Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Physica A 341 (2004) 677 – 701

PHYSICA A

www.elsevier.com/locate/physa

Distributive routing and congestion control in wireless multihop ad hoc communication networks

Ingmar Glauche[a,b,c], Wolfram Krause[a,d], Rudolf Sollacher[a], Martin Greiner[a,*]

[a] *Corporate Technology, Information and Communications, Siemens AG, D-81730 München, Germany*

[b] *Institut für Theoretische Physik, Technische Universität Dresden, D-01062 Dresden, Germany*

[c] *Institut für Medizinische Informatik, Statistik und Epidemiologie, Universität Leipzig, Liebigstr. 27, D-04103 Leipzig, Germany*

[d] *Institut für Theoretische Physik, Johann Wolfgang Goethe-Universität Frankfurt, Postfach 11 19 32, D-60054 Frankfurt am Main, Germany*

Received 25 February 2004

Abstract

Due to their inherent complexity, engineered wireless multihop ad hoc communication networks represent a technological challenge. Having no mastering infrastructure the nodes have to selforganize themselves in such a way that for example network connectivity, good data traffic performance and robustness are guaranteed. In this contribution the focus is on routing and congestion control. First, random data traffic along shortest path routes is studied by simulations as well as theoretical modeling. Measures of congestion like end-to-end time delay and relaxation times are given. A scaling law of the average time delay with respect to network size is revealed and found to depend on the underlying network topology. In the second step, a distributive routing and congestion control is proposed. Each node locally propagates its routing cost estimates and information about its congestion state to its neighbors, which then update their respective cost estimates. This allows for a flexible adaptation of end-to-end routes to the overall congestion

[*] Corresponding author. Institut für Medizinische Informatik, Statistik und Epidemiologie, Universität Leipzig, Liebigstr. 27, D-04103 Leipzig, Germany.

E-mail addresses: ingmar.glauche@imise.uni-leipzig.de (I. Glauche), krause@th.physik.uni-frankfurt.de (W. Krause), rudolf.sollacher@siemens.com (R. Sollacher), martin.greiner@siemens.com (M. Greiner).

doi:10.1016/j.physa.2004.04.126

state of the network. Compared to shortest-path routing, the critical network load is significantly increased.

PACS: 05.10.Ln; 05.65.+b; 84.40.Ua; 89.20.−a; 89.75.Fb

Keywords: Dynamics on complex networks; Information and communication technology; Wireless multihop ad hoc networks; Data traffic; Distributive congestion control

1. Introduction

In two previous Papers [1,2] we have already discussed the so-called wireless multihop ad hoc networks. They represent an engineered communication network, which reveals many facets of very intriguing and complex behavior. In this respect they fit nicely into the cross-disciplinary realm of the Statistical Physics of complex networks [3–5], which has already opened its doors for other communication networks like the Internet, but also for biological and social networks.

Wireless multihop ad hoc networks represent an infrastructureless peer-to-peer generalization of todays wireless cellular phone networks. Instead of being slaved to a central control authority, each node not only sends or receives packets, but also forwards them for others. Consequently, communication packets hop via inbetween ad hoc nodes to connect the initial sender to the final recipient. A lot of coordination amongst the nodes is needed for the overall network to perform well. They have to ensure network connectivity, good data-traffic performance and robustness against various forms of perturbations, just to name but the most important issues. Because of this intrinsic coordination, wireless multihop ad hoc networks represent an excellent example of what is called a selforganizing network. However, their biggest challenge is yet to come, how to get selforganization to work.

The connectivity issue has already been discussed quite extensively [1,6–9], also addressing interference effects [10,11]. In one form or the other all these efforts relate to continuum percolation [12–14]. An interesting distributive scheme has been put forward in Ref. [1], which turned out to be amazingly robust, guaranteeing strong network connectivity almost surely; we will briefly touch upon this scheme again in Section 2.1. The robustness issue with respect to selfish users has received inspirations from the biological immune system and distributive algorithmic suggestions have been put forward [15].

As to data-traffic performance, estimates on the throughput, i.e., the capacity of how much end-to-end traffic the network is able to handle without overloading, have been given. In Ref. [16] a rigorous upper bound has been derived to scale with the square root of the network size. Refined estimates have been given in [2], revealing that the scalability of the throughput depends on the underlying network structure. Besides several other idealistic assumptions, these estimates have employed shortest-path routing. Although several proactive and reactive routing schemes have already been discussed

[17–20], we are not aware of any selfoptimizing scheme, which also accounts for congestion avoidance.

In this paper we will propose a prototype for such a highly wanted distributive and adaptive routing and congestion control. The idea is that every node keeps an estimate of how much it costs to send packets to final destinations and to update these estimates in a distributive manner. The latter can be achieved in a very elegant way without any additional exchange of control information. Whenever a node is actively involved in a forwarding one-hop transmission, it silences its neighbors anyhow, so that those do not interfere on the shared wireless propagation medium. This blocking is called medium access control. Upon blocking its neighbors, the node is able to distribute its routing cost estimates and its congestion state to them, which those then use to update their cost estimates. In the technical jargon of engineers, this distributive scheme corresponds to a coupling of the medium access control layer with the routing layer.

The structure of this paper is as follows. Section 2 summarizes the key operational features of wireless multihop ad hoc networks, introduces plausible simplifications and describes the setup of generic simulations with random data traffic. Shortest-path routing is used in Section 3 to investigate certain fingerprints of congestion like end-to-end time delay and single-node relaxation times. Whenever possible, the numerical simulations are accompanied with analytic modeling. Section 4 presents the details of the proposed distributive routing and congestion control and compares its results to those obtained with the shortest-path routing. The conclusion and a short outlook are given in Section 5.

2. Some basics on wireless multihop ad hoc networks

We explain the key features of wireless multihop ad hoc networks hand in hand with some simplifications.

2.1. Geometric ad hoc graphs

The first simplification is to neglect mobility and to distribute N nodes onto the unit square in a random and homogeneous way. Then, according to a simple isotropic propagation-receiver model, a unidirectional link from node i to node j exists, if

$$\frac{P_i/R_{ij}^\alpha}{\mathrm{NOISE} + \sum_{\mathrm{active}k} P_k/R_{kj}^\alpha} \geqslant \mathrm{SNR}\,. \tag{1}$$

P_i denotes the transmission power of node i and R_{ij} represents the Euclidean distance between i and j. The path-loss exponent α is assumed to be constant. Without any loss of generality the variables NOISE and SNR are set equal to one. Condition (1) guarantees that j is able to listen to i, i.e., $i \to j$. Throughout this paper we will neglect the interference sum over other active nodes k in the denominator of Eq. (1); this is justified once α is not too close to 2 [11].

With these simplifications, wireless communication networks can be modeled as graphs $\mathscr{G} = (\mathscr{N}, \mathscr{L})$, where $\mathscr{N}$ refers to the set of nodes and $\mathscr{L}$ to the set of links.

In general these links in $\mathscr{L}$ are directional links $i \to j$. The subset $\mathscr{L}^{\text{bidir}} \subset \mathscr{L}$ represents the complete set of all bidirectional links $i \leftrightarrow j$. Although not strictly required, bidirectional links are preferred for the operation of wireless ad hoc networks because many communication protocols require instant feedback. The subset $\mathscr{N}_i \subset \mathscr{N}$ is called the communication neighborhood of node i and represents the complete set of nodes $j \in \mathscr{N}_i$ that all have bidirectional links $j \leftrightarrow i$ in $\mathscr{L}^{\text{bidir}}$ with node i. The node degree k_i of node i is the number of nodes contained in $\mathscr{N}_i$. In a similar fashion $\mathscr{N}_i^{\text{out}}$ defines the set of nodes that have at least an unidirectional link $i \to j$ from i. A communication route or path is a sequence of nodes such that there are bidirectional links in $\mathscr{L}^{\text{bidir}}$ between all consecutive pairs of nodes. A shortest path between two nodes $i, f \in \mathscr{N}$ is a route containing fewest possible number of nodes. The average length of all shortest paths over all node pairs i, f is referred to as the diameter D of the network.

One further step is needed in order to fully specify wireless multihop ad hoc network graphs: assignment of the transmission power P_i for all nodes. Most widely used is the constant transmission power rule, where the same transmission power $P_i = P$ is assigned to all nodes $i \in \mathscr{N}$ [6,8,16]. All existing links are then bidirectional. Once the transmission power is chosen such that $P = (k_{\text{target}}/\pi N)^{\alpha/2}$, an average node will have k_{target} bidirectional neighbors. This target degree has to be $k_{\text{target}} \geqslant 4.52$ for a connected giant component to exist independent of the network size [14], but for the entire network to be strongly connected it needs to be larger [1,7,9]. We adopt the value $k_{\text{target}} = 24$, which guarantees strong network connectivity almost surely for network sizes up to several thousands [1]. We will call wireless multihop ad hoc network graphs generated with this power assignment as const-P networks.

A different power assignment has been presented in Ref. [1], which is more energy efficient. It is based on a distributive assignment. In a nutshell, each node i forces the $k_{\min}$ closest nodes j to adjust their transmission powers to $P_j = R_{ij}^{\alpha}$, while adopting the value $P_i = \sup_j P_j$ for itself. Its own value can be increased further whenever another close-by node forces i in return to have an even larger transmission power. In this respect each node has at least $k_{\min}$ bidirectional neighbors. We adopt the value $k_{\min} = 8$, which guarantees strong network connectivity almost surely for network sizes up to several thousand nodes [1]. We will call wireless multihop ad hoc network graphs generated with this heterogeneous power assignment as minimum-node-degree networks.

