

Load Balancing by Network Curvature Control

M. Lou, E. Jonckheere, F. Bonahon, Y. Baryshnikov, B. Krishnamachari

Mingji Lou

Western Digital
Lake Forest, CA 92692, USA
E-mail: mingjilou@gmail.com

Edmond Jonckheere, Francis Bonahon, Bhaskar Krishnamachari

University of Southern California
Los Angeles, CA 90089-2563, USA
E-mail: jonckhee@usc.edu, fbonahon@math.usc.edu, bkrishna@usc.edu

Yuliy Baryshnikov

Alcatel-Lucent
Murray Hill, NJ 07974-0636, USA
E-mail: yuliy.baryshnikov@alcatel-lucent.com

Abstract: The traditional heavy-tailed interpretation of congestion is challenged in this paper. A counter example shows that a network with uniform degree can have significant traffic congestion when the degree is larger than 6. A profound understanding of what causes congestion is reestablished, based on the network curvature theorem. A load balancing algorithm based on curvature control is presented with network applications.

Keywords: network congestion, curvature, inertia, Gromov hyperbolic graphs, Poincaré disk, load balancing, Yamabe flow.

1 Introduction

One of the most important challenges in networking systems, especially in large and wide area networks, is the traffic congestion problem. The queuing feature between two routers can create a logical bottleneck between two users. Correspondingly, insufficient bandwidth on the physical links between routers is a contributor to congestion. The current congestion control technologies in communication networks are based on the feedback from the congested node to the source to slow down the packet flow rate, such as bidirectional congestion control and Random Early Detection (RED). However, these technologies can only be applied once the congestion has happened to some degree, and it is only based on the *local* point of view of some queue overflow along the source to target path. This paper investigates the deeper reason behind the congestion *in the large scale*, and will challenge the current least-cost-path algorithms, such as Dijkstra's shortest path algorithm. It will indeed be established that these least-cost-path algorithms aggravate the congestion, especially in negatively curved networks.

A fundamental ingredient in this paper is that, in order to get a large-scale view on the congestion problem, we utilize the coarse approximation of a network graph by a Riemannian manifold. A graph as a mathematical idealization of a network is completely different than a Riemannian manifold; however, the recent development of the so-called coarse geometry under the leadership of Mikhael Gromov has given the two mathematical structures—graphs and manifolds—the unifying framework of geodesic spaces. As a corollary, the concept of curvature has become applicable to graphs [2, 8–10]. The fundamental mathematical idea behind this unification is to realize that the traditional Riemannian curvature, which relies on the differentiable structure of the manifold, can be rephrased in terms of the more primitive concept of distance [3, 6]. Since a

communication network can be endowed with a distance, which represents communication cost, delay, outage, etc., its curvature can be defined. The positively curved versus negatively curved network dichotomy roughly corresponds to the more traditional meshed (decentralized) versus core-concentric (centralized) network dichotomy [14].

It has been experimentally observed that, on the Internet and other networks, traffic seems to concentrate quite heavily on some very small subsets. As shown in Fig. 1, congestion could occur at the “core” through an easy mechanism. However, one extremely important point that will be made in this paper is that, contrary to traditional belief, congestion is not necessarily a manifestation of the heavy-tailed behavior related to high node degree, but is a manifestation of a more subtle process that can be traced back to the curvature.

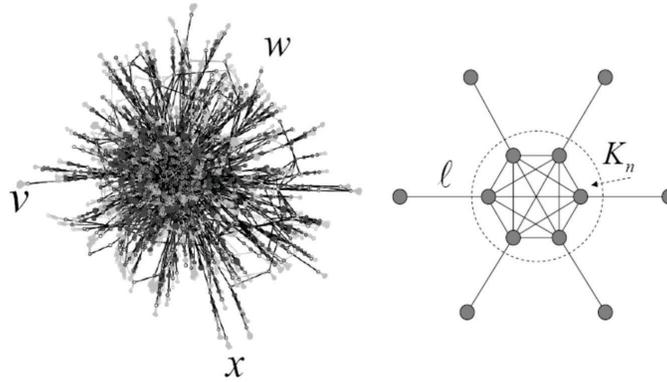


Figure 1: Traditional understanding of congestion occurring at the “core.” left, Internet Service Provider (ISP) graph; right: idealized model.

The mathematical apparatus and computer simulations will unveil this striking traffic pattern in negatively curved networks from both a theoretical and a practical point of view. In further, this paper studies another fundamental question: if congestion does not necessarily occur at vertices of high degree, nor at the so-called highly connected “core,” then what are the congestion points? This paper shows that congestion occurs at the points of least inertia of the network. Last but not least, a curvature-based load-balancing routing is proposed, and its performance is compared with a shortest-path routing in multicast communication network.

2 General Concepts and Conjectures

2.1 Curvature, Traffic, Betweenness, Inertia, and Centroid

Let $G = (V, E)$ be a graph specified by its vertex set V and its edge set E and endowed with a (symmetric) distance function $d : G \times G \rightarrow \mathbb{R}^+$. A path $p(s, t)$ from s to t is a continuous map $[0, l] \rightarrow G$ such that $p(s, t)(0) = s$ and $p(s, t)(l) = t$. The weight of an edge $e = xy$ is defined as $w(e) = d(x, y)$. The **length** of the path is defined as $\ell(p(s, t)) = \sum_{e \subseteq p(s, t)} w(e)$. A **geodesic** $[s, t]$ is a path such that $\ell([s, t]) \leq \ell(p(s, t))$, $\forall p(s, t)$. A geodesic **triangle** is defined as $\Delta abc = [a, b] \cup [b, c] \cup [c, a]$.

