

SCIENTIFIC REPORTS



OPEN

Controlling congestion on complex networks: fairness, efficiency and network structure

Luboš Buzna¹ & Rui Carvalho²

We consider two elementary (max-flow and uniform-flow) and two realistic (max-min fairness and proportional fairness) congestion control schemes, and analyse how the algorithms and network structure affect throughput, the fairness of flow allocation, and the location of bottleneck edges. The more realistic proportional fairness and max-min fairness algorithms have similar throughput, but path flow allocations are more unequal in scale-free than in random regular networks. Scale-free networks have lower throughput than their random regular counterparts in the uniform-flow algorithm, which is favoured in the complex networks literature. We show, however, that this relation is reversed on all other congestion control algorithms for a region of the parameter space given by the degree exponent γ and average degree $\langle k \rangle$. Moreover, the uniform-flow algorithm severely underestimates the network throughput of congested networks, and a rich phenomenology of path flow allocations is only present in the more realistic α -fair family of algorithms. Finally, we show that the number of paths passing through an edge characterises the location of a wide range of bottleneck edges in these algorithms. Such identification of bottlenecks could provide a bridge between the two fields of complex networks and congestion control.

Twenty-first century life depends on the reliability of critical infrastructure networks. Because of high construction costs, these networks often end up operating close to the unstable region: a small increase in flow leads to network congestion or even shut-down^{1–4}. Without properly designed congestion control, the consequences can be catastrophic, as in the congestion collapse on the Internet⁵. In 1986, the Internet (then ARPANet) was a slow (56 Kbps) and small network with a large number of hosts (5,089)⁶. In October that year, the link between University of California, Berkeley and Lawrence Berkeley National Laboratory (360 m long) suffered a drop in flow rate by three orders of magnitude from 32 Kbps to 40 bps. The reason for the collapse is the control mechanism implemented at the time, which focused on congestion at the receiver. The bottleneck, however, was congestion on the network. Two years later, Van Jacobson redesigned the TCP congestion control algorithm⁷, enabling the Internet to expand in size and speed. Today, we need algorithms to share scarce network resources during times of crises^{8,9}. In the future, we will require algorithms to share the capacity of electrical distribution networks for the charging of electric vehicles⁴. Moreover, when transport becomes autonomous, we may need algorithms to ease traffic congestion^{10,11}, and an understanding of the role of fairness, efficiency and network structure on such algorithms could improve the way society manages transportation. These challenges revive the topic of congestion control and uncover a new range of problems of network design at the interface between physics, engineering and the social sciences^{4,8,12–15}. Although much work has been done to characterise congestion control mechanisms^{5,16–18} and the topology of large random networks^{19–23}, little is known about the effect of network structure on congestion control. Furthermore, while congestion control methods have been in operation in communication networks since the 1980s, the relative performance of these algorithms on large random networks remains elusive.

A network is at the onset of congestion when at least one edge is carrying traffic at its capacity²⁴. When there is an attempt to increase traffic on that edge beyond its capacity, the network becomes congested, and the flow on the edge does not increase any further, even if the traffic load presented to the edge increases. In modelling congested complex networks, researchers typically look for the value of a control parameter for which the network reaches the onset of congestion. Studies have focused on the onset of congestion as a function of network structure and parameters²⁵, optimal topologies for local search with congestion^{26–28}, scaling of fluctuations in a

¹University of Žilina, Univerzitná 8215/1, 01026, Žilina, Slovakia. ²School of Engineering and Computing Sciences, Durham University, Lower Mountjoy, South Road, Durham, DH1 3LE, UK. Correspondence and requests for materials should be addressed to L.B. (email: lubos.buzna@fri.uniza.sk)

model of an $M/M/1$ queueing system²⁹, improved routing protocols³⁰, the impact of community structure on the transport of information³¹, an edge weighting rule to lower costs with node capacity and increase the packet generation rate at the onset of congestion³², and the emergence of extreme events in interdependent networks³³. These studies have the limitation that the sending frequency of packets (or rate) is uniform on the network and, consequently, the transition from free flow to congestion is determined by the nodes with the largest betweenness centrality. Hence, only the node(s) with the largest betweenness are fully utilised at the onset of congestion, and thus this method considerably underestimates the flow that congested networks can transport (see Methods, Section ‘Uniform-flow’). While traditionally network flows are modelled by maximising the network throughput (max-flow) or minimising the costs (minimum-cost), such efficient allocations can leave some users with zero flow, an unfair solution from the user point of view. Congestion control algorithms solve these problems by achieving cost-effective and scalable network protocols that well utilise the network capacity, sharing it among users in a fair way. These algorithms allocate path flows to paths connecting source to sink nodes. In doing so, they capture fairness by a family of user utility functions, called α -fair^{18,34}:

$$U_j(f_j, \alpha) = \begin{cases} \frac{f_j^{1-\alpha}}{1-\alpha} & \text{for } \alpha \geq 0, \alpha \neq 1 \\ \log(f_j) & \text{for } \alpha = 1, \end{cases} \quad (1)$$

where $j = 1, \dots, R$ is a path (or user), and f_j is the path flow assigned to path j . The algorithms maximise the aggregate utility $U(\alpha) = \sum_{j=1}^R U_j(f_j, \alpha)$, under the constraint that the path flows are feasible, i.e., all path flows are non-negative and no edge flow exceeds edge capacity.

For $\alpha = 0$, we recover the *max-flow* (MF) allocation that maximises the network throughput³⁵ $U(0) = \sum_{j=1}^R f_j$. For $\alpha = 1$, we find the *proportional fairness* (PF) allocation, an algorithm that manages congestion via Lagrange multipliers, which can be interpreted as an edge price. The proportional fairness optimisation problem is convex, and Slater’s qualification constraint implies that its primal and dual formulations are equivalent³⁶. The primal problem is solved for the path flows, whereas the dual is solved for the Lagrange multipliers or shadow prices. Both the primal and the dual problems can be posed as decentralised optimisation problems and solved as a system of coupled ODEs³⁷, which is much more efficient in large real-world networks than centralised control. Algorithmically, in the primal problem, source nodes ramp up the path flow additively but decrease it multiplicatively if at least one edge of the path is used close to capacity. The size of the system of coupled ODEs in the primal is determined by the number of paths in the network; in contrast, the number of ODEs in the dual is given by the number of network edges and is thus only dependent on network structure. Hence, if the number of paths is much larger than the number of network edges it is preferable to solve the dual instead of the primal^{5,8,37}. The *max-min fairness* (MMF) allocation is defined by $\alpha = \infty$ in Eq. (1); it is typically found, however, with a more efficient algorithm that maximises the use of network resources by users with the minimum allocation. Once these ‘poor’ users get the largest possible allocation, the process repeats iteratively for the next less well-off users^{16,38}. Intuitively, a set of path flows is max-min fair if the wealthy can only get wealthier by making the poor even poorer. The *uniform-flow* (UF) problem is determined by the maximisation of the aggregate utility $U(\alpha)$ any $\alpha \geq 0$, with the added constraint that all path flows are the same, which implies the optimum is independent of α . The α -fairness utility function provides a social planner with a way to understand the trade-off between efficiency ($\alpha = 0$) and a continuum of models of fairness, such as proportional fairness ($\alpha = 1$) and max-min fairness ($\alpha = \infty$)^{16-18,37-42}. The proportional fairness allocation is especial because the system and the users simultaneously maximise their utility functions, and because it is implemented in communication networks⁵ (see Methods, Section ‘The mathematics of congestion control’).

Results

To gain insights into the behaviour of the α -fairness family of algorithms, and to illustrate the phenomenon of congestion collapse, we first analyse the network throughput on a ring lattice. We consider a simple protocol that distributes edge capacity proportionally to flows on the paths that pass through an edge (see Methods, Section ‘Avoiding congestion collapse on the ring lattice’). A long path, which uses all network edges, competes for flow with a set of short paths that use only two edges each. Individual paths may increase the flow they inject into the network with the aim of raising their edge capacity quota; queues then build up at the nodes, and the lattice becomes congested. Surprisingly, as the injected flow grows, the network throughput does not converge to an upper bound as intuitively expected, but to zero. This collapse, however, can be avoided if we control congestion with the α -fair family of algorithms of Eq. (1). Intuitively, network throughput should decrease with an increase in α , so that it is larger or equal for max-flow than for proportional fairness, greater or equal for proportional fairness than for max-min fairness, and in turn larger or the same for max-min fairness than for uniform-flow. In other words: we expect that the price to pay for increasing equity is a decrease in throughput, such that the proportional fairness allocation is a trade-off between efficiency (max-flow) and fairness (max-min fairness, and uniform-flow). Our intuition is right for small ring lattices, but as the number of nodes in the ring grows throughput in the proportionally fair and max-flow allocations converge. Indeed, proportional fairness penalises long paths because these use more network resources than short paths. As the size of the ring grows, the long path uses a higher proportion of network capacity, thus getting less and less flow, and proportional fairness converges to max-flow. In contrast, max-min fairness yields a lower throughput than these two protocols because it assigns the same allocation to all paths (see Methods, Section ‘Avoiding congestion collapse on the ring lattice’). Hence, the ring lattice illustrates the counter-intuitive phenomena of congestion collapse, as well as, in the presence of congestion control, the surprising converge of proportional fairness to max-flow as the ring size grows. These