Fig. 1 illustrates two random geometric graphs, one obtained with the const-P assignment and the other with the minimum-node-degree assignment. We explicitly point out that the existence of links in $\mathscr{L}$ is a direct result of the spatial positions and the transmission powers of the nodes. In that sense not every possible set of links $\mathscr{L}$ can be realized by an appropriate power assignment. This is in clear contrast to any wired network where such restrictions do not influence the existence of connections between different nodes.

2.2. Generic data traffic

In order to study the statistical properties of data traffic on wireless multihop ad hoc networks, the generic simulation model as already presented in Ref. [2] has been

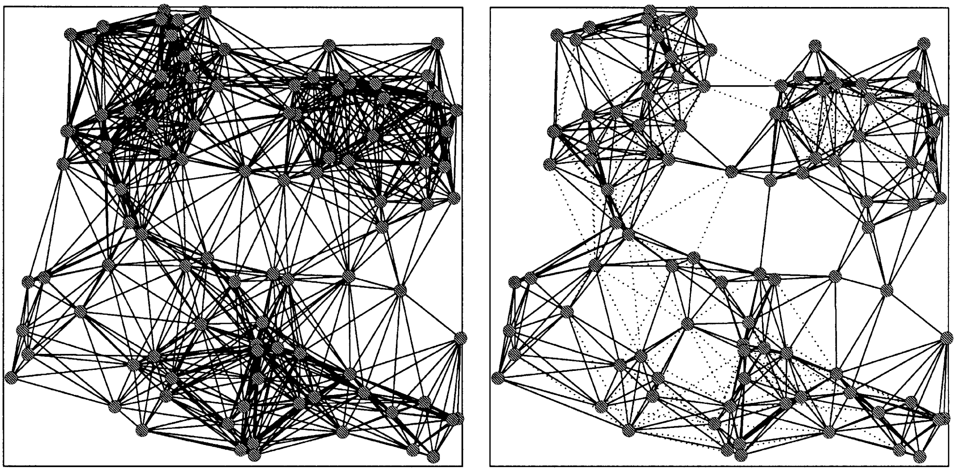

Fig. 1. Random geometric wireless multihop ad hoc graphs obtained with (left) const-P and (right) minimum-node-degree transmission power assignment. $N = 100$ nodes have been randomly and homogeneously distributed onto a unit square. Solid/dotted links are bi-/unidirectional.

applied. For the sake of illustration of the key mechanisms, we now give again a short outline. The simulation is based on discrete time steps. At the very beginning of a time step a new data packet of fixed size can be generated at each node $i \in \mathcal{N}$ with a probability $\mu_i = \mu < 1$, which is also referred to as the packet creation rate. In case of creation at a certain node i, a destination f is randomly chosen among $\{\mathcal{N} \setminus i\}$ and the packet is put at the end of i's buffer queue, assumed to have infinite capacity. Nodes, for which a new packet has been created, are blocked and are not involved in any further communication action for the remainder of this time step. During a short contention phase following the packet creation phase the non-blocked nodes with a non-zero queue compete for gaining sender status. A competing node i is randomly picked first and obtains permission to transmit its first-in-line packet. It then makes a decision, to which neighbor $j \in \mathcal{N}_i$ the packet with final destination f is forwarded. In its simplest form, this could be shortest-path routing or, in a more sophisticated form, routing depending on the congestion state. In order to reduce mutual interference within the shared communication medium, the sending as well as receiving node block their respective outgoing neighbors $\{(N_i^{\mathrm{out}} \setminus j) \cup (\mathcal{N}_j^{\mathrm{out}} \setminus i)\}$ for the remainder of this time step; this blocking is called medium access control (MAC). Only then another node with non-zero queue that has not been blocked so far is chosen at random to attempt the transmission of its first-in-line packet. If the intended receiver has already been blocked before, the node tests its second-in-line packet and so on, until either the first idle recipient is found or the end of its queue is reached. This service discipline is denoted as first-in-first-possible-out. If this node succeeds to gain sender status, it then MAC-blocks again its remaining outgoing neighbors as well as those of the receiving node. This iteration is repeated until no free one-hop transmissions are left. Finally, all nodes with sending permission then submit their selected packet and remove it from their queue. The receiving nodes either add the incoming packet to the end of their queue or, if they are the final recipient, destroy the packet.

3. Fixed shortest-path routing: properties of data traffic

A particular simple form of routing uses shortest paths with respect to the hop-metric. For the network to learn about the shortest routes all by itself, each node is required to flood discovery information into the whole network, collect the feedback and store those routes with the shortest hop distance. In principle degeneracy might occur, which would allow to pick the least-congested shortest path between initial sender i and final recipient, f as a modest form of congestion control; we will come back to this in Section 4.1. For all of this section we prefer to discard degeneracy, pick one of the shortest degenerate paths at random and forward all packets originating in i and destined for f along it. This restriction allows for several analytical insights and points to the specific needs for improvement which a more sophisticated routing and congestion control has to take care of.

3.1. *End-to-end time delay I: simulation results*

We define the end-to-end (e2e) time delay of a packet to be the number of time steps between the generation at the originating node and the destruction at the final receiver. Its temporal and network-ensemble average provides a measure of the network performance. The direct sampling of end-to-end times within the generic data traffic simulations implies a tagging of each packet with its creation time and is needed to extract respective distributions. Although average e2e-time delays can also be determined via this route, an indirect sampling procedure is more efficient in that case. In the subcritical regime $\mu < \mu_{\text{crit}}$ the average number μN of packets created within the overall network per time step must be equal to the number of packets delivered per unit time. Since the average time a packet spends in the network is $\langle t_{\text{e2e}} \rangle$ we can assume that $\langle N_{\text{active}} \rangle / \langle t_{\text{e2e}} \rangle$ packets are delivered to their final destination per unit time, where $\langle N_{\text{active}} \rangle = \langle \sum_i n_i \rangle$ represents the average total number of active packets after a stationary network state has been reached. This leads to Little's Law,

$$\frac{\langle N_{\text{active}} \rangle}{\langle t_{\text{e2e}} \rangle} = \mu N \ , \tag{2}$$

well known in queuing theory [21]. Since the network size N and the packet creation rate μ are known to us and $\langle N_{\text{active}} \rangle$ is easily sampled, Little's Law allows for the indirect determination of the average end-to-end time delay.

Fig. 2 illustrates the average e2e-time delay as a function of the packet creation rate μ. As expected, it increases with the network load μ and diverges at the critical packet creation rate μ_{crit}, which is a clear phase-transition-like sign that the system is about to leave its uncongested subcritical state and to enter its congested supercritical state for $\mu > \mu_{\text{crit}}$. We also observe that μ_{crit} is different for the const-P and the minimum-node-degree networks; see Ref. [2] for more details on the dependence of the end-to-end throughput $\mu_{\text{crit}} N$ on the underlying network structure.

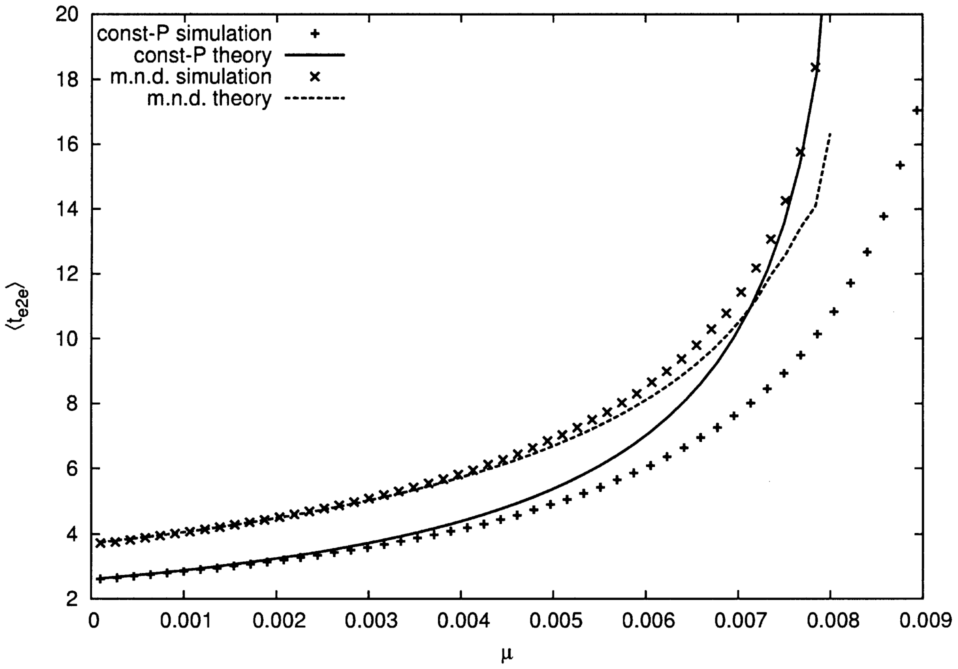

Fig. 2. Sample- and node-averaged end-to-end time delay $t_{\mathrm{e2e}}(\mu)$ determined from generic data traffic simulations (symboled curves) and the analytic estimate using Eqs. (2), (7), (10) and (11) (curves without symbols). The network size has been fixed to $N=100$. The two transmission power assignments are const-P (vertical crosses) with $k_{\mathrm{target}}=24$ and minimum-node-degree (rotated crosses) with $k_{\min}=8$.