For the sake of simplicity, the network curvature concept is restricted to planar communication graphs and is based on Alexandrov angles [1, 3, 6]. Let $(ab_1 = ab_{\deg(a)+1}, ab_2, \dots, ab_{\deg(a)})$ be a cyclic ordering of the set of edges attached to the vertex a . The Alexandrov angle α_k at the vertex a of the geodesic triangle $\Delta ab_k b_{k+1}$ is defined as $\alpha_k = \cos^{-1} \frac{w(ab_k)^2 + w(ab_{k+1})^2 - w(b_k b_{k+1})^2}{2w(ab_k)w(ab_{k+1})}$.

and the (Gauss) *curvature* at the vertex a is defined as

$$\kappa(a) = \frac{2\pi - \sum_{i=1}^{\deg(a)} \alpha_k}{\sum_{k=1}^{\deg(a)} A(\Delta ab_k b_{k+1})} = \frac{K(a)}{\sum_{k=1}^{\deg(a)} A(\Delta ab_k b_{k+1})} \quad (1)$$

where $A(\Delta ab_k b_{k+1})$ denotes the area of the geodesic triangle $\Delta ab_k b_{k+1}$ easily computable via Heron's formula. It is easily seen that, for the number of hops metric ($w(e) = 1$), $\alpha_k = \pi/6$; therefore, $\kappa(a) < 0$, $\kappa(a) = 0$, or $\kappa(a) > 0$ depending on whether $\deg(a) > 6$, $\deg(a) = 6$, or $\deg(a) < 6$, respectively.

An infinite negatively curved graph has the property that it is isometric to a negatively curved manifold up to a bounded distortion (see [5] for precise statement). Since this graph-manifold identification entails a bounded error, large scale problems on graphs can be mapped to more manageable continuous geometry problems on manifolds (see Sec. 4).

The traffic on the graph is driven by a *demand measure* $\Lambda_d : V \times V \rightarrow \mathbb{R}^+$, where the demand $\Lambda_d(s, t)$ is the traffic rate (e.g., number of packets per second) to be transmitted from the source s to the destination target t . Assume that the routing protocol sends the packets from the source s to the target t along the path $p(s, t)$ with probability $\pi(p(s, t))$. As such, the path $p(s, t)$ inherits a traffic rate measure $\tau(p(s, t)) = \Lambda_d(s, t)\pi(p(s, t))$. An edge e laying on the path $p(s, t)$ inherits from that path a traffic $\tau(p(s, t))$. Aggregating this traffic over all source-target pairs and all paths traversing the edge e yields the traffic rate sustained by the edge e , $\tau(e) = \sum_{(s,t) \in V \times V} \sum_{p(s,t) \supseteq e} \tau(p(s, t))$. The *traffic rate* at a vertex v is defined as

$$\begin{aligned} b(v) &= \frac{1}{2} \sum_{e \supseteq v} \tau(e) + \sum_{s \neq v} \Lambda_d(s, v) + \sum_{t \neq v} \Lambda_d(v, t) \\ &= |\{[s, t] : v \in [s, t]\}| \quad (\text{if } \Lambda_d(x, y) = 1, \forall x \neq y) \end{aligned}$$

The notation $b(\cdot)$ stands for *betweenness* [3], as for a uniformly distributed demand, the traffic at v is the number of geodesics passing through v . Given a connected subgraph $X \subseteq G$, we define its *traffic load* to be representative of the number of packets in it:

$$\Lambda_t(X) = \sum_{s,t \in V} \left(\sum_{e \in p(s,t) \cap X} w(e) \right) \Lambda_d(s, t) \pi(p(s, t)) \quad (2)$$

The *inertia* of a (connected) graph G with respect to a vertex v is defined as $\phi_G(v) = \sum_{v^i \in V} d^2(v, v^i)$. Observe that this inertia may be infinite. A *center of mass* or *centroid* of the graph G is defined as a vertex relative to which the inertia is minimum: $\text{cm}(G) = \arg \min_{v \in V} \phi_G(v)$. The global minimum need not be unique.

2.2 Conjectures

Conjecture 1. \mathbf{G}^- . *If the graph $G = (V, E)$ is negatively curved along with a demand measure Λ_d uniformly distributed over $V \times V$, the least-cost protocol that sends packets over optimal routes leads to a very high traffic rate $b(v)$ (traffic load $\Lambda_t(X)$) over a very small number of vertices v (over a very small subset X).*

Conjecture 2. \mathbf{G}^+ . *If the graph $G = (V, E)$ is nonnegatively curved along with a demand measure Λ_d uniformly distributed over $V \times V$, the protocol that sends packets over optimal routes leads to a nearly uniform traffic rate $b(v)$.*

We can now formulate our third conjecture, saying that the maximum traffic load for uniformly distributed demand occurs near the center of mass of the network relative to uniformly distributed weight:

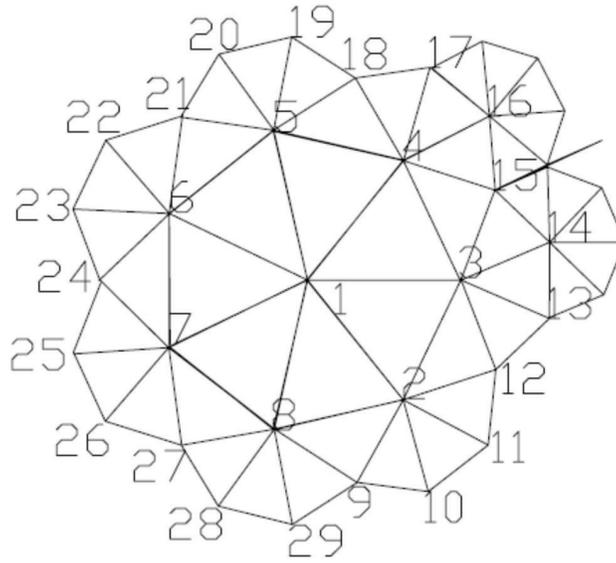


Figure 2: Simple hyperbolic graph with node degree seven and uniform edge length.

Conjecture 3. G . $\arg \max_v b(v) \approx \text{cm}(G)$. (Equality failed only in one positively curved example.)

Conjecture 4. M^- . If the graph G is negatively curved, the inertia $\phi_G(v)$ has a unique global minimum and $\text{cm}(G)$ is unique. (This result is already known for global Busemann nonpositively curved spaces [12].)

Conjecture 5. M^+ . If G is nonnegatively curved, $\text{cm}(G)$ is not uniquely defined. (In a real-life, massive, nonnegatively curved network, the inertia $\phi_G(v)$ is nearly constant with v and $\text{cm}(G)$ might be hard to identify.)

Clearly, Conjecture M^\pm along with Conjecture G would yield Conjecture G^\pm .