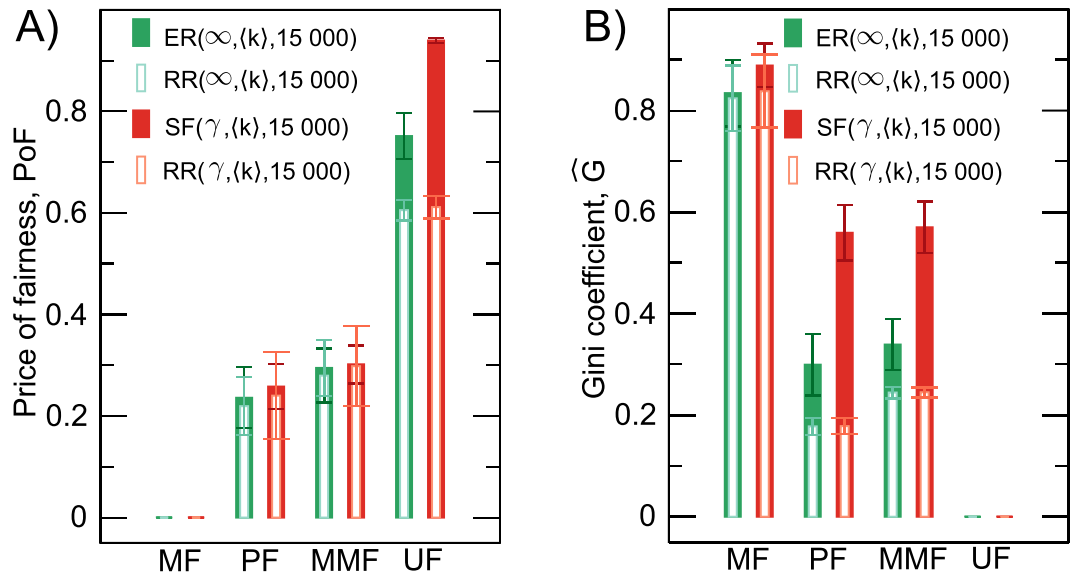


Figure 1. (A) Mean of the price of fairness (PoF) and (B) Gini coefficient of path flows for each congestion control algorithm in Erdős-Rényi and scale-free networks (and their random regular counterparts) with $R = 15000$ paths. For each network structure, we average over 20 randomly generated networks with $N = 2000$ nodes. Whiskers show the 95% confidence intervals. We average the price of fairness and Gini coefficient over the mean degree $\langle k \rangle = \{3, 4, 5, 6, 7, 8\}$ and the power-law exponent $\gamma = \{2.1, 2.3, 2.5, 2.7, 2.9\}$ for scale-free ($SF(\gamma, \langle k \rangle, R)$) and the corresponding random regular ($RR(\gamma, \langle k \rangle, R)$) networks, but only over $\langle k \rangle$ for Erdős-Rényi ($SF(\gamma = \infty, \langle k \rangle, R)$) and the corresponding random regular ($RR(\infty, \langle k \rangle, R)$) networks.

observations made on a regular network structure with a regular structure of paths are in sharp contrast with our findings on random networks.

We next study the effect of controlling congestion on scale-free (SF) (with exponent $2 < \gamma < 3$), Erdős-Rényi (ER), and random regular (RR) substrate networks with average node degree $3 \leq K \leq 8$. Flows take place on a *transport overlay network*, which is the subgraph formed by a set of R shortest paths, chosen with uniform probability among all possible shortest paths on the substrate network (see Methods, Section ‘Network Models’). From now on, we consider only the ‘transport overlay network’ when we refer to random networks and omit this term from the text. We now ask the question: to what extent is being fair compatible with maximising network throughput on random networks?

To analyse the interplay between algorithms and network structure, we next compute ‘the price of fairness’⁴³, that is the relative system efficiency loss under a ‘fair’ allocation compared to the one that maximises the sum of user utilities:

$$PoF(\mathcal{F}, \mathcal{N}) = 1 - \frac{F(\mathcal{F}, \mathcal{N})}{F(\text{MF}, \mathcal{N})}, \quad (2)$$

where $\mathcal{F} \in \{\text{MF}, \text{PF}, \text{MMF}, \text{UF}\}$ is the algorithm (max-flow, proportional fairness, max-min fairness, or uniform-flow), and $F(\mathcal{F}, \mathcal{N})$ is the throughput of the algorithm \mathcal{F} for the chosen network structure. We denote the network structure by \mathcal{N} , such that for scale-free networks we write $\mathcal{N} = SF(\gamma, \langle k \rangle, R)$, and we characterise Erdős-Rényi networks ($\gamma = \infty$) by $\mathcal{N} = ER(\infty, \langle k \rangle, R)$. Moreover, we write $\mathcal{N} = RR(\gamma, \langle k \rangle, R)$ to denote the corresponding random regular networks both for scale-free and Erdős-Rényi networks (see Methods, Section ‘Network Models’). The efficient algorithm (max-flow) has a price of fairness of zero, whereas an algorithm that results in zero network throughput has a price of fairness of one. Figure 1(A) shows that in contrast to the ring lattice, the price of fairness of proportional fairness in random networks is larger than zero, and of comparable magnitude to the price of fairness of max-min fairness, for all network structures we analysed, showing that the throughput of proportional fairness now approaches max-min fairness. To characterise the fairness of each algorithm, we show the inequality of path flows in Fig. 1B by the Gini coefficient (see Methods, Section ‘Gini coefficient’).

An ideal congestion control algorithm would have high throughput (low price of fairness) and low inequality (low Gini coefficient) for any network structure. However, such general algorithm does not exist, because the maximisation of throughput leads to inequality. Indeed, to maximise throughput in a network with constant edge capacity, a few paths receive all the network capacity of the edges they pass through, whereas a majority of paths will be allocated zero path flow (see Fig. 2C). The coexistence of both types of paths leads to the vast inequality in path flows. The α -fairness family of algorithms increases the equity of path flows with increasing α . As a consequence, however, α -fairness lowers network throughput as α increases from $\alpha = 0$ (max-flow) to $\alpha = \infty$ (max-min fairness), and this mechanism captures the efficiency-fairness trade-off.

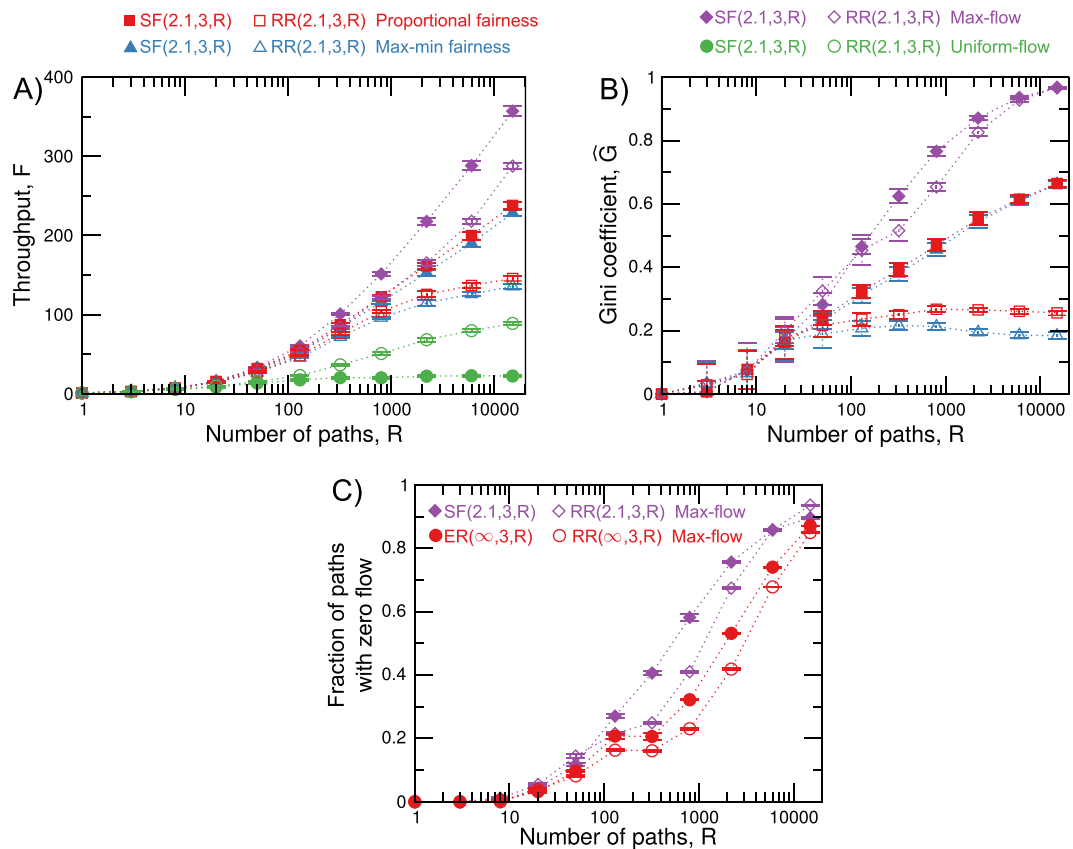


Figure 2. (A) Average network throughput and (B) Average Gini coefficient of path flows in scale-free and random regular networks for $\gamma = 2.1$ and $\langle K \rangle = 3$ as a function of the number R of shortest paths for max-flow, proportional fairness, max-min fairness, and uniform-flow. (C) The average fraction of paths that are allocated zero flow by the max-flow algorithm as a function of the number R of shortest paths in scale-free networks ($\gamma = 2.1$, $\langle K \rangle = 3$), Erdős-Rényi networks ($\langle K \rangle = 3$), and their random counterparts. For each network structure, we average over 20 randomly generated networks with $N = 2000$ nodes. Whiskers show the 95% confidence intervals.

Figure 1(A) and (B) show that the efficient algorithm (max-flow) has a high level of inequality (high Gini coefficient) and uniform-flow has low throughput (high price of fairness), thus illustrating why there is little incentive to implement these algorithms in the real-world, despite the recent interest on uniform-flow^{25–33}. In contrast, proportional fairness and max-min fairness are trade-offs between efficiency and fairness, as illustrated by the mid-range values of the price of fairness and Gini coefficient for all network structures analysed. Taken together, these features uncover the effect in network throughput and fairness of elementary (max-flow and uniform-flow) versus elaborate (proportional fairness and max-min fairness) congestion control algorithms.