In the limit $\mu \to 0$ the average e2e-time delay converges to the network diameter $\langle t_{\mathrm{e2e}}\rangle \to D$. Although this observation is intuitively clear, we will give an analytic support in the next subsection. Due to the differing network diameter, it appears that with respect to e2e-time delay the const-P networks perform better than the minimum-node-degree networks. Note however, that their respective parameters k_{target} and $k_{\min}$ have been chosen from the connectivity perspective only. A larger $k_{\min}$ would lower the network diameter for the minimum-node-degree networks. In Section 3.3 we will present another form of performance comparison between the two network models.

The distribution $p(t_{\mathrm{e2e}})$ of the e2e-time delay obtained from the generic data-traffic simulation with fixed shortest-path routing is shown in Fig. 3. It is a network-wide distribution, which has been sampled over all generated packets. The employed const-P network realization consists of $N=100$ nodes. Safely inside the subcritical phase ($\mu=0.005$) the sampled distribution can be fitted well with a generalized exponential

$$p_{\mathrm{exponential}}(t_{\mathrm{e2e}})=\frac{1}{b}\,\mathrm{e}^{-(t_{\mathrm{e2e}}-a)/b}\ . \tag{3}$$

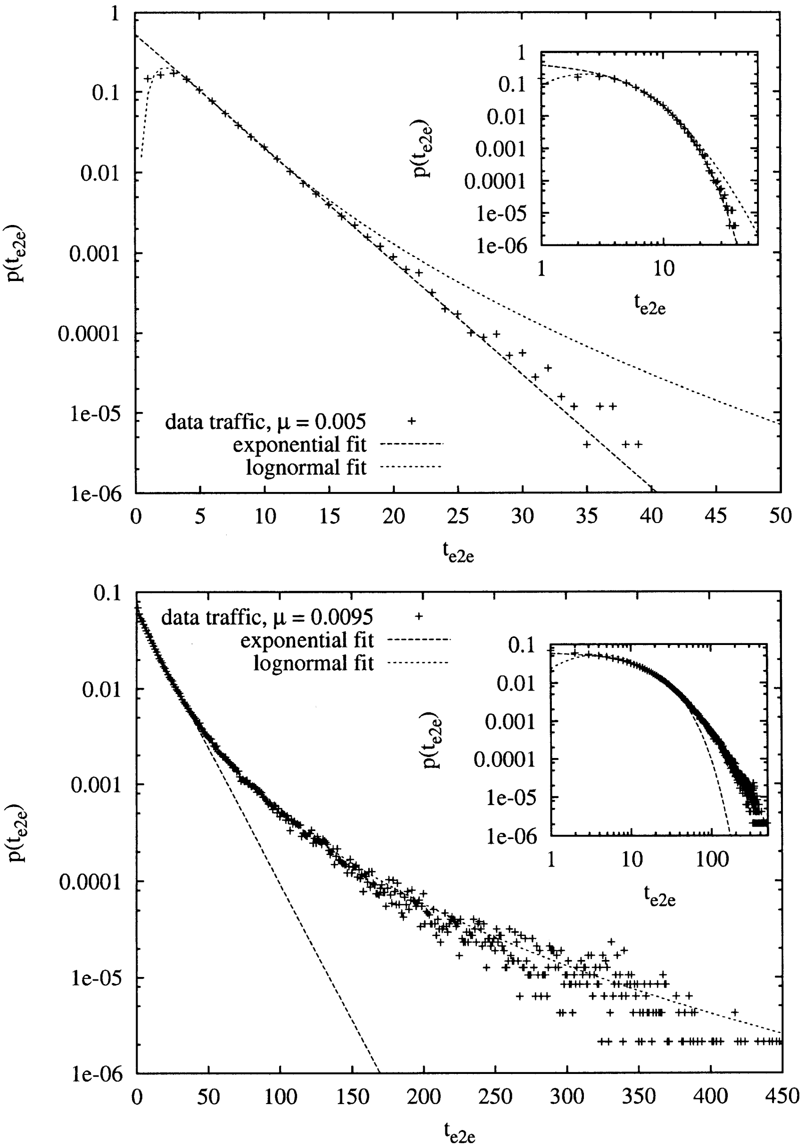

Fig. 3. Distribution of the end-to-end time delay obtained from a generic data traffic simulation with fixed shortest-path routing on a typical const-P network with $N = 100$ nodes. (Top) $\mu = 0.005$ well below and (bottom) $\mu = 0.0095$ close to the critical packet creation rate μ_{crit}. Best fits with an exponential and a log-normal distribution are also shown. The insets represent log–log plots.

However, for packet creation rates ($\mu = 0.0095$) close to $\mu^{\text{crit}} = 0.0101$ of the particular used network realization a log-normal distribution

$$p_{\text{lognormal}}(t_{\text{e2e}}) = \frac{1}{\sqrt{2\pi}\sigma t_{\text{e2e}}}\, \mathrm{e}^{-(\ln t_{\text{e2e}} - \tau)^2/(2\sigma^2)} \tag{4}$$

provides a better fit. The emergence of a log-normal distribution close to the critical packet creation rate appears to be an inherent feature of communication networks [22,23], when nodes are strongly and collectively coupled via heavy data traffic.

The tendency to have a small but non-negligible number of packets with a rather high end-to-end time delay is an undesirable feature for communication networks. A suppression of this tail in the distribution of the end-to-end time delay is regarded as one goal of any advanced routing and congestion control. Of course, another goal is to increase the critical packet creation rate.

3.2. End-to-end time delay II: analytic estimate

In this subsection we give an analytic estimate of the mean end-to-end time delay. The starting point is Little's Law (2), which demands to model the average number N_{active} of active data packets traveling on the network. Upon assuming inter-node correlations to be small, we adopt the single-node picture and reduce $N_{\text{active}} = \sum_i n_i$ to the modeling of a single-node queue length n_i. The latter itself depends on the in- and out-flux rates μ_i^{in} and μ_i^{out} of data packets to node i. If this turns out to be the only dependence, then the probability for node i to have n_i packets in its queue can be described by the rate equation

$$p(n_i, t+1) = \mu_i^{\text{out}} p(n_i+1, t) + (1 - \mu_i^{\text{in}} - \mu_i^{\text{out}}) p(n_i, t) + \mu_i^{\text{in}} p(n_i - 1, t) \, . \tag{5}$$

Reducing to the stationary limit $p(n_i, t+1) = p(n_i, t)$ and taking the boundary condition $p(n_i < 0) = 0$ into account, its normalized solution is

$$p(n_i) = \left(\frac{\mu_i^{\text{in}}}{\mu_i^{\text{out}}}\right)^{n_i} \left(1 - \frac{\mu_i^{\text{in}}}{\mu_i^{\text{out}}}\right) \, . \tag{6}$$

As a direct consequence, the relations

$$\langle n_i \rangle = \frac{\mu_i^{\text{in}}/\mu_i^{\text{out}}}{1 - \mu_i^{\text{in}}/\mu_i^{\text{out}}} \tag{7}$$

and

$$p(n_i \geqslant 1) = \mu_i^{\text{in}}/\mu_i^{\text{out}} \tag{8}$$

are derived, which will be of later use.

We have carefully checked the assumption going into Eq. (5); see also Ref. [24]. If the queue length distribution only depends on the in- and out-flux rates, which is filed as case $M/M/1$ in queuing theory [21], then the interarrival and sending times statistics should both obey the geometric distribution

$$p(t_i^{\text{arrive/send}} = t) = (1 - \mu_i^{\text{in/out}})^{t-1} \mu_i^{\text{in/out}} \tag{9}$$

which comes with mean $\langle t_i^{\text{arrive/send}} \rangle = 1/\mu_i^{\text{in/out}}$ and reflects the independence of subsequent packet arrival and departure events. The interarrival time t_i^{arrive} is defined as the time between two successive arrivals of data packets that are put at the end of node i's queue. The sending time t_i^{send} is defined as the time between two successive sending events from node i to any of its neighboring nodes $j \in \mathcal{N}_i$ given that the queue is non-empty. Fig. 4 illustrates results obtained from the generic data traffic simulation, which show that, when focusing on a single node of an arbitrary network realization, the distributions of interarrival and sending time nicely follow the parameterization (9) for various packet creation rates and that the extracted in- and out-flux rates lead to a good agreement between the queue length distribution (6) and its simulated counterpart.

The modeling is now open for the in- and out-flux rates. We briefly outline the results, which have already been derived in a previous Paper [2]. The in-flux rate

$$\mu_i^{\text{in}} = \frac{\mu B_i}{N-1} \tag{10}$$

is proportional to the rate μN of newly created packets, of which the fraction $B_i/N(N-1)$ will be routed via node i during later time steps. The node inbetweeness B_i and, equivalently, the link inbetweeness $B_{i\leftrightarrow j}$ count the number of shortest paths, which go over node i and link $i \leftrightarrow j$, respectively. The modeling of the out-flux rate $\mu_i^{\text{out}} = 1/\tau_i$ is equivalent to the modeling of the mean sending time $\tau_i = \langle t_i^{\text{send}} \rangle$:

$$\begin{aligned}\tau_i &= 1 + \sum_{j_1 \in \mathcal{N}_i^{in}} p_{j_1}(n_{j_1} \geqslant 1) + \sum_{j_2 \in \mathcal{N}(\mathcal{N}_i^{in}) \backslash \mathcal{N}_i^{in}} p_{j_2}(n_{j_2} \geqslant 1) \sum_{j_1 \in \mathcal{N}_i^{in}} \frac{B_{j_2 \leftrightarrow j_1}}{2B_{j_2}} \\ &= 1 + \sum_{j_1 \in \mathcal{N}_i^{in}} \frac{\mu B_{j_1}}{(N-1)} \tau_{j_1} + \sum_{j_2 \in \mathcal{N}(\mathcal{N}_i^{in}) \backslash \mathcal{N}_i^{in}} \frac{\mu B_{j_2}}{(N-1)} \tau_{j_2} \sum_{j_1 \in \mathcal{N}_i^{in}} \frac{B_{j_2 \leftrightarrow j_1}}{2B_{j_2}} \,. \end{aligned} \tag{11}$$

The first sum in the first line represents those one-hop neighbors j_1, which also want to transmit a packet at the same time. Also two-hop neighbors j_2 contribute, once they have a packet to transmit to a one-hop neighbor, which would then MAC-block node i; this is described by the last term of the first line. For the second step, Eqs. (8) and (10) have been used, leading to N coupled linear inhomogeneous equations for the sending times, which are then solved numerically for a given network realization. Note, that the expression (11) overestimates the actual sending time to some extend, because one- and two-hop neighbors might have already been blocked by previously assigned one-hop transmissions.