3 Benchmark Examples

Several benchmark examples in here will provide support for these conjectures: a set of planar graphs in which the curvature is dictated by the valence (degree) of the nodes, as shown in Fig. 2. As shown in Fig. 3, we examine the negatively curved cases of valence 7, 8 and 9, hence of curvature $\frac{1}{2\pi} (2\pi - 7\frac{\pi}{3}) = -\frac{1}{6}$, $-\frac{1}{3}$ and $-\frac{1}{2}$, respectively, in which significant traffic congestion occurs at the *centroid* of the graph when least cost routing (Dijkstra's) algorithm is applied. Then we contrast the results with those of a vanishing curvature graph of valence 6 (curvature=0), in which the congestion is more smoothly distributed over all nodes. Towards a more realistic situation, we then look at a case of mixed valence. We then proceed to positively curved graph of valence <6 , in which the situation is drastically different than in negative curvature, as the traffic is uniformly distributed!

Clearly, as shown in Fig. 4, as the curvature becomes more and more negative, the vertex carrying the heaviest traffic in the graph becomes more and more congested relative to the other nodes, consistently with the inertia at the center of mass becoming smaller and smaller. By increasing the node degree, with the same number of nodes and traffic demand, there are more

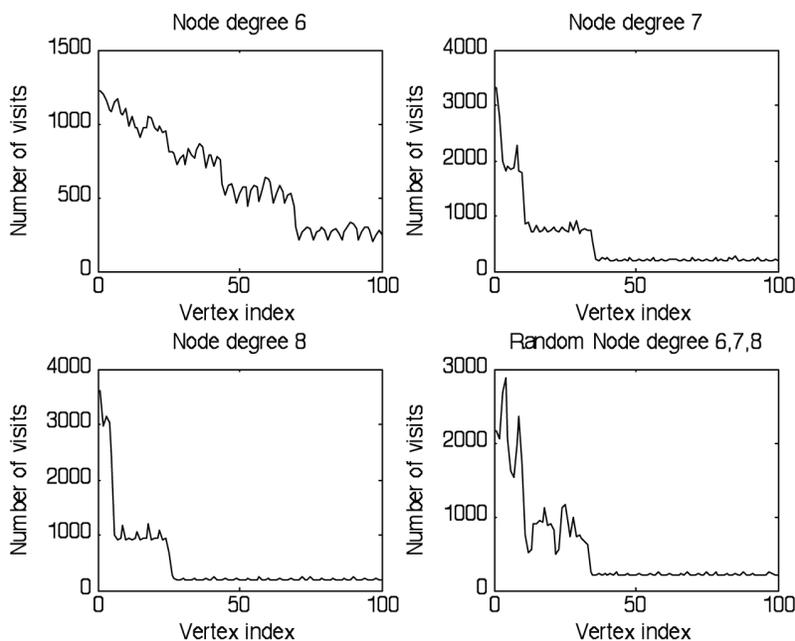


Figure 3: Traffic distribution in different planar networks: degree 6 everywhere, degree 7 everywhere, degree 8 everywhere, and random degree 6/7/8. The routing algorithm implements random pickup of equal cost paths.

connections in the network, so that the total traffic in the network decreases, but the traffic is heavier in the congestion center.

To further test Conjecture \mathbf{M}^- , especially to eliminate a possible contribution of the symmetrical structure of the graph to the congestion cases of the previous paragraph, we simulate the traffic and inertia distribution in a highly unsymmetrical network as shown in Fig. 5. The position of the heaviest traffic congestion point matches the node with minimum inertia. These results, in further, confirm our conjecture \mathbf{M}^- .

4 Proofs of Conjectures

4.1 Proof Conjecture \mathbf{G}^+ : Traffic in Positive Curvature

Consider the Platonic solids. All of these Platonic solids have their symmetry group Γ . This symmetry group Γ acts on the vertex set V_P of the Platonic solid P as a map $\Gamma \times V_P \rightarrow V_P, (g, v) \mapsto g(v)$, where g is an element of the symmetry group. Recall that the action of a symmetry group on a space is *vertex-transitive* if $\forall v, w \in V_P$ there exists a $g \in \Gamma$ such that $w = g(v)$.

It is easy to see that the action of the symmetry groups on the Platonic solids is vertex-transitive. We prove this as follows: Observe that all Platonic solids except the tetrahedron have dihedral (rotation) symmetries about axes joining the centers of pairs of opposite faces, while the tetrahedron has D_3 symmetry about the axis joining a vertex to the center of the opposite face. Then consider two vertices v, w on a Platonic solid. Join them by a sequence of consecutive edges. It is easy to see that the beginning vertex of an edge can be moved to the end vertex by a symmetry about the axis perpendicular to the center of a face comprising the edge.

With the above concept, it is easy to prove that the betweenness is uniform. Let $b_G(w)$

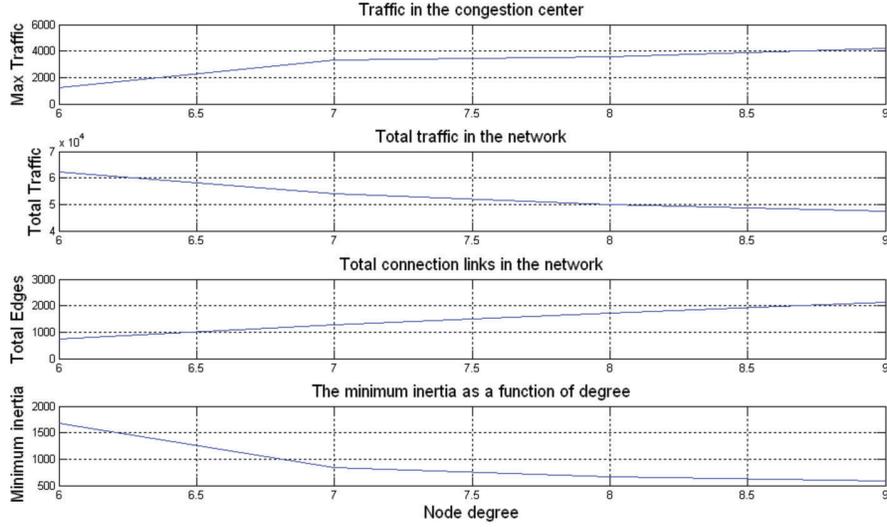


Figure 4: Traffic at the center of mass, total traffic in the network, and the total connection in the network.

denote the betweenness of the node v in the graph G . Then we have

$$b_G(w) = b_G(gv) = b_{g^{-1}G}(v) = b_G(v)$$

The only nontrivial part in the above string of equalities is the second one, where it is essential that the edge length be uniform. Indeed if (s, t) is a pair communicating via gv , we have $d(s, g(v)) + d(g(v), t) = d(s, t)$, from which it follows that $d(g^{-1}s, v) + d(v, g^{-1}t) = d(g^{-1}s, g^{-1}t)$, hence there is a pair $(g^{-1}s, g^{-1}t)$ communicating via v .