We observe in Fig. 1A that for max-flow, max-min fairness and proportional fairness, the price of fairness is largely independent of network structure. Similarly, Fig. 1B shows that for max-flow, the inequality of path flows (measured by the Gini coefficient) is also largely independent of network structure. These observations suggest that proportional fairness and max-min fairness are similar algorithms with only minor dependence on network structure. Surprisingly, however, the inequality of path flow allocations for proportional fairness and max-min fairness depends mainly on network structure (see Fig. 1B). Hence, network designers that implement congestion control should be aware that scale-free and random regular network structures have similar throughput, but scale-free topologies induce larger inequality in path flows. This is especially important, because proportional fairness is often implemented in real-world networks (e.g., the Internet), and the effect of network structure on the inequality of path flows is revealed by our study of the α -fairness family of algorithms, but cannot be disentangled from an analysis of max-flow or uniform-flow only. Thus, previous studies of max-flow (large inequality)^{44,45} and uniform-flow^{25–33} (no inequality) miss the effect of network structure on the inequality of path flow allocations, and our study is a natural extension to congestion control algorithms of the body of work in the complex networks literature.

To study the effect of demand on the throughput and inequality of path flows, we analyse how these quantities vary with the number R of shortest paths in the network, and we study networks with $k = 3$ and $\gamma = 2.1$. Figure 2A is a plot of network throughput as the number R of shortest paths grows. The Gini coefficient, plot in Fig. 2B, quantifies the growth in the inequality of path flows as a function of R (see Methods, Section ‘Gini coefficient’). Network throughput increases with the number of paths, since the capacity of more edges is used. Because the network size is fixed, however, the growth in throughput slows down inevitably as more paths are

added to the network. The asymptotic value of throughput, and the way this slowing down takes place characterises the efficiency of the algorithm and network structure. Figure 2A, shows that the increase in throughput with R is much slower for uniform-flow than for the other algorithms. This result illustrates the poor performance of uniform-flow for a broad range of R values and thus complements Fig. 1, which compares algorithms only for $R = 1500$. The throughput and Gini coefficient curves do not intersect in Fig. 2, and thus the relative performance of algorithms does not change much with R .

In max-flow, the path flow allocations share edge capacity on min-cuts among a relatively small number of paths, leaving most paths with zero flow (see Fig. 2C), thus creating a large inequality in the assignment of path flows. Although max-flow is an extreme case, because it is the only analysed algorithm that can leave paths with zero flow, inequality is present in all congestion control algorithms. Indeed, the increase in throughput with R is also accompanied by a raise in the inequality of path flow allocations in max-flow, proportional fairness and max-min fairness.

Traditionally, congestion control algorithms have been designed to counter-balance the phenomenon that max-flow may exclude some paths (i.e., users) from using the network. However, little is known about the behaviour of these algorithms as a function of network structure. Here we take a step towards filling this gap by analysing the effect on network throughput of varying γ and $\langle k \rangle$. To do this, we consider the relative throughput,

$$\rho(\mathcal{F}, \mathcal{N}) = \frac{F(\mathcal{F}, \text{SF}(\gamma, \langle k \rangle, R))}{F(\mathcal{F}, \text{RR}(\gamma, \langle k \rangle, R))}, \quad (3)$$

where $F(\cdot)$ is the network throughput, \mathcal{F} is the fairness algorithm, and \mathcal{N} identifies the network structure and parameters ($\gamma = \infty$ denotes Erdős-Rényi and their random regular networks). The ratio $\rho(\mathcal{F}, \mathcal{N})$ isolates the effect of node degree distribution in throughput by comparing scale-free and Erdős-Rényi networks against the null model of random regular networks (see Methods, Section ‘Network Models’). We observe that $\lim_{k \rightarrow (N-1)} \rho(\mathcal{F}, \mathcal{N}) = 1$ because both networks in the ratio converge to fully connected graphs in this limit. Together with the relative network throughput, we consider the number $\phi(\mathcal{N}, i)$ of paths passing through edge i :

$$\phi(\mathcal{N}, i) = \sum_{j=1}^R H_{ij}, \quad (4)$$

where H is the edge-path incidence matrix (see Methods, Section ‘The mathematics of congestion control’). Because edge capacity is one ($c_i = 1$), the path flows assigned by uniform-flow are given by $1/\max\{\phi(\mathcal{N}, i) | i = 1, \dots, E\}$. Thus we have the exact relation between ρ and ϕ for uniform-flow:

$$\rho(\text{UF}, \mathcal{N}) = \frac{\max_i \phi(\text{RR}(\gamma, \langle k \rangle, R), i)}{\max_i \phi(\text{SF}(\gamma, \langle k \rangle, R), i)}. \quad (5)$$

We found, $\rho(\text{UF}, \mathcal{N}) < 1$ for all γ and $\langle k \rangle$, due to the higher maximum concentration of paths in scale-free networks than in random regular networks, i.e. $\max_i \phi(\text{SF}(\gamma, \langle k \rangle, R), i) > \max_i \phi(\text{RR}(\gamma, \langle k \rangle, R), i)$, as illustrated in Fig. 3A.

Similarly to uniform-flow, we could expect $\rho(\mathcal{F}, \mathcal{N}) < 1$ for all values of γ and $\langle k \rangle$. Surprisingly, however, as Fig. 3B–D show, $\rho(\mathcal{F}, \mathcal{N})$ can be smaller or larger than one depending on the region of parameter space $(\gamma, \langle k \rangle)$. Moreover, the dividing line $\rho(\mathcal{F}, \mathcal{N}) = 1$ is largely independent of the algorithm, indicating that network structure is the primary factor behind the relative throughput $\rho(\mathcal{F}, \mathcal{N})$ in the α -fairness family. The network structure is, however, not the only parameter influencing $\rho(\mathcal{F}, \mathcal{N})$. To show the effect of algorithms on $\rho(\mathcal{F}, \mathcal{N})$, we analyse small values of γ and $\langle k \rangle$ in Fig. 2A and Fig. 3B–D. For $\gamma = 2.1$, $\langle k \rangle = 3$ and $R = 15000$, throughput in max-flow is 24% higher for scale-free than for random regular networks. For proportional fairness and max-min fairness, however, this value increases to 63% and 68%, respectively, disentangling flow in scale-free and random regular networks in this region of parameter space, as can be observed on the highlighted cells of the heatmaps in Fig. 3B–D. Figure 2A shows that this happens because for $R = 15000$ proportionally fair and max-min fair throughput saturate in random regular networks, while throughput steadily grows with R in max-flow and all algorithms in scale-free networks.

Figure 3B–D show the dividing line between $\rho(\mathcal{F}, \mathcal{N})$ smaller and larger than one in parameter space. This dividing line is approximately the same in all α -fairness algorithms. Hence, we make use of structural network measures to gain insights on system behaviour on both sides of the line. We consider two main factors that influence network throughput. First, path length affects network throughput because paths transport a constant path flow on each of their edges. Hence, paths consume capacity from each edge they pass through, and thus the longer they are the larger the number of edges that have their available capacity reduced. Second, the pattern of path intersections influences throughput because these networks have limited edge capacity ($c = 1$). Indeed, if a large number of paths pass through a limited set of edges, network throughput is restricted by the pattern of path intersections, because the limited capacity of these edges is shared among this large set of paths. In contrast, path flows and network throughput are larger if routing is such that paths broadly avoid each other. We use $\phi(\mathcal{N}, i)$ as a simple measure to characterise the pattern of path intersections. To uncover the behaviour of path length and path intersections, we select two cells in the heatmap: ($\gamma = 2.1$, $\langle k \rangle = 3$) to represent $\rho(\mathcal{F}, \mathcal{N}) < 1$ and ($\gamma = 2.5$, $\langle k \rangle = 8$) for $\rho(\mathcal{F}, \mathcal{N}) > 1$.