Upon putting Eqs. (2), (7), (10) and (11) together, an analytic estimate of the average end-to-end time delay can finally be given. It is interesting to mention two limiting cases. In the limit $\mu \to 0$ of very small packet creation rates, the estimated sending times (11) converge to $\tau_i \to 1$, so that the average end-to-end time delay becomes

$$\lim_{\mu \to 0} \langle t_{\text{e2e}} \rangle = \frac{1}{\mu N} \sum_i \mu_i^{\text{in}} = D \,, \tag{12}$$

where the sum rule $\langle B_i \rangle = (N-1)D$ has been used to express the mean node inbetweeness in terms of the network diameter [2]. This shows that for weak data traffic loads the network diameter determines the end-to-end time delay. In the other limit, $\mu \to \mu_{\text{crit}}$, the average end-to-end time delay is dominated by the most critical node. This node i is the first, for which the in- and out-flux rates become identical and which determines the critical packet creation rate $\mu_{\text{crit}} = (N-1)/(\sup_j B_j \tau_j)$. The sum over all nodes

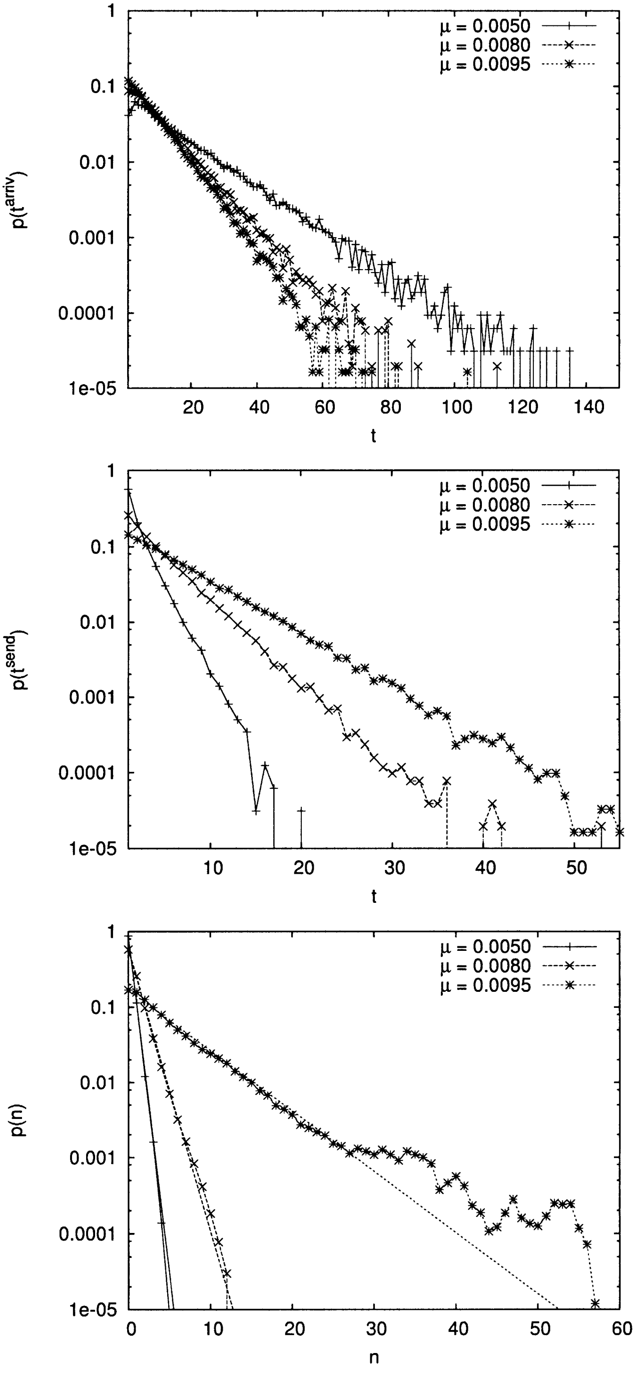

Fig. 4. Distributions for (top) interarrival times, (middle) sending times and (bottom) queue lengths as observed at the most-critical node of a const-P network realization with $N = 100$. All shown packet creation rates belong to the subcritical phase. The thin lines in (bottom) represent the expression (6), where according to (9) the in- and out-flux rate have been taken from (top) and (middle) as the reciprocal of the mean interarrival and sending times.

$\sum_j n_j \approx n_i$ breaks down and we arrive at

$$\lim_{\mu \to \mu_{\rm crit}} \langle t_{\rm e2e} \rangle = \frac{1}{\mu N} \frac{1}{1 - \mu B_i \tau_i(\mu)/(N-1)} \sim \frac{1}{\mu_{\rm crit} - \mu} \, . \tag{13}$$

In the last step we have made use of the fact that $\tau_i(\mu)$ is bounded from above by the magnitude of node i's one- and two-hop neighborhood. As the packet creation rate approaches $\mu_{\rm crit}$, the number of packets within the network explodes to infinity. The critical exponent turns out to be 1.

For the const-P and minimum-node-degree networks, the analytic estimate of the average end-to-end time delay is shown in Fig. 2 and compared to the respective results obtained from the generic data traffic simulations. Note that both the analytic as well as the simulation estimate have been averaged over a large sample of network realizations. For μ safely below $\mu_{\rm crit}$ a good agreement is found. Since the analytic estimate and the generic data-traffic simulation produce a slightly different $\mu_{\rm crit}$, divergence of $t_{\rm e2e}$ sets in at different μ, so that the quality of the comparison declines for μ close to $\mu_{\rm crit}$.

3.3. End-to-end time delay III: scalability

Instead of comparing the load-dependent end-to-end time delay between the two network models of fixed size, a comparison within a given network model, but of varying size allows to address the scalability issue. Fig. 5 illustrates $\langle t_{\rm e2e} \rangle$ as a function of $\mu/\mu_{\rm crit}$ for various sizes $N \geqslant 100$ of const-P and minimum-node-degree networks. For each $\mu/\mu_{\rm crit}$ and fixed N generic data traffic simulations have been run with a sample of 100 network realizations. Note also, that for each realization the critical packet creation rate fluctuates to some small degree. For a fixed $\mu/\mu_{\rm crit}$ the end-to-end time delay increases with network size. In the limit $\mu \ll \mu_{\rm crit}$ this increase roughly scales as $\sqrt{N}$, which is in accordance with $t_{\rm e2e}(\mu = 0|N) = D(N) \sim \sqrt{N}$, reflecting the scaling behavior of the network diameter.

In order to make scalability statements also for $0 < \mu/\mu_{\rm crit} < 1$, the following scaling ansatz is proposed:

$$\left(\frac{t_{\rm e2e}(\mu/\mu_{\rm crit}|N)}{t_{\rm e2e}(0|N)} \right)^{\delta} = \frac{t_{\rm e2e}(\mu/\mu_{\rm crit}|N_0)}{t_{\rm e2e}(0|N_0)} \, . \tag{14}$$

N_0 is some reference network size. An exponent $\delta = 1$ would imply that the relative increase of the end-to-end time delay with respect to relative network load is independent of the network size. For $\delta > 1$, the relative increase would decrease with network size and for $\delta < 1$ it would be the other way around. In fact, the scaling ansatz (14) leads to an excellent curve collapse, also shown as the lowest curve in Fig. 5(top)+(bottom): the solid curve corresponds to the right-hand side of (14) with $N_0 = 200$ and the collapsing symboled curves correspond to the left-hand side of (14). The fitted exponent δ is shown in the inset figure and reveals an N-dependence of the form $\delta(N) = (N/N_0)^{\beta}$. For const-$P$ networks with $k_{\rm target} = 24$ and minimum-node-degree networks with $k_{\rm min} = 8$ we find $\beta = 0.11$ and 0.25, respectively. This outcome shows, that from the perspective of the relative end-to-end time delay minimum-node-degree networks scale better with increasing network size than const-P networks.

