochastic and transient analyses of the network behavior.

The instinctive drive to model the network as realistic as possible, opposes to the ability to interpret this model and to derive analytic solutions. The objective of this paper is to clarify certain aspects of the network dynamics by means of queueing and graph theory. This is schematically displayed in Figure 1, where common performance evaluation techniques are shown at various levels of modeling. The network model that we use is presented in Section 2 where we will motivate the assumptions and usability. In Section 3 we analyze the network model with use of queueing theory. Queueing theory facilitates, to some extent, the distillation of the network behavior caused by topological features and traffic conditions. In Section 6 we calculate blocking probabilities based on topology information. Section 5 discusses how we can estimate the capacity of networks. Section A illustrates the difficulties that are introduced into the network by path allocation. We conclude our paper with a summary in Section 7.

2 Network Model

Let us consider a fixed network consisting of N nodes and L links. The term "network dynamics" is here broadly understood as the set of network properties such as the blocking and loss rate, the number of allocated paths, the hopcount of these paths, etc. that change over time when loading the network with traffic.

The loading of a network with traffic needs to be detailed. In general, traffic is injected into the network and leaves the network elsewhere. First, we assume that traffic is only injected in one node, the source, and that it leaves the network in one other node, the destination. In other words, we confine to unicast. The source-destination pair is assumed to be uniformly chosen over all N nodes of the network. This assumption for the Internet is quite realistic as argued in [11, pp. 340]. A slightly more realistic setting is to take the density population of users on earth and measurements of traffic matrices into account, at the expense of a considerably more complex model. Second, we confine to flows. A flow can be regarded as a connection set-up between a source and a destination node in the network. The ensemble of all packets of that flow follow the same path from a source to destination during the life-time of that flow. The motivation to confine to flows lies in optical networks where light paths are set-up in that manner. Alternatively, a flow can be an accumulation of connections from source to destination, e.g. packets originated at various sources entering and leaving the backbone at the same ingress and egress routers or, a flow can represent a multimedia streaming. In addition, since only flows are observed in our setting, the details of the packet level such as packet inter-arrival times and packet correlations can be omitted, but only the flow arrival rate plays a role. We assume a Poisson arrival process with average rate λ. The Poisson assumption for flows is commonly regarded as realistic: it is very precise for telephony, and, on the aggregate level, also for the Internet. The flow duration is less universally agreed upon. For simplicity and as a first step in modeling the "network dynamics", we assume that the flow duration is exponentially distributed with mean $\frac{1}{\mu}$. Below, we return to this assumption.

Next, we need to specify the capacity consumption of a flow. In this paper, we focus on an extreme scenario in which a flow consumes the entire link capacity. In addition, we assume that all links have a unit capacity, are unweighed and undirected. We exclude the existence of self-loops and multiple links between two nodes. Thus, the network is homogeneous: for example, a same optical technology with same link drivers. Since each flow consumes the full capacity of a link, an allocated link remains unavailable for future request until the flow has been terminated or released. Upon the arrival of a flow set-up request, a minimum-hop path is computed between the source-destination pair using Dijkstra's shortest path algorithm. If a path is found, the flow is set-up by allocating the required capacity on the links that constitute the path. If multiple shortest paths are found, one of these paths is chosen randomly. If the network lacks resources to provide a feasible path, the flow request is rejected. Figure 2 illustrates a schematic representation of the network model with the arrival process. Flow requests arrive at random intervals (Poisson assumption) and are then routed. If the flow request is accepted, the connection is established.

The major reason for choosing such an extreme scenario was inspired by the question: "How many flows (light paths) can be set-up in a network?" The maximal possible number of flows that can be set-up in our setting appears in the complete graph because any other graph is a subgraph of the complete graph. Hence, we first focus on the complete graph. The description of the network model is now complete.

Although the network model may seem overly simplified with respect to reality – a fact that we do not deny –, this comment should be placed in some perspective. The even simpler model of a

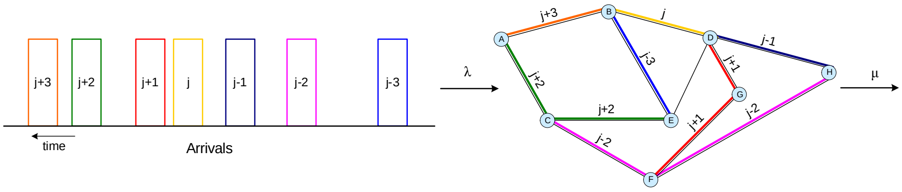


Figure 2: The network model with flow arrivals. Arrivals occur at random intervals and the requests are routed in the network. E.g. flow request request j corresponds to a connection between nodes B and D, while flow $j+2$ follows the path $A-C-E$.

K	queue buffer capacity	$G_p(N)$	E-R random graph
N	nodes	K_N	complete graph
L	links	N_S	system size of the queue (buffer plus server) / flows in the network
p	link density or probability		
p_c	critical link density	T_S	sojourn time of jobs in the queue / time flows resides in the network
D	nodal degree		
j	flow or path index	H_N	hopcount
λ	average flow arrival rate	τ_r	relaxation time
μ	average flow service rate	R	flow requests
ρ	traffic load	r	flow rejection rate

Table 1: Table of notation.

lattice in which each flow just consists of 1 link already leads to a difficult percolation problem [4]. Indeed, when removing random links in a lattice, the lattice is disconnected with high probability if the link density $p = \frac{L}{L_{\text{total}}} \to \frac{1}{2}$ for large N, where L_{total} is the total number of links in the lattice. When considering the complete graph instead of the lattice, the complete graph is disconnected by removing random links when $p < p_c$ and the threshold link density is $p_c \approx \frac{\log N}{N}$ for large N. This is a key result in the theory of the Erdös-Rényi (E-R) or classical random graph $G_p(N)$ (see e.g. [6]). In the first stages of loading the network – equivalent to removing paths, a set of correlated links –, our network model shows resemblance with the random graph $G_p(N)$. However, the fact that paths and not random links are removed, is shown to considerably complicate the understanding of our results.

3 A queueing model of our network model

Before turning to simulations, in this section, we analyze our network model explained in Section 2 with queueing theory. We regard the network as a single system at which flows arrive and depart at random times. We assume that the network functionalities such as routing, signalling and admission-control, are invisible outside the network. Hence, the network can be viewed as a black box, which is illustrated as a cloud at the top in Figure 3. We compare the network model with a single-server queue with finite buffer K as visualized at the bottom in Figure 3 at which jobs (flow requests) arrive and depart. The correspondence between the network model and its queueing analogue requires that the network and the queue experience a same in- and output. The arrival of a flow at the network coincides with adding a new job to the queue's buffer. The analogue of the life time of a flow is the sojourn

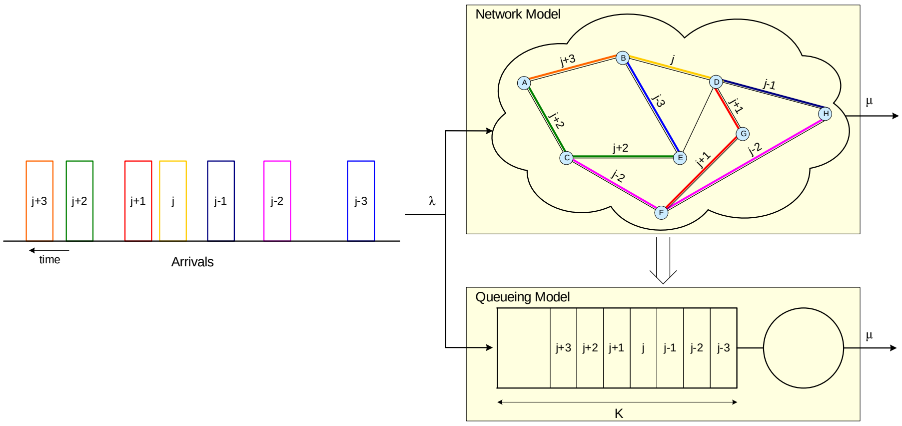


Figure 3: The network is modeled by a single queue. The flow arrivals at the network correspond to arriving jobs at the queue.

time of a job at the queue, where each job consists of (i) selecting an arbitrary source-destination pair, (ii) performing a shortest path computation, and (iii) setting-up of the path/connection or, if not possible, announcing an error/rejection message. The average service rate at the queue is equal to the average release or termination rate of a flow in the network. Since flows are terminated arbitrarily in the network, the service discipline in the queueing system should be random, in the sense that the server selects a uniformly chosen packet in the buffer. The service discipline is not FIFO order. Our network modeling assumption have been taken in such a way that the corresponding queue arrival process is a Poisson process with rate λ and the service process is exponentially distributed with mean service rate μ. Hence, the analogue of the network model is of the M/M/1 queuing family. The sequel of this section is devoted to further discuss and motivate the analogy.

The M/M/1 queue is one of the few queueing systems for which a time-dependent analysis is available. The mean system size $N_S(t)$ (buffer plus server) of the M/M/1 queue as a function of time is given for $\rho < 1$ by [9],

$$\mathrm{E}\left[N_{S_{M/M/1}}(t)|N_{S_{M/M/1}}(0)=0\right] = \mathrm{E}\left[N_{S_{M/M/1}}\right] - \frac{2}{\pi}\int_0^{\pi} \frac{e^{-\gamma(y)\mu t}\sin^2 y}{\gamma(y)^2}dy \tag{3.1}$$

where $\mathrm{E}\left[N_{S_{M/M/1}}\right] = \frac{\rho}{1-\rho}$ is the steady state system size and $\gamma(y) = 1+\rho-2\sqrt{\rho}\cos y$. Abate and Whitt [1] and Sharma *et al.* [8, 7] provide alternate expressions for the mean system size. Using Little's Law [11, Sec. 13.6], the average system time – the time each jobs spends in the system – is $\mathrm{E}\left[T_S(t)\right] = \mathrm{E}\left[N_S(t)\right]/\lambda$. One important difference between the network model and the classical M/M/1 queue is the service order: FIFO whereas the network releases flows arbitrarily. Since we are interested in average quantities to first order, Little's law is not affected by the service discipline, as long as the service discipline does not control the arrivals. Furthermore, Little's Law assumes a work-conserving system where all offered load is serviced. If the network induces blocking, a fraction of the flows offered to the network are rejected which affects the actual arrival rate seen by the network. Hence, Little's Law must be applied with use of the effective arrival rate.

When the network lacks resources to accommodate a new flow request, the request is rejected.

The rejection rate is equal to the loss probability for each new flow request and is defined as

$$r = \frac{\mathrm{E}\left[R_{\text{reject}}\right]}{R_{\text{total}}}$$

where R_{reject} and R_{total} represent the sum of rejected requests and the total number of requests, respectively. In the queueing analogue, rejections are modeled by assuming a finite buffer. The size of the buffer then relates to the maximum number of concurrent flows in the network. If we assume an M/M/1/K model as analogy for the network, then K corresponds to maximum number of flows that the network can accommodate, which we will refer to as the "network capacity". The limited capacity affects the rejections by network and consequently the load. The steady state rejection rate for the M/M/1/K queue is found as the probability that the system contains K items,

$$r = \Pr\left[N_{S_{M/M/1/K}} = K\right] = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} \tag{3.2}$$

Analytic solutions for the M/M/1/K queue in both the transient and stationary domain are found by Tarabia [10]. Equations (3.3)–(3.5) give the first and second order moment of the system size in steady state and the first order moment in the transient domain when $\rho \neq 1$.

$$\mathrm{E}\left[N_{S_{M/M/1/K}}\right] = \mathrm{E}\left[N_{S_{M/M/1}}\right] - \frac{(K+1)\,\rho^{K+1}}{1-\rho^{K+1}} \tag{3.3}$$

$$\mathrm{E}\left[N^2_{S_{M/M/1/K}}\right] = \frac{\rho C(\rho)}{(1-\rho)^2\,(1-\rho^{K+1})} \tag{3.4}$$

$$\mathrm{E}\left[N_{S_{M/M/1/K}}(t)\right] = \mathrm{E}\left[N_{S_{M/M/1/K}}\right] - \frac{1}{K+1}\sum_{j=1}^{K} A(\rho,\nu)e^{-(\lambda+\mu)t+2t\sqrt{\rho}\cos\nu} \tag{3.5}$$

with $\nu = \frac{\pi j}{K+1}$ and where

$$\begin{aligned} A(\rho,\nu) &= \frac{2\rho\left(\sin\nu + \rho^{\frac{K+1}{2}}\sin K\nu\right)\sin\nu}{\left(1+\rho-2\sqrt{\rho}\cos\nu\right)^2} \\ &\quad + \rho^{\frac{K+2}{2}}\left(2+4K+\rho\left(2K-1\right)\right)\sin K\nu \\ C(\rho) &= 1+\rho-(K+1)^2\rho^K + \left(2K^2+2K-1\right)\rho^{K+1} - K^2\rho^{K+2} \end{aligned}$$