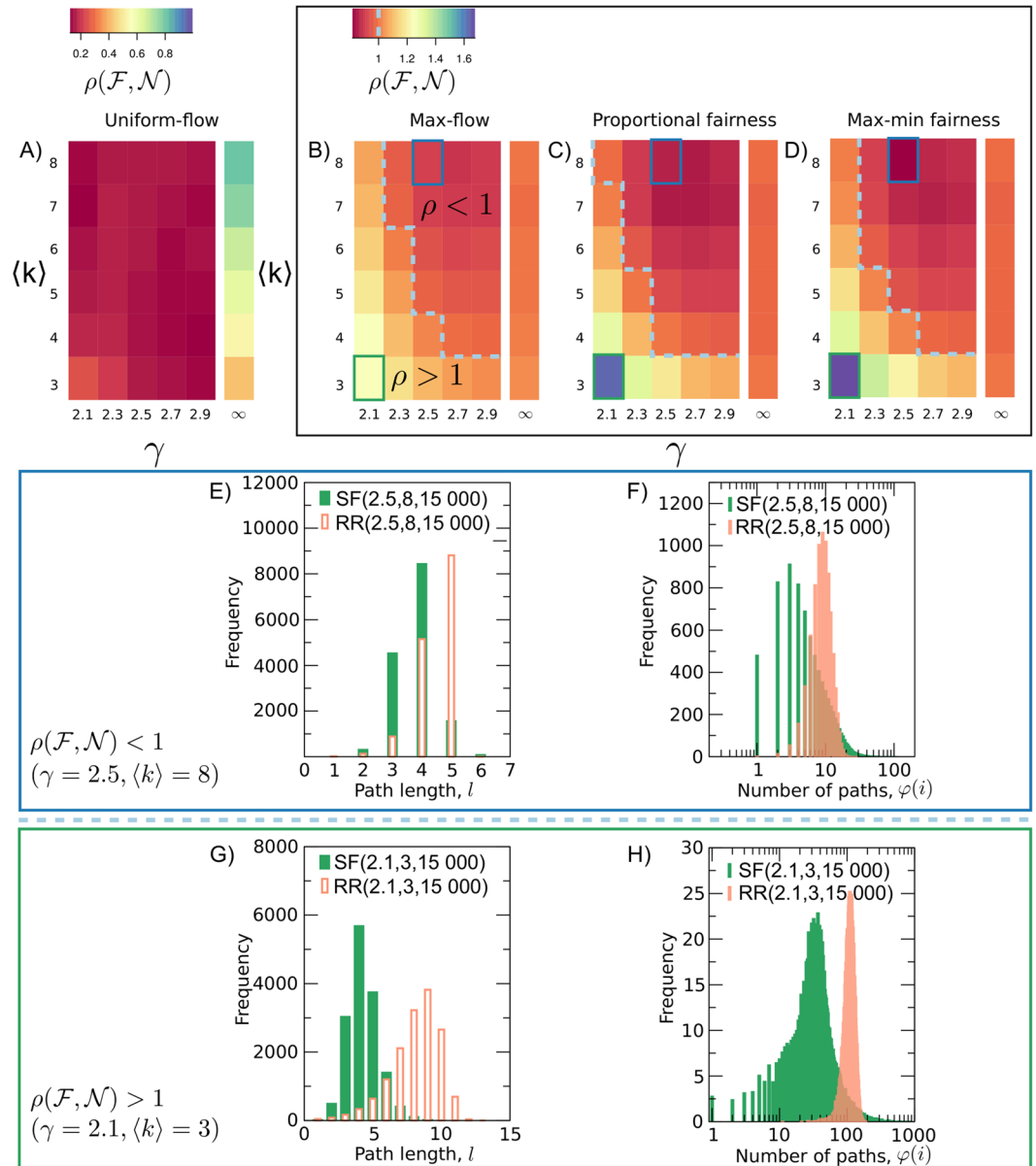


Figure 3. Heatmaps of the relative throughput $\rho(\mathcal{F}, \mathcal{N})$ for uniform-flow, panel (A), and the α -fairness family of algorithms (max-flow, proportional fairness and max-min fairness), panels (B) to (D). We highlight the (light blue dashed) line that separates the regions $\rho(\mathcal{F}, \mathcal{N}) < 1$ and $\rho(\mathcal{F}, \mathcal{N}) > 1$. Moreover, we highlight two heatmap cells in all α -fairness algorithms, to represent the regions $\rho(\mathcal{F}, \mathcal{N})$ smaller ($\gamma = 2.5, \langle k \rangle = 8$) and larger ($\gamma = 2.1, \langle k \rangle = 3$) than one. Panels (E) to (H) show the histogram of path length and the number $\phi(\mathcal{N}, i)$ of paths passing through edge i . Panels (E) and (F) examine the heatmap cell $(\gamma = 2.5, \langle k \rangle = 8)$, for which $\rho(\mathcal{F}, \mathcal{N}) < 1$, whereas panels (G) and (H) analyse the heatmap cell $(\gamma = 2.1, \langle k \rangle = 3)$, for which $\rho(\mathcal{F}, \mathcal{N}) > 1$. For each network structure, we average over 20 randomly generated networks with $N = 2000$ nodes and $R = 15000$ shortest paths.

To shed light on the mechanisms that explain the surprising $\rho(\mathcal{F}, \mathcal{N}) > 1$ region, we show in Fig. 3E and G the histogram of path length, and in Fig. 3F and H the histograms of $\varphi(\mathcal{N}, i)$ for two selected representative cells. Why do random regular networks accommodate higher flow than scale-free for $\rho(\mathcal{F}, \mathcal{N}) < 1$? An analysis of the cell $(\gamma = 2.5, \langle k \rangle = 8)$ shows the probability distribution of path length is similar in scale-free and random regular networks (see Fig. 3E). However, the distribution of the number $\varphi(\mathcal{N}, i)$ of paths passing through an edge i is heavy-tailed for scale-free, but not for random regular networks: a relatively large number of edges are crossed by many paths in scale-free than in random regular networks (see Fig. 3F). This heavy-tailed distribution of $\varphi(\mathcal{N}, i)$ is an indicator of edge congestion in scale-free networks. Moreover, in random regular networks, we observe that a small number of paths pass through many edges, but not, as one would expect in congested networks, that a large number of paths pass through a limited number of edges. Hence, the distribution of $\varphi(\mathcal{N}, i)$ illustrates why congestion tends to be higher in scale-free than random regular networks for $\rho(\mathcal{F}, \mathcal{N}) < 1$. Why do scale-free

networks accommodate higher flow than random regular for $\rho(\mathcal{F}, \mathcal{N}) > 1$? An analysis of the cell ($\gamma = 2.1$, $\langle k \rangle = 3$) shows two effects. First, paths are significantly longer in random regular ($\langle l \rangle = 8.1$) than in scale-free networks ($\langle l \rangle = 4.3$). Longer paths in random regular networks consume more network resources and also intersect with other paths more often than in the region $\rho(\mathcal{F}, \mathcal{N}) < 1$, and thus will be more congested than shorter paths. Second, we only observe higher values of $\varphi(\mathcal{N}, i)$ in scale-free networks than in random regular networks for a small number of edges. This small number of congested edges is not, however, large enough to invert the ratio $\rho(\mathcal{F}, \mathcal{N})$. Taken together, these two effects make possible that scale-free networks accommodate larger flow than random regular for low values of γ and $\langle k \rangle$.

Currently, researchers find the onset of congestion in complex networks from betweenness centrality^{26–33}, a measure that captures the number of paths that cross through nodes or edges²². This uniform-flow approach finds the onset of congestion by locating the node or edge with the highest betweenness centrality, which is crossed by the largest number of paths. Figures 1 and 2 illustrate, however, that uniform-flow severely underestimates throughput in the α -fair family of algorithms at the onset of congestion, because it allocates path flows by sharing only the capacity of the most congested node or edge among the paths that pass through it. The uniform-flow algorithm thus allocates path flows globally by maximising locally the path flows that cross through the most congested edge. Hence structural measures, such as betweenness centrality, that determine the uniform-flow allocation analytically, might not be good predictors of network throughput in more realistic congestion algorithms.

Here we are interested in the question of whether the number of paths passing through individual edges can be used to locate bottleneck edges in the α -fair family of algorithms. To investigate this problem, we use $\varphi(\mathcal{N}, i)$ as a measure of edge load that, similarly to betweenness centrality, captures the interaction between paths on network edges. Edge betweenness centrality counts the number of shortest paths, which connect all possible source-sink pairs, passing through an edge. In contrast, $\varphi(\mathcal{N}, i)$ counts the number of paths passing through the edge, which depends on the particular routing used, but not just on shortest path routing. Here, edge betweenness correlates with $\varphi(\mathcal{N}, i)$, however, because we select shortest paths with uniform probability from all shortest paths.

Figure 4(A–C) show that if the number R of paths is sufficiently large, edges with high value $\varphi(\mathcal{N}, i)$ are used up to capacity (i.e., are bottleneck edges), with negligible standard deviation. To further relate the number of paths that cross through each edge with the location of bottleneck edges, we show in Fig. 4D–F the frequency of bottleneck (shaded) and non-bottleneck (clear) edges as a function of $\varphi(\mathcal{N}, i)$ for proportional fairness (see Fig. S1 of the Supplementary Information for max-min fairness and max-flow). An analysis of the shaded area on the tail of the distributions shows that a large percentage of edges with high $\varphi(\mathcal{N}, i)$ are bottlenecks. For example, if we consider the 10% of edges with the largest value of $\varphi(\mathcal{N}, i)$ in SF(2.1, 3, 15 000) (ER(∞ , 4, 15 000)), we find that on average [MF = 95.3, PF = 95.3, MMF = 95.1]% ([MF = 99.0, PF = 99.3, MMF = 95.6]%) of these are bottlenecks, representing [12.8, 21.5, 22.8] ([12.4, 21.5, 27.7]%) of all bottleneck edges (the first value, enclosed in squared brackets, corresponds to max-flow, the second value to proportional fairness and the third value to max-min fairness). Thus, we find that, apart from a few exceptions, edges with high $\varphi(\mathcal{N}, i)$ are bottlenecks. These results are largely independent of the congestion control algorithm (parameter α) and of the network topology (exponent γ and average node degree $\langle k \rangle$). Our numerical analysis generalises the reasoning that congestion can be characterised by the structure of paths, and can be interpreted as an extension of the analytical results for uniform-flow^{26–33} to the more realistic α -fair family of congestion control algorithms combined with routing that is determined or approximated by shortest paths.

The relation between the routing of paths and the location of congested edges is crucial for both network designers and operators. Network designers wish to anticipate the location of bottlenecks during the design stage, so as to avoid weak links in the areas with the highest expected traffic, and to place the sensor and communication network infrastructure so as to minimise expenses with the overlaid control network. Likewise, network operators wish to determine the links that require a capacity upgrade if routing changes. Hence, predicting the location of bottleneck edges from the routing of paths, may be important in real-world networks that implement congestion control algorithms (e.g., the TCP/IP Internet congestion control protocol implements proportional fairness).

Discussion

We first analysed the trade-off between the efficiency and fairness of the α -fair family of congestion control algorithms in random networks. We found that the proportional and max-min fairness algorithms generate similar throughput when results are averaged over the range of network parameters and benchmarked against the null model of random regular networks. This is significant because, in real-world systems that resemble random networks, a network operator can choose to implement proportional fairness instead of max-min fairness (the fair algorithm) with little sacrifice in fairness and throughput, and with surprisingly simple decentralised algorithms⁵. We also found that the inequality of path flows in proportional and max-min fairness depends on the structure of the network: path flows are considerably more unequal in scale-free than in random regular networks. Moreover, we showed that max-flow creates high inequality in path flow allocations and uniform-flow generates low throughput, and thus these two algorithms are too elementary to be implemented in real-world networks.

