

RESEARCH

Open Access



An analytical solution to the multicommodity network flow problem with weighted random routing

Onuttom Narayan¹ and Iraj Saniee^{2*} 

*Correspondence:

iis@research.bell-labs.com

² Mathematics & Algorithms

Group, Bell Labs, Nokia,

Murray Hill, NJ 07974, USA

Full list of author information is available at the end of the article

Abstract

We derive an analytical expression for the mean load at each node of an arbitrary undirected graph for the non-uniform multicommodity flow problem under weighted random routing. We show the mean load at each node, net of its demand and normalized by its (weighted) degree, is a constant equal to the trace of the product of two matrices: the Laplacian of the demand matrix and the generalized inverse of the graph Laplacian. For the case of uniform demand, this constant reduces to the sum of the inverses of the non-zero eigenvalues of the graph Laplacian. We note that such a closed-form expression for the network capacity for the general multicommodity network flow problem has not been reported before, and even though (weighted) random routing is not a practical procedure, it enables us to derive a (tight) upper bound for the capacity of the network under more standard routing policies. Using this new expression, we compute network capacity for a sample of demand matrices for some prototypical networks, including uniform demand (one unit between all node pairs) and broadcast demand (one unit between a source node and each other node as destination), and finally derive estimates of the mean load in some asymptotic cases.

Keywords: Multicommodity flow, Network load, Steady state, Laplacian of a graph, Laplacian of the demand matrix, Weighted random walk

Introduction

The study of network capacity, or load, for arbitrary demand matrices is over half a century old, and goes back to the pioneering work of Ford and Fulkerson (1956) and Elias et al. (1956) for the single commodity and to early attempts (Hu 1963; Lomonosov 1985; Shahrokhi and Matula 1990; Papernov 1976; Okamura and Seymour 1981) for the multicommodity flow solutions of the problem. This rather large literature aims to provide characterizations of the load, or the minimal capacity required to route the specified demand, in terms of nodal or link capacities needed, based on cut values. For the case of the single commodity model cut values provide both necessary and sufficient conditions, while for the multicommodity case, at best they provide necessary conditions.

The existing literature on multicommodity flow problem typically deals with either necessary or sufficient conditions for the existence of solutions or, more frequently,

computational schemes for the minimum cost routing of demands. There are no known closed-form solutions, or complete characterizations, of the solution to this problem under standard routing policies, such as shortest path or minimum cost routing. The reader is invited to consult multiple surveys of the state of the art in this literature (Kennington 1978; Assad 1978; Ouorou et al. 2000; Awerbuch and Leighton 2005; Wang 2018a, b). By contrast, in this paper, we report on a closed-form solution that provides a complete description of the mean load at each node and each link of a network under stochastic routing of demand governed by the weights of the edges of the network, as we describe below.

Single commodity or multicommodity network flow models in communication, transportation and numerous other settings typically assume shortest path routing. There are natural settings in which alternative routing not involving shortest paths may be required. For example, it may happen that longer routes are used for load balancing or, in the case of capacitated networks, to avoid network expansion (Minou 1989; Magnanti and Wong 1984). Or an inverse problem may be posed: to determine edge weights so that shortest path routes identified by these weights result in the smallest load across the network (Applegate and Cohen 2003). Given the universality of the network flow model, there is a vast number of applications of this model to numerous areas, in particular network design, bandwidth and server capacity computation (Davis et al. 2002; Bienstock et al. 2006), network coding (Bassoli et al. 2013; Jalali et al. 2016) and search algorithms (Chung 1997).

But a key application of the multicommodity flow problem remains the same: determining the network capacity that meets a specified demand matrix, given the connectivity of the network. This is a problem of general interest to which there are no closed-form solutions that compactly and efficiently leverage the intrinsic features of the network and the demand matrix to give a direct analytical answer, the focus of this paper. Determination of even (tight) upper or lower bounds today requires a use of numerical computation. In this report, we provide one such solution that gives (tight) upper bounds for nodal and edge capacities.

As stated above, there are few analytical results concerning the multicommodity flow problem with shortest path routing, in the sense of having a closed form solution as a function of a small number of parameters characterizing the network and the commodities or point to point demands. These include the characterization of the maximal load for hyperbolic graphs (Narayan and Saniee 2011; Jonckheere et al. 2011). In this setting, for a network of N nodes one assumes 1 unit of (directed) flow between all $N(N - 1)$ node pairs, and then asks how the load scales due to shortest path routing as a function of N . This measure is sometimes referred to as the *betweenness centrality*, see Newman (2010) used for ranking of nodes for search algorithms.

In this paper, we study the near opposite of shortest path routing: when flows are routed in a uniformly or weighted random manner, each flow starting from its source and moving at each step randomly to a neighboring node and only stopping when the destination of the flow is reached. An analytical solution to this problem, if one could be found, would serve as an upper bound for the network capacity under various more realistic shortest path routing models discussed above. In a previous paper (Narayan et al. 2020), this problem was solved for the special case of the *uniform* multicommodity

flow problem, that is, when all source-destination demands are present and are equal to one. Further, it was assumed that there were no biases in random routing. Here we generalize this result further by allowing (1) arbitrary and non-symmetric demand and (2) probabilistic routing that is driven by weights on the network edges. More specifically, we consider the case when an arbitrary amount of traffic d_{ij} , in discrete units of messages, or *packets* from hereon, is injected into the network at every time step and at each node i for each possible destination node $j \neq i$. Thus there are $\sum_{i \neq j} d_{ij}$ units of traffic (packets/messages) injected into the N -node network at every time step, and this traffic is removed from the network when it reaches its destination. The network is assumed to be connected, i.e., have a single component.