As mentioned previously, to correctly apply Little's formula to the case of the M/M/1/K queue, the effective arrival rate must be considered,

$$\mathrm{E}\left[T_{S_{M/M/1/K}}\right] = \frac{\mathrm{E}\left[N_{S_{M/M/1/K}}\right]}{\lambda(1-r)} = \frac{1-\rho^{K+1}}{\lambda(1-\rho^K)}\mathrm{E}\left[N_{S_{M/M/1/K}}\right]$$

A non-trivial issue is the determination of K. In the M/M/1/K queueing model, K is simply a parameter that is fixed. In the network perspective, K is a random variable that, in a complicated manner, depends on the topology and traffic parameters. Hence, we use simulation to obtain estimates for K and investigate how K scales with the network size N and traffic load ρ. A novel point of our analysis is the relation (3.6) between K and the network capacity, at least for the complete graph K_N where $\beta \approx 0.42$,

$$K = \beta\binom{N}{2} \tag{3.6}$$

1 INITIALIZE(K_N)
2 $t_{arrivals} \leftarrow t_{departures} \leftarrow 0$
3 queue $Q_{flows} \leftarrow \varnothing$
4 **for** $j \leftarrow 1$ **to** j_{max}
5   **do while** $Q_{flows} \neq \varnothing$ **and** $t_{departures} \leq t_{arrivals}$
6   **do** $t_{departures} \leftarrow t_{departures} +$ RANDOM-EXPONENTIAL($\frac{1}{\mu}$)
7   RELEASE-FLOW(K_N, Q_{flows})
8     $t_{arrivals} \leftarrow t_{arrivals} +$ RANDOM-EXPONENTIAL($\frac{1}{\lambda}$)
9   **if** $Q_{flows} = \varnothing$
10   **then** $t_{departures} \leftarrow t_{arrivals}$
11     $f \leftarrow$ GENERATE-FLOW(K_N)
12   **if** $length[f] > 0$
13   **then** ALLOCATE-FLOW(K_N, Q_{flows}, f)

Figure 4: Meta-code for network simulation.

4 Simulations

We have evaluated network performance measures via simulations. The meta-code for a simulation is presented in Figures 4 to 7. The network is initialized in line 1 as the complete graph K_N with unweighed, bi-directional links with unit capacity. The network is empty and unloaded. In line 2, variables $t_{arrivals}$ and $t_{departures}$ are initialized that correspond to the flow-arrival and -departure times. In line 3 the queue Q_{flows} is initialized, which stores the active flows in the networks. The simulation starts in line 4, where it enters a loop that is executed for each flow-arrival. The loop repeats j_{max} times, corresponding to the maximum number of flow-arrivals. In lines 5–7, flows are released from the network until the network is empty or until $t_{departures} > t_{arrivals}$. Next, $t_{arrivals}$ is updated and if the network is empty, $t_{departures}$ is given the same value in lines 9–10. Finally, in lines 11–13 a new flow is generated and allocated.

Figures 5, 6 and 7 contain meta-code for three subroutines used in Figure 4. Figure 5 illustrates how a flow is generated. Two (different) nodes are chosen randomly from K_N in lines 2 and 3, after which Dijkstra's shortest path is computed and inserted into f. In Figure 6 the meta-code is displayed for allocating flow f into K_N. In lines 1–4 each link along flow f is traversed and allocated in K_N. When the link is allocated, the full capacity on that link is reserved. In line 5 the flow is added to the queue Q_{flows}. Figure 7 explains the release of a flow. In line 1 a random flow is chosen and removed from Q_{flows}. In lines 2–5 the links used by this flow are released.

At the start of the simulation, the network resides in a temporal warm-up phase, where the average amount of incoming traffic exceeds the average amount of traffic being served. The load gradually increases until a steady state is reached where the average number of flows entering and leaving the network are in balance. The speed of convergence to the steady state is compared for different network sizes. Furthermore, we are interested how the average number of flows in steady state is a function of N. We have simulated 10^3 network realizations and issued 10^5 flow requests per realization.

```
GENERATE-FLOW(K_N)
flow f ← ∅
node v ← RANDOM-NODE(K_N)
node u ← RANDOM-NODE(K_N)
f ← DIJKSTRA(K_N, v, u)
return f
```

Figure 5: Meta-code for GENERATE-FLOW subroutine.

```
ALLOCATE-FLOW(K_N, Q_flows, f)
for i ← 1 to length[f] − 1
   do node v ← f[i]
      node u ← f[i + 1]
      ALLOCATE-LINK(K_N, v, u)
ENQUEUE(Q_flows, f)
```

Figure 6: Meta-code for ALLOCATE-FLOW subroutine.

```
RELEASE-FLOW(K_N, Q_flows)
flow f ← DEQUEUE-RANDOM(Q_flows)
for i ← 1 to length[f] − 1
   do node v ← f[i]
      node u ← f[i + 1]
      RELEASE-LINK(K_N, v, u)
```

Figure 7: Meta-code for RELEASE-FLOW subroutine.

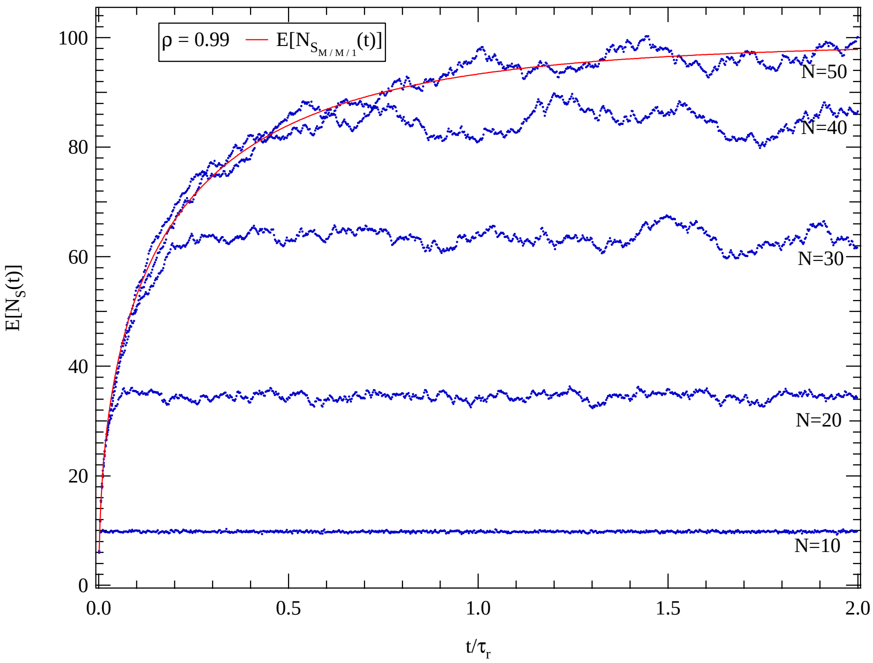


Figure 8: Simulations of the average number of flows for networks of various sizes and average number of jobs in the M/M/1 queue as a function of time, normalized with respect to the relaxation time ($\tau_r \approx 39800$). Computations of (3.1) are added for reference.

4.1 Number of flows in the network

Figure 8 shows the average number of flows as function of time, normalized with respect to the relaxation time. The relaxation time reflects the convergence rate to the stationary regime and is given for the M/M/1 queue by [11, pp. 217],

$$\tau_r = \frac{1}{\mu\left(1 - \sqrt{\rho}\right)^2} \tag{4.1}$$

The arrival and service rate are chosen as $\lambda = 0.99$ and $\mu = 1$, respectively. Hence, with $\rho = \frac{\lambda}{\mu} = 0.99$, the mean steady state system size of the M/M/1 queue is $\mathrm{E}\left[N_{S_{M/M/1}}\right] = 99$. Since the M/M/1 queue does not induce rejections, $\mathrm{E}\left[N_{S_{M/M/1}}\right]$ can be seen as an upper-bound for the average number of flows in the network; which is also evident from (3.3). Figure 8 shows that the average number of flows tends towards the upper-bound for $N = 50$. Networks of more than 50 nodes have therefore not been considered with this load. Choosing ρ closer to 1 will increase the variance of the average throughput and will lead to large fluctuations in the flow arrivals. Moreover, an extremely high load may result in numerical instability.

For $N = 50$ the average number of flows in Figure 8 closely follows equation (3.1). The fluctuations around the average number of flows in steady state increase as the network size grows, which agrees with (3.3) and (3.4). Smaller networks lack resources to accommodate each request, resulting in early rejections. To get an impression of the magnitude of the variance as compared to the mean number of flows in the M/M/1/K queue, the average system size as a function of K has been plotted in Figure 9 with error bars ($\pm$ one standard deviation). Due to the high load, the standard deviation nearly equals

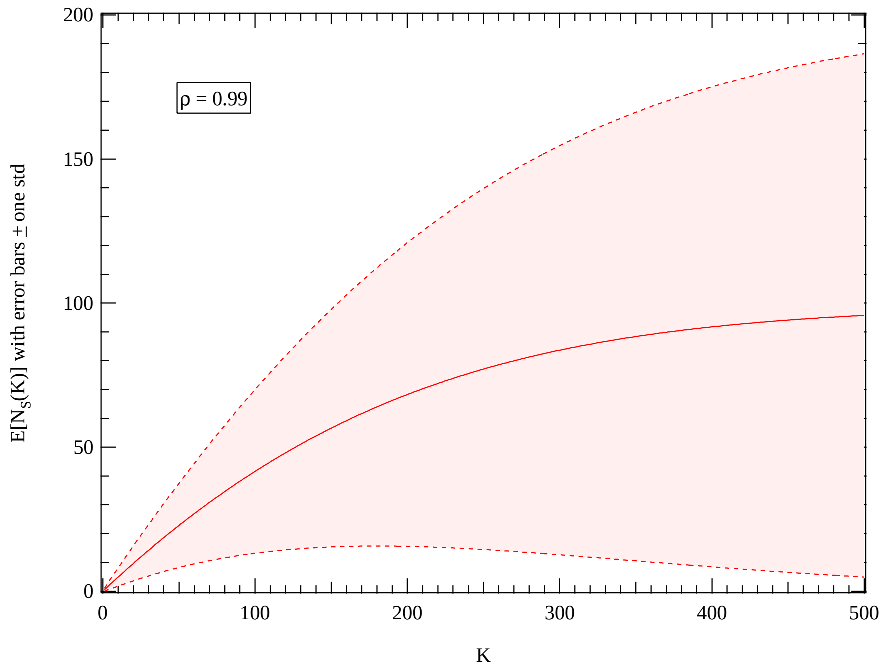


Figure 9: Computations of the mean system size of the M/M/1/K queue as a function of K with error bars ($\pm$ one standard deviation).

the average system size.

To find a relation between the network size N and buffer size K, we have computed the sample mean of the number of flows in the steady state from Figure 8 over an interval of 10^4 arrivals. The mean has been plot against the total number of links $L_{total} = \binom{N}{2}$ in Figure 10. The values for $N = 1$ and $N = 2$ are obtained analytically: for $N = 1$, the network does not have links and the average number of flows will be zero. For $N = 2$ the network contains a single link, which is either occupied or not. This can be modeled by a two state continuous time Markov chain, where the probability rate from the free to the occupied state is λ and the reverse transition occurs with rate μ. The steady state probability that the link is occupied is found [11, pp. 195–196] as $\frac{\lambda}{\lambda+\mu} \triangleq \frac{\rho}{1+\rho}$. This probability is precisely the long run average occupation time of this link. Since only one flow is allowed to travel over that link, it is also equal to the average number of flows on this link. The sample mean $\mathrm{E}\left[N_S(N)\right]$ has been fitted with (3.3) where $K = \beta\binom{N}{2}$. The best value for $\beta = 0.42$ (see the legend of Figure 10). Figure 10 shows that (3.3) agrees remarkably well with the simulations. The spread of the samples that occurs for larger N is due to the high variance, as mentioned previously. A remarkable result is $K = \beta\binom{N}{2}$: the linearity between the network capacity and the number of links in K_N. The number of links that effectively contributes to the capacity appears constant at some 42 percent of the total number of links, independent of the network size. Figure 11 illustrates some examples where (3.5) with $K = \beta\binom{N}{2}$ has been applied to the results in Figure 8 for various N.