We next characterised the growth in the network throughput and Gini coefficient as a function of the number R of shortest path for a chosen scale-free network structure ($\gamma = 2.1$, $\langle k \rangle = 3$) and the corresponding random regular structure. We found that the price to pay for the increase in throughput as we independently increase R or decrease α is an increase in the inequality of path flow allocations. We found inequality present in all algorithms, but prevalent in max-flow. Indeed, we showed that max-flow assigns zero path flow to a substantial fraction of paths, thus creating a significant inequality in the allocation of path flows. Our analysis indicates that these results are consistent across a wide range of the number R of paths in the network.

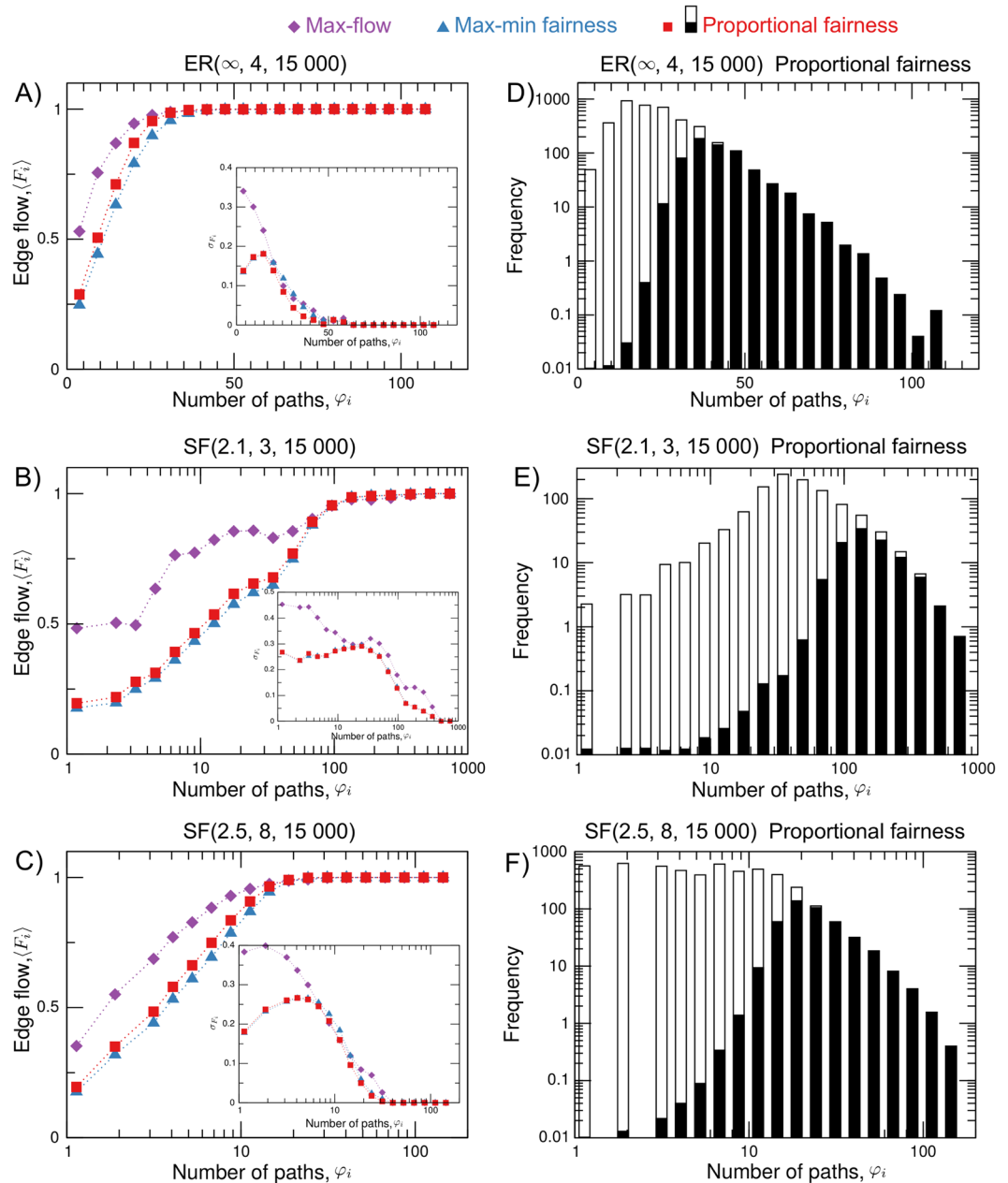


Figure 4. (A–C) Average edge flow vs. the number of paths $\varphi(\mathcal{N}, i)$ computed from 20 random realisations with $N=2000$ for Erdős–Rényi networks ER(∞ , 4, 15 000), and for the two classes of scale-free networks on the highlighted cells in Fig. 3B–D, i.e., SF(2.1, 3, 15 000) and SF(2.5, 8, 15 000). The insets show the corresponding values of the standard deviation, illustrating that the predictability of bottleneck location improves with the increase in $\varphi(\mathcal{N}, i)$. Panels D–F show the histograms of $\varphi(\mathcal{N}, i)$ for proportional fairness, where the shaded area of each bin is the proportion of bottleneck edges in the bin (see Fig. S1 in the Supplementary Information for max-flow and max-min fairness). Bottlenecks are saturated edges, that is edges for which $F_i \geq 0.999c$, where c is edge capacity.

Whereas this broad analysis over network parameters or a chosen network structure starts to disentangle the fairness of algorithms as a function of network structure, we next showed it is not enough to fully describe the network throughput. We compared the network throughput in congestion control algorithms with the null model of random regular networks in the parameter space formed by the node degree distribution exponent γ , and the average node degree $\langle k \rangle$. For the uniform-flow algorithm, we found that random regular networks, which have a more homogeneous node degree distribution than scale-free, systematically transport less flow than scale-free networks on the onset of congestion. Surprisingly, for the α -fair family of algorithms, we found that random regular networks can support less or more flow than scale-free, depending on the region of the parameter space. Moreover, we showed that the dividing line between these two regions of parameter space can be justified by

structural network measures, but that it is broadly independent of the congestion algorithm. Real-world networks could uncover further insights about the interplay between the α -fair family of algorithms and network topology.

An analysis of the effect of network structure based solely on uniform-flow would conclude that random regular networks have higher throughput than scale-free networks for all values of γ and $\langle k \rangle$. Our results show that this conclusion is misleading. The uniform-flow approach leaves networks severely under-utilised in comparison with more elaborate congestion control algorithms. We showed that uniform-flow is a crude algorithm to gain insights about the network throughput of complex networks and our findings highlight the limitations of the current line of work^{25–33} on complex networks. Congestion control protocols such as max-min fairness or proportional fairness avert congestion by allocating path flows that are determined as an outcome of an optimisation procedure. Although the result is a higher level of inequality than in uniform-flow, these protocols significantly increase the network throughput and thus are superior to uniform-flow. The price to pay for elaborate algorithms for congestion control is that the rate λ of packet production becomes source node dependent, and the critical rate is no longer found analytically. It would be hard to argue, however, that these are important factors in the modelling of real-world congested networks. Previous work on congested complex networks with uniform-flow^{25–33} identifies congestion with the appearance of the first bottleneck. Inspired by this idea, we investigated whether the number of paths passing through individual edges can locate bottleneck edges in the more realistic α -fair family of algorithms. We found that congestion on complex networks can be found not only on the edge with the largest number of paths, but on a bigger set of edges. Such edges are crossed by a high number of paths, and thus have high edge betweenness, if the routing also follows the shortest paths.

In summary, we combined two very well established and related, but so far separated, research areas: congestion control and complex networks. We explained the main milestones in the more than 30-year old line of work in congestion control, and we compared the results from this body of literature with congestion control algorithms studied in the complex networks community in the last 15 years, which identifies the onset of congestion by considering homogeneous (uniform) path flows. On the one hand, our results show the severe limitations of the uniform-flow approach, which is the conventional algorithm to study congestion in the complex networks literature. On the other hand, we illustrated that structural characteristics typically favoured in complex networks can characterise congested edges for the α -fair family of control algorithms, an approximation that has not received enough attention in the field of congestion control. We believe that our paper has potential to open the work in congestion control to complex networks scientists and, vice-versa that it will reveal the rich field of network science to researchers working on congestion control.

Methods

The mathematics of congestion control. Let $\mathcal{G} = (\mathcal{V}, \varepsilon)$ be an undirected and connected graph, with node-set \mathcal{V} and edge-set ε , such that edge $i \in \varepsilon$ has capacity c_i . The network has N nodes and E edges, and a set of R source and sink pairs (s_j, t_j) with $s_j, t_j \in \mathcal{V}$ for $j = 1, \dots, R$. Each source and sink pair (s_j, t_j) is connected by a path r_j , such that $\mathcal{R} = \cup_{j=1}^R \{r_j\}$ is the set of all source to sink paths on the network. The relationship between edges and paths is given by the *edge-path incidence matrix* H , such that $H_{ij} = 1$ if edge i belongs to path r_j , and $H_{ij} = 0$ otherwise. Matrix H has dimensions $E \times R$, and maps paths to the edges contained in these paths. All edges of a path r_j transport the same *path flow* f_j . The flow F_i on edge i is then the sum of path flows over all paths that cross the edge:

$$F_i = \sum_{j=1}^R H_{ij} f_j. \quad (6)$$

A vector f of path flows is *feasible* if $Hf \leq c$ and $f_j \geq 0$ for $j = 1, \dots, R$, where c is the vector of edge capacities. An edge is a *bottleneck* if the flow passing through it is equal to the edge capacity. We define the network congestion control problem⁵:

$$\begin{aligned} & \underset{f}{\text{maximise}} && U(f) = \sum_{j=1}^R U_j(f_j, \alpha) \\ & \text{subject to} && Hf \leq c \\ & && f_j \geq 0 \quad \text{for } j \in 1, \dots, R, \end{aligned} \quad (7)$$

where $\alpha \geq 0$ is a parameter and $U_j(f_j, \alpha)$ is defined by Eq. (1).