We first demonstrate that a steady-state distribution is achieved, and then derive a closed-form expression for the expected flow, or the average number of packets or messages passing through each node, in terms of both the graph and demand (matrix) Laplacians. Next, we show how the result simplifies considerably when the demand matrix is symmetric. Following that, we derive a closed-form expression for when we each message only once for a node it passes through, even if it passes through multiple times, thus providing a tighter upper bound on the network capacity needed to process the required load. Finally, to illustrate the results more concretely, we use the derived analytical results to estimate the largest mean load for a few archetypal demand matrices over prototypical networks.

Weighted random routing with repetition

We shall use the following notation in the sequel.

Time evolution equations

As described in the previous section, we consider an undirected connected graph $G(N, E)$ with N nodes, in which packets of traffic are injected at various nodes in a deterministic manner and move towards specified destinations. The dynamics are discrete time, i.e. packets of traffic move from node to node at time $t = 0, 1, 2, 3 \dots$. At each time step, therefore, the number of packets at a node depends on the number of packets at all its adjacent nodes at the previous time step, and any packets injected at their source or removed at their destination. We first obtain the equations for this time evolution.

At every time step, the number of packets injected at a node k with destination node l is given by T_{kl} , which is the kl th element of the demand matrix T . Furthermore, any packet at node i at time t that has not yet reached its destination (i.e. whose destination is not the node i) moves to one of the nodes, j , adjacent to node i at time $t + 1$. The move from node i to an adjacent node j is chosen randomly, with probability

$$P_{i \rightarrow j} = \frac{A_{ij}}{\sum_{k \sim i} A_{ik}}. \quad (1)$$

Here we use the notation $i \sim j$ if the nodes i and j have an edge connecting them, and $i \not\sim j$ if they do not. A is the weighted adjacency matrix defined in Table 1, with $A_{ij} = 0$ if $i \not\sim j$. If $i \sim j$, the matrix element A_{ij} is a positive real number which reflects the weight

Table 1 Notation

G	$G(N, E)$ is a graph with N nodes and E edges
A	(A_{ij}) is the weighted adjacency matrix with $A_{ij} = 0, A_{ij} \geq 0$ for all $i, j \leq N$
d_i	$\sum_j A_{ij}$ is the degree of node $i \leq N$
D	(d_i) is the $N \times N$ diagonal matrix of weighted nodal degrees
L	$D - A$ is the graph Laplacian
M	$(M_{ij}) = L + \mathbf{1}_{N \times N}/N$ is the uniformly perturbed graph Laplacian
T	(T_{ij}) is the $N \times N$ demand matrix
\tilde{T}	diagonal matrix of nodal demands $\tilde{T}_j = \sum_i T_{ij}$
L_T	$\tilde{T} - T$ is the demand Laplacian for symmetric T

of the edge (ij) , i.e. whether the edge should be favored or disfavored when traffic flows through the network. We assume that the graph is undirected, i.e. that the matrix A is symmetric.

As stated before, any packet of traffic that is at its destination at time t is removed from the network and is no longer present at time $t + 1$. On the other hand, if a packet returns to its source as it moves around randomly, it continues as it would from any other node. The same is true if it returns to a node it has previously visited, i.e. if its path includes a closed loop. We return to this point in the next section of this paper.

We are interested in the expected value of the number of packets at each node. We expect that in steady state, if and when it exists, packets are removed from any node i at the rate of $\sum_j T_{ji}$ per time step. Recall the number of packets injected at the node i is $\sum_j T_{ij}$ per time step. We seek to find the steady state load, i.e. the average number of packets at each and every node of the network. This load may be interpreted as how fast on average each node needs to route packets it received net of those terminating at it. We first need to ensure there is a single and unique equilibrium distribution for the load in the network under the above conditions. The reader may refer to a simplified version of this model, in which $T_{ij} = 1$ for all $i \neq j$ and $A_{ij} = 1$ for $i \sim j$, which was discussed in earlier work (Narayan et al. 2020).

Theorem 1 *For a connected graph $G(N, E)$, with a deterministic injection rate of T_{ij} packets per time step at node i destined for node j for each pair of nodes (ij) , where each packet is routed randomly from its current node to its neighbors according to the weights in the adjacency matrix A until it reaches its destination, there exists a unique steady state number of packets at each node.*

Proof Let us first consider the case when only one element in the demand matrix is non-zero, and this element is unity. That is, $T_{ij} = \delta_{ik}\delta_{jl}$ for some choice of k and l . With this demand matrix, let $p_i^{kl}(t)$ be the ensemble average—i.e. averaged over many realizations of the same network—of the number of packets at node i at time t , except that $p_l^{kl}(t) = 0$, reflecting the fact that packets are immediately removed when they reach the destination node. The rate equation for the p_i 's is then

$$p_i^{kl}(t+1) = \delta_{ik} + \sum_j A_{ij} \frac{p_j^{kl}(t)}{d_j} \quad (2)$$

for $i \neq l$, with the boundary condition $p_l^{kl}(t+1) = 0$. Here d_j is the (weighted) degree of node j , as defined in Table 1. The boundary condition p_l^{kl} is an example of a Dirichlet boundary condition, where a function is defined in a region and is specified to be zero on the boundary of the region; in this case, the boundary is the node l and the region is all the other nodes in the graph.

We now show that, under the time evolution of Eq. (2), the function $p_i^{kl}(t)$ reaches a t -independent unique steady state. Let $p_i^{kl(1)}(0)$ and $p_i^{kl(2)}(0)$ be two initial configurations at $t = 0$, that evolve according to Eq. (2). Define $r_i^{kl}(t)$ to be equal to $[p_i^{kl(1)}(t) - p_i^{kl(2)}(t)]/d_i$. Then r^{kl} satisfies

$$d_i r_i^{kl}(t+1) = \sum_j A_{ij} r_j^{kl}(t) \quad (3)$$

with the Dirichlet boundary condition at $i = l$. This is equivalent to

$$r_i^{kl}(t+1) = \sum_j (\delta_{ij} - L_{ij}) r_j^{kl}(t) \quad (4)$$

where L is the Laplacian:

$$L_{ij} = d_i \delta_{ij} - A_{ij}. \quad (5)$$

Note that the matrix A is the *weighted* adjacency matrix. The real symmetric matrix L has a complete set of eigenvectors, with eigenvalues $\lambda \geq 0$. The eigenvalue $\lambda = 0$ only exists if one can construct a function f on the graph for which $f_i = f_j$ for all the nodes, in which case it is nondegenerate. This is not possible with Dirichlet boundary conditions, so that $\lambda > 0$. Thus the operator $I - L$ (with Dirichlet boundary conditions) is a contraction. Therefore $r^{kl}(t \rightarrow \infty) \rightarrow 0$, and as $t \rightarrow \infty$ all initial configurations tend to the same t -independent steady state configuration. \square