The proof that the inertia is uniform is essentially the same:

$$\phi_G(w) = \phi_G(gv) = \phi_{g^{-1}G}(v) = \phi_G(v)$$

The proof that $\tau(v)$ is uniform involves the edge-transitivity of the symmetry group. Let $\tau_G(e)$ be the traffic rate on edge e in the graph G . Then

$$\tau_G(e_2) = \tau_G(ge_1) = \tau_{g^{-1}G}(e_1) = \tau_G(e_1)$$

Again, in the second inequality, it is essential that the demand be uniform. From the above, it easily follows that $\tau_G(e_1) = \tau_G(e_2)$.

Hence we have the following result: For a uniformly distributed demand measure $\Lambda_d : V_P \times V_P \rightarrow \mathbb{R}^+$ on the squared power of the vertex set of one of the 5 Platonic solids, the traffic load $b : V_P \rightarrow \mathbb{R}^+$ is uniform, for a geodesic routing and provided the traffic is equally distributed among pairs of nodes.

More generally, by Higuchi's theorem [17], positively curved graphs are finite; next, the valence can only take values 3, 4, 5; therefore, the inertia and the congestion remain bounded from above and from below.

4.2 Quantitative Measure of Traffic in Disks

In general, the **traffic load density** in a convex subset X of a surface Σ is defined in a way inspired from (2), except for some normalization,

$$\lambda_t(X) = \frac{1}{A(\Sigma)^2 A(X)} \iint_{(s,t) \in \Sigma \times \Sigma} \ell(X \cap [s, t]) d\Lambda_d(s, t)$$

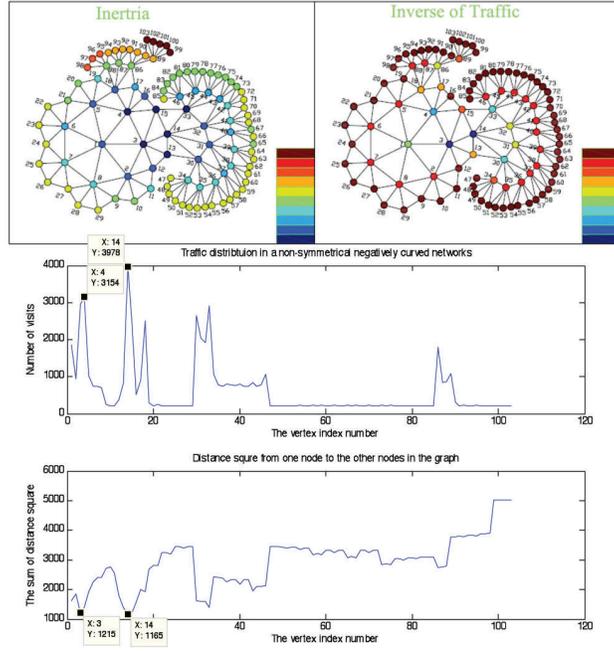


Figure 5: Traffic and inertia distribution in an unsymmetrical network. (Vertex #3 has maximum traffic and minimum inertia.)

Here Σ is the “network” that comprises all sources s and all targets t ; the normalization by the squared area $A(\Sigma)^2$ is justified by $d\Lambda_d(s, t) = A(ds)A(dt)$, and the normalization $A(X)$ is for obvious reasons. From here on, we specialize the computations to $\Sigma = B(R)$ and $X = B(r)$, a large ball and a small ball ($R \gg r$), respectively, with their common center at the origin of the Euclidean space \mathbb{E}^2 or the hyperbolic space \mathbb{H}^2 .

4.3 Conjecture G^+ : Traffic at the Center of a Euclidean Disk

The above double integral can be rewritten as the following:

$$\lambda_t(X) = \frac{1}{A(B(R))^2 A(B(r))} \int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{2\pi} \int_0^r \ell \{X \cap [s, t]\} \dots \\ \dots \times |\text{Jacobian}| \, du \, d\theta \, dl \, dl'$$

We first compute the Jacobian relative to the change of variables from Cartesian coordinates to polar coordinates. Assume the points s and t are at (x, y) and (x', y') in Cartesian coordinates. As shown in Fig. 6, their corresponding representations in polar coordinates with (u, θ, l, l') are the following:

$$\begin{aligned} x &= u \cos \theta + (l + \sqrt{r^2 - u^2}) \cos \left(\theta + \frac{\pi}{2} \right) = u \cos \theta - (l + \sqrt{r^2 - u^2}) \sin \theta \\ y &= u \sin \theta + (l + \sqrt{r^2 - u^2}) \sin \left(\theta + \frac{\pi}{2} \right) = u \sin \theta + (l + \sqrt{r^2 - u^2}) \cos \theta \\ x' &= u \cos \theta + (l' + \sqrt{r^2 - u^2}) \cos \left(\theta - \frac{\pi}{2} \right) = u \cos \theta + (l' + \sqrt{r^2 - u^2}) \sin \theta \\ y' &= u \sin \theta + (l' + \sqrt{r^2 - u^2}) \sin \left(\theta - \frac{\pi}{2} \right) = u \sin \theta - (l' + \sqrt{r^2 - u^2}) \cos \theta \end{aligned}$$

Long but elementary calculations show that $|\text{Jacobian}| = l + l'$. Then

$$\begin{aligned} \lambda_t(X) &= \lim_{R \rightarrow \infty} \frac{\int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{2\pi} \int_0^r 2\sqrt{r^2-u^2}(l+l') \, du \, d\theta \, dl \, dl'}{(\pi R^2)^2 (\pi r^2)} \\ &= \frac{1}{\pi R} \end{aligned}$$