In max-flow ($\alpha=0$), to increase a path flow by ε , we have to decrease a set of other power path flows, such that the sum of the decreases is larger or equal to ε . In contrast, in max-min fairness ($\alpha \rightarrow \infty$), to increase a path flow by ε , we have to decrease at least by ε a set of other path flows that are less or equal to the former. Finally, to increase a path flow by a percentage ε in proportional fairness ($\alpha = 1$), we have to decrease a set of other power path flows, such that the sum of the percentage decreases is larger or equal to ε ⁵.

Max-min fairness. Formally, a vector f of path flows is max-min fair, if it is feasible and if for any other feasible vector f' of path flows, there exists a path $r_j \in \mathcal{R}$: $f'_j > f_j$ implies that there exists another path $r_l \in \mathcal{R}$: $f'_l < f_l$ and $f_l \leq f'_l$ ¹⁷. The max-min fairness allocation is the solution of problem⁷ for $\alpha \rightarrow \infty$. The allocation is typically found, however, with an iterative algorithm¹⁷ that locates the bottleneck edges. The algorithm first increases all path flows uniformly from zero until it maximises the smallest path flows, that is until it finds the first bottleneck edges. The path flows on paths that pass through these bottlenecks cannot be increased because the edges are used

to their full capacity, and hence the algorithm fixes these path flows, and updates the residual capacity still available to other paths. Next, the process is repeated for the paths that do not have yet a fixed path flow. To describe the algorithm formally, we define $\mathcal{R}^{(m)}$ to be the set of paths on the network at iteration m , and $\mathcal{R}_i^{(m)}$ to be the subset of paths in $\mathcal{R}^{(m)}$ that cross through edge i . Before we start the algorithm, we assign $\mathcal{R}^{(1)} = \mathcal{R}$ and $c_i^{(1)} = c_i$ for all edges, and a path flow $f_j^{(0)} = 0$ to each path $r_j \in \mathcal{R}^{(1)}$. Next, we initialise the iteration counter $m = 1$. In the first step of the MMF algorithm, for each edge i with non-zero capacity that belongs to at least one path, we define the edge capacity divided equally among all paths that pass through the edge at iteration m of the algorithm as:

$$s_i^{(m)} = c_i^{(m)} / |\mathcal{R}_i^{(m)}|, \tag{8}$$

for all $c_i^{(m)} \neq 0$. We then find the minimum of $s_i^{(m)}$, given by

$$\Delta f^{(m)} = \min_{i \in \varepsilon, c_i^{(m)} \neq 0} s_i^{(m)}. \tag{9}$$

In the second step of the MMF algorithm, we increase all path flows of paths in $\mathcal{R}^{(m)}$ by $\Delta f^{(m)}$, such that

$$f_j^{(m)} = \begin{cases} f_j^{(m-1)} + \Delta f^{(m)} & \text{if } r_j \in \mathcal{R}^{(m)} \\ f_j^{(m-1)} & \text{if } r_j \in \mathcal{R} \setminus \mathcal{R}^{(m)} \end{cases} \tag{10}$$

The effect is to saturate the set of bottleneck edges $\varepsilon_B^{(m)} = \{i \in \varepsilon: \sum_{j=1}^R H_{ij} \Delta f^{(m)} = c_i^{(m)}\}$, and consequently also to saturate the set of paths that contain at least one bottleneck edge. Next, we create a residual network, by subtracting the capacity used by the path flows,

$$c_i^{(m+1)} = c_i^{(m)} - \sum_{j=1}^R H_{ij} \Delta f^{(m)}. \tag{11}$$

Note that all bottleneck edges will be saturated, that is each will have $c_i^{(m+1)} = 0$ after this step. We also say that all paths that contain at least one bottleneck edge are saturated paths, to mean that their path flow will not be increased in subsequent iterations of the MMF algorithm. We say that $\mathcal{R}^{(m+1)}$ is the set of augmenting paths because the path flows of paths in $\mathcal{R}^{(m+1)}$ can still be increased in subsequent iterations of the algorithm, and update it following:

$$\mathcal{R}^{(m+1)} = \mathcal{R}^{(m)} \setminus \bigcup_{i \in \varepsilon_B^{(m)}} \mathcal{R}_i^{(m)}. \tag{12}$$

Finally, if $\mathcal{R}^{(m+1)}$ is not empty, we increase the iteration counter $m \leftarrow m + 1$, and go back to the first step, otherwise we stop.

Proportional fairness. A vector of path flows $f^* = (f_1^*, \dots, f_R^*)$ is *proportionally fair* if it is feasible and if for any other feasible vector of path flows f , the sum of proportional changes in the path flows is non-positive^{37,46}:

$$\sum_{j=1}^R \frac{f_j - f_j^*}{f_j^*} \leq 0. \tag{13}$$

The proportionally fair allocation is found from problem⁷ with the utility function in Eq 1. for $\alpha = 1$, and we refer to this problem as the primal⁵. The optimisation problem is convex because the aggregate utility $U(f)$ is concave and the inequality constraints are convex. Thus, any locally optimal point is also a global optimum, and we can use results from the theory of convex optimisation to find the proportional fair flow allocation (see refs 47 and 48 for a brief introduction to Lagrange multipliers, and ref. 36 on convex optimisation). The Lagrangian is given by refs 37 and 46.

$$L(f, \mu) = \sum_{j=1}^R \log(f_j) + \mu^T (c - Hf), \tag{14}$$

where $\mu = (\mu_1, \dots, \mu_E)$ is a vector of Lagrange multipliers. The Lagrange dual function³⁶ is then given by $\sup_f L(f, \mu)$, which is easily determined analytically by $\partial L(f^*, \mu^*) / \partial f = 0$ as

$$\frac{\partial L(f^*, \mu^*)}{\partial f_j^*} = \frac{1}{f_j^*} - \sum_{i=1}^E H_{ij} \mu_i^* = 0 \Leftrightarrow f_j^* = \frac{1}{\sum_{i=1}^E H_{ij} \mu_i^*} \text{ for } j \in 1, \dots, R, \tag{15}$$

and thus

$$\sup_f L(f, \mu) = - \sum_{j=1}^R \log \left(\sum_{i=1}^E H_{ij} \mu_i \right) + \sum_{i=1}^E \mu_i c_i - R. \tag{16}$$

After removing the constant term in equation (16) and converting to a maximisation problem, we obtain the dual problem^{37,46}

$$\begin{aligned} & \underset{\mu}{\text{maximise}} \quad V(\mu) = \sum_{j=1}^R \log \left(\sum_{i=1}^E H_{ij} \mu_i \right) - \sum_{i=1}^E \mu_i c_i \\ & \text{subject to} \quad \mu_i \geq 0 \quad \text{for } i \in 1, \dots, E, \end{aligned} \quad (17)$$

where $\mu = (\mu_1, \dots, \mu_E)$ is a vector of dual variables. The primal problem is convex and the inequality constraints are affine. Hence, Slater's condition is verified and thus strong duality holds. This means that the *duality gap*, i.e., the difference between the optimal of the primal problem⁷ and the optimal of the dual problem¹⁷, is zero³⁶. The primal objective function depends on R variables (the path flows) and is constrained by an affine system of equations, whereas the dual objective function depends on E variables (the edges) and is constrained only by the condition that the dual variables are non-negative. Thus, the dual problem¹⁷ is more efficient to solve than the primal when the number of paths exceeds number of network edges. The optimal path flows can then be recovered from the optimal Lagrange multipliers with Eq. 15.

The decentralized implementation of proportional fairness relies on a feed-back mechanism on path flows⁵: multiplicatively decrease path flows of paths that pass through bottlenecks and additively increase all other path flows. The combination of the fast correction (multiplicative decrease) and slow ramp-up (additive increase) is the mechanism behind the TCP internet congestion control protocol. Crucially, this mechanism requires each bottleneck to send a feedback signal to the sender of each path, with the information that the path flow should be additively increased or multiplicatively decreased. Knowledge of where to place sensors and where to connect to the communication network that sends the feedback signals is thus important for the network designer and operator.

Uniform-flow. The uniform-flow allocation can be found for any $\alpha \geq 0$ from problem⁷ and Eq. (1), with an additional set of constraints ensuring that path flows are uniform:

$$f_1 = f_j \text{ for } j \in 2, \dots, R. \quad (18)$$

The optimal uniform-flow allocation is α -invariant, because $U_j(f_j, \alpha)$ is a monotonically increasing function of f_j for any $\alpha \geq 0$. Algorithmically, the uniform-flow allocation can also be found as the solution to the first iteration ($m = 1$) of the max-min fairness algorithm, since the algorithm maximises the minimum path flow allocation, and all path flows are the same at the end of the first iteration.

The onset of congestion in complex networks is often determined by the uniform-flow allocation^{25–33}. At each time step, source node n generates a packet with probability λ and sends it towards the sink node along a shortest path. The expected number of packets in the network at each time step is λND , where D is the average shortest path length. Moreover, the probability that a packet will pass through a node n_{max} with the largest betweenness is $B_{n_{max}} / \sum_{n=1}^N B_n$ (here, the betweenness centrality B_n of a node n equals the number of shortest paths between all pairs of nodes in the network going through node n [22, page 28]). The average number of packets that node n_{max} receives at each time step is thus $Q_{in} = \lambda D B_{n_{max}} / ((N - 1)D)$, where we used the simplification that the sum of the betweenness values of all nodes is the number of pairs of nodes on the network multiplied by the average path length, $\sum_{n=1}^N B_n = N(N - 1)D$. At each time step, the node with the highest betweenness can deliver $Q_{out} = c_{n_{max}}$ packets, and hence the onset of congestion is given by $Q_{out} = Q_{in}$, that is

$$\lambda_c = c_{n_{max}} (N - 1) / B_{n_{max}}. \quad (19)$$

This deduction considers a network of capacitated nodes, but we can have capacity constraints on the links instead, and packets may queue at the nodes for service. Congestion control algorithms are similar for node and link capacity, and here we analyse random networks with link capacity, because this is the standard in the modelling of communication networks⁵.