Steady state solution

Having obtained the time evolution equation Eq. (2), we find its fixed point in this section, with Dirichlet boundary conditions as introduced in the proof of Theorem (1). As before, $\{\lambda_\alpha, \alpha < N\}$ represent the eigenvalues of the graph Laplacian.

In steady state, the arguments t and $t+1$ in Eq. (2) can be dropped. Moreover, we know that, of the traffic flowing from the node k to the node l , the load that flows into l at any time step must be equal to the load injected into the node k , i.e. unity. Therefore $\sum A_{ij} p_j^{kl}/d_j = 1$, and we can extend Eq. (2) as

$$p_i^{kl} = \delta_{ik} - \delta_{il} + \sum_j A_{ij} \frac{p_j^{kl}}{d_j} \quad (6)$$

for all i , with the additional condition $p_i^{kl} = 0$. It may seem that we have gained nothing by restricting our analysis to the steady state configuration, since we still have to impose Dirichlet boundary conditions at the l 'th node. However, as we shall see immediately, the solution to Eq. (6) can easily be found in terms of the eigenvectors of the Laplacian without the Dirichlet boundary condition, i.e. independent of k and l . We rewrite Eq. (6) in vector form as

$$\mathbf{p}^{kl} = \mathbf{v}^k - \mathbf{v}^l + AD^{-1}\mathbf{p}^{kl} \quad (7)$$

where \mathbf{p}^{kl} is a column vector whose i 'th entry is p_i^{kl} , the matrix D is a diagonal matrix with $D_{ij} = d_i\delta_{ij}$, and \mathbf{v}^k and \mathbf{v}^l are unit column vectors whose i 'th elements are δ_{ik} and δ_{il} respectively. Defining $\mathbf{p}^{kl} = D\mathbf{r}^{kl}$, this equation is equivalent to

$$L\mathbf{r}^{kl} = \mathbf{v}^k - \mathbf{v}^l \quad (8)$$

with $r_i^{kl} = 0$.

If the matrix L were invertible, we could obtain $\mathbf{r}^{kl} = L^{-1}(\mathbf{v}^k - \mathbf{v}^l)$, but L has a zero eigenvalue. So we define the matrix $M = L + P$, where P is a constant matrix with entries $P_{ij} = 1/N$. Noting that the eigenvector ξ^0 of L with $\lambda = 0$ is a constant vector, with entries equal to $1/\sqrt{N}$ to ensure normalization, the matrix P projects onto ξ^0 . The eigenvalues of the matrix M are then equal to $\lambda_\alpha + \delta_{\alpha 0}$ in terms of the eigenvalues of the Laplacian. In terms of this matrix, and its inverse M^{-1} ,

$$\mathbf{r}^{kl} = M^{-1}[\mathbf{v}^k - \mathbf{v}^l - P\mathbf{r}^{kl}] = M^{-1}[\mathbf{v}^k - \mathbf{v}^l] - P\mathbf{r}^{kl} \quad (9)$$

where we have used the fact that $MP = LP + P^2 = P^2 = P$, i.e. $P = M^{-1}P$. The second term on the right hand side, $P\mathbf{r}^{kl}$, is a column vector with all entries identical. Therefore in component form,

$$r_j^{kl} = [M^{-1}]_{jk} - [M^{-1}]_{jl} + \text{const} = [M^{-1}]_{jk} - [M^{-1}]_{jl} - [M^{-1}]_{lk} + [M^{-1}]_{ll}. \quad (10)$$

For a general demand matrix, we arrive at the result:

Theorem 2 *The total steady state load at the j 'th node is*

$$\begin{aligned} \Lambda_j &= \sum_k \sum_l T_{kl} p_j^{kl} + \sum_k T_{kj} \\ &= d_j \sum_k \sum_l T_{kl} ([M^{-1}]_{jk} - [M^{-1}]_{jl} - [M^{-1}]_{lk} + [M^{-1}]_{ll}) + \tilde{T}_j. \end{aligned} \quad (11)$$

□

The term $\tilde{T}_j = \sum_k T_{kj}$ is added to account for the processing of the traffic at its destination node, based on the assumption that this is the same as the amount of processing at any intermediate node on its path; this term can be easily modified if the assumption is changed. If the demand matrix is symmetric, this simplifies to a more compact form with a natural interpretation.

Theorem 3 When the demand matrix is symmetric, the total steady state load at the j 'th node is

$$\begin{aligned}\Lambda_j &= d_j \text{Tr}[(\tilde{T} - T)M^{-1}] + \tilde{T}_j \\ &= d_j \text{Tr}[L_T M^{-1}] + \tilde{T}_j.\end{aligned}\quad (12)$$

□

It is also possible to write Eq. (11) in terms of the eigenvectors and eigenvalues of the Laplacian, L . Using the result $[M^{-1}]_{ij} = \sum_{\alpha} \xi_i^{\alpha} (\lambda_{\alpha} + \delta_{\alpha 0})^{-1} \xi_j^{\alpha}$, which is in turn equal to $\sum_{\alpha \neq 0} \xi_i^{\alpha} \lambda_{\alpha}^{-1} \xi_j^{\alpha} + 1/N$, we obtain

$$\Lambda_j = d_j \sum_k \sum_l T_{kl} \sum_{\alpha \neq 0} \frac{1}{\lambda_{\alpha}} (\xi_j^{\alpha} - \xi_l^{\alpha})(\xi_k^{\alpha} - \xi_l^{\alpha}) + \tilde{T}_j. \quad (13)$$