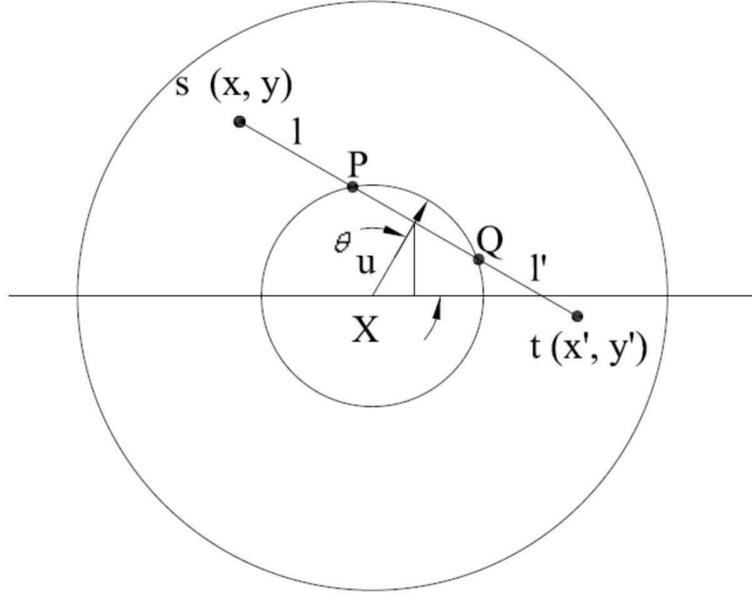


Figure 6: Traffic in Euclidean disk.

4.4 Conjecture G^- : Traffic at the Center of a Hyperbolic Disk

As shown in Fig. 7, the points s and t are at (x, y) and (x', y') , respectively, in the Cartesian coordinates of the Poincaré disk, $D = \{x + jy \in C : |x + jy| = r < 1\}$. It is, however, useful to parameterize the source and target by their representations in “polar” coordinates $(u, \theta, \bar{r}, \bar{r}')$, where the distances u, \bar{r}, \bar{r}' are hyperbolic. If r, \bar{r} represent the Euclidean and hyperbolic measurements, respectively, of the radius, then $r = \tanh(\frac{1}{2}\bar{r})$. As is well known, the area element is given by

$$dA = \frac{4dx dy}{(1 - |r|^2)^2} = \frac{4dx dy}{(1 - \tanh^2 \frac{1}{2}\bar{r})^2} = 4 \cosh^4 \left(\frac{1}{2}\bar{r} \right) dx dy$$

The Cartesian (x, y, x', y') versus polar-hyperbolic $(u, \theta, \bar{r}, \bar{r}')$ coordinate transformation is the following:

$$\begin{aligned} x &= \cos(\lambda + \theta) \cdot \tanh\left(\frac{1}{2}\bar{r}\right), & x' &= \cos(\theta - \lambda') \cdot \tanh\left(\frac{1}{2}\bar{r}'\right) \\ y &= \sin(\lambda + \theta) \cdot \tanh\left(\frac{1}{2}\bar{r}\right), & y' &= \sin(\theta - \lambda') \cdot \tanh\left(\frac{1}{2}\bar{r}'\right) \end{aligned}$$

where $\cos \lambda = \frac{\tanh u}{\tanh \bar{r}}$ and $\cos \lambda' = \frac{\tanh u}{\tanh \bar{r}'}$, per hyperbolic trigonometry in square angle triangles. Next,

$$\begin{aligned} |\text{Jacobian}|_{\bar{r}, \bar{r}' \rightarrow \infty} &= O\left(\frac{1}{\cosh^2\left(\frac{1}{2}\bar{r}\right) \cosh^2\left(\frac{1}{2}\bar{r}'\right)}\right) \\ \lambda_t(X) &= \frac{1}{A(B(R))^2 A(B(r))} \iint_{(s,t) \in B(R) \times B(r)} \ell\{X \cap [s, t]\} A(ds) A(dt) \\ &= \frac{1}{A(B(\bar{R}))^2 A(B(\bar{r}))} \iint \iint_{B(\bar{R}) \times B(\bar{r})} \ell\{X \cap [s, t]\} \times (16 \cosh^4 \bar{r} \cdot \cosh^4 \bar{r}') dx dx' dy dy' \\ &= \int_r^{\bar{R}} \int_r^{\bar{R}} \int_0^{2\pi} \int_0^{\bar{r}} \ell\{X \cap [s, t]\} \times (16 \cosh^4 \bar{r} \cdot \cosh^4 \bar{r}') \times |\text{Jacobian}| du d\theta d\bar{r} d\bar{r}' \\ &\approx O\left[\frac{\sinh^2\left(\frac{\bar{R}}{2}\right) \cosh^2\left(\frac{\bar{R}}{2}\right)}{\sinh^2\left(\frac{\bar{R}}{2}\right) \sinh^2\left(\frac{\bar{R}}{2}\right)}\right] \quad (\bar{R} \rightarrow \infty) \\ &= O(\text{constant}) \end{aligned}$$

(The reader is referred to [15] for the details.)

Moreover, since $\Delta d^2(\cdot, z) > 0$, it follows that ϕ is subharmonic in the strong sense that $\Delta\phi > 0$. By rotational symmetry, $\phi(v)$ is constant on $v \in \partial B(r)$, $r < R$. Write this value as $\phi(\partial B(r))$. By the subharmonic property, it follows that $\phi(0) \leq \phi(\partial B(r)), \forall r < R$. For any point $v \in B(R)$, we obviously have $v \in \partial B(|v|)$, with $|v| < R$, so that $\phi(0) \leq \phi(\partial B(|v|)) = \phi(v)$. Hence the minimum is reached at $v = 0$. It remains to show that the minimum is unique, that is, to show that the preceding inequality can be strengthened to a strict inequality. From the strengthened subharmonic property $\Delta\phi > 0$ and the Green function argument of [11, page 9], it follows that $\phi(\partial B(|v|))$ is *strictly* increasing with $|v|$. Hence $\phi(v)$ reaches its (unique) minimum at $v = 0$. \square

5 Shortest-Path Routing vs. Curvature Based Load Balancing

5.1 Traditional Shortest-Path Routing

In the previous sections, we have shown, from a theoretical point of view, that for uniformly distributed demand the shortest path length routing in negatively curved networks causes congestion over a small number of vertices; moreover, these vertices with heavy traffic rate occur near the center of mass of the network. In this section, we more specifically look at this congestion phenomenon in the practical setting of traffic overload in communication network. To make this problem more specific and straightforward, we focus our attention on multicasting traffic, even though our theorem can be applied to more general communication network paradigms, such as VoIP and multimedia networking, mobile Ad-Hoc networks, wireless sensor networks, etc., where traffic congestion and routing algorithms are the big concerns in the design of those communication networks.