The reasoning leading to Eq. (19) assumes that the number $\lambda_c / (N - 1)$ of packets injected into the network per path at each time step is the same for all paths, and that it is determined by the ratio between the node capacity $c_{n_{max}}$ and the number of paths $B_{n_{max}} / (N - 1)$ passing through node n_{max} . The obvious drawback of this approach is that the estimate of λ_c in Eq. (19) considers only the first bottleneck to appear in the network, and thus underestimates the load typically present in congested networks.

Avoiding congestion collapse on the ring lattice. Consider a ring lattice of N nodes, each connected to its nearest neighbours by an edge with finite capacity c , as illustrated in Fig. 5. We relax the constraint that all edges of path r_j transport the same path flow f_j , allow instead queues to build up at the nodes, and thus edge flows to differ on edges along a path. User j injects a flow $f^{(0)}$ into the network at node $j \in \{1, \dots, N\}$ on a *short* path j ; the flow $f_j^{(1)}$ passes from node j to node $(j + 1) \pmod{N}$; the flow $f_j^{(2)}$ passes from node $(j + 1) \pmod{N}$ to node $(j + 2) \pmod{N}$ and exits the network. Consider also that node 1 is a source and sink of a *long* path, user $N + 1$, that passes through all nodes, with flow $f_{N+1}^{(j)}$ on the edge linking node j to node $(j + 1) \pmod{N}$. The subscript j identifies the short path, as well as its first node. The superscripts (1) and (2) index the edges the short paths pass through, and superscript (j) indexes edges the long path cross.

To illustrate the mechanism of congestion collapse, we assume a simple congestion control scheme that distributes the total flow $F_j = f^{(0)} + f_{(N+j-2) \pmod{N}+1}^{(1)} + f_{N+1}^{(j-1)}$ at node j proportionally to the paths that pass through the node:

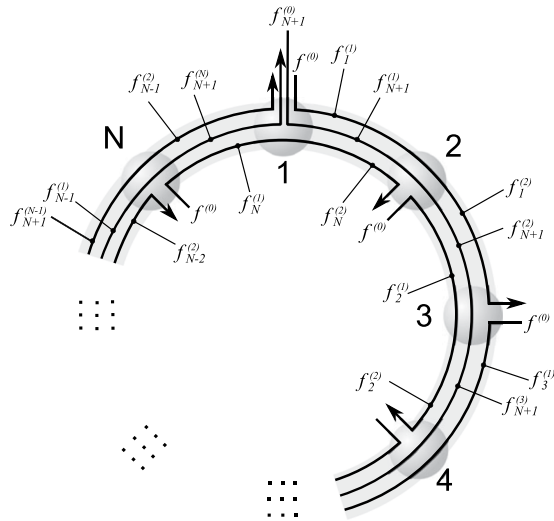


Figure 5. Ring lattice with N nodes and edges, such that each edge has capacity c , with paths and nodes indexed clockwise from the top. Short paths inject a flow $f^{(0)}$ at each node and use the network for two edges before exiting, and a long path injects a flow $f_{N+1}^{(0)}$ at node 1 and uses all edges of the network. If we distribute edge capacity proportionally to flows entering the edge, the network throughput goes to zero as the flow injected at each node increases, because queues of increasing length build up at each node.

$$f_j^{(1)} = \min\left\{f^{(0)}, \frac{f^{(0)}}{E_j} c\right\} \tag{20}$$

$$f_{(N+j-2)(\text{mod } N)+1}^{(2)} = \min\left\{f_{(N+j-2)(\text{mod } N)+1}^{(1)}, \frac{f_{(N+j-2)(\text{mod } N)+1}^{(1)}}{E_j} c\right\} \tag{21}$$

$$f_{N+1}^{(j)} = \min\left\{f_{N+1}^{(j-1)}, \frac{f_{N+1}^{(j-1)}}{E_j} c\right\}. \tag{22}$$

The network is not congested for $f^{(0)} < (c - f_{N+1}^{(0)})/2$, and in this case the throughput is $Nf^{(0)} + f_{N+1}^{(0)}$. If the network is congested, flows decrease along a path, that is $f^{(0)} > f_j^{(1)} > f_j^{(2)}$, and queues build up at each node. The proportional allocation of edge capacities may motivate individual users to increase $f^{(0)}$ in order to receive a larger share of network capacity. However, as the flow $f^{(0)}$ injected at each short path grows, the length of queues at the nodes also grow and the network throughput decreases. This *congestion collapse* is a consequence of the collapse of throughput both for short and long paths. Indeed, in the limit $\{f^{(0)}, f_{N+1}^{(0)}\} \rightarrow \infty$, the system of Eqs (20–22) yields $f_1^{(1)} = c/2$, $f_j^{(1)} = c$ for $j \geq 2$ and $f_j^{(2)} = 0$ for all short paths; and $f_{N+1}^{(1)} = c/2$, and $f_{N+1}^{(j)} = 0$ for $j \geq 2$ on the long path. Hence, $\sum_{j=1}^N f_j^{(2)} + f_{N+1}^{(N)} \rightarrow 0$.

Let us now assume the more restrictive condition that $N \geq 2$ paths carry a path flow, and consider the effect of controlling congestion. Because the path flow is constant on all edges along a path, we have a path flow f for every short path, and a path flow f_{N+1} for the long path, such that $f_{N+1} + 2f = c$ at every edge, and thus

$$f = (c - f_{N+1})/2. \tag{23}$$

One max-flow allocation is $f = c/2$ and $f_{N+1} = 0$, leaving user $N + 1$ with no access to the network, with a network throughput of $Nc/2$. The max-min fair solution is $f = f_{N+1} = c/3$, with a network throughput of $(N + 1)c/3$. The proportionally fair solution is found by maximising $U = \log f_{N+1} + \sum_{j=1}^N (\log(c - f_{N+1}) - \log 2)$ over f_{N+1} , yielding the path flow on the long path:

$$f_{N+1}^* = \frac{c}{N + 1}. \tag{24}$$

Combining Eqs 23 and 24 yields the proportionally fair path flow on the short paths:

$$f^* = \frac{N}{2(N+1)}c. \quad (25)$$

Hence, the proportional fair network throughput is $(N^2 + 2)c/(2(N + 1))$. As the size of the network increases, the proportional fair allocation approaches the max-flow solution, leaving the long path with ever smaller flow. In contrast, the max-min fair protocol assigns the same allocation to all paths independently of network size, at the expense of having a lower throughput than proportional fairness and max-flow. Thus, this example shows that proportional fairness and max-flow generate the same throughput on an infinitely size ring lattice, and that this is higher than the throughput provided by max-min fairness.

Network Models. We are interested in global congestion patterns, and thus require connected networks. We generate undirected, unweighted (i.e., unit capacity) and connected scale-free (SF) networks following the static model⁴⁹, and call these the *substrate networks*. We start with $N = 2000$ disconnected nodes and assign a weight $w_i = i^{-\beta}$ to each node i ($i = 1, \dots, N$), where $\beta \in [0, 1)$. We randomly select two nodes i and j with probability proportional to w_i and w_j , respectively, and connect them if they are not yet connected, avoiding self-loops and multiedges. We repeat this procedure until the average node degree of the largest connected component is $\langle k \rangle$, and keep only the largest connected component. The degree distribution follows a power-law with exponent $\gamma = (1 + \beta)/\beta$. We generate scale-free networks with average degree $\langle k \rangle \in \{3, 4, 5, 6, 7, 8\}$, and $\gamma = \{2.1, 2.3, 2.5, 2.7, 2.9, 3.1\}$. We treat Erdős-Rényi networks as a special case of scale-free networks for $\beta = 0$ ($\gamma = \infty$). This procedure generates networks with different number of nodes and edges, dependent on γ and $\langle k \rangle$. To overcome this limitation, we compare each network generated by the static model with a connected random regular (RR) graph with the same average degree as the scale-free (SF) or Erdős-Rényi (ER) network. This RR network has the same number of nodes and edges as the corresponding SF network generated with the static model, and thus can be seen as a rewired graph, such that each node has a fixed degree. We then use the RR network as a null model, against which we analyse the features of the corresponding SF or ER network.

In real-world flow networks, most flow between a pair of source and sink nodes will be located over only one route⁵⁰, and that is typically the shortest path because it minimises the cost of transport^{51,52}. Researchers have explored a variety of alternatives to shortest path routing^{30,51,53}, yet there is no clear alternative to shortest path routing from this effort and these algorithms have often been designed for specific models and scale-free networks. An alternative way to determine the routing would be to find all elementary paths (paths that do not traverse any node more than once) between source and sink node pairs, but this is only practical for small networks because the number of paths grows exponentially with network size⁵⁴. Hence, we analyse routing between a source and sink pair along shortest paths only. We chose R shortest paths with uniform probability from the set of all shortest paths, and extract the *transport overlay network*⁵⁵ composed of the edges that are crossed by at least one path on the substrate network. This transport overlay network is the union of all R shortest paths, and is the subgraph of the substrate network that carries flow.