When the demand matrix is symmetric, this reduces to

$$\Lambda_j = d_j \sum_k \sum_l T_{kl} \sum_{\alpha \neq 0} \frac{1}{\lambda_{\alpha}} \xi_l^{\alpha} (\xi_l^{\alpha} - \xi_k^{\alpha}) + \tilde{T}_j. \quad (14)$$

Further, when the demand matrix is uniform, i.e., $T_{kl} = 1 - \delta_{kl}$, since the first summed expression in Eq. (14) is clearly zero when $k = l$, we can drop the factor of T_{kl} . Since $\sum_l \xi_l^{\alpha} = 0$ for $\alpha \neq 0$ and $\sum_l (\xi_l^{\alpha})^2 = 1$ by normalization, this reduces still further to

$$\Lambda_j = Nd_j \sum_{\alpha \neq 0} \frac{1}{\lambda_{\alpha}} + (N - 1) \quad (15)$$

which was derived in earlier work (Narayan et al. 2020).

Remark 1 For a symmetric demand matrix, the result that Λ_j is linearly dependent on d_j can be obtained directly. The traffic from node k to node l can be represented as a stream of random walkers that diffuse through the network at discrete time steps. At every time step, in addition to the diffusive dynamics, T_{kl} walkers are introduced at node k , and all the walkers at node l are removed. Comparing with Eq. (2), the expected number of random walkers at node j at time t is equal to $p_j^{kl}(t)$. If the random walks corresponding to all source destination pairs take place simultaneously, with each walker labelled with an index corresponding to its destination, we have random walkers with N different labels moving through the network. In addition to the random walk dynamics, walkers are created and destroyed at their sources and destinations respectively. In steady state, the number of walkers created and destroyed at the j 'th node at any time step are equal to $\sum_i T_{ji}$ and $\sum_i T_{ij}$ respectively, but they have different labels. If we ignore the labels on the random walkers, the creation and destruction of random walkers can be ignored. The steady state solution for $\sum_k \sum_l p_j^{kl}(t)$ is proportional to the steady state solution for a diffusion process on the graph with no sources or sinks. It is easy to verify that, in this steady state, the number of random walkers at the j 'th node is proportional to $\sum_i A_{ji} = d_j$. Although this tells us that $[\Lambda_j - \tilde{T}_j]$ is a constant, independent of j , it does not tell us that this constant is equal to $\text{Tr}[(\tilde{T} - T)M^{-1}]$.

Remark 2 So far we have dealt with connected undirected graphs. We point out that when the graph is directed, then assuming that steady state distribution is achievable, Remark 1 implies that the expected load $\Lambda_j = \tilde{T}_j + C\pi_j$ where C is some constant independent of the node and (π_j) is the right principal eigenvector (with eigenvalue 1) of the random walk matrix for the directed graph, which for undirected graphs is equal to (d_j) .

Remark 3 We observe that the proofs of both theorems carry through essentially unchanged if we replace the deterministic arrival of one packet at each source node for each destination node at each time step with a Poisson arrival process with a mean of one packet arrival per node per unit time for each destination node.

Remark 4

Specializing to the case of a uniform demand matrix, where Eq. (15) applies, in the large- N limit, the spectral density of the Laplacian $\sum_{\alpha} \delta(\lambda - \lambda_{\alpha})$ tends to $N\rho(\lambda)$, where $\rho(\lambda)$ is smooth, and

$$N \sum 1/\lambda_{\alpha} \rightarrow N^2 \int \frac{\rho(\lambda)}{\lambda} d\lambda \quad (16)$$

which is $\sim N^2$ if the integral has a spectral gap or is otherwise convergent. On the other hand, if $\rho(\lambda \rightarrow 0)$ is non-zero, the spectral gap for large N is proportional to $1/N$, and $N^2 \int (\rho(\lambda)/\lambda) d\lambda \sim N^2 \ln N$. The maximum load at any node in the network is, up to an additive constant, $N d_{\max} \sum 1/\lambda_{\alpha}$, which scales differently than the average load if d_{\max} diverges as $N \rightarrow \infty$.

Random routing without repetition

With different edges in the graph weighted differently, random routing preferentially follows paths that have edges with high weights. However, this generalized model suffers from an important shortcoming: if A_{ij} is large for an edge connecting a node i to a node j , then a packet that reaches node i is likely to move to j at the next time step, and vice versa. The packet may spend several time steps going to and fro between the two nodes. If A_{ij} is increased with the weights of other edges held constant, this behavior becomes more pronounced. The multiple visits to nodes i and j increase the load at these nodes. Clearly, this is not how any reasonable routing scheme, even with randomness, would function. One way in which to eliminate the enhancement of the load at such nodes is to count each node on the path of a packet only once, regardless of how many times it is visited. In effect, this is as if the back and forth travel between the two adjoining nodes were to be eliminated. This is the strategy that we use in this section of the paper.

Counting each node on a path only once is an imperfect solution, since in addition to eliminating such back and forth paths, it also affects paths that include closed loops. The node at which the path enters and exits the loop, through which it goes twice, is

only counted once even though the loop is not being eliminated. (If the loop were to be eliminated, all the other nodes in the loop would be counted zero times, not once.) However, this solution eliminates the greatest shortcoming of random routing for a weighted graph in the previous section of this paper.