Multicasting could involve almost all layers of a communication network. A multicast task can be performed at the application layer, where a hybrid network is a good model for this application, as will be presented later. A multicast task can also go systematically through the physical, link, and network layers. The increasing popularity of group communication applications such as teleconference and information dissemination services has led to an increasing interest for the development of multicast transport protocols. However, these transport protocols could cause congestion collapse if they are widely used, as they ignore the curvature and are hence prone to the related congestion problems discussed above.

Two basic multicast tree algorithms are currently available in the industry: one is the dense-mode algorithm; the other is the sparse-mode algorithm. Both multicast tree algorithms are at the heart of the multicast protocols, such as the Distance Vector Multicast Routing Protocol (DVMRP) in dense-mode, and the Protocol-Independent Multicast (PIM) operating in both dense mode and sparse mode. As shown in Fig. 8a, the dense-mode uses the source-based tree. It determines a shortest-path tree to all destinations first, and then uses a reverse shortest-path tree rooted at a source. So the spanning tree starts at the source and guarantees the lowest cost from a source to all leaves of the trees. The sparse-mode algorithm uses a shared-tree technique which uses a rendezvous point (RP) to connect sources and receivers. This rendezvous point acts as the core or root to coordinate forwarding packets from source to receivers under its distribution subtrees, as shown in Fig. 8b.

We used the Network Simulator (**ns-2**) to build up the traffic congestion environment in multicast communication. To make our simulation straightforward, we focus our attention on the congestion versus network curvature issue by ignoring the dynamic change in the group membership and using User Datagram Protocol (UDP) as the sources. In further, we apply the same topology structure (uniform node degree 6, 7 or 8) into the **ns-2** simulation as the one already used in Section 3. A snapshot of the visualization with **ns-2** NAM (the Network

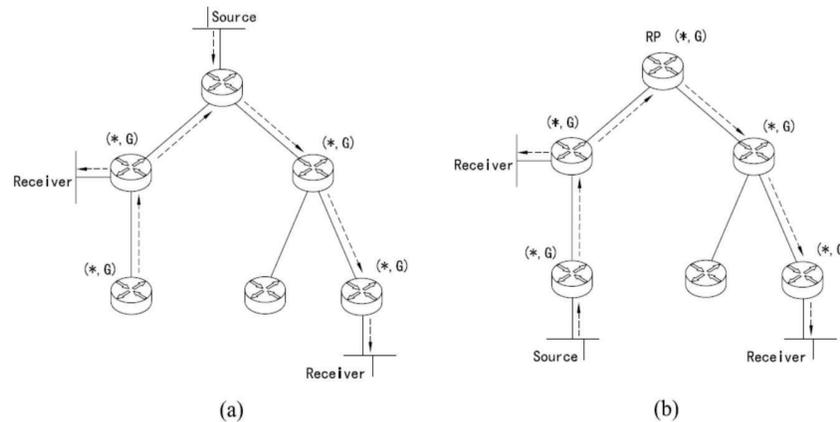


Figure 8: Two methods for constructing multicast algorithms: (a) dense mode, using a source-based tree; (b) sparse-mode, using a shared tree.

Animator) is shown in Fig. 9. In this figure, the node degree is 8 with a total of 100 nodes in the graph with node #0 at the centroid (a similar layout as the one shown in Fig. 2).

Other important networking settings are the followings:

- Every node in this graph is multicasting to all the other nodes in the network.
- The maximum buffer size of the queue in every link between two nodes is 1000 bytes pks, and every link is a duplex-link with 1Mb bandwidth, with a response time of 2ms.
- The size of every file is fixed to 2000 bytes.
- The start time of every UDP source is an exponential random variable with average value 0.01, and the interval time between two successive UDP packets for the source is 2.0 seconds.

As shown in Fig. 9, with the above setting, the network with 100 nodes and node degree 8 has congestion at nodes #0, #2, #3, #4, #5, as revealed by heavy packet drops. There is no such congestion for the network with node degree 6.

5.2 Load Balancing Routing

Load balancing algorithms are widely used to curb the congestion. For example, Cisco IOS router software has built-in load balancing functionality, and is available across all router platforms. It allows a router to use multiple equal cost paths to a destination when forwarding packets. The fundamental mechanism is as follows: When the router must select a route from many with the same administrative distance, the routers choose the path with the lowest congestion cost to the destination. In further, one can select load-balancing to work per-destination or per-packet. As shown in Table 1, from [7], the position of the asterisk (*) points to the interface over which the next packet/destination-based flow is sent; and the asterisk (*) keeps rotating among the equal cost paths each time a packet/flow is served.

However, in most cases, negatively curved networks have worst congestion problem, and the current load-balancing algorithms cannot alleviate it. The reason is this: first of all, there are not many multiple paths with the same administrative distance, since negatively curved manifolds have no conjugate points as the positively curved manifolds have. Second, even if we allow for quasi-optimal paths, there are still too close to the optimal one to bypass the congestion points.

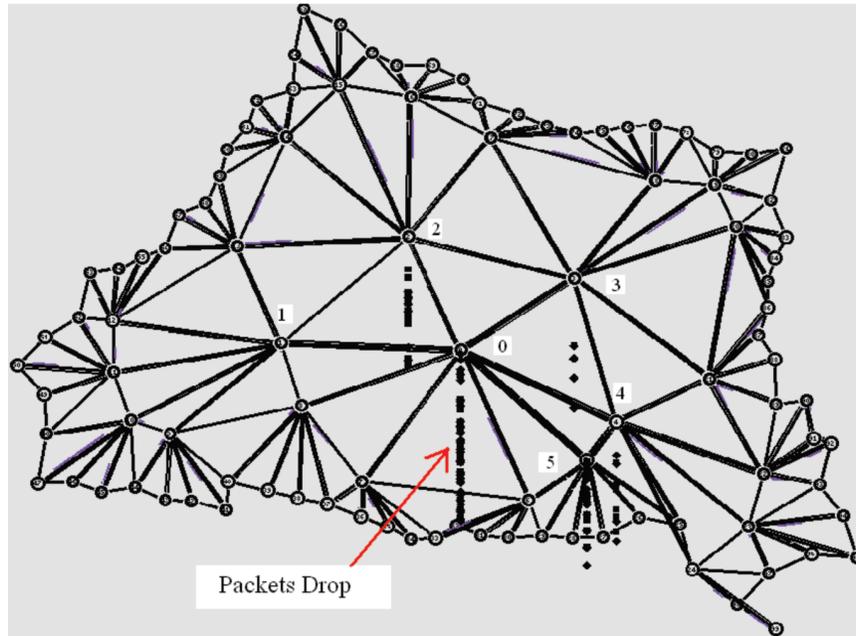


Figure 9: Snapshot of the visualization of network with node degree 8 and its packet loss using ns-2 Nam.