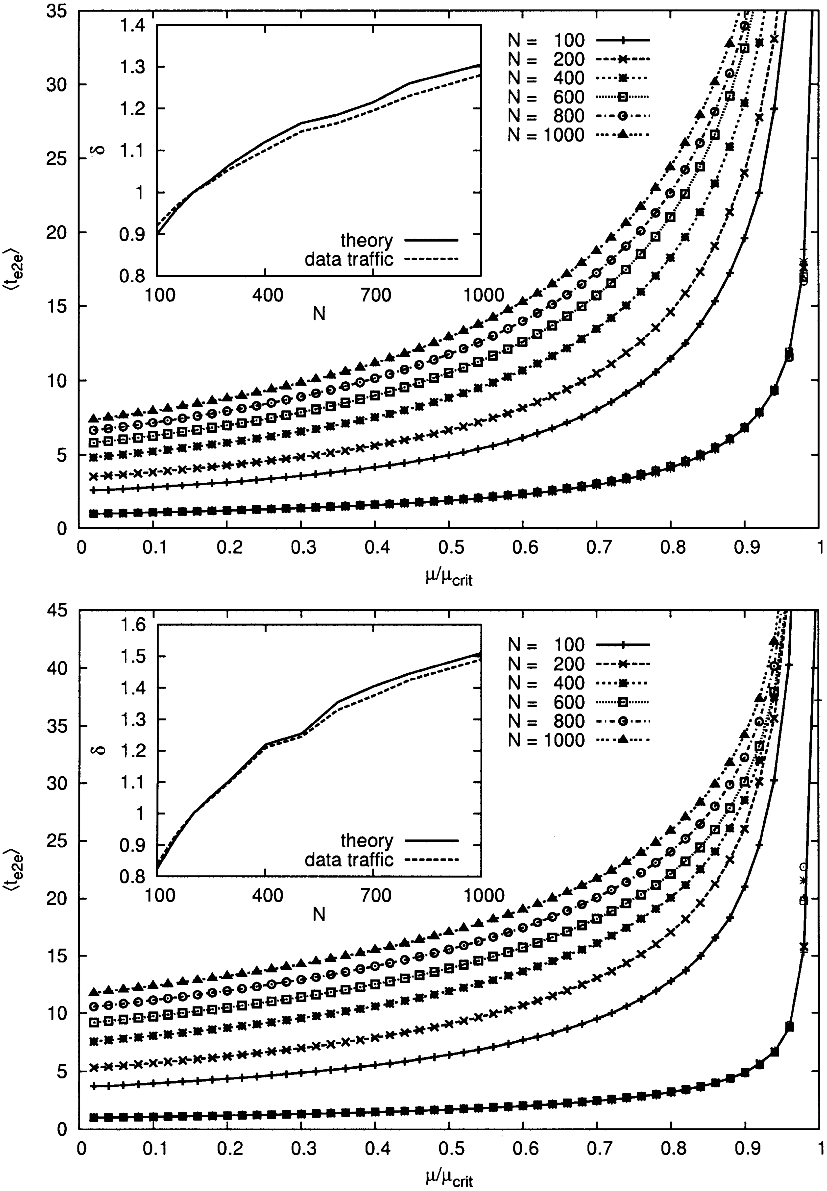

Fig. 5. Average end-to-end time delay as a function of $\mu/\mu_{\rm crit}$ for various sizes of (top) const-P and (bottom) minimum-node-degree networks. The lowest curve represents the curve collapse (14) with $N_0 = 200$; the exponent δ is shown in the inset figure.

Focusing on the size-dependence of the network models, the analytic estimate of the average end-to-end time delay also confirms the scaling ansatz (14) with $\delta(N) = (N/N_0)^\beta$; see insets of Fig. 5(top)+(bottom). For minimum-node-degree networks the found value $\beta = 0.25$ perfectly matches the outcome from the generic data traffic simulations, whereas for const-P networks a small discrepancy remains between the theoretically found $\beta = 0.15$ and its simulation counterpart 0.11.

3.4. Single-node correlation time: simulation results

Another reason for the emergence of congestion is that the queue length at specific nodes may fluctuate strongly around its mean. For sure, the occurrence of $n_i \gg \langle n_i \rangle$ with non-negligible probability enhances congestion. More than this, it is also the enhanced time it takes in such cases to relax back to the mean. A long relaxation time would mean that the specific node as well as its surrounding part of the network stay in a congested state for quite some time. A straightforward measure of such a relaxation time is given by the first moment $\langle n(t+\Delta t)|n(t)\rangle$ of the conditional probability $p(n(t+\Delta t)|n(t))$ to have a queue length $n(t+\Delta t)$ at time $t+\Delta t$ given $n(t)$ packets at time t. It is related to the correlation function $r(\Delta t)$ by averaging over all possible $n(t)$:

$$r(\Delta t) = \frac{\langle n(t+\Delta t) n(t)\rangle}{\langle n\rangle^2} = \frac{1}{\langle n\rangle^2} \sum_{n(t)} \langle n(t+\Delta t)|n(t)\rangle \, n(t) p(n(t)) \, . \tag{15}$$

Thus, the correlation function can also be seen as an averaged measure of relaxation times.

Fig. 6 illustrates the sampled single-node temporal correlation function $r_i(\Delta t)$ for a characteristic node in a typical const-P network. The convergence of $r_i(\Delta t)$ to 1 for large time differences indicates that correlations between queue lengths $n_i(t)$ and $n_i(t+\Delta t)$ no longer exist for large Δt. As the packet creation rate grows, this decorrelation is shifted towards larger time differences.

A non-standard way to extract characteristic time scales results from the observation, that the various curves of Fig. 6 for differing packet creation rates all appear to have a similar functional form. This motivates the curve collapse

$$r_i(\Delta t) = \left[f\left(\frac{\Delta t}{T_{\text{collapse}}(\mu)} \right) \right]^{\zeta} , \tag{16}$$

where the condition $r_i(\Delta t = 0) = \langle n_i^2(t)\rangle / \langle n_i(t)\rangle^2$ fixes the exponent to $\zeta = \ln(\langle n_i^2(t)\rangle / \langle n_i(t)\rangle^2)/\ln f(0)$; for later convenience we choose $f(0) = 2$. A suitable tuning of $T_{\text{collapse}}(\mu)$ then leads to a perfect curve collapse; see the inset of Fig. 6. The extracted time scale is shown in Fig. 7 as a function of the packet creation rate. Another, now standard way to extract a characteristic time scale uses

$$\int_0^{\infty} (r(t)^{1/\zeta} - 1) \, \mathrm{d}t = (r(0)^{1/\zeta} - 1) T_{\text{int}} \tag{17}$$

to define an integral time scale T_{int}. Its dependence on the packet creation rate is also shown in Fig. 7.

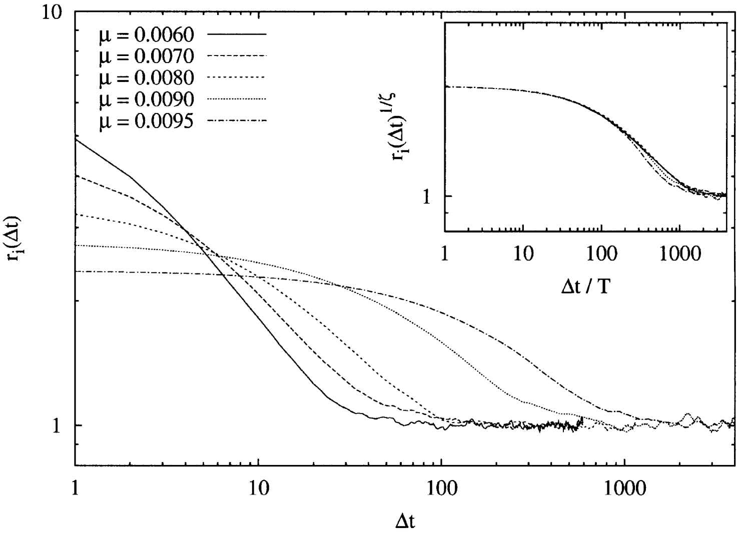

Fig. 6. Single-node temporal correlation $r_i(\Delta t)$ for the most-critical node of a const-P reference network with $N = 100$, obtained from a simulation covering 5×10^5 time steps. Different line types correspond to different packets creation rates, all below $\mu_{\rm crit} = 0.0101$. The inset shows the curve collapse (16) after appropriate rescaling.

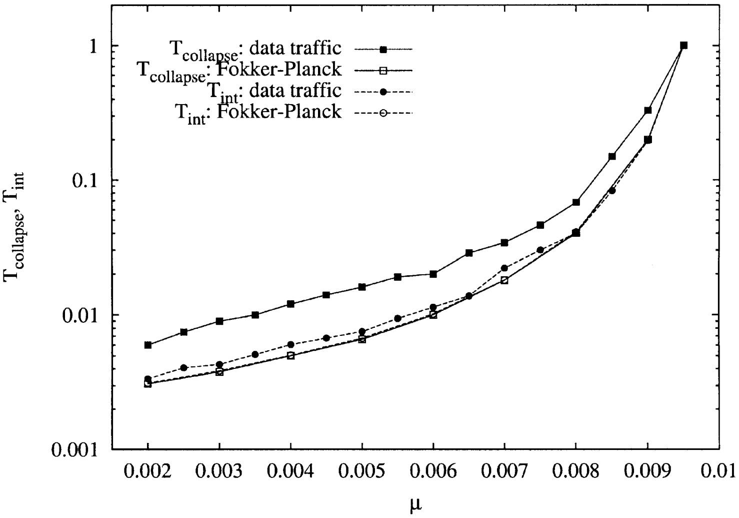

Fig. 7. Time scales $T_{\rm collapse}$ and $T_{\rm int}$ as a function of the packet creation rate. The same const-P reference network with $N = 100$ has been used as in the previous figure. The results directly obtained from a data traffic simulation are shown with filled symbols. The curves with open symbols have been derived from Eq. (21) with in- and out-flux rates sampled from data traffic simulations. Note that the two time scales have been normalized such that $T_{\rm collapse/int} = 1$ for $\mu = 0.0095$.