The analysis with an unweighted network and uniform demand between all source-destination pairs has been reported in Narayan and Saniee (2018), which we extend here. As before, we start with packets that are sourced at node k and have their destination as node l , and perform a random walk through the network. At each time step, a packet is removed from the network if it has reached the destination l or a query node m . The steady state equation is then a modification of Eq. (7):

$$\mathbf{p}^{kl} = \mathbf{v}^k - (1 - \mu_m^{kl})\mathbf{v}^l - \mu_m^{kl}\mathbf{v}^m + AD^{-1}\mathbf{p}^{kl} \quad (17)$$

where μ_m^{kl} is determined by the condition that $p_l^{kl} = p_m^{kl} = 0$. The fraction μ_m^{kl} of the traffic between k and l that is removed at node m is the fraction that passes through the node m at some point in its path from k to l . In reality, the packet continues on after it encounters m , perhaps passing through m multiple times, before it reaches its destination l , but removing it from the network ensures that it is counted only once at the node m .

Defining \mathbf{r}^{kl} as before, we obtain

$$(D - A)\mathbf{r}^{kl} = \mathbf{v}^k - (1 - \mu_m^{kl})\mathbf{v}^l - \mu_m^{kl}\mathbf{v}^m. \quad (18)$$

The solution to this equation is a modification of Eq. (10):

$$r_j^{kl} = [M^{-1}]_{jk} - (1 - \mu_m^{kl})[M^{-1}]_{jl} - \mu_m^{kl}[M^{-1}]_{jm} + \text{const}. \quad (19)$$

Applying the condition $r_l^{kl} = r_m^{kl} = 0$, and subtracting one from the other, we obtain

$$\begin{aligned} & [M^{-1}]_{mk} - [M^{-1}]_{lk} - [M^{-1}]_{ml} + [M^{-1}]_{ll} \\ &= \mu_m^{kl} \left([M^{-1}]_{mm} - [M^{-1}]_{ml} - [M^{-1}]_{lm} + [M^{-1}]_{mm} \right). \end{aligned} \quad (20)$$

Note that $\mu_k^{kl} = 1$, but μ_l^{kl} is indeterminate because the terms with \mathbf{v}^l and \mathbf{v}^m in Eq. (17) merge together. Since all paths from k to l pass through both k and l , we fix $\mu_l^{kl} = 1$.

As noted earlier, μ_m^{kl} is the fraction of the traffic from node k to node l that passes through the node m , possibly multiple times. Summing over all source-destination pairs, the load at node m with repeated traversals counted only once is given by the following.

Theorem 4 *The load at node m with repeated traversals counted only once is given by*

$$\begin{aligned} \tilde{\Lambda}_m &= \tilde{T}_m + \sum_k \sum_{l \neq m} T_{kl} \mu_m^{kl} \\ &= \tilde{T}_m + \sum_k \sum_{l \neq m} T_{kl} \frac{[M^{-1}]_{mk} - [M^{-1}]_{lk} - [M^{-1}]_{ml} + [M^{-1}]_{ll}}{[M^{-1}]_{mm} - [M^{-1}]_{ml} - [M^{-1}]_{lm} + [M^{-1}]_{mm}}. \end{aligned} \quad (21)$$

As before, this can be expressed in terms of the eigenvectors and eigenvalues of the Laplacian, yielding

$$\tilde{\Lambda}_m = \tilde{T}_m + \sum_k \sum_{l \neq m} T_{kl} \left[\sum_{\alpha \neq 0} \frac{(\xi_k^\alpha - \xi_l^\alpha)(\xi_m^\alpha - \xi_l^\alpha)}{\lambda_\alpha} \right] \left[\sum_{\alpha \neq 0} \frac{(\xi_m^\alpha - \xi_l^\alpha)^2}{\lambda_\alpha} \right]^{-1}. \quad (22)$$

Numerical results

In this section, we provide numerical results for three different types of graphs: square lattice graphs, hyperbolic grids, and Erdős–Rényi random graphs (Erdős and Rényi 1959) for two types of demand regimes: broadcast with constant value and broadcast exponential decay. Numerical results for the uniform demand matrix for hyperbolic grids, Erdős–Rényi graphs, and scale-free networks (Barabási and Albert 1999); Dorogovtsev et al. (2000) have been presented earlier (Narayan et al. 2020).

Square lattice

We consider a $L_0 \times L_0$ square lattice with $N = L_0^2$ nodes. Periodic boundary conditions are used, so that every node has exactly four neighbors. It is convenient to choose the normalized eigenfunctions of the Laplacian as the complex functions

$$\xi_{\mathbf{q}}(\mathbf{r}) = \frac{1}{L_0} \exp[i\mathbf{q} \cdot \mathbf{r}] \quad (23)$$

where $\mathbf{r} = (x, y)$ is the location of the node, with x and y integers, and $\mathbf{q} = (q_x, q_y) = 2\pi(m, n)/L_0$, with $n, m = 0, 1, 2 \dots L_0 - 1$, and $m = n = 0$ not allowed. For such complex eigenfunctions, Eq. (13) is generalized to

$$\Lambda_j = d_j \sum_k \sum_l T_{kl} \sum_{\alpha \neq 0} \frac{1}{\lambda_\alpha} (\xi_j^{*\alpha} - \xi_l^{*\alpha})(\xi_k^\alpha - \xi_l^\alpha) + \tilde{T}_{jj}. \quad (24)$$

For the case of unit demand between all node pairs, from Eq. (16) and the following discussion, the load at any node grows as $\sim N^2 \ln N$ for large N . If we change the normalization so that the total demand between all node pairs adds up to 1, i.e. $T_{kl} = (1 - \delta_{kl})/[N(N - 1)]$, the load at any node grows as $\sim \ln N$ for large N .

Instead, we consider the case when all the traffic is broadcast from a single node k , which we choose to have coordinates $x = y = 0$. (Because of periodic boundary conditions, all nodes are equivalent.) Eq. (24) then becomes

$$\Lambda_j = \frac{4}{N} \sum_l T_{0l} \sum_{\mathbf{q} \neq 0} \frac{1}{\lambda_{\mathbf{q}}} (1 + \exp[-i\mathbf{q}_j \cdot \mathbf{q}]) (1 - \exp[i(\mathbf{r}_l \cdot \mathbf{q})]) + \tilde{T}_{jj} \quad (25)$$

where we have used the fact that $\lambda_{\mathbf{q}} = \lambda_{-\mathbf{q}}$.