To examine to what extend the network capacity is influenced by the traffic load, we have simulated networks of $10, 20, 30$ and 40 nodes and traffic load ranging from 0.85 to 0.99. Figure 12 compares the simulation results with computations of (3.3) using corresponding load and buffer sizes. Figure 12 points out that the network capacity is not dependent on the load, i.e. β and consequently K are not

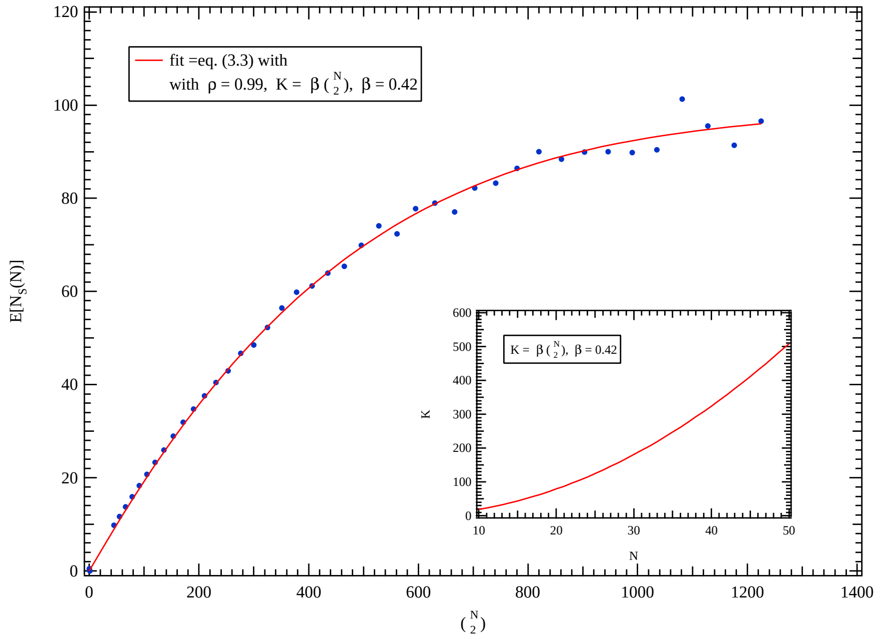


Figure 10: The average number of flows in steady state as a function of the network size $\binom{N}{2}$ (see also Figure 8). The simulation (dots) has been fitted with (3.3) (line). The inset shows the relation between the number of nodes and the network capacity $K = \beta\binom{N}{2}$.

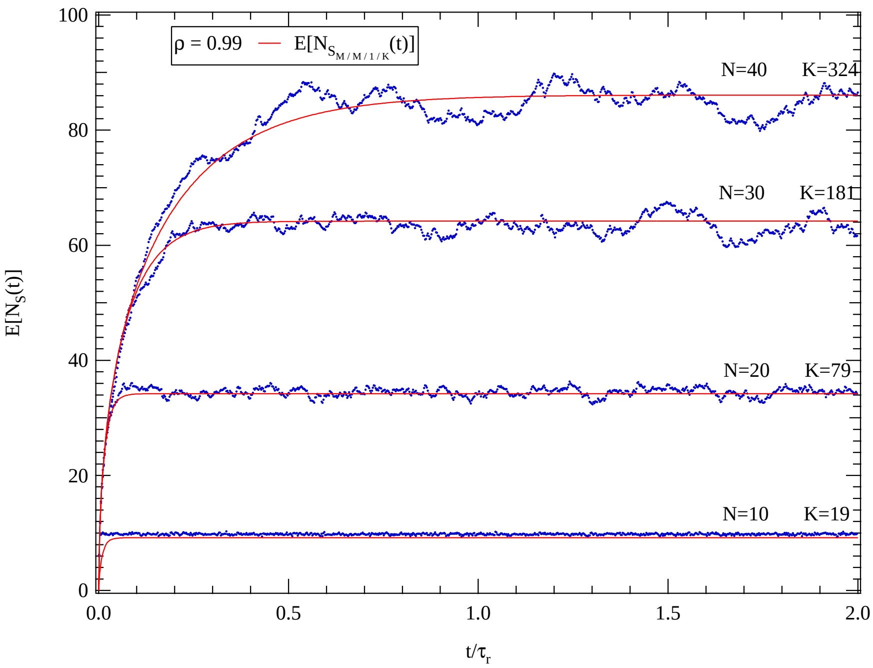


Figure 11: The average number of flows in the network as a function of time, compared with computations of (3.5).

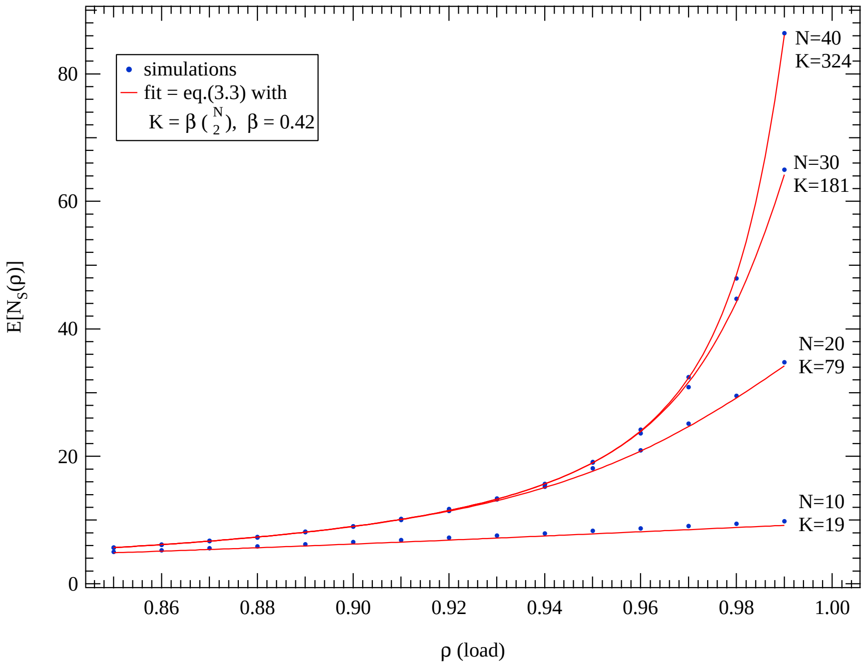


Figure 12: The average number of flows in steady state as a function of the traffic load for networks of various sizes. The simulation results (dots) are compared with computations of (3.3) (line) using $K = 0.42\binom{N}{2}$.

dependent on the traffic load. Moreover, the network capacity appears to be fixed for a given topology and the average number of flows is in accordance with queueing theory.

4.2 Hopcount

The hopcount is the number of links (hops) a flow must traverse to reach a destination from a given source. It is a random variable that depends on many variables in the network, e.g. the number of nodes, the link density and the topological structure. The average hopcount indicates how many links are allocated per flow on average. In our model, the allocation of a link consumes the full capacity of that link and is analogous to the (temporal) removal of that link. Provided that the average path length (in hops) mostly spans a single link and seldom more than two, the process of allocating flows is quite well modeled by the construction of the E-R random graph, where links are removed at random with probability p from K_N. This observation suggests us to compare the topology that originates after flow allocations with the E-R random graph. The average hopcount for the E-R random graph is, for large N, approximated by [11, pp. 346],

$$\mathrm{E}\left[H_N\right] \approx 2 - p + (1-p)(1-p^2)^{N-2} \tag{4.2}$$

$$\approx 2 - p \qquad \text{for } N \to \infty \tag{4.3}$$

where we have used that $\Pr\left[H_N = 3\right] \approx \Pr\left[H_N > 2\right]$. The average hopcount of the shortest path for flows entering the network is displayed in Figure 13 as a function of time. Contrary to the average number of flows, we observe that the variance decreases for growing N. The explanation is that, when N grows, the number of routes between a particular source and destination pair increases. The

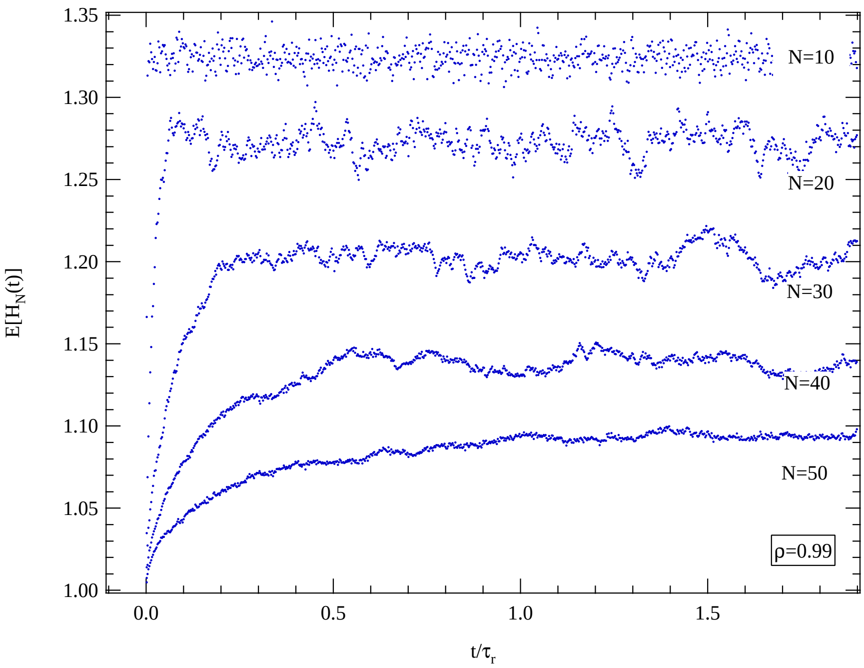


Figure 13: Average hopcount of the allocated flows as a function of time. The time axis has been normalized with respect to the relaxation time.

probability to find a short route increases with N, yielding a lower hopcount (see Figure 13) and less variance (the probability that the hopcount equals 2 or even 3 drops exponentially with N). The average hopcount in steady state, computed over an interval of 10^4 arrivals, is shown in Figure 14 as a function of N. We have made computations of (4.2) with the simulation results for the average link density from Figure 15 and added the results to Figure 14. For small N, the average hopcount is well above 1 and the probability of finding a shortest path of at least 3 hops is not negligible. Since equation (4.2) has assumed large N, the correspondence in Figure 14 improves for increasing N.

4.3 Link density

The number of allocated links in the network is directly proportional to the number of flows currently in the network. The allocation of a flow consumes the (full) capacity of the links on the path, making the links unavailable for future requests until the links are released. The ratio of allocated links by the total number of links relates to the link density as $p = 1 - \frac{\mathrm{E}[L_{allocated}]}{L_{total}}$. The number of allocated links equals the sum of the length (in hops) of all the flows in the network,

$$\mathrm{E}\left[L_{allocated}(t)\right] = \sum_{j=1}^{N_S(t)} H_j$$

where H_j equals the number of hops of flow j. After taking the expectation of both sides, invoking Wald's identity [11, pp. 34] and assuming that each H_j is i.i.d. as H_N and that $N_S(t)$ is independent from H_j, we obtain

$$\mathrm{E}\left[L_{allocated}(t)\right] = \mathrm{E}\left[N_S(t)\right] \mathrm{E}\left[H_N\right]$$

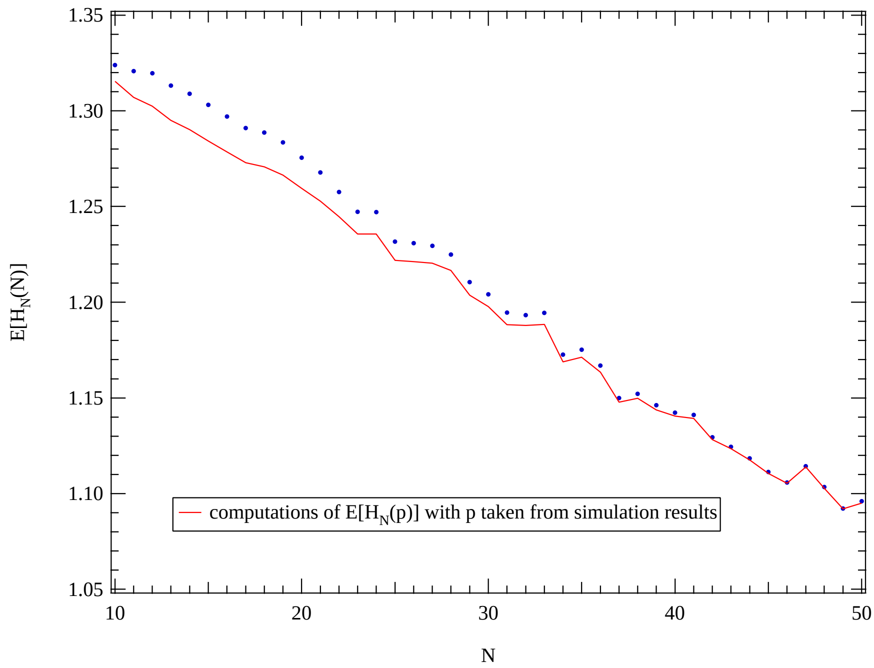


Figure 14: Simulation results of the average hopcount of the shortest path between any two nodes for networks of various sizes during steady state (dots). Computations of (4.2) have been added where the link density is used from the results in Figure 15 (line).