3.5. Single-node correlation time: analytic estimate

We will now give some semi-analytic understanding of the simulation findings of the previous subsection. In the rearranged form,

$$\begin{aligned} p(n,t+1) - p(n,t) &= \frac{(\mu^{\text{out}} - \mu^{\text{in}})}{2}\,[p(n+1,t) - p(n-1,t)] \\ &\quad + \frac{(\mu^{\text{out}} + \mu^{\text{in}})}{2}\,[p(n+1,t) - 2p(n,t) + p(n-1,t)]\,, \end{aligned} \tag{18}$$

the rate equation (5) transforms into a Fokker–Planck equation

$$\frac{\partial p(n,t)}{\partial t} = -(\mu^{\text{in}} - \mu^{\text{out}})\frac{\partial p(n,t)}{\partial n} + \frac{(\mu^{\text{in}} + \mu^{\text{out}})}{2}\,\frac{\partial^2 p(n,t)}{\partial n^2}\,. \tag{19}$$

The invoked continuum limit from discrete to continuous n is justified for long queue lengths $\langle n\rangle \gg 1$, which is the case when the in- and out-flux rates $\mu^{\text{in}} \approx \mu^{\text{out}}$ are almost the same. The latter also determine the drift and diffusion coefficients

$$-\gamma = \mu^{\text{in}} - \mu^{\text{out}}, \quad D = \mu^{\text{in}} + \mu^{\text{out}} \tag{20}$$

which are both constant. Note, that a minus sign has been introduced in the definition of the drift coefficient, which guarantees $\gamma > 0$ for the subcritical traffic regime $\mu^{\text{in}} < \mu^{\text{out}}$.

The solution of this Fokker–Planck equation with given initial condition $n(t=0)$ and reflecting boundaries at $n=0$ and ∞ requires an expansion into eigenfunctions [25]. The eigenfunction method also allows a direct calculation of the correlation function. This calculation is rather lengthy, but straightforward. Details are given in Ref. [24]. Here, we state only the final result

$$\begin{aligned} \langle n(t)n(0)\rangle = \frac{1}{4}\left(\frac{D}{\gamma}\right)^2 + \frac{4}{\pi}\left(\frac{D}{\gamma}\right)^2 \mathrm{e}^{-\frac{\gamma^2}{2D}t}\Bigg[&\frac{\pi}{16}M\left(\frac{3}{2},-\frac{1}{2},\frac{\gamma^2}{2D}t\right) \\ &+\frac{2}{3}\sqrt{\pi}\left(\frac{\gamma^2}{2D}t\right)^{3/2} M\left(3,\frac{5}{2},\frac{\gamma^2}{2D}t\right)\Bigg]\,, \end{aligned} \tag{21}$$

expressed in terms of confluent hypergeometric functions [26]. It comes with the property $\langle n^2(0)\rangle = 2\langle n(0)\rangle^2$, which is in agreement with the simulation results of the previous subsection for packet creation rates close to the critical one (see again Fig. 6), where $\langle n\rangle \gg 1$. It also explains the choice $f(0)=2$ introduced for the curve collapse (16).

Expression (21) only depends on the in- and out-flux rates (20), otherwise its functional form is fixed. Upon taking the sampled μ_i^{in} and μ_i^{out} from the generic data traffic simulation, expression (21) can be compared with the directly sampled correlation functions; this is done in Fig. 8. For all packet creation rates, the decorrelation time obtained via (21) is systematically somewhat larger than for the directly sampled correlation functions. A possible reason for this discrepancy might be the invoked

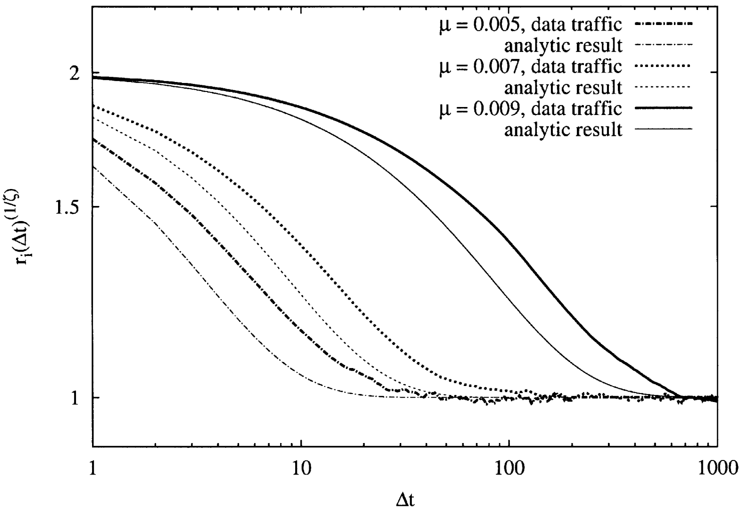

Fig. 8. Comparison of the rescaled correlation functions $(r_i(\Delta t))^{1/\xi}$ of Fig. 6 with the analytic expression (21). For the latter, the in- and out-flux rates have been sampled from the same generic data traffic simulations, the same network realization and the same node, which have been used for the former.

rescaling $(r_i(\Delta t))^{1/\xi}$ of the latter. However, the functional form looks the same and also the respective time scales T_{collapse} and T_{int} as a function of the packet creation rate show a good agreement with the previously obtained results of Fig. 7. The overall good correspondence between the simulation and the semi-analytic results more or less explains the increase of the correlation time scales with growing packet creation rate as an inherent feature of the underlying single-node queuing behavior.

4. Routing and congestion control

From the previous section on the generic data traffic with shortest-path routing we have learned several things. It is the most critical node, which gets overloaded first among all other nodes and which determines the critical packet creation rate of the overall network. This limits the network's e2e-throughput capacity to $\mu_{\text{crit}}N$. Moreover, for network loads close to μ_{crit} a good fraction of nodes belonging to the greater neighborhood of the critical one will also come with large queue lengths $n_i \gg 1$. This congestion cluster then gives rise to large average e2e-time delays and to large fluctuations of the latter, which in turn result in large relaxation times. It is the goal of any routing and congestion control to avoid such congested network areas and to detour the data traffic around. Such actions are likely to be rewarded with an increase of the e2e-throughput capacity, a decrease of the average e2e-time delay as well as its fluctuations and the related relaxation times. In this section we discuss three different routing and congestion controls of increasing sophistication. The first one exploits the degeneracy of shortest e2e-routes. The other two approaches modify the distance

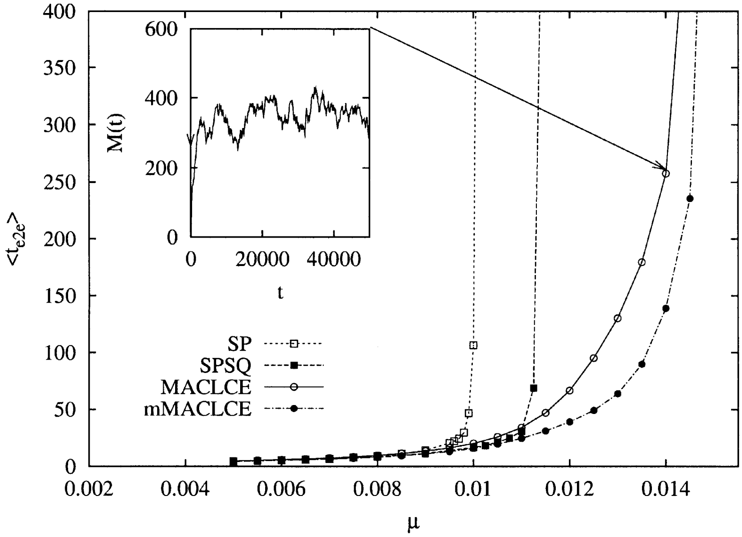

Fig. 9. Average end-to-end time delay as a function of the packet creation rate for various routing and congestion controls: SP, SPSQ, MACLCE with $\nu = 0$, and memory-based MAC-distributed lowest-cost-estimate (mMACLCE) with $\nu = 0.65$. Respective generic data traffic simulations have been run on an identical const-P network realization of size $N = 100$.

metric to include each node's congestion state and adapt the routing decisions according to updated cost estimates, which are locally exchanged with every MAC-blocking.

4.1. Simple congestion control with degenerated shortest paths

The shortest-path (SP) routing used in Section 3 does not take advantage of the route degeneracy between an arbitrary sender and a final receiver. Randomly choosing one out of several degenerate shortest routes for each new packet will already give some relief to the most congested nodes. However, a bias on the actual congestion state would do even better. A simple extension in this direction is shortest-path shortest-queue (SPSQ) routing. Instead of randomly choosing one next-hop neighbor out of the several degenerate shortest routes that specific node is picked which in addition has the shortest queue length in its buffer for this very moment. If more than one node qualifies, one of them is again picked randomly. Note, that in order to make such a routing decision, the forwarding node needs to have information from its neighbors about their congestion state. A very elegant way to provide this information without sending additional control packets is to include it into the MAC-blockings, in which the neighbors had been actively involved during previous one-hop transmissions.