Gini coefficient. To characterise inequalities in the flow allocations, we analyse the Gini coefficient of path flows. The Gini coefficient is defined as⁵⁶

$$G = \frac{1}{2\mu} E[|u - v|] = \frac{1}{2\mu} \int_0^\infty \int_0^\infty |u - v| g(u) g(v) du dv, \quad (26)$$

where u and v are independent identically distributed random variables with probability density g and mean μ . In other words, the Gini coefficient is one half of the mean difference in units of the mean. The difference between the two variables receives a small weight in the tail of the distribution, where $g(u)g(v)$ is small, but a relatively large weight near the mode. Hence, G is more sensitive to changes near the mode than to changes in the tails. For a random sample $(x_l, l = 1, 2, \dots, n)$, the empirical Gini coefficient, \hat{G} , may be estimated by a sample mean

$$\hat{G} = \frac{\sum_{l=1}^n \sum_{o=1}^n |x_l - x_o|}{2n^2 \mu}. \quad (27)$$

The Gini coefficient is used as a measure of inequality, because a sample where the only non-zero value is x has $\mu = x/n$ and hence $\hat{G} = (n - 1)/n \rightarrow 1$ as $n \rightarrow \infty$, whereas $\hat{G} = 0$ if all data points have the same value

References

- Ottino, J. M. Engineering complex systems. *Nature* **427**, 399–399 (2004).
- Scala, A. *et al.* *Power Grids, Smart Grids and Complex Networks*, 97–110. NATO Science for Peace and Security Series C: Environmental Security (Springer Netherlands, 2014).
- Rosas-Casals, M. & Solé, R. Analysis of major failures in Europe's power grid. *International Journal of Electrical Power & Energy Systems* **33**, 805–808 (2011).
- Carvalho, R., Buzna, L., Gibbens, R. & Kelly, F. Critical behaviour in charging of electric vehicles. *New J. Phys.* **17**, 095001 (2015).
- Kelly, F. & Yudovina, E. *Stochastic Networks* (Cambridge University Press, 2014).
- Bidgoli, H. *The Internet Encyclopedia*, vol. II (John Wiley & Sons Inc, 2004).
- Jacobson, V. Congestion avoidance and control. In *Proceedings of SIGCOMM* **88**, 314–329 (ACM, 1988).
- Carvalho, R. *et al.* Resilience of Natural Gas Networks during Conflicts, Crises and Disruptions. *PLoS One* **9**, e90265 (2014).
- Carvalho, R. *et al.* Robustness of trans-European gas networks. *Phys. Rev. E* **80**, 0161069 (2009).
- Giridhar, A. & Kumar, P. R. Scheduling Automated Traffic on a Network of Roads. *IEEE Transactions on Vehicular Technology* **55**, 1467–1474 (2006).
- Tachet, R. *et al.* Revisiting Street Intersections Using Slot-Based Systems. *PLOS ONE* **11**, 1–9 (2016).
- Liu, Y.-Y., Slotine, J.-J. & Barabási, A.-L. Controllability of complex networks. *Nature* **473**, 167–173 (2011).

13. Devine, M. T., Gleeson, J. P., Kinsella, J. & Ramsey, D. M. A Rolling Optimisation Model of the UK Natural Gas Market. *Networks and Spatial Economics* **14**, 209–244 (2014).
14. Nepusz, T. & Vicsek, T. Controlling edge dynamics in complex networks. *Nature Physics* **8**, 568–573 (2012).
15. Szolnoki, A., Perc, M. & Szabo, G. Accuracy in strategy imitations promotes the evolution of fairness in the spatial ultimatum game. *Epl* **100**, 28005 (2012).
16. Bertsekas, D. P. & Gallager, R. *Data Networks* (Prentice Hall, 1992).
17. Pioro, M. & Medhi, D. *Routing, Flow, and Capacity Design in Communication and Computer Networks* (Morgan Kaufmann, 2004).
18. Srikant, R. *The Mathematics of Internet Congestion Control* (Birkhäuser, Boston, 2004).
19. Albert, R. & Barabasi, A. L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47–97 (2002).
20. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D. U. Complex networks: Structure and dynamics. *Phys. Rep.-Rev. Sec. Phys. Lett.* **424**, 175–308 (2006).
21. Caldarelli, G. *Scale-Free Networks: Complex Webs in Nature and Technology* (Oxford University Press, New York, 2007).
22. Cohen, R. & Havlin, S. *Complex Networks: Structure, Robustness and Function* (Cambridge University Press, New York, 2010).
23. Newman, M. *Networks: An Introduction* (Oxford University Press, New York, 2010).
24. Kobayashi, H., Mark, B. L. & Turin, W. *Probability, Random Processes, and Statistical Analysis: Applications to Communications, Signal Processing, Queueing Theory and Mathematical Finance* (Cambridge University Press, 2011).
25. Zhao, L., Lai, Y. C., Park, K. & Ye, N. Onset of traffic congestion in complex networks. *Phys. Rev. E* **71**, 026125 (2005).
26. Guimerà, R., Díaz-Guilera, A., Vega-Redondo, F., Cabrales, A. & Arenas, A. Optimal network topologies for local search with congestion. *Physical Review Letters* **89**, 248701 (2002).
27. Guimera, R., Arenas, A., Diaz-Guilera, A. & Giral, F. Dynamical properties of model communication networks. *Phys. Rev. E* **66**, 026704 (2002).
28. Cholvi, V., Laderas, V., Lopez, L. & Fernandez, A. Self-adapting network topologies in congested scenarios. *Phys. Rev. E* **71**, 035103(R) (2005).
29. Duch, J. & Arenas, A. Scaling of fluctuations in traffic on complex networks. *Physical Review Letters* **96**, 218702 (2006).
30. Sreenivasan, S., Cohen, R., Lopez, E., Toroczkai, Z. & Stanley, H. E. Structural bottlenecks for communication in networks. *Phys. Rev. E* **75**, 036105 (2007).
31. Danon, L., Arenas, A. & Diaz-Guilera, A. Impact of community structure on information transfer. *Phys. Rev. E* **77**, 036103 (2008).
32. Yang, R., Wang, W. X., Lai, Y. C. & Chen, G. R. Optimal weighting scheme for suppressing cascades and traffic congestion in complex networks. *Phys. Rev. E* **79**, 026112 (2009).
33. Chen, Y. Z. *et al.* Extreme events in multilayer, interdependent complex networks and control. *Scientific Reports* **5**, 17277–17277 (2015).
34. Mo, J. H. & Walrand, J. Fair end-to-end window-based congestion control. *IEEE-ACM Trans. Netw.* **8**, 556–567 (2000).
35. Ahuja, R. K., Magnanti, T. L. & Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications* (Prentice Hall, 1993).
36. Boyd, S. & Vandenberghe, L. *Convex Optimization* (Cambridge University Press, New York, 2004).
37. Kelly, F. P., Maulloo, A. K. & Tan, D. K. H. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* **49**, 237–252 (1998).
38. Carvalho, R., Buzna, L., Just, W., Helbing, D. & Arrowsmith, D. K. Fair sharing of resources in a supply network with constraints. *Phys. Rev. E* **85**, 046101 (2012).
39. Chiu, D. M. & Jain, R. Analysis of the increase and decrease algorithms for congestion avoidance in computer-networks. *Computer Networks and ISDN Systems* **17**, 1–14 (1989).
40. Johnson, S. D. & D'Souza, R. M. Inequality and Network Formation Games. *Internet Mathematics* **11**, 253–276 (2015).
41. Low, S. H., Paganini, F. & Doyle, J. C. Internet congestion control. *IEEE Control Systems Magazine* **22**, 28–43 (2002).
42. Massoulié, L. & Roberts, J. Bandwidth sharing: Objectives and algorithms. *IEEE-ACM Trans. Netw.* **10**, 320–328 (2002).
43. Bertsimas, D., Farias, V. F. & Trichakis, N. The Price of Fairness. *Oper. Res.* **59**, 17–31 (2011).
44. Carmi, S., Wu, Z., López, E., Havlin, S. & Eugene Stanley, H. Transport between multiple users in complex networks. *The European Physical Journal B* **57**, 165–174 (2007).
45. Carmi, S., Wu, Z., Havlin, S. & Stanley, H. E. Transport in networks with multiple sources and sinks. *Epl* **84**, 28005 (2008).
46. Tan, D. K. H. *Mathematical Models of Rate Control for Communication Networks*. Ph.D. thesis, Statistical Laboratory, University of Cambridge (1999).
47. Courant, R. & Hilbert, D. *Methods of Mathematical Physics*. vol. 1 (Wiley-Interscience, 1989).
48. Ball, K. *Optimization and Lagrange Multipliers*, chap. III. **64**, 255–257 (Princeton University Press, New Jersey, 2008).
49. Goh, K. I., Kahng, B. & Kim, D. Universal behavior of load distribution in scale-free networks. *Physical Review Letters* **87**, 278701 (2001).
50. Nace, D. & Pioro, M. M.-M. Fairness and Its Applications to Routing and Load-Balancing in Communication Networks: A Tutorial. *IEEE Commun. Surv. Tutor.* **10**, 5–17 (2008).
51. Danila, B., Yu, Y., Marsh, J. A. & Bassler, K. E. Optimal transport on complex networks. *Phys. Rev. E* **74**, 046106 (2006).
52. Barthelemy, M. Spatial networks. *Phys. Rep.-Rev. Sec. Phys. Lett.* **499**, 1–101 (2011).
53. Goh, K. I., Noh, J. D., Kahng, B. & Kim, D. Load distribution in weighted complex networks. *Phys. Rev. E* **72**, 4 (2005).
54. Ogryczak, W., Luss, H., Pioro, M., Nace, D. & Tomaszewski, A. Fair Optimization and Networks: A Survey. *J. Appl. Math.* **25** (2014).
55. Wang, H. J., Hernandez, J. M. & Van Mieghem, P. Betweenness centrality in a weighted network. *Phys. Rev. E* **77**, 046105 (2008).
56. Ullah, A. & Giles, D. E. A. *Handbook of Applied Economic Statistics*. (CRC Press, New York, 1998).

Acknowledgements

We thank Matej Cebecauer for conducting numerical experiments and analysing results in early versions of the manuscript. We thank Dirk Helbing for granting access to the ETHZ Brutus high-performance cluster. This work was supported by the Alan Turing Institute, call for collaboration in the Lloyd's Register Foundation Programme to support data-centric engineering under grant number LRF16-05, by the Engineering and Physical Sciences Research Council under grant number EP/I016023/1, by VEGA (project 1/0463/16), APVV (project APVV-15-0179) and by FP 7 (project ERAdiate 621386).

Author Contributions

L.B. and R.C. conceived the experiment(s), L.B. and R.C. wrote the manuscript

Additional Information

Supplementary information accompanies this paper at doi:10.1038/s41598-017-09524-3

Competing Interests: The authors declare that they have no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2017