Since $L_{available} = L_{total} - L_{allocated}$ and $\mathrm{E}\left[L_{available}\right] = pL_{total}$, the link density becomes

$$p = 1 - \frac{\mathrm{E}\left[N_S\right]\mathrm{E}\left[H_N\right]}{L_{total}}$$

Using the average hopcount (4.3), the steady state link density converts to

$$p \approx 2 - \frac{L_{total}}{L_{total} - \mathrm{E}\left[N_S\right]} \tag{4.4}$$

Figure 15 shows the link density as a function of the number of nodes in steady state. Computations of (4.4) have been added using $\mathrm{E}[N_S]$ data from Figure 10. Figure 15 illustrates that the first order estimate (4.4) agrees well. Equation (4.4) slightly overestimates the simulations for small N, which is caused by disregarding the probability of paths longer than 2 hops and the assumption that H_j is independent of $N_S(t)$. As the size N of the network grows, the probability of longer paths decreases: Figure 15 shows that (4.4) better matches the simulations.

The link density is important as it inherently determines the blocking probability within the network. If the link density reaches beyond a critical value, the graph is almost surely disconnected and virtually each flow is blocked. Comparing the network with the E-R random graph, facilitates to express the critical link density p_c as a function of the network size N: $p_c(N) \approx \frac{1}{N}\log N$. Computations of the critical link density have been added to Figure 15.

Although the average link density is well above the critical threshold, the network still experiences blocking. Figure 16 explains this phenomenon. For a single sample path, i.e. a single network realization, the link density is plotted on the left axis and the number of rejected requests on the right

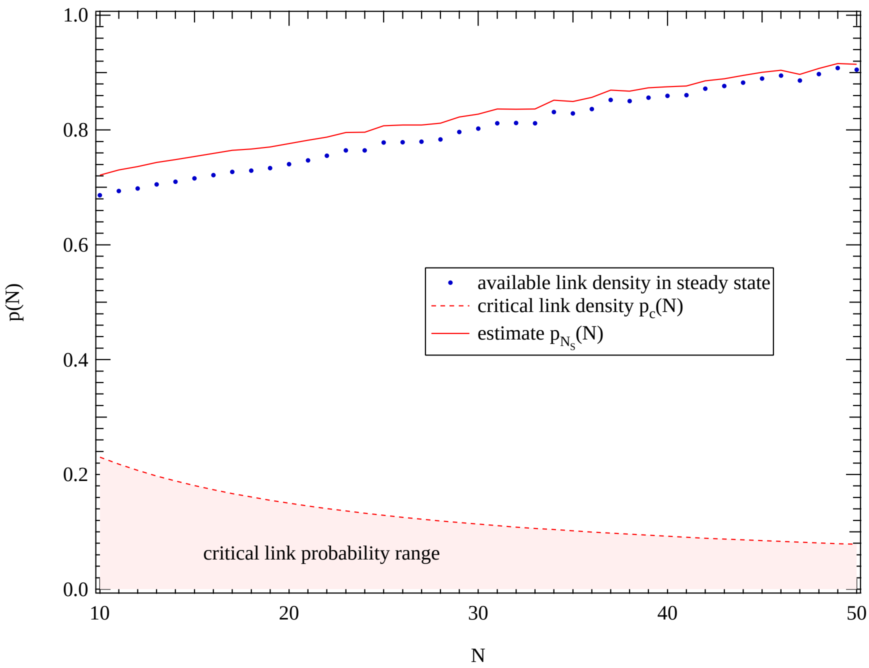


Figure 15: The link density in steady state. The simulations (dots) are compared with computations (solid line) of (4.4) using $\mathrm{E}[N_S]$ results from Figure 10. The link density threshold for the E-R random graph p_c has been added to the figure. Below $p_c(N)$ the E-R random graph is a.s. disconnected.

axis as a function of time. Figure 16 shows that the average link density is well above the random graph's critical density. However, rejections still take place due to the high variations in the traffic. A few short periods of strongly increased arrivals contribute most to the rejections by the network. This observation implies that, beside the mean, also the tail probabilities are crucial.

In addition to the link density, we have also monitored the degree $D(t)$ and degree distribution $\Pr[D(t) = k]$ during the same simulation as that in Figure 16. At each time step, the degree is sampled for a different, arbitrary node. The result is displayed in Figure 17. The right-hand side of Figure 17 shows $D(t)$, while the left-hand side presents the probability distribution function of $D(t)$ (on lin-log-scale).

4.4 Blocking probability

When a flow is rejected by the network, the network lacks resources (capacity) to accommodate the flow. With respect to the M/M/1/K model, this is analogous to the situation where the buffer is full. Hence, the rejection rate in the network can be approximated by (3.2). The average rejection rate r in steady state is displayed in Figure 18. The average rejection rate has been computed by first computing the ratio of average rejected requests by total requests as a function of time and then averaging over an interval of 10^4 arrivals during steady state. Figure 18 also shows computations of (3.2) with $\rho = 0.99$, $K = \beta\binom{N}{2}$ and $\beta = 0.42$.

Figure 18 confirms that the M/M/1/K queue is a suitable model to clarify network rejections. The good match again underlines the importance of the relationship $K = \beta\binom{N}{2}$ between the network capacity and the number of links in the network, as explained in the previous section.

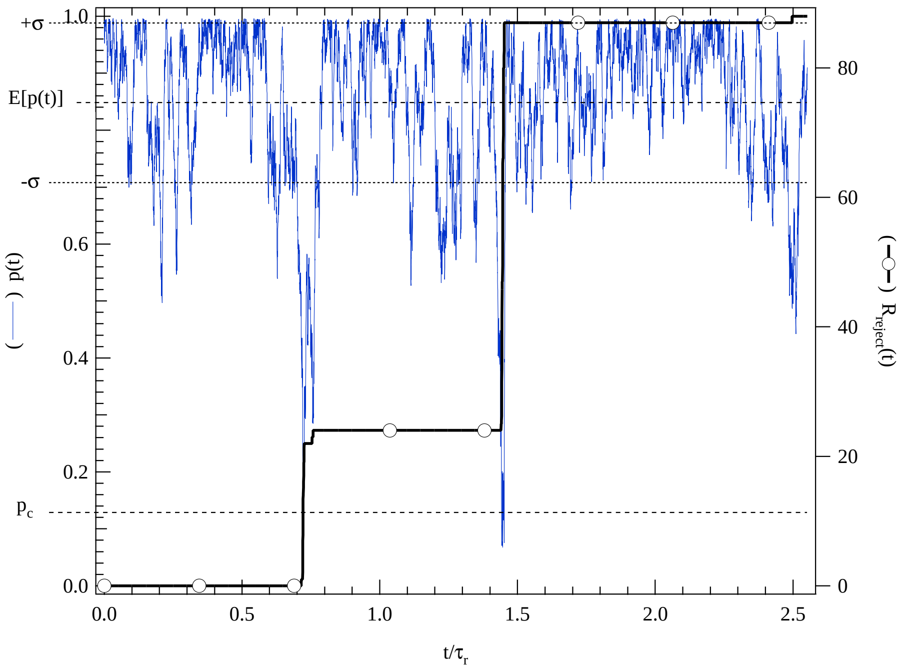


Figure 16: The link density and number of rejected requests as a function of time, for a single sample path, where $N = 25$ and $\rho = 0.99$. The upper dashed lines indicate the sampled mean $\mathrm{E}[p(t)] \pm \sigma$ (one standard deviation). The lower dashed line illustrates p_c for the E-R random graph with $N = 25$.

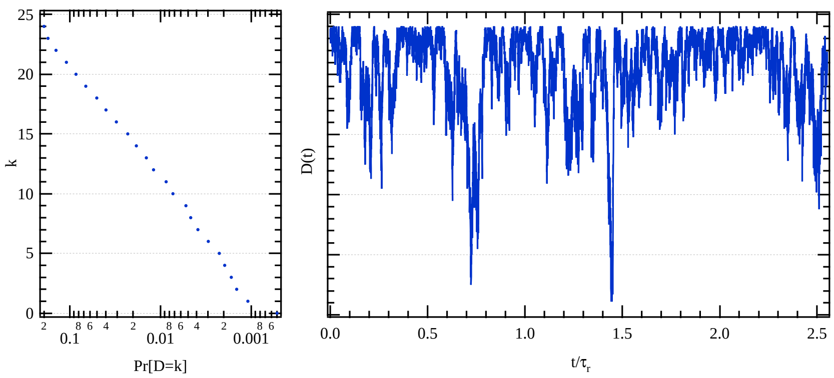


Figure 17: On the right-hand side, the evolution of the nodal degree $D(t)$ in a network of 25 nodes. At each time step, the degree of a random node is sampled. On the left-hand side, the probability distribution of $D(t)$ (on lin-log scale) is plotted.

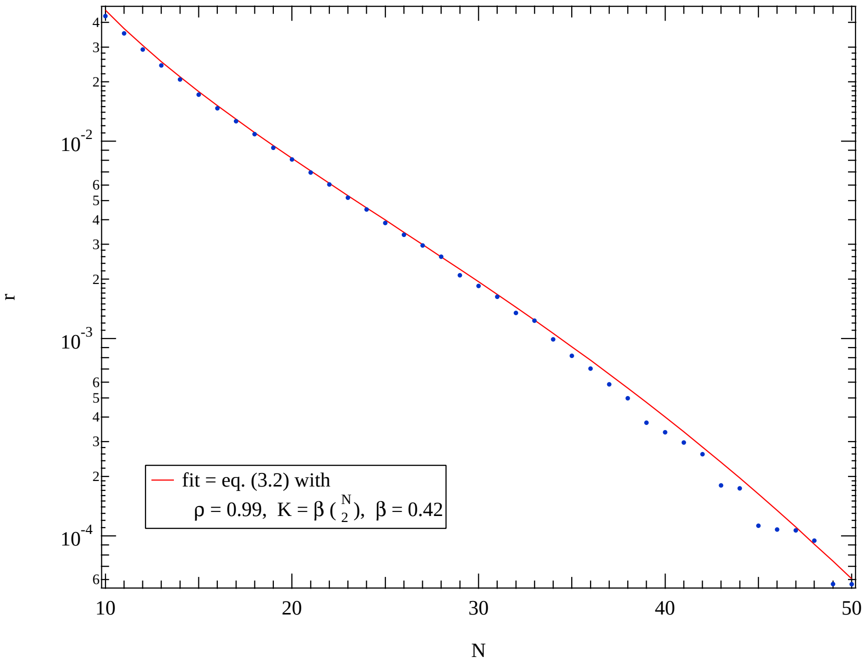


Figure 18: Simulation results of the average rejection rate in steady state as a function of the network size (dots). Computations of (3.2) have been added, with $K = \beta\binom{N}{2}, \beta = 0.42$ (line).

4.5 Lifetime

In Figure 19 the flow lifetime distribution is given for stationary networks. The average lifetime for the various networks has been added in square brackets. Figure 19 reveals that the lifetime distribution is not exponential, which is explained by the service process. The flows are serviced in random order and as a result, the probability a particular flow is served next, decreases with the number of flows in the network. Hence, we see that the lifetime distribution exhibits a longer tail as compared to the M/M/1 FIFO queue [2].