As can be seen from Figs. 9–11, the simple SPSQ routing and congestion control already leads to some noticeable improvements. The critical packet creation rate, where

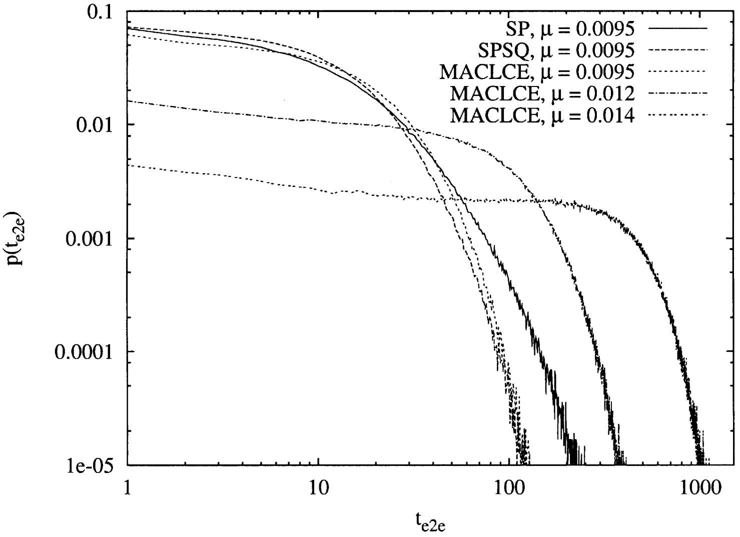

Fig. 10. Distribution of the end-to-end time delay for various routing and congestion schemes and packet creation rates. For $\mu = 0.0095$ the schemes SP, SPSQ and MACLCE are compared; for $\mu = 0.012$ and 0.014 only the MACLCE scheme is shown. Generic data traffic simulations have been run on an identical const-P network realization of size $N = 100$.

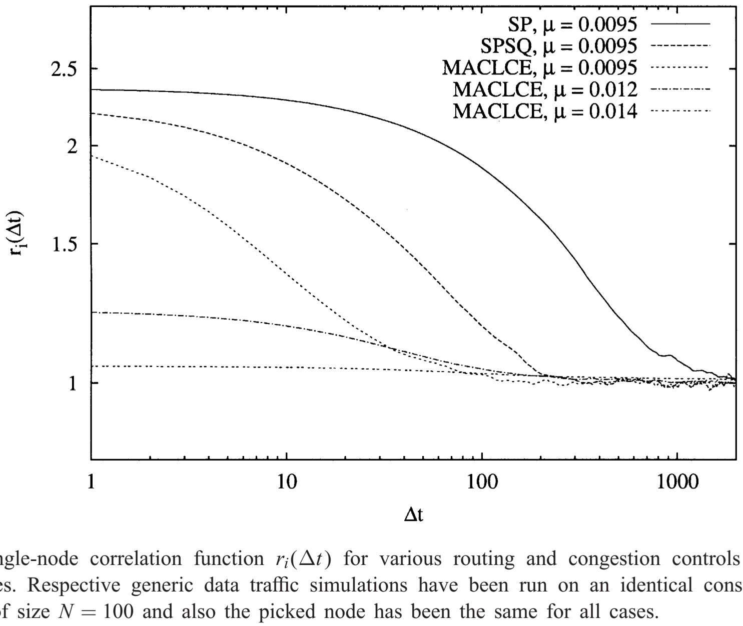

Fig. 11. Single-node correlation function $r_i(\Delta t)$ for various routing and congestion controls and packet creation rates. Respective generic data traffic simulations have been run on an identical const-P network realization of size $N = 100$ and also the picked node has been the same for all cases.

the average e2e-time delay diverges, increases. The tail of the e2e-time-delay distribution is suppressed to some extend. Last but not least, also the correlation time scales have become smaller.

4.2. Congestion control with instantaneous adaptive routing

If a node would have the complete information on the instantaneous congestion state of all other nodes belonging to the network, then it could determine the shortest route to the intended final recipient with a modified metric, which does not only take the hop distance into account, but also counts the queue lengths of all inbetween nodes. However, first of all the node does not have this information, and even if it would, then the shortest path at decision time needs not be equal to the shortest path for delivery due to the always changing congestion state of the network during delivery time t_{e2e}. Instead, the node could try to get some sort of cost estimate it takes to send the packet to the final destination via this or that neighbor, and to constantly update these estimates. This idea is already known as asynchronous vector distance routing [27] and has found its way into Internet routing at the autonomous-system level. Next, we give an outline of this approach, modified and tuned to the specificity of wireless multihop ad hoc communication.

Pick a node i that has a packet to forward to the final destination f. Node i has to decide to which neighbor $j \in \mathcal{N}_i$ it is going to forward the packet. For each of them it has a cost estimate $W_{if,j}$. It chooses node $j_{\min}$, providing the lowest cost estimate $W_{if,j_{\min}} = \min_{j \in \mathcal{N}_i} W_{if,j} = W_{if}$. Before i starts transmitting its packet to $j_{\min}$, it has to MAC-block its neighbors. Hand-in-hand with the blocking signal it tells them about the minimum cost W_{if} and its future queue length $n_i - 1$. While then being blocked, those neighbors have enough time to process this information and update their estimated cost

$$W_{jf,i} \leftarrow w_{ji} + W_{if} \tag{22}$$

to send a packet via i to f during a future time step. w_{ji} is the cost to send a packet from j to its neighbor i. Since the queue length $n_i - 1$ is a reasonable measure of how busy node i is [27,28], we set

$$w_{ji} = (n_i - 1) + 1 \,, \tag{23}$$

the one at the end takes care of the hop-distance between the two nodes. As the intended receiver of i's to-be-transmitted packet, node $j_{\min}$ also blocks its neighbors $k \in \mathcal{N}_{j_{\min}}$ and takes this chance to inform those about its estimated cost $W_{j_{\min}f}$ and its future queue length $n_{j_{\min}} + 1$. Those neighbors then process this information analogous to Eq. (22), but with the modification

$$w_{kj_{\min}} = (n_{j_{\min}} + 1) + 1 \tag{24}$$

for the future link cost.

Updating the estimated costs after receipt of an extended MAC-signal appears to be a natural thing for wireless multihop ad hoc communication. It couples the MAC-layer and the routing layer in a very elegant way and gives rise to a distributive congestion control. Note that beyond MAC no additional congestion control signals have to be exchanged. This is in contrast to other communication networks like the Internet, which first probe the congestion state with additional dedicated signals before they adapt to it.

A few more technical words are in order. The congestion information distributed by the MAC-signals may either include only those cost estimates belonging to the final destination of the currently transmitted packet, or it may include much more than this, namely the cost estimates to every single node of the network. Both cases certainly represent two extremes, the former focusing on keeping the distributed signal small and the latter allowing for a faster spread of the congestion information over the entire network. For the remainder of this paper, we will only concentrate on the latter case.

Implementing the proposed congestion control for the generic data traffic simulations, requires to specify the initialization. The initial cost estimates have been set to

$$W_{if,j} = \begin{cases} 1 & (f \in \mathcal{N}_i)\,, \\ 0 & (i = f)\,, \\ \infty & (\text{else})\,. \end{cases} \tag{25}$$

The simulations themselves then have shown that for not-too-small packet creation rates the number of time steps it takes to distribute the initial cost estimates over the entire network and to reach a kind of steady state is of the order of the network size N. Furthermore, test simulations have revealed that a relaxation of the deterministic lowest-cost-neighbor choice for forwarding the data packet, such that also the other higher-cost neighbors become eligible in some probabilistic form, always leads to a degradation in performance.

The simulations of the generic data traffic for the routing and congestion control with the MAC-distributed lowest-cost estimates (MACLCE) have turned out to be quite time-consuming. For this reason we present results only for one realization of a constant-P network with size $N = 100$. The simulation run took 5×10^5 time steps. As a function of the packet creation rate the average e2e-time delay is illustrated in Fig. 9. For moderate packet creation rates the various routing and congestion schemes SP, SPSQ and MACLCE perform about equally well, but when it comes to the critical regime, the control with MAC-distributed lowest-cost estimates yields by far the largest critical packet creation rate. Although more packets are present in the network, such a routing and congestion scheme is able to handle packet creation rates, exceeding the maximum rate obtained with the classical shortest-path routing by a factor of about 1.4. The inset of this figure shows the number of active packets $M(t) = N_{\text{active}}(t)$ for MACLCE at $\mu = 0.014$, indicated by the arrow. There is no linear dependence on time, which is a clear sign that the network is still operating in the subcritical regime.

Another goal for the new routing and congestion algorithm has been to reduce the pronounced tail in the distribution $p(t_{\text{e2e}})$ of the end-to-end time delay. Fig. 10 demonstrates that for $\mu = 0.0095$, which is very close to the SP critical packet creation rate, the MACLCE and SPSQ distributions are about the same and, when compared to the SP distribution, come with a suppressed tail. Note, that log-scales have been used in Fig. 10. For larger packet creation rates, where the SP and SPSQ schemes are already in the supercritical regime, the routing and congestion control with MACLCE results in rather flat distributions, which come with a relative sharp cutoff at larger end-to-end time delays. Of course, this cutoff increases with the network load.