So, in here, we propose a curvature based load-balancing algorithm. The system diagram is shown in Fig. 10. The curvature κ of the network is used to control a switch. If the network is nonnegatively curved, $\kappa \geq 0$, the weight of the edges in the network is the administrative distance; and the shortest path is calculated based on that. Therefore, traditional load-balancing is used as we mentioned above. If the network is negatively curved, the weight of the edge between two directly connected vertex v_i and vertex v_j is reassigned to be:

$$\tilde{w}(v_i v_j) = \underbrace{\left(\sum_k d^2(v_k, v_i) \right)^{-1}}_{u(v_i)} w(v_i v_j) \underbrace{\left(\sum_j d^2(v_j, v_k) \right)^{-1}}_{u(v_j)} \quad (3)$$

where d is the administrative distance.

A modified graph is generated with these edge weights instead of the administrative distances. The curvature will be positive in this modified graph since $\chi = 2 > 0$. The inertia distribution will tend to be flat since the edges close to minimum inertia vertices (with heaviest traffic) of the original graph are assigned larger weights to increase the inertia so that the routing curbs the traffic along those edges. Fig. 11 compares the traffic distribution with and without the curvature based load-balancing. In this experiment, we use the node degree 7 network. The heaviest traffic drops from 3340 to 1756 after the curvature based load-balancing. It is a 47% decrease. Since the paths have to be detoured from the congestion vertex through extra routers, it will cause an increase of the total traffic in the network. The total traffic with the load-balancing is 69126 compared with 53964 without the load-balancing. It is a 28% increase. Fig. 12 compares the typical routings with and without load-balancing.

Table 1: In load balancing [7], the asterisk (*) keeps rotating among equal cost paths.

```

M2515-B# show ip route 1.0.0.0
Routing entry for 1.0.0.0/8
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Advertised by rip (self originated)
  Last update from 192.168.75.7 on Serial1, 00:00:00 ago
  Routing Descriptor Blocks:
  * 192.168.57.7, from 192.168.57.7, 00:00:18 ago, via Serial0
    Route metric is 1, traffic share count is 1
    192.168.75.7, from 192.168.75.7, 00:00:00 ago, via Serial1
    Route metric is 1, traffic share count is 1
    
```

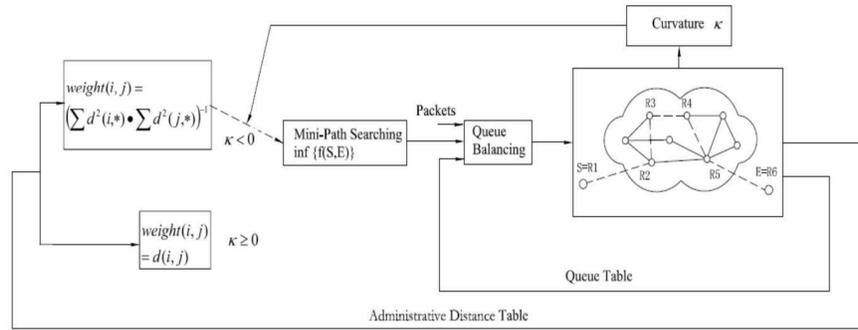


Figure 10: The system level diagram of curvature based load balancing.

5.3 Load Balancing by Yamabe Flow

The link weight reassignment (3) fundamentally smoothes over the inertia of the graph and, as a corollary of the various conjectures and results, alleviates the congestion by distributing the traffic more uniformly. From a deeper mathematical viewpoint, the new link weight \tilde{w} is in fact a *conformal transformation* [16] of the original weight w . The *combinatorial Yamabe flow* [16] is a refined procedure that iterates on the conformal factor $u : V \rightarrow \mathbb{R}^+$ to produce, subject to no obstruction, a metric of uniformly positive curvature. More specifically, the *combinatorial Yamabe flow* on a triangulated surface is the system of ODE's

$$\frac{du(v_i, t)}{dt} = -K_{u*d}(v_i)u(v_i, t), \quad u(v_i, 0) = 1$$

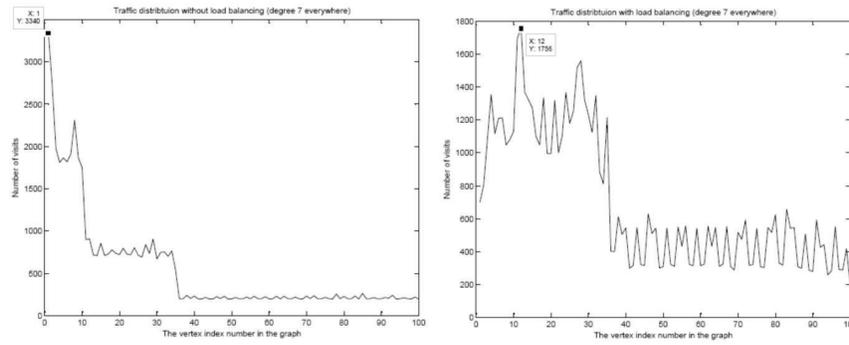


Figure 11: Traffic distribution of node degree 7 network; left: without load-balancing; right: with load-balancing.