4.6 Conclusions

A general conclusion is that the M/M/1/K queue appears to be suitable to model the network behavior. The (transient) solutions for the M/M/1/K queue can be applied to describe network behavior and provide insight into the loading of the network as well as blocking. A new finding is the linear relation $K = \beta\binom{N}{2}$ between the number of links in the network and the buffer capacity K. The origin of the scaling factor β and the sensitivity to network scaling, e.g. increasing the number of channels per link so that more flows are allowed per link, deserve further investigation. The meaning of β can be explained as the "efficiency" with which the links are allocated. By allocating links, parts of the network become mutually unreachable, compromising the maximum number of sustainable flows. Additionally, we can conjecture that the network capacity K (and therefore also β) is not determined by the traffic load. The network capacity in K_N seems embedded in the topological properties and does not change with the traffic load.

Furthermore, we have observed that the average hopcount can be explained if we model the network

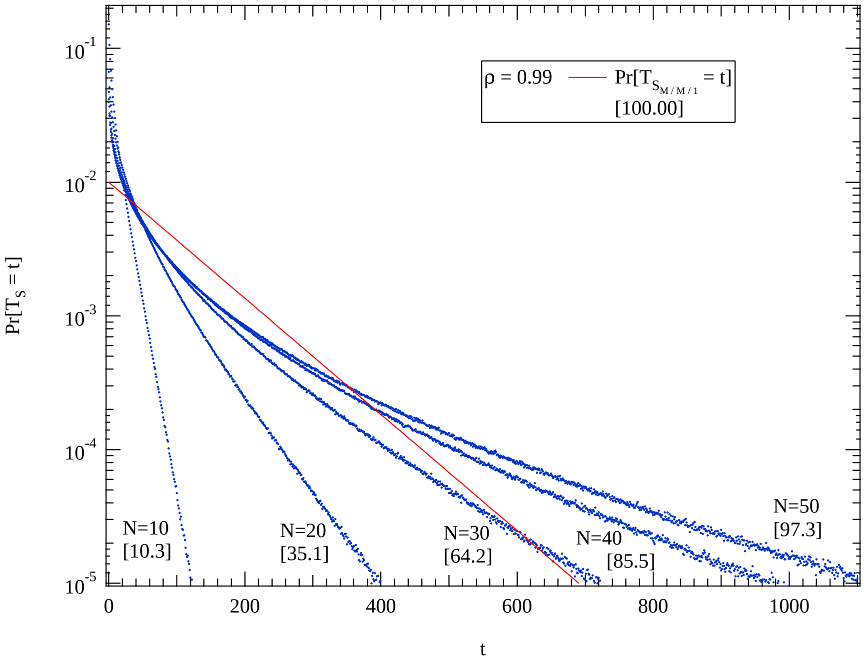


Figure 19: Lifetime distribution for various networks in steady state. The system time distribution for the M/M/1 FIFO queue is added as reference. The mean lifetimes for the corresponding network size and the system time for the M/M/1 queue are shown in square brackets.

as an E-R random graph. Open challenges in this perspective are studies on how this assumption will hold when we consider other topological structures than the full mesh as initial network. Finally, we conclude by noting that the traffic fluctuations are indeed of importance when evaluating and modeling the network performance. When only the mean is considered, important aspects of network dynamics are missed as illustrated in Figure 9 and Figure 16. Even though the mean link density would plead for negligible blocking probability, the burstiness of the traffic may still induce rejections.

5 Network Capacity Approximation and Degree Distribution

In Section 4 we have studied the capacity of a network with continuously arriving and departing flows. In this section, the average number of flows that can be allocated in a network before rejection occurs, is addressed. Additionally, we are interested in the effect of removing paths in the graph. More precise, we analyze the evolution of the connectivity and the rejection rate as flows are allocated in the network. Starting from the complete graph with unit link weights, we choose two nodes uniformly and compute the shortest path using Dijkstra's algorithm. The main difference with the previous sections is that the path is allocated for an infinite duration, which is equivalent to removing the path's links from the graph. We repeat the process of choosing nodes and removing links until the graph becomes disconnected and the simulation ends. The graph after removal of j paths is denoted by $\hat{G}(j, N)$. Figure 20 presents the meta-code of the subsequent steps to construct $\hat{G}(j, N)$.

Our aim is to compute the average number of paths that is allocated before $\hat{G}(j, N)$ becomes disconnected. Additionally, the degree distribution and average hopcount are studied and compared with the Erdös-Rényi random graph $G_p(N)$. The degree distribution of $G_p(N)$ is the binomial distribution with mean $p(N-1)$. The average hopcount of $G_p(N)$ for $p > p_c$, approximated by (4.2), can be written as

$$\mathrm{E}\left[H(p)\right] \geq 2 - p + \epsilon \tag{5.1}$$

where (5.1) follows from (4.2) for large N and ϵ is the correction for ignoring paths of more than 2 hops.

In the initial phase of the evolution of $\hat{G}(j,N)$, the average length of the computed paths equals one with high probability and single links are removed independently from the graphs. Hence, $\hat{G}(j,N)$ agrees precisely with $G_p(N)$. The agreement with $G_p(N)$ facilitates to find an upper-bound for the link density in $\hat{G}(j,N)$ as a function of j and the total initial number of links L_{total}. By definition of the link density $p[j]$, the average number of links of $\hat{G}(j,N)$ is $\mathrm{E}\left[L[j]\right] = L_{total}p[j]$. The initial conditions for $p[j]$ are $p[0] = 1$ and $p[1] = 1 - \frac{1}{L_{total}}$. Using (5.1) with $\epsilon = 0$, the average number of links at stage $j+1$ can be written as

$$\begin{aligned} \mathrm{E}\left[L[j+1]\right] &= L_{total}p[j] - \mathrm{E}\left[H\left(p[j]\right)\right] \\ &\leq L_{total}p[j] - \left(2 - p[j]\right) \end{aligned} \tag{5.2}$$

This difference equation converts to

$$p[j+1] = \left(1 + \frac{1}{L_{total}}\right)p[j] - \frac{2}{L_{total}} \tag{5.3}$$

Solving (5.3) yields

$$p[j] \leq 2 - \left(1 + \frac{1}{L_{total}}\right)^j \tag{5.4}$$

Equation (5.4) forms a tight upper-bound for the true link density, because it follows from (5.1) that $\mathrm{E}\left[H(p[j])\right]_{\epsilon=0} > \mathrm{E}\left[H(p[j])\right]_{\epsilon>0}$ and consequently from (5.2) it follows that $p[j]_{\epsilon=0} \geq p[j]_{\epsilon>0}, \forall j$.

5.1 Simulations

For various N ranging from 25 to 100 we compare (5.4) with simulations. Figure 21 shows that $\hat{G}(j,N)$ agrees very well with $G_p(N)$ and (5.4) perfectly matches the simulations, implying a strong resemblance between $\hat{G}(j,N)$ and $G_p(N)$. The effect of removing paths, instead of links, becomes

INITIALIZE(K_N)
while $\hat{G}(j,N)$ is connected
  do node $v \leftarrow$ RANDOM-NODE($\hat{G}(j,N)$)
    node $u \leftarrow$ RANDOM-NODE($\hat{G}(j,N)$)
    path $P \leftarrow$ DIJKSTRA($\hat{G}(j,N), v, u$)
    remove P from $\hat{G}$.

Figure 20: Meta-code for the construction of $\hat{G}(j,N)$.

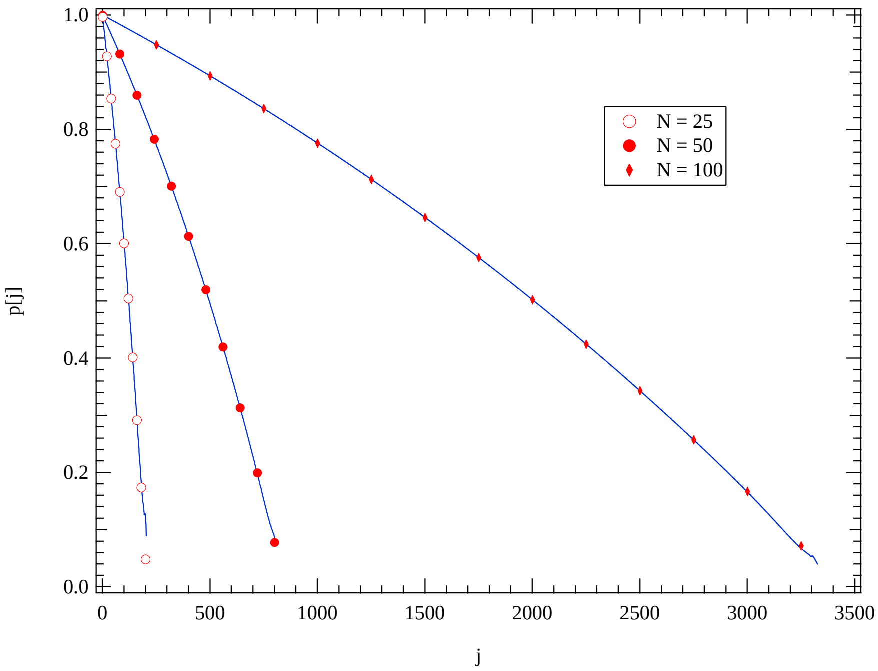


Figure 21: Effective link probability for 25, 50 and 100 node graphs after allocating j paths. Simulations (lines) are compared with (5.4) (markers).

visible when we consider the connectivity threshold of $\hat{G}(j, N)$. The maximum number of paths, j_{max}, that can be removed from $\hat{G}(j, N)$ before becoming disconnected, is a stochastic variable and may differ for each realization of $\hat{G}(j, N)$. The link density threshold $p[j_{max}]$ then corresponds to the link density at which $\hat{G}(j_{max}, N)$ is precisely disconnected. We have plotted $\Pr[j_{max} = x]$ in Figure 22 and fitted the result with the Gumbel distribution [11, pp. 54],

$$\Pr[X = x] = \frac{1}{a} \exp\left(\frac{x-b}{a}\right) \exp\left(-\exp\left(\frac{x-b}{a}\right)\right) \tag{5.5}$$

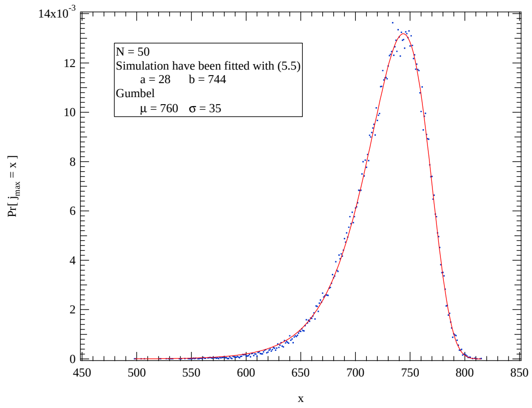
where a and b are referred to as the "scale" and "location" parameter, respectively. The Gumbel distribution fits very well. The mean and standard deviation for the Gumbel distribution equal $\mu = b + 0.5772a$ and $\sigma = \frac{a\pi}{\sqrt{6}}$, respectively. From the fits in Figure 22 we can deduce that the standard deviation is very small as compared to mean and therefore the mean $E[j_{max}]$ sufficiently describes the random variable j_{max}.

Since $\Pr[j_{max} = x]$ very likely follows the Gumbel distribution and while using (5.4), which is a monotonously decreasing function for j, we can write the following,

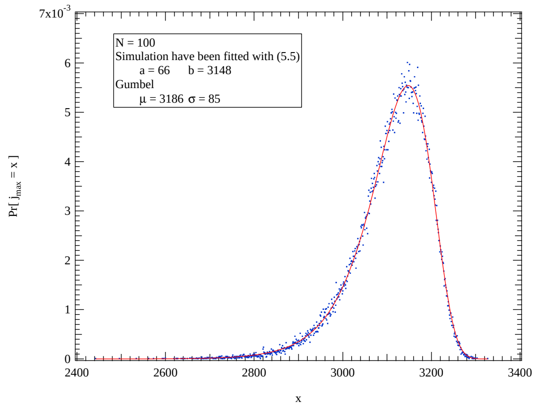
$$\Pr[p[j_{max}] \leq x] = 1 - \Pr[j_{max} \leq y(x)] = \exp\left(-\exp\left(\frac{y(x)-b}{a}\right)\right) \tag{5.6}$$

where $y(x) = p^{-1}(x)$ is the inverse of (5.4), thus $y(x) = \frac{\ln(2-x)}{\ln\left(1+\frac{1}{L}\right)}$. Equation (5.6) then becomes,

$$\Pr[p[j_{max}] \leq x] = \exp\left(-\exp\left(-\frac{b}{a}\right)(2-x)^{\frac{1}{a\ln\left(1+\frac{1}{L}\right)}}\right) \tag{5.7}$$



(a) $N = 50$



(b) $N = 100$

Figure 22: This figure shows the probability distribution function $\Pr[j_{max} = x]$ of $\hat{G}(j, N)$. The simulations (dots) have been fitted with the Gumbel distribution from equation (5.5) (line).