Compared to the SP routing significant changes can be observed in the distribution $p(n_i)$ for the single-node queue length; see Fig. 12. At $\mu = 0.0095$ the SP distribution is a broad exponential (see also bottom of Fig. 4), whereas the MACLCE distribution is

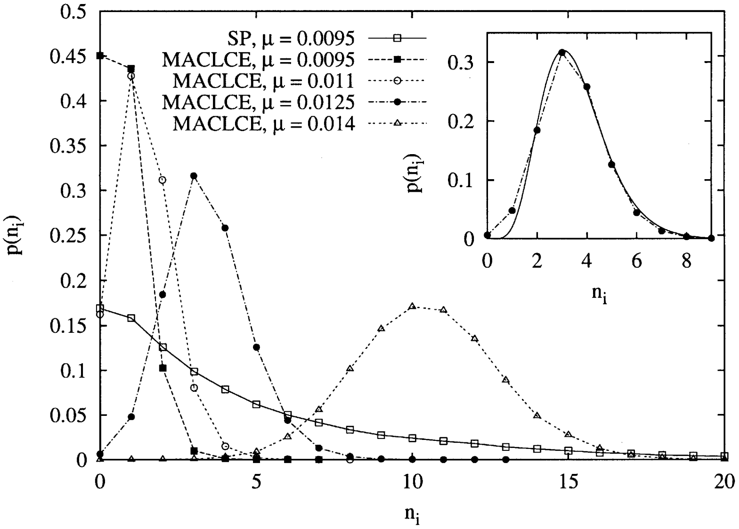

Fig. 12. Single-node distribution $p(n_i)$ of the queue length for various routing and congestion controls and packet creation rates. Respective generic data traffic simulations have been run on an identical const-P network realization of size $N = 100$ and also the picked node has been the same for all cases. For the MACLCE-curve with $\mu = 0.0125$ the inset illustrates a fit with the Gamma-distribution (26).

confined to basically $n_i \leqslant 3$. For larger packet creation rates the MACLCE distributions become bell-shaped. Their mean and variance increase with μ. Good fits can be obtained with a two-parametric continuous Gamma distribution

$$p_{\mathrm{gamma}}(n) = \frac{1}{\Gamma(a)b^a}\, n^{a-1} \mathrm{e}^{-n/b} , \tag{26}$$

consult the inset figure. For MACLCE-controlled networks close to μ^{crit} this bell-shaped single-node queue-length distribution is a result of the flexible routing scheme, which adapts well to the current congestion state, and is key to their improved operational functionality.

A direct consequence of the bell-shaped distributions $p(n_i)$ is that $\langle n_i^2(t)\rangle \approx \langle n_i(t)\rangle^2$. This then keeps the correlation function (15) close to one, even for small Δt. Thus with the packet creation rate approaching its critical limit, the single-node temporal correlations are expected to be drastically reduced. The simulation results illustrated in Fig. 11 confirm this view.

4.3. Congestion control with memory-based adaptive routing

So far the cost estimate $W_{if,j}$ of node i is updated according to Eq. (22) as soon as the neighboring node $j \in \mathcal{N}_i$ MAC-reports its change of either w_{ij} or W_{jf}. This immediate update might be too fast to be optimal. It could be wise to keep at least in parts some of the old cost estimate. Introducing a kind of memory parameter $0 \leqslant v \leqslant 1$, a proposition for a modified update rule would be

$$W_{if,j} \leftarrow v\, W_{if,j} + (1-v)[w_{ij} + W_{jf}]\ . \tag{27}$$

For $v > 0$ a fraction of the old cost estimate is kept as a part of the new estimate. This modification includes a memory of formerly used routes that are only updated if significant changes in the network's congestion state have occurred. This approach, which is inspired by reinforcement learning [27], is known as Q-routing [29].

The simple extension (27) proved to cause a small but clearly measurable further improvement in network performance. As indicated in Fig. 9 the memory-based mMACLCE scheme with $v = 0.65$ increases the critical packet-creation rate a little further when compared to the instantaneous MACLCE scheme with $v = 0$. It turns out that $v \approx 0.65$ is the optimal choice. As a function of v the critical packet creation rate first increases for values rising from $v = 0$ to 0.65, takes its maximum at $v \approx 0.65$ and then decreases for values above. It is intuitive that in case of a strong memory ($v \approx 1$) the update rule collapses and only keeps very old and out-fashioned cost estimates, which are not suited to adapt to the always changing congestion state of the network.

5. Conclusion and outlook

This paper has focused on routing and congestion control in wireless multihop ad hoc communication networks. Simulations with random data traffic have been accompanied with analytic estimates, whenever possible. The focus has first been on shortest-path routing, to understand certain data traffic characteristics like end-to-end time delay and correlation time scales, and to find fingerprints of congestion once the network is operating close to its critical load. A scaling law has been found for the average end-to-end time delay with respect to network size, which also revealed a dependence on the underlying network topology. In a second step and going beyond shortest-path routing, a distributive routing and congestion control has been proposed, which couples the MAC- and routing layer of wireless multihop ad hoc communication. Before one-hop forwarding a packet, the sending as well as receiving node MAC-block their respective neighbors and distribute information about their congestion state and routing cost estimates, which the latter then use for updates. This distributive scheme turned out to be very efficient. Compared to shortest-path routing, the critical network load increased noticeably. Routes are constantly adapting to the prevailing congestion state of the network. With other words, routes selforganize themselves.

The proposed prototype routing and congestion control needs of course further testing and extensions. Other than simple random data traffic has to be looked at, such as for example selfsimilar [30], self-organized-critical [31] and spatially localized. Congestion updates with different forms of cost metrics are important to investigate as well as the sparsity issue, i.e., which information is important to be distributed to other nodes and which is negligible. At the end, the biggest challenge is yet to come, to turn good ideas into real-life-functioning implementations. This is where Physics and Engineering should meet again.

Acknowledgements

W.K. gratefully acknowledges support from the Ernst von Siemens-Scholarship and the Frankfurt Center for Scientific Computing.

References

[1] I. Glauche, W. Krause, R. Sollacher, M. Greiner, Continuum percolation of wireless ad hoc communication networks, Physica A 325 (2003) 577–600.

[2] W. Krause, I. Glauche, R. Sollacher, M. Greiner, Impact of network structure on the performance of wireless multihop ad hoc communication, Physica A 338 (2004) 633–658.

[3] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (2002) 47–97.

[4] S.N. Dorogovtsev, J.F.F. Mendes, Evolution of Networks—From Biological Nets to the Internet and WWW, Oxford University Press, Oxford, UK, 2003.

[5] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (1) (2003) 167–256.

[6] P. Gupta, P. Kumar, Critical power for asymptotic connectivity in wireless networks, in: Stochastic Analysis, Control, Optimization and Applications, Birkhauser, Boston, 1998, pp. 547–566.

[7] C. Bettstetter, On the minimum node degree and connectivity of a wireless multihop network, in: Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2002, ACM, Lausanne, Switzerland, 2002, pp. 80–91.

[8] O. Dousse, P. Thiran, M. Hasler, Connectivity in ad hoc and hybrid networks, in: Proceedings of IEEE Infocom, New York, NY, USA, 2002, pp. 1079–1088.

[9] F. Xue, P.R. Kumar, The number of neighbors needed for connectivity of wireless networks, Wireless Networks 10 (2) (2004) 169–181.

[10] O. Dousse, F. Baccelli, P. Thiran, Impact of interference on connectivity in ad hoc networks, in: Proceedings of IEEE Infocom, San Francisco, CA, USA, 2003.

[11] R. Sollacher, M. Greiner, I. Glauche, Impact of interference on the wireless ad hoc networks capacity and topology, Wireless Networks, in press.

[12] R. Meester, R. Roy, Continuum Percolation, Cambridge University Press, Cambridge, UK, 1996.

[13] D. Stoyan, W.S. Kendall, J. Mecke, Stochastic Geometry and Its Applications, 2nd Edition, Wiley, Chichester, UK, 1996.

[14] J. Dall, M. Christensen, Random geometric graphs, Phys. Rev. E 66 (2002) 016121–016129.

[15] J.-Y.L. Boudec, S. Sarafijanovic, An artificial immune system approach to misbehavior detection in mobile ad hoc networks, in: A.J. Ijspeert, D. Mange, M. Murata, S. Nishio (Eds.), Bio-ADIT 2004, Lausanne, Switzerland, 2004, pp. 96–111.

[16] P. Gupta, P.R. Kumar, The capacity of wireless networks, IEEE Trans. Info. Theory IT 46 (2) (2000) 388–404.

[17] Mobile ad hoc networks (manet) working group, http://www.ietf.org/html.charters/manet-charter.html.

[18] Wireless ad hoc networks bibliography, http://w3.antd.nist.gov/wctg/manet/manet_bibliog.html.

[19] Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2002, ACM, Lausanne, Switzerland, 2002.

[20] Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2003, ACM, Annapolis, MD, USA, 2003.

[21] E. Gelenbe, G. Pujolle, Introduction to Queueing Networks, 2nd Edition, Wiley, Chichester, UK, 1999.

[22] B.A. Huberman, R.M. Lukose, Social dilemmas and internet congestion, Science 277 (1997) 535–537.

[23] R.V. Solé, S. Valverde, Information transfer and phase transitions in a model of internet traffic, Physica A 289 (3–4) (2001) 595–605.

[24] I. Glauche, Distributive flow control in multihop ad hoc communication networks, Diploma Thesis, Technische Universität, Dresden, 2003.

[25] C. Gardiner, Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences, Springer, Berlin, 1983.

[26] M. Abramowitz, Handbook of Mathematical Functions, Dover Publications, New York, 1972.

[27] M.L. Littman, J.A. Boyan, A distributed reinforcement learning scheme for network routing, in: J. Alspector, R. Goodman, T.X. Brown (Eds.), Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications, Erlbaum, Hillsdale, NJ, USA, 1993, pp. 45–51.

[28] M. Heusse, S. Guerin, D. Snyers, P. Kuntz, Adaptive agent-driven routing and load balancing in communication networks, Adv. Complex Systems 1 (1998) 237–254.

[29] S.P.M. Choi, D.-Y. Yeung, Predictive q-routing: a memory-based reinforcement learning approach to adaptive traffic control, in: D.S. Touretzky, M.C. Mozer, M.C. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, NIPS*95, MIT Press, Cambridge, 1996, pp. 945–951.

[30] K. Park, W. Willinger (Eds.), Self-Similar Network Traffic and Performance Evaluation, Wiley, New York, NY, USA, 2000.

[31] S. Valverde, R.V. Solé, Self-organized critical traffic in parallel computer networks, Physica A 312 (3–4) (2002) 636–648.