If T_{0l} is independent of the destination node l , normalized to $T_{0l} = (1 - \delta_{l0})/(N - 1)$ to ensure unit total demand, we can replace this with $T_{0l} = 1/(N - 1)$ inside the sum, because $1 - \exp[i(\mathbf{r}_l \cdot \mathbf{q})] = 0$ when l is at the origin. The sum over l can easily be performed, and the l -dependent term cancels out when summed, resulting in

$$\Lambda_j = \frac{4}{(N - 1)} \sum_{\mathbf{q} \neq 0} \frac{1}{\lambda_{\mathbf{q}}} (1 + \exp[-i\mathbf{r}_j \cdot \mathbf{q}]) + \frac{1}{N - 1} (1 - \delta_{j0}). \quad (26)$$

In the large- N limit, this approaches

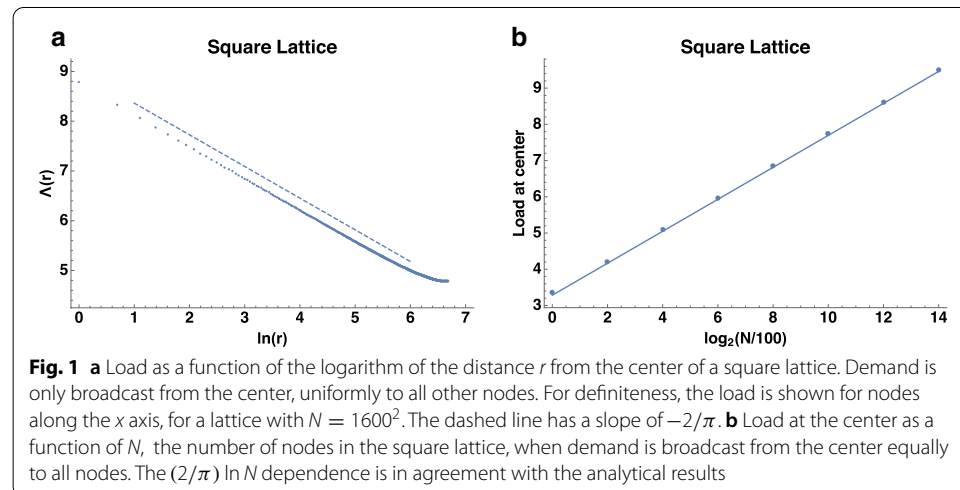
$$\frac{4}{N} \sum_{\mathbf{q} \neq 0} \frac{2}{\lambda_{\mathbf{q}}} + 4 \int \frac{\cos(\mathbf{q} \cdot \mathbf{r}_j) - 1}{4 - 2 \cos(q_x) - 2 \cos(q_y)} \frac{dq_x}{2\pi} \frac{dq_y}{2\pi} \quad (27)$$

where the sum diverges as $N \rightarrow \infty$, and the integral is finite. It is possible to argue, and to verify numerically in Fig. 1, that the large- N behavior of the sum is $(2/\pi) \ln N + \text{const.}$ Numerically, the constant is $0.3901 \dots$. It is similarly possible to obtain that the integral approaches $-(2/\pi) \ln |\mathbf{r}_j| + \text{const.}$ for large $|\mathbf{r}_j|$. Noting that the maximum distance between any pair of nodes is $O(\sqrt{N})$, the load decreases as one moves away from the source, with a singular part that ranges from $(2 \ln N)/\pi$ near the source to $(\ln N)/\pi$ far from the source. This singular term is not present with geodesic routing.

Alternatively, if the demand falls off exponentially with distance from the source node, it is possible to take the large- N limit without any singular terms. We assume that the demand between the source node at $(0, 0)$ and a destination node at $(x, y) \neq (0, 0)$ is proportional to $p^{|x|+|y|}$. A normalization constant of $(1-p)^2/(4p)$ ensures unit total demand, matching the previous paragraph. Then

$$\Lambda_j = \frac{(1-p)^2}{Np} \sum_l \sum_{\mathbf{q} \neq 0} p^{|l_x|+|l_y|} \frac{1}{\lambda_{\mathbf{q}}} (1 + \exp[-i\mathbf{r}_j \cdot \mathbf{q}]) (1 - \exp[i(l_x q_x + l_y q_y)]) + \tilde{T}_{jj} \quad (28)$$

where, as in the previous paragraph, it is not necessary to specify $(l_x, l_y) \neq (0, 0)$ in the sum. The sums over l_x and l_y are then independent and easy to perform. In the large- N limit, we can then replace $N^{-1} \sum_{\mathbf{q}}$ with $\int_0^{2\pi} \int_0^{2\pi} dq_x dq_y / (4\pi)^2$, since the zero in $\lambda_{\mathbf{q}}$ is canceled by a zero in $(1 - \exp[i(l_x q_x + l_y q_y)])$ for small- q , resulting in the absence of singular terms in the large- N limit. The resulting integrals can be evaluated numerically, but an inspection of Eq. (28) shows that the j -dependent term $(1 + \exp[-i\mathbf{r}_j \cdot \mathbf{q}])$ is a rapidly oscillating function of \mathbf{q} , averaging to unity, when $|\mathbf{r}_j| \rightarrow \infty$. Thus Λ_j far from the source is half its value near the source, even though the demand is concentrated near the source.



The r -dependence of the load can be understood using a continuum approximation, using an analogy to electrostatics. The source and destination of traffic map to positive and negative charges respectively, and the load maps to the electrostatic potential. The details of the analysis are outside the scope of this paper, but for demand broadcast equally to all nodes, the approximation predicts a $\sim \ln r$ dependence of the load, while for demand decaying exponentially with distance, $\Lambda(r) - \Lambda(\infty) \sim 1/r^4$ for large r . Both of these are consistent with our numerical results.