Apart from a shift, the probability density function (5.7) follows the Weibull-distribution [11, pp. 55]. The result for $\Pr[p[j_{max}] = x]$ is shown in Figure 23. By definition of j_{max}, there holds that

$$\Pr[p[j_{max}] \leq x] = \Pr\left[\hat{G}(j, N) \text{ is connected at } p[j] = x\right] \tag{5.8}$$

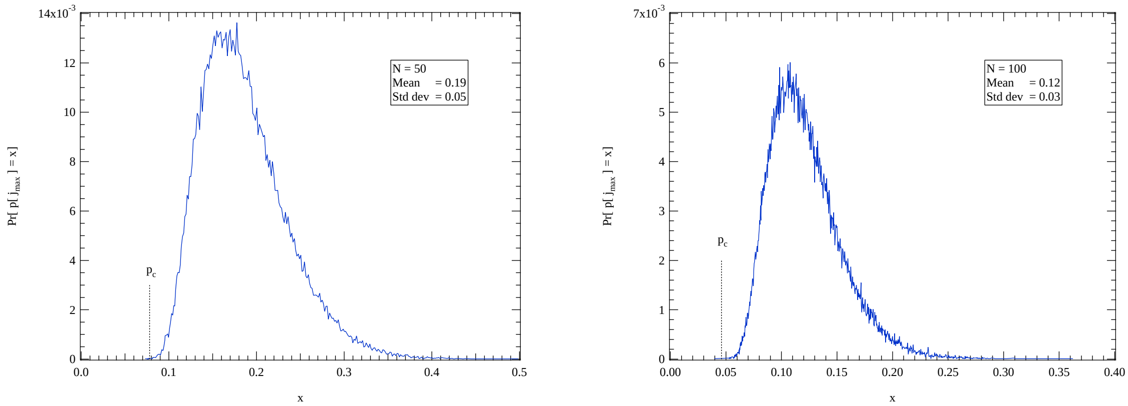
Figure 24 compares (5.8) with the connectivity of $G_p(N)$ as a function of the link density. Computations of $\Pr[G_p(N) \text{ is connected}]$ are obtained from [11, pp. 334–337],

$$\begin{aligned} \Pr[G_p(N) \text{ is connected}] &\approx \left(1 - (1-p)^{N-1}\right)^N \\ &\approx \exp^{-N \exp^{-p(N-1)}} \end{aligned} \tag{5.9}$$

The dependencies between the links have a clear effect on the connectivity threshold of $\hat{G}(j, N)$. Not only is the critical density higher, the transition region is also wider. Figure 24 can be explained when considering the nodal degree. Revisiting [11, pp. 337] we see that $\Pr[\text{G is connected}] = \Pr[D_{min} \geq 1]$ a.s., where $D_{min} = \min_{\text{all nodes} \in G} D$ is the minimal nodal degree in G.

In Figure 25 the degree distribution is shown for $N = 50$ and various j. Results for $N = 25$ and $N = 100$ are placed in Section B of the Appendix. Figure 25 reveals that the degree distribution increasingly deviates from the binomial distribution of $G_p(N)$ as more paths are allocated. The degree distribution has a higher variance as a result of the correlation that exists between links when removing multi-hop paths from $\hat{G}(j, N)$ (as opposed to the removal of single, independent link in the case of $G_p(N)$). The links removed in $\hat{G}(j, N)$ are correlated because they belong to the same path. The degree of intermediate nodes along the path is decreased by two, while the source and destination nodes only loose a single neighbor. Nodes are not treated identically and consequently dependencies are introduced and aggravated during the removal process. The increased variance implies a higher probability of disconnectedness of $\hat{G}(j, N)$ as compared to $G_p(N)$ at equal average degree value. Since the degree distribution is wider, the transition region in Figure 24 is also wider.

Using the average link density $p[j]$ from Figure 21, the computations of expected hopcount (4.2) are compared with simulation results of $\hat{G}(j, N)$. The result is displayed in Figure 26. The dispersion in the tails in Figure 26 stems from the stochastic nature of the hopcount. The number of samples that



(a) $N = 50$. The link density threshold for a 50-node E-R random graph is $p_c \approx 0.08$.

(b) $N = 100$. The link density threshold for a 100-node E-R random graph is $p_c \approx 0.05$.

Figure 23: This figure shows the probability distribution function $\Pr[p[j_{max}] = x]$ of the link density at which the graph $\hat{G}(j, N)$ becomes disconnected due to the removal of paths. Figure 24 shows the cumulative distribution function $\Pr[p[j_{max}] \leq x]$.

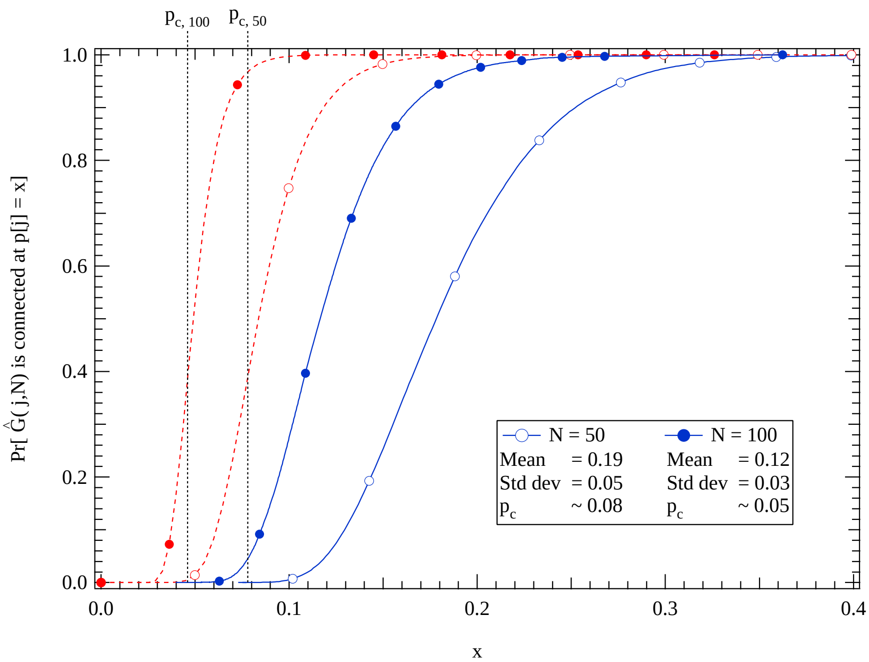


Figure 24: $\Pr\left[\hat{G}(j, N) \text{ is connected at } p[j] = x\right]$ for $N = 50$ and $N = 100$ (solid lines). As a reference, computations of (5.9) have been added (dashed lines) for corresponding network sizes.

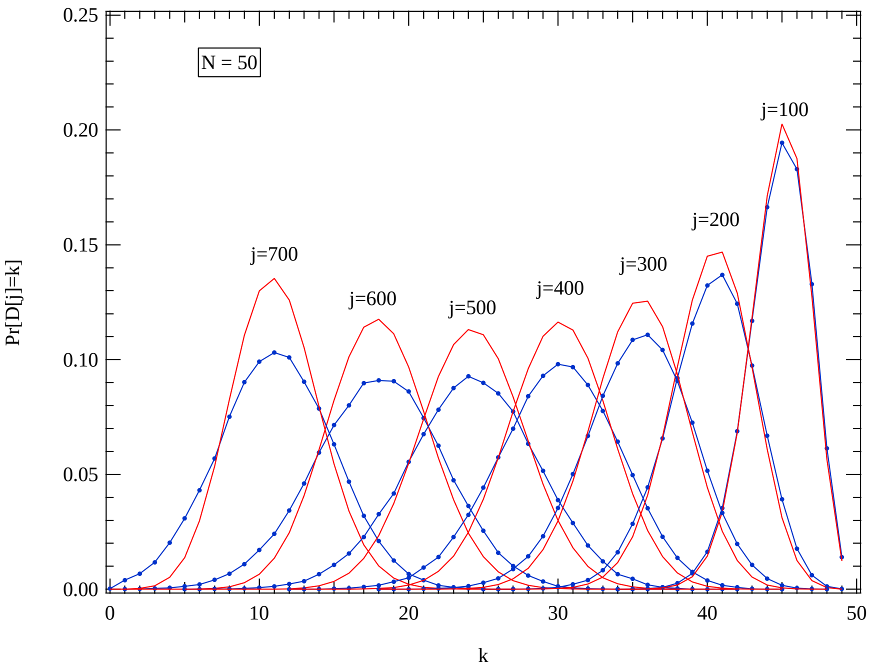


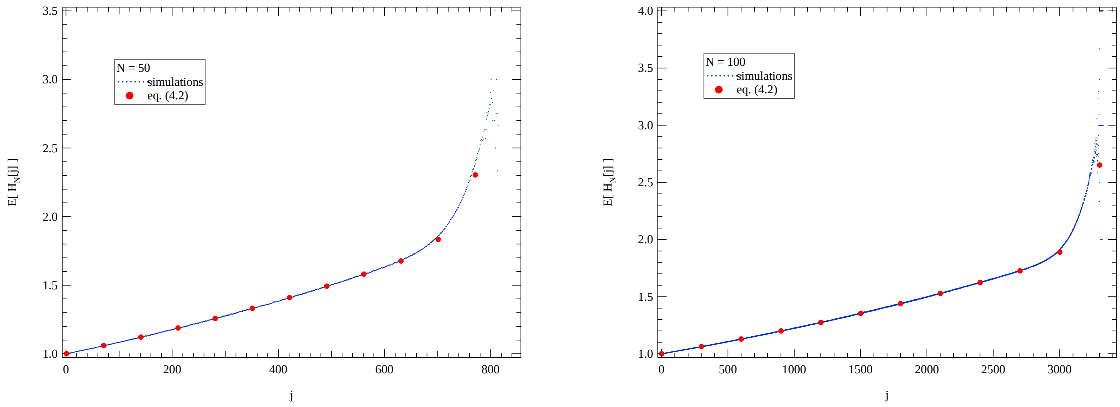
Figure 25: Degree distribution of $\hat{G}(j, N)$ for $N = 50$ and various j. As a reference, the degree distribution for the random graph has been added for each j with mean identical to the simulation. The dotted lines are simulation results, the normal lines are computations of the binomial distribution.

is available to compute the mean hopcount is determined by j_{max}. Because j_{max} is a random variable, less samples are available at the tail, yielding a higher variance. The increase of the mean hopcount is evident from (4.2); as more paths are allocated, the number of remaining links are decreased and the probability of finding a direct link, or a short path, between source and destination decreases. We realize that the degree distribution alone is not a sufficient measure to fully investigate graph behavior, yet it gives insight into important issues as connectivity. Section 6 directly relates clustering, rejection and connectivity through the graph's average degree. Section A will elaborate on the dependence of the degree distribution and gives an analytic approximation for a similar process as that in Figure 25.

6 Rejection Rate and Disconnectedness

The previous Section 5 has focussed on connectivity of $\hat{G}(j, N)$ and its relation with the nodal degree. In this section we study the blocking probability of $\hat{G}(j, N)$ in more detail. The graph $\hat{G}(j, N)$ is defined as the graph that evolves after j random, minimum-hop paths have been removed from the complete graph K_N. In addition to $\hat{G}(j, N)$, we define $G^*(p[j], N)$ as the graph that evolves after an arbitrary number of random, minimum-hop paths have been removed, such that the final link density is p. Figure 27 presents the meta-code for the construction of $G^*(p[j], N)$. The difference between $\hat{G}(j, N)$ and $G^*(p[j], N)$ is that the latter does not require connectedness.