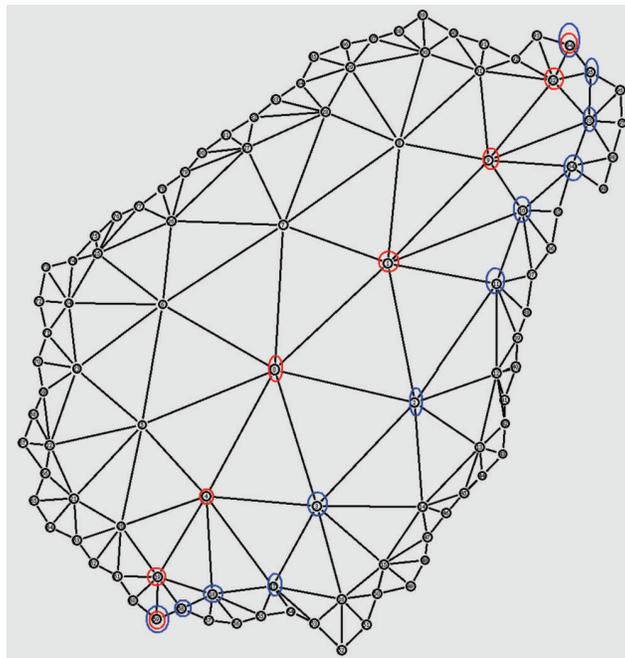


Figure 12: Routing with and without the curvature-based load balancing (Red circle: without load balancing; Blue circle: with load balancing.)

where $u(v_i, t)$ is the *conformal factor* associated with the vertex v_i at time t , $u * d$ is the *conformally modified administrative distance* defined as $u * w(v_i v_j) = u(v_i)w(v_i v_j)u(v_j)$, and $K_{u*d}(v_i)$ is the combinatorial curvature (the numerator of (1)) at v_i for the metric $u * w$. By the piecewise linear (PL) Gauss-Bonnet theorem, $\sum_i K(v_i) = 2\pi\chi$, where $\chi = |F| - |E| + |V|$ is the Euler characteristic. Thus a metric of uniformly positive curvature exists only if $\chi > 0$, which is the case for the triangulation of Fig. 12. On such a triangulation, the Yamabe flow will converge to a metric of constant positive curvature, unless it reaches a removable singularity. The latter is a degenerate triangle $\Delta v_i v_j v_k$ of the triangulation, that is, $u * w(v_i v_j) + u * w(v_j v_k) = u * w(v_i v_k)$. This singularity is easily removed by deleting the link $v_i v_k$. Thus, to alleviate congestion, some links might have to be removed, a phenomenon otherwise referred to as *Braess paradox*.

6 Conclusion and Future Work

We have proposed to utilize coarse geometry concepts to analyze the traffic pattern in networks, especially in negatively curved networks. We have found that the Alexandrov angles provide relevant curvature information, consistently with the Gauss concept. The latter provides the quintessence of the topological structure of a network. Networks with different curvatures have drastically different behaviors as far as traffic, random walk, percolation processes, etc. are concerned. Negatively curved networks are prone to congestion. Because of the pervasive implication of negative curvature, the congestion cannot be really alleviated, unless more drastic action—curvature control—is implemented by a Yamabe-like scheme. The remaining challenge is to implement the Yamabe flow in some flooding scheme.

7 Acknowledgment

This research was partially supported by National Science Foundation grant NetSE 1017881.

Bibliography

- [1] F. Ariaei and E. Jonckheere, Cooperative ‘curvature-driven’ control of mobile autonomous sensor agent network, *IEEE Conference on Decision and Control*, New Orleans, LA, December 2007, pp. 1453-1458.
- [2] Y. Baryshnikov, On the curvature of the Internet, in *Workshop on Stochastic Geometry and Teletraffic*, Eindhoven, The Netherlands, April 2002.
- [3] L. M. Blumenthal, *Theory and Applications of Distance Geometry*, Oxford at the Clarendon Press, London, 1953.
- [4] M. Boguna, F. Papadopoulos and D. Krioukov, Sustaining the Internet with hyperbolic mapping, *Nature Communications*, volume 1, article number 62, September 2010.
- [5] M. Bonk and O. Schramm, Embeddings of Gromov hyperbolic spaces, *Geom. Funct. Analysis*, volume 10, pp. 266-306, 2000.
- [6] D. Burago, Y. Burago, and S. Ivanov, *A Course in Metric Geometry*, volume 33 of Graduate Study in Mathematics, American Mathematical Society, Providence, Rhode Island, 2001.
- [7] Cisco, How Does Load Balancing Work? *Cisco Document ID: 5212,2005*. http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml.
- [8] E. Ghys and P. de la Harpe, editors. Sur les groupes hyperboliques d’après Mikhael Gromov, Number 83 in *Progress in Mathematics*, Birkhauser, Boston, MA, 1990. (Papers from the Swiss Seminar on Hyperbolic Groups held in Bern, 1988.)
- [9] M. Gromov, Hyperbolic groups, in S. M. Gersten, editor, *Essays in Group Theory*, volume 8 of Mathematical Sciences Research Institute Publication, pages 75-263, Springer-Verlag, New York, 1987.
- [10] M. Gromov, *Metric Structures for Riemannian and Non-Riemannian Spaces*, volume 152 of Progress in Mathematics, Springer-Verlag, 1999.
- [11] S. D. Fisher, *Function Theory on Planar Domains: A Second Course in Complex Analysis*, John Wiley & Sons, New York, 1983.

-
- [12] J. Jost, *Nonpositive Curvature: Geometric and Analytic Aspects*, Birkhauser, Lectures in Mathematics, Basel-Boston-Berlin, 1997.
 - [13] J. Jost, *Riemannian Geometry and Geometric Analysis*, Second Edition, Springer, Universitext, Berlin, Heidelberg, New York, 1998.
 - [14] M. Lou and E. A. Jonckheere, Tracking and mitigating piracy, *American Control Conference*, Minneapolis, MN, June 14-16, 2006, paper WeA20.2, Session: Networks and Control, pp. 656–661.
 - [15] M. Lou, Traffic pattern analysis in negatively curved network, Ph.D. dissertation, Department of Electrical Engineering, University of Southern California, Los Angeles, Calif, USA, 2008.
 - [16] F. Luo, Combinatorial Yamabe flow on surfaces, *Communications in Contemporary Mathematics*, volume 6, number 5, pp. 765-780, 2004.
 - [17] L. Sun and X. Yu, Positively curved cubic plane graphs are finite, *Journal of Graph Theory*, volume 47, number 4, pp. 241-274, 2004.