Hyperbolic grids

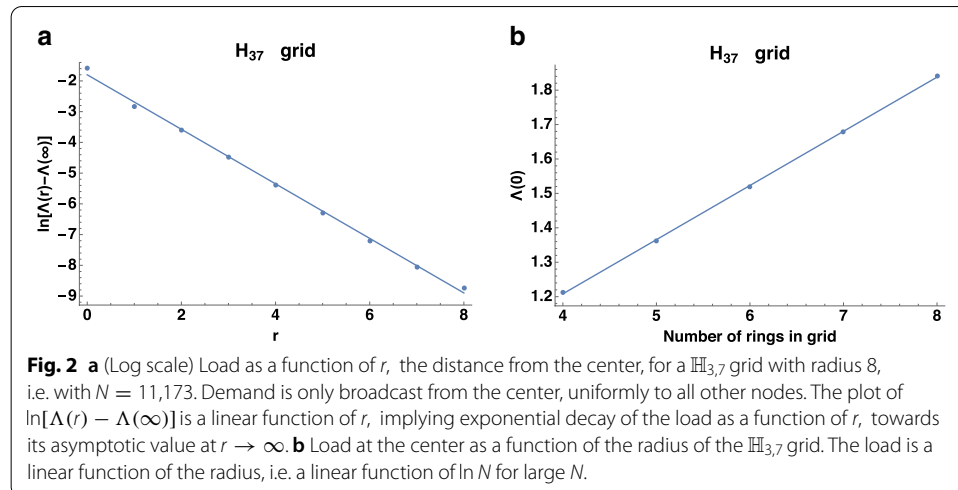
A hyperbolic grid $\mathbb{H}_{p,q}$ is an infinite regular planar graph with constant degree q , and each face a p -sided polygon, satisfying the condition $(p-2)(q-2) > 4$. If $(p-2)(q-2) = 4$, it is possible to make all the polygons as regular polygons of the same size. This is no longer possible if $(p-2)(q-2) > 4$, but one can interpret the graph as a planar projection of a graph on a surface of constant negative curvature, and on that surface, the polygons are regular and identical.

When the demand between all pairs of nodes (i, j) with $i \neq j$ is equal to $1/(N(N-1))$, so that the total demand is 1, Eq. (15) is adjusted to

$$\Lambda_j = \frac{d_j}{N-1} \sum_{\alpha \neq 0} \frac{1}{\lambda_\alpha} + \frac{1}{N}. \quad (29)$$

The second term is negligible when $N \rightarrow \infty$, while the first term was shown numerically Narayan et al. (2020) to scale linearly with $\ln N$ for large N when $(p, q) = (3, 7)$.

Here, we show numerical results when demand is only broadcast from the node at the center, equally to all other nodes. To have the same total demand as in the previous case, the demand for each node is $1/(N-1)$. As seen in Fig. 2, the load decays as $\Lambda(r) = C \exp[-cr] + \Lambda(\infty)$, where r is the geodesic distance between the center and the node. Furthermore, the load at the center grows linearly with the radius of the grid, i.e. $\sim \ln N$, which is the same as for uniform demand on the hyperbolic grid, and also similar



to the analytical results for the square lattice, when demand is broadcast from the center and is independent of the distance to the destination.

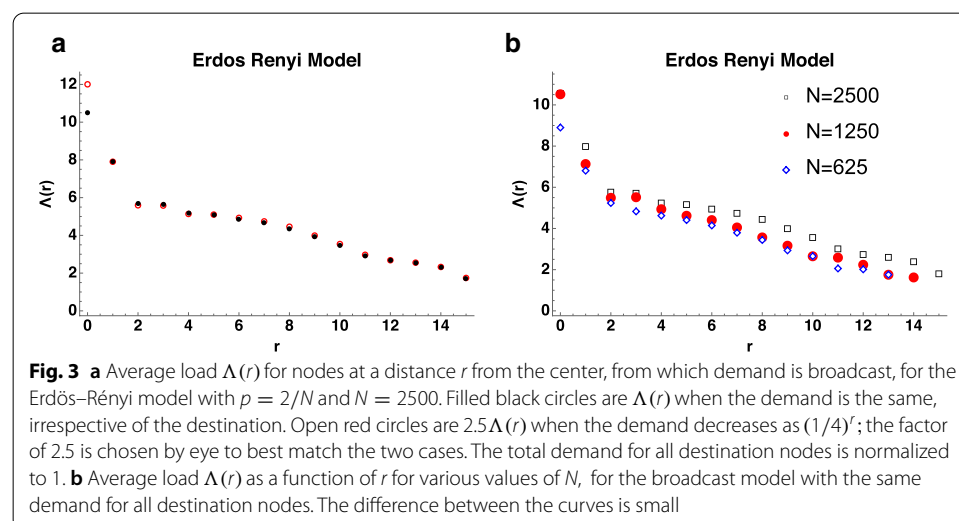
We also consider the case when the demand is broadcast from the central node, but drops off exponentially as a function of the distance r from the center, i.e. proportional to p^r , with the proportionality constant chosen so that the total demand is unity, irrespective of N . We choose $p = 0.25$; the total demand on an infinite grid is finite as long as $p < (3 - \sqrt{5})/2$. The results for this case are not shown here, but within numerical accuracy, $\Lambda(r) - \Lambda(\infty)$ decreases exponentially as a function of r . The load is found to be almost independent of N , similar to the analytical results for exponentially decaying demand on the square lattice.

As with the square lattice, the r -dependence of the load can be understood in a continuum approximation with an electrostatic analogy, which predicts that $\Lambda(r) - \Lambda(\infty)$ will decay exponentially with r .