When a graph is disconnected, it is fragmented into groups of nodes, which are referred to as "clusters". For two nodes to be connected in any graph, they must belong to the same cluster. If the



(a) $N = 50$ (b) $N = 100$

Figure 26: The average hopcount of $\hat{G}(j, N)$ as function of j compared with computations of (4.2).

nodes are uniformly chosen, then the probability that one belongs to a cluster with size N', equals

$$\Pr[\text{A node belongs to a cluster of } N' \text{ nodes}] = S = \frac{N'}{N} \tag{6.1}$$

The probability that both nodes belong to the same cluster, is S^2. Hence, the probability that a source and destination node are not connected equals the rejection rate,

$$\begin{aligned} r &= 1 - \Pr[\text{Source node in } N']\Pr[\text{Destination node in } N'] \\ &= 1 - S^2 \end{aligned} \tag{6.2}$$

Clustering in the E-R random graph has been studied in detail by Janson *et al.* [6]. The transition from a group of solitaire nodes to a connected cluster is very steep for increasing link density. The majority of the nodes belongs to a single cluster, which is called the "giant component" (GC). The remaining nodes are clustered in groups of size $O(\log N)$. The size of GC (S_{GC}) in the E-R random graph, is found as (see [12]),

$$S_{GC}(\overline{D}_{G_p(N)}) = 1 - e^{-\overline{D}_{G_p(N)}} \sum_{n=0}^{\infty} \frac{(n+1)^n}{(n+1)!} (\overline{D}_{G_p(N)} e^{-\overline{D}_{G_p(N)}})^n \tag{6.3}$$

where $\overline{D}_{G_p(N)} = p(N-1)$ is the mean nodal degree of $G_p(N)$. For N sufficiently large, we can assume that two random nodes in the network are connected if they belong to GC.

```
INITIALIZE(K_N)
while linkdensity[G*(p[j], N)] > p
  do node v ← RANDOM-NODE(G*(p[j], N))
     node u ← RANDOM-NODE(G*(p[j], N))
     path P ← DIJKSTRA(G*(p[j], N), v, u)
     if length[P] > 0
        remove path P from G*(p[j], N).
```

Figure 27: Meta-code for the construction of $G^*(p[j], N)$.

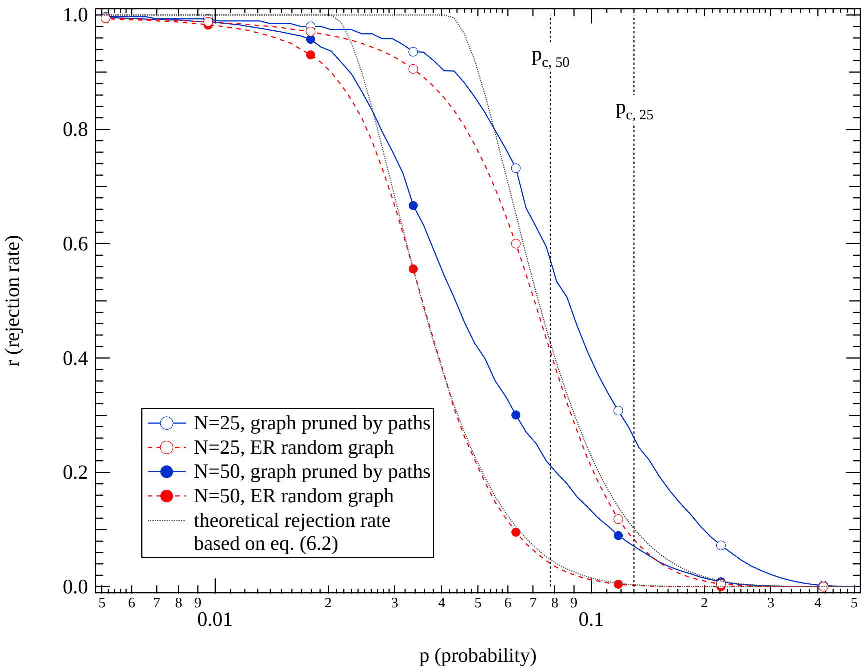


Figure 28: Average blocking probability compared with computations of (6.2). The dashed lines are results for the E-R random graph, while the solid lines show the simulation results for $G^*(p[j], N)$. In addition, computations of (6.2) and density threshold values for the E-R random graph have been added.

6.1 Simulations

We computed the blocking probability for $G_p(N)$ and $G^*(p[j], N)$ for $N = 25$ and $N = 50$ through simulations. We have generated 10^6 different realizations of $G_p(N)$ and $G^*(p[j], N)$ for each combination of N and p. For each realization we tried to route a path between uniformly chosen source and destination nodes. The ratio of average failed attempts by the total number of attempts gives us the rejection rate,

$$r = \frac{\mathrm{E}\left[\sum \text{failed attempts}\right]}{\sum \text{total attempts}}$$

The results are combined in Figure 28, which also contains computations of (6.2), where the cluster size is computed through (6.2). The results in Figure 28 are in conformance with the results in the previous Section 5. The rejection rate of $G^*(p[j], N)$ is higher than $G_p(N)$, which directly implies that $G^*(p[j], N)$ becomes disconnected at higher values of p. At an early stage in the evolution of $G^*(p[j], N)$, a solitary node disconnects from the graph such that rejection already takes place at higher density values. The explanation for this can be derived from Figure 25, as has been done previously in Section 5.1. When the link density further decreases, the giant component steadily reduces as small groups of nodes detach from the giant cluster, causing a gradual increase in the rejection rate of $G^*(p[j], N)$. Finally, for extreme low link densities, the giant cluster in $G^*(p[j], N)$ has vanished and the remaining clusters are similar to those of $G_p(N)$. Likewise, the rejection rate of $G^*(p[j], N)$ converges with $G_p(N)$. In addition, Figure 28 reveals that the rejection rate is still reasonably low at p_c, which is explained by the fact that nodes are still connected as long as they

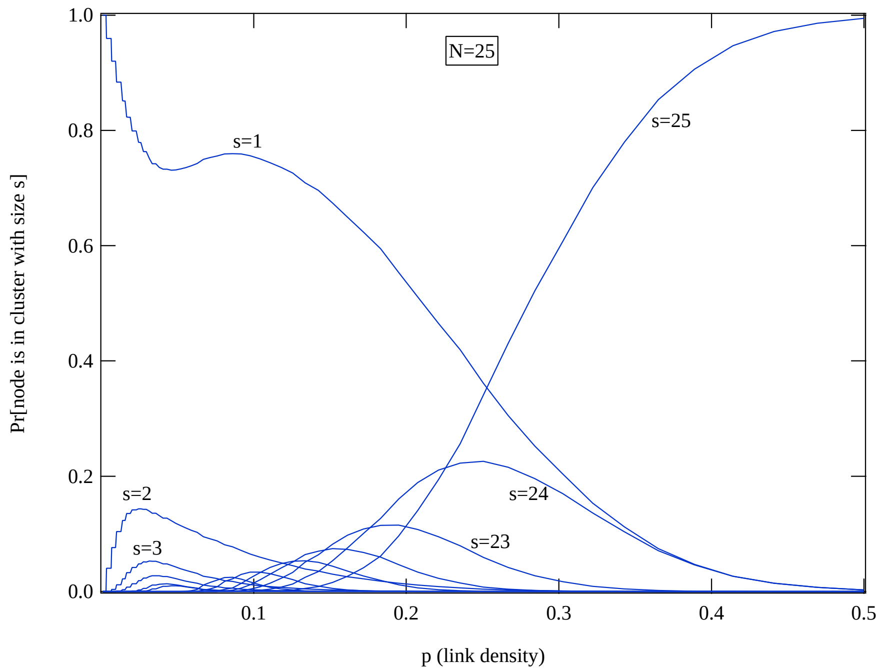


Figure 29: The probability that a node belongs to a cluster of size s is plotted against the link density for $G^*(p[j], N)$, where $N = 25$. Several traces for different s have been omitted from the figure to enhance the readability.

reside within the same cluster. Figure 28 also shows that the phase transition for the E-R random graph is more abrupt than for $G^*(p[j], N)$. The transition region is more narrow due to the smaller variance of the nodal degree distribution of $G_p(N)$ as compared to $G^*(p[j], N)$. To get a better understanding of the clustering occurring in the network, we have plotted the clustersize distribution in $G^*(p[j], N)$ at various link densities in Figure 29. In Figure 29 we have plotted the evolution of the clustering process for a network of 25 nodes. Each line shows the portion of nodes that belong to a cluster of size s. Several lines have been omitted for clarity. Figure 29 confirms our assumption that the network becomes disconnected because of some solitary node instead of the formation of two large clusters. Steadily, the giant cluster reduces and more isolated nodes and small clusters emerge. Until finally the giant cluster has vanished and the network is divided into a multitude of clusters and solitary nodes.

7 Conclusions

The results from Section 4 show that modeling the network by the M/M/1/K queue provides remarkable insights into the relation between the "network capacity" and the number of nodes. For the scenarios we have studied, we have found the linear relation $K = \beta\binom{N}{2}$. The value of β is the outcome of various network properties and the interaction of processes during the network operation, e.g. the routing process, the selection of source and destination nodes and of course topological properties. Simulations with different traffic intensities suggest that β seems not affected by the traffic load. We can regard β as the "efficiency" with which the network can be utilized. With $\beta \approx 0.42$, the maximum number of links that is used during operation is only 42 percent of the total.

Another interpretation on the "network capacity" has been introduced in Sections 5 and 6, where we have studied the number of flows we can accommodate in a network before rejection occurs or before the network becomes disconnected. We have found that the maximum number of flows that can be allocated in a connected network is distributed according a Gumbel distribution (5.5). The maximum number of flows that can be allocated, is a counting process of the number of allocated paths. The allocation of paths can be considered as a sequence of i.i.d. random variables, such that the distribution follows the Gumbel distribution [11, pp. 107]. The link density of the network just before disconnectedness then follows a Weibull distribution.

The removal of paths from the network results in a wider degree distribution as compared the E-R random graph. The difference is introduced by the correlation that exists between the links of each path. The degree of intermediate nodes in a path, is decreased by two, while the degree of the endpoints is only decreased by one. The nodes are treated differently and their distribution increasingly deviates from the binomial distribution of the E-R random graph. The increased variance in the degree distribution manifests itself when considering the rejection rate and connectivity: the transition from no rejection (full connectivity) to full rejection (no connectivity) as a function of the link density is less step for $G^*(p[j], N)$ as compared to $G_p(N)$.

References

[1] J. Abate and W. Whitt. Calculating time-dependent performance measures for the M/M/1 queue. *IEEE Transactions on Communications*, 37(10):1102–1104, Oct 1989.

[2] L. Flatto. The waiting time distribution for the random order service M/M/1 queue. *The Annals of Applied Probability*, 7(3):382–409, 1997.

[3] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, Aug 2001.

[4] G. Grimmett. *Percolation*. Springer-Verlag, New York, 1989.

[5] ITU-T. Recommendation E.800. *Terms and Definitions Related to Quality of Service and Network Performance including Dependability*, Aug 1994.

[6] S. Janson, D. E. Knuth, T. Luczak, and B. Pittel. The birth of the giant component. *Random Structures Algorithms 4*, 3:231–358, 1993.

[7] O. P. Sharma. *Markovian Queues*. Series in Mathematics and its Applications. Ellis Horwood, 1990.

[8] O. P. Sharma and A. M. K. Tarabia. A simple transient analysis of an M/M/1/N queue. *Sankhyā: The Indian Journal of Statistics*, 62(A, Pt. 2):273–281, 2000.

[9] L. Takács. *Introduction to the Theory of Queues*. Oxford University Press, New York, 1962.

[10] A. M. K. Tarabia. Transient analysis of an M/M/1/N queue - an alternative approach. *Tamkang Journal of Science and Engineering*, 3(4):263–266, 2000.

[11] P. Van Mieghem. *Performance Analysis of Communications Networks and Systems*. Cambridge University Press, 2006.

[12] P. Van Mieghem and S. van Langen. Influence of the link weight structure on the shortest path. *Physical Review E*, 71(056113), May 2005.

A The Degree Distribution in K_N after Removing Links

In this section we examine the effect of removing correlated links on the nodal degree distribution. To understand why a small deviation of the E-R model may lead to large differences observed in Section 5, we assume that *any path between two random nodes consists of two links.* We consider the process where, at each stage j, precisely two links of an arbitrary node in the complete graph K_N are removed. We compute the degree $D[j]$ of an arbitrary node at stage j in the thinned complete graph K_N. From that process, we derive a recursion relation

$$D[j] = D[j-1]\,1_{\text{no link removed}} + (D[j-1]-1)\,1_{\text{1 link removed}} + (D[j-1]-2)\,1_{\text{2 links removed}} \tag{A.1}$$

and $D[0] = N-1$. Ignoring the boundary restrictions[1] when $D[j]=1$ and $D[j]=0$, the probability density function of $D[j]$ obeys the equation

$$\begin{aligned}\Pr[D[j]=k] = {} & \Pr[D[j-1]=k]\Pr[\text{no link removed}] + \\ & + \Pr[D[j-1]=k+1]\Pr[\text{1 link removed}] + \\ & + \Pr[D[j-1]=k+2]\Pr[\text{2 links removed}]\end{aligned}$$

[1] These seriously complicate the analysis and prevent the derivation of analytic expressions

with $p_1 = \Pr[1 \text{ link removed}] = \frac{2}{N}$, $p_2 = \Pr[2 \text{ links removed}] = \frac{1}{N}$ and $\Pr[\text{no link removed}] = 1 - \frac{3}{N}$. The corresponding probability generating function

$$\varphi_{D[j]}(z) = \sum_{k=0}^{N-1} \Pr[D[j] = k] z^k$$

obeys the functional equation

$$\varphi_{D[j]}(z) = \left(1 - \frac{3}{N}\right) \varphi_{D[j-1]}(z) + \frac{2}{N} \sum_{k=0}^{N-1} \Pr[D[j-1] = k+1] z^k + \frac{1}{N} \sum_{k=0}^{N-1} \Pr[D[j-1] = k+2] z^k$$

After simplification, we obtain

$$\varphi_{D[j]}(z) = \left(\left(1 - \frac{3}{N}\right) + \frac{2}{Nz} + \frac{1}{Nz^2}\right) \varphi_{D[j-1]}(z) - \left(\frac{2}{Nz} + \frac{1}{Nz^2}\right) \varphi_{D[j-1]}(0) - \frac{\Pr[D[j-1] = 1]}{Nz^2}$$

Since $\varphi_{D[j]}(1) = 1$, it follows that both $\Pr[D[j-1] = 1] = \varphi'_{D[j-1]}(0) = 0$ and $\Pr[D[j-1] = 0] = \varphi_{D[j-1]}(0) = 0$. The functional equation reduces to

$$\varphi_{D[j]}(z) = \frac{(N-3) z^2 + 2z + 1}{Nz^2} \varphi_{D[j-1]}(z)$$

With the initial condition $\varphi_{D[0]}(z) = E\left[z^{D[0]}\right] = z^{N-1}$, the solution is

$$\varphi_{D[j]}(z) = \left(\frac{(N-3) z^2 + 2z + 1}{Nz^2}\right)^j z^{N-1} \tag{A.2}$$

The mean and the variance of $D[j]$ are most efficiently computed from $L_{D[j]}(z) = \log \varphi_{D[j]}(z)$ as (see [11])

$$\begin{aligned} E[D[j]] &= L'_{D[j]}(1) = N - 1 - \frac{4j}{N} \\ \text{Var}[D[j]] &= L''_{D[j]}(1) + L'_{D[j]}(1) = \frac{2j(3N-8)}{N^2} \end{aligned}$$

The mean $E[D[j]]$ follows from the general law for the degree

$$\sum_{n=1}^{N} d_n[j] = 2L[j] = 2\left(\binom{N}{2} - 2j\right) \tag{A.3}$$

and the definition of the mean $E[D[j]] = \frac{1}{N}\sum_{n=1}^{N} d_n[j]$ resulting again in $E[D[j]] = N - 1 - \frac{4j}{N}$. However, the requirement that $\varphi'_{D[j-1]}(0) = 0$ and $\varphi_{D[j-1]}(0) = 0$ implies that $N - 2 - 2j > 0$ or that $\frac{N-2}{2} > j$. The major reason for this artifact is the neglect of the boundary equations.

By expanding the probability generating function in a Taylor series around $z = 0$, the probability density function $\Pr[D[j] = k]$ can be obtained. In general, the power series of $\left(z^2 + bz + c\right)^n$ with $n \in \mathbb{N}$ is derived as,

$$\left(z^2 + bz + c\right)^n = \sum_{j=0}^{n} \binom{n}{j} c^{n-j} \left(z^2 + bz\right)^j = \sum_{j=0}^{n} \binom{n}{j} c^{n-j} \sum_{k=0}^{j} \binom{j}{k} b^{j-k} z^{k+j}$$

```
INITIALIZE(K_N).
while K_N is connected
  do node v ← RANDOM-NODE(K_N)
    link l_1 ← RANDOM-LINK(v)
    remove l_1 from K_N
    if degree[v] > 0
      then link l_2 ← RANDOM-LINK(v)
        remove l_2 from K_N
```

Figure 30: Meta-code for simulating effect of path removal on graph properties.

Let $m = k + j$, then $0 \leq m \leq 2n$ and, from $0 \leq j = m - k \leq n$, it follows that $k \leq m$. Hence,

$$\left(z^2 + bz + c\right)^n = \sum_{m=0}^{2n} \sum_{k=0}^{m} \binom{n}{m-k} \binom{m-k}{k} c^{n-m+k} b^{m-2k} z^m \tag{A.4}$$

Using (A.4) with $b = \frac{2}{N-3}$, $c = \frac{1}{N-3}$ and $n = j$ leads to

$$\begin{aligned}
\varphi_{D[j]}(z) &= \left(\frac{N-3}{N}\right)^j \left(z^2 + \frac{2}{N-3} z + \frac{1}{N-3}\right)^j z^{N-1-2j} \\
&= \frac{1}{N^j} \sum_{m=0}^{2j} \left(2^m \sum_{l=0}^{m} \binom{j}{m-l} \binom{m-l}{l} \left(\frac{N-3}{4}\right)^l\right) z^{m+N-1-2j} \\
&= \frac{1}{N^j} \sum_{m=N-1-2j}^{N-1} \left(2^{m-N+1+2j} \sum_{l=0}^{m-N+1+2j} \binom{j}{m-N+1+2j-l} \times \right. \\
&\quad \left. \times \binom{m-N+1+2j-l}{l} \left(\frac{N-3}{4}\right)^l\right) z^m
\end{aligned}$$

from which it follows that

$$\begin{aligned}
\Pr\left[D\left[j\right] = k\right] = \frac{2^{k-N+1+2j}}{N^j} \sum_{l=0}^{k-N+1+2j} &\binom{j}{k-N+1+2j-l} \times \\
&\times \binom{k-N+1+2j-l}{l} \left(\frac{N-3}{4}\right)^l
\end{aligned} \tag{A.5}$$

Unfortunately, we cannot evaluate the above sum. We have performed simulations to test the validity of (A.5).

Figure 30 presents the meta-code of the simulation. First the complete graph K_N is initialized. While K_N remains connected, a random node is chosen uniformly and two of its links are removed from the graph. The simulation results for $N = 50$ are presented in Figure 31, which illustrates that deviations appear for increasing k due to the fact that the analysis above only applies for small $j < \frac{N-2}{2}$. Figure 31 is "similar" to Figure 25.

B Degree Distribution of $\hat{G}_j(N)$ as function of j

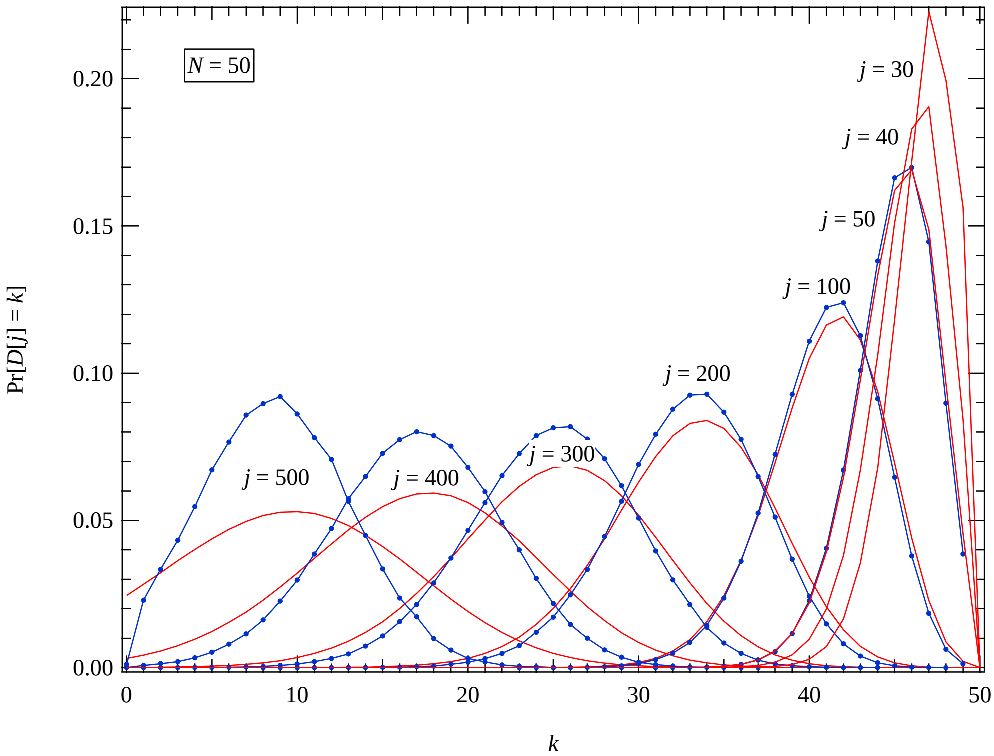
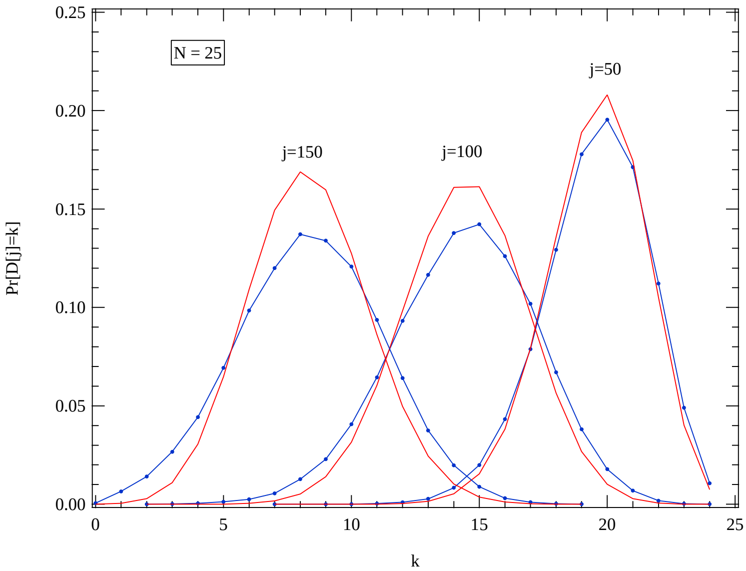
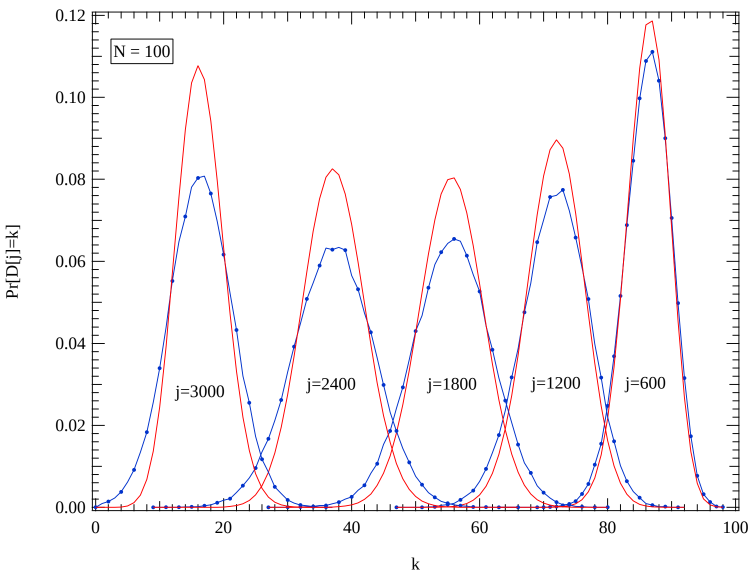


Figure 31: The probability density function of $D[j]$ for $N = 50$ and various j. The line that connects dots are simulations, while the other line are computations of (A.2).



(a) N=25



(b) N=100

Figure 32: Degree distribution of $\hat{G}(j, N)$ for various N and j. As a reference, the degree distribution for the random graph has been added for each result with mean identical to the simulation. The dotted lines are simulation results, the normal lines are computations of the E-R graph degree distribution.