Random graphs

We performed numerical simulations on Erdős–Rényi random graphs, with the probability of an edge being connected being $p = 2/N$ for a graph with N nodes. We only retain the giant component of the graph, which has $O(N)$ nodes, numerically found to be $\approx 0.8N$. For each value of N , several random graphs were generated, so that the total number of nodes in the giant components of all the graphs was approximately 10^4 . Demand was broadcast from the node at the center of each graph, either uniformly to all other nodes, or decaying as $\sim (1/4)^r$ with the distance r from the center to the destination.

As shown in Fig. 3, rather surprisingly, the average load at the nodes at a distance r from the center has the same form when the demand is uniform or exponentially decaying, up to an overall proportionality constant. The average load $\Lambda(r)$ for nodes at a distance r from the center does not have a simple functional form. There is also very little change in the plots as a function of N , consistent with our earlier result Narayan et al.



(2020) that the load at the center with uniform demand between all node pairs is independent of N .

Conclusions

We have shown that, for the general non-uniform multicommodity flow problem on an arbitrary connected graph under weighted random routing, the mean load at each node of the graph exists and is unique. We have also derived an explicit expression for it in terms of the graph and demand (matrix) Laplacians. Further, we have derived a closed form expression for the mean load without counting repeated visits to a transit node. Using these explicit expressions, we have obtained numerical estimates for the mean load for the regular and hyperbolic lattices in the large-size regime, and have numerically computed the mean load for Erdős–Rényi random graphs. This has been done for uniform and for broadcast demand. For the latter (broadcast demand), we have observed that the average load decreases as a function of distance from the broadcast source, ranging from exponential decrease (for regular hyperbolic graphs), to logarithmic decrease (for regular grids), to staged decrease (for Erdős–Rényi random graphs).

Acknowledgements

O. Narayan acknowledges useful conversations with Prof. Josh M. Deutsch.

Authors' contributions

The authors contributed equally to all aspects of this paper. Both authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Physics, University of California, Santa Cruz, CA 95064, USA. ²Mathematics & Algorithms Group, Bell Labs, Nokia, Murray Hill, NJ 07974, USA.

Received: 23 February 2021 Accepted: 1 June 2021

Published online: 16 June 2021

References

- Applegate D, Cohen E (2003) Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. *ACM Sigcomm'03*, pp 313–324
- Assad A (1978) Multicommodity network flows—a survey. *Networks* 8(1):37–91
- Awerbuch B, Leighton T (2005) Multicommodity flows: a survey of recent research. In: *Proceedings of ISAAC 1993: algorithms and computation*. Springer lecture notes in computer science, pp 297–302
- Barabási AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286:509–512
- Bassoli R et al (2013) Network coding theory: a survey. *IEEE Commun Surv Tutor* 15(4):1950–1978
- Bienstock D, Raskina O, Saniee I, Wang Q (2006) Combined network design and multiperiod pricing: modeling, solution techniques, and computation. *Oper Res* 54(2):261–276
- Chung FRK (1997) *Spectral Graph Theory* American Mathematical Society. CBMS Series, No, p 92
- Davis RD, Kumaran K, Liu G, Saniee I (2002) SPIDER: a simple and flexible tool for design and provisioning of protected lightpaths in optical networks. *Bell Labs Tech J* 6(1):82–97
- Dorogovtsev SN, Mendes JFF, Samukhin AN (2000) Structure of growing networks: exact solution of the Barabási–Albert model. *Phys Rev Lett* 85:4633–4636
- Elias P, Feinstein A, Shannon C (1956) A note on the maximum flow through a network. *IEEE Trans Inf Theory* 2:117–119
- Erdős P, Rényi A (1959) On random graphs. *Publ Math* 6:290–297
- Ford LR, Fulkerson DR (1956) Maximal flow through a network. *Can J Math* 8:399–404
- Hu TC (1963) Multicommodity network flows. *Oper Res* 11:344–360

- Jalali S, Effros M, Ho T (2016) On the impact of a single edge on the network coding capacity. [arXiv:1607.06793](https://arxiv.org/abs/1607.06793)
- Jonckheere EA, Lou M, Bonahon F, Baryshnikov Y (2011) Euclidean versus hyperbolic congestion in idealized versus experimental networks. *Internet Math* 7:1–27
- Kennington JL (1978) A survey of linear cost multicommodity network flows. *Oper Res* 26(2):209–236
- Lomonosov MV (1985) Combinatorial approaches to multiflow problems. *Discrete Appl Math* 11:1–94
- Magnanti TL, Wong RT (1984) Network design and transportation planning: models and algorithms. *Transp Sci* 18:1–55
- Minou M (1989) Network synthesis and optimum network design problems: models, solution methods and applications. *Networks* 19:313–360
- Narayan O, Saniee I (2011) Large-scale curvature of networks. *Phys Rev E* 84:066108
- Narayan O, Saniee I, Marbukh V (2020) Congestion due to random walk routing. In: *The 9th international conference on complex networks and their applications*, vol 1. Springer, pp 556–567
- Narayan O, Saniee I (2018) Scaling of Random Walk Betweenness in Networks. In: *The 7th international conference on complex networks and their applications*, vol 1. Springer, pp 41–51
- Newman MEJ (2010) *Networks: an introduction*. Oxford University Press, Oxford
- Okamura H, Seymour PD (1981) Multicommodity flows in planar graphs. *J Combin Theory Ser B* 31:75–81
- Quorou A, Mahey J-Ph, Vial P (2000) A survey of algorithms for convex multicommodity flow problems. *Manag Sci* 46(1):126–147
- Papernov BA (1976) Feasibility of multicommodity flows. In: Friedman A (ed) *Studies in discrete optimization*. Izdat. "Nauka", Moscow, pp 230–261 **(in Russian)**
- Shahrokhi F, Matula DW (1990) The maximum concurrent flow problem. *J ACM* 37:318–334
- Wang I-L (2018a) Multicommodity network flows: a survey, Part I, applications and formulations. *Int J Oper Res* 15(4):145–153
- Wang I-L (2018b) Multicommodity network flows: a survey, Part II, solution methods. *Int J Oper Res* 15(4):155–173

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
