Chapter 7

Scaling of Congestion in the Internet

The Internet grows in size every day. As time progresses, more end-hosts are added to the edge of the network. Correspondingly, to accommodate these new end-hosts, ISPs add more routers and links. History has shown that the addition of these links maintains certain macroscopic properties of the Internet. For example, the interconnection between ISPs in the Internet continues to follow a power law graph structure [32]. Also, the addition of new end-hosts over time places a greater load on the network as a whole. Fortunately, the improvement of network technology operates over the same time period. We expect the network links at the edge and core of the network to improve by a similar performance factor, since they both typically follow similar Moore’s Law-like technology trends.

Unfortunately, due to the topology of the network and behavior of Internet routing, the increase in load may be different on different links of the network. As a result, it may be necessary for the speed of some key “centrally located” hot-spot links in the network to improve much more quickly than others. If this is true, then these key parts of the network will eventually emerge into persistent bottlenecks. Under these circumstances, routing-based mechanisms can no longer be employed by end-networks to improve their performance. Consider route control, for example. Multihoming paths will still have to traverse links in tier-1 ISPs (due to the global reach of the ISPs). Similarly, for most placements of overlay nodes, even overlay paths will traverse tier-1 networks (although the likelihood of this happening is lower compared to route control). Therefore, if links belonging to such backbone carriers emerge as hot-spots, neither approach can help end-networks overcome these inevitable hot-spots. We can then say that the network has poor scaling properties. In such a situation, we would either need to drastically adjust the Internet’s routing behavior or change the structure of the network (for example, the interconnections between ISPs) to ensure good future performance.

On the other hand, if the worst congestion scales well with the network size then we can expect the network to continue to operate as it does now. Also, routing-based mechanisms will continue to be effective at improving end-point performance. In this chapter, we perform a preliminary study of the scaling properties of the Internet. Using reasonably realistic theoretical models of network
evolution and inter-domain routing, we seek to answer the following question:

How does the maximum congestion in the Internet scale with the network size?

Our analysis focuses on the Internet AS-level interconnection. We consider a model of network evolution based on Preferential Connectivity [18], and a simple model of traffic in which a unit amount of flow between every pair of nodes is routed along the shortest path between them. We employ simple combinatorial/probabilistic arguments to give bounds on the maximum congestion in the AS-level graph. We also conduct simulations of the congestion on the links in the network, based both on real and on synthetically generated AS-level topologies and synthetic traffic matrices. Through our simulations, we also investigate the impact of several key factors on the worst congestion in the network, such as variants of the inter-domain routing algorithm, alternate traffic matrices, and finally, alternate topologies for the AS-level interconnection.

The key result in this chapter is that the maximum congestion in Internet-like graphs scales poorly with the growing size of the graph. Specifically, the maximum congestion for shortest path routing and uniform traffic matrices is worse than \( n^{1+\Omega(1)} \), with the exponent depending on the degree of “skew” in the power law degree distribution of the graph\(^1\). Our simulations show that policy routing in the AS graph results in roughly the same maximum congestion as shortest path routing, but certainly not worse. When alternate, non-uniform traffic models are considered, the congestion scaling properties of power law graphs worsen substantially. We also show that in terms of the maximum congestion, power law trees are considerably worse than power law graphs. In contrast, graphs with exponential degree distribution have very good congestion properties.

Further, we also discuss simple guidelines to change the ISP-level interconnections that result in a dramatic improvement in the congestion scaling properties of Internet-like graphs. We show that when parallel links are added between adjacent nodes (i.e., ASes) in the network according to simple functions of their degrees or the number of neighboring ASed (e.g., the minimum of the two degrees), the maximum congestion in the resulting graph scales linearly. This heuristic for adding edges reflects the desired amount of peering between neighboring ASes in the Internet in order to guarantee good overall congestion scaling properties.

**Chapter outline.** In Section 7.1, we formalize our analytical approach and discuss our simulation set-up. The analysis is presented in Section 7.2. Section 7.3 presents the results from our simulations. In Section 7.4, we discuss the implications of our results on network design and present mechanisms to alleviate the poor congestion scaling of the Internet graph. In Section 7.5, we survey past work on modeling the Internet and results for deriving congestion scaling properties of general\(^1\)

---

\(^1\)There is some disagreement about whether a power law correctly models the degree distribution of the Internet graph. However, it is widely agreed that the distribution is heavy-tailed. While our main results (specifically, simulation results) focus on power law distributions, we believe that they hold equally well for other such heavy-tailed distributions (e.g. Weibull).
graphs. Finally, in Section 7.6, we summarize the key observations in this chapter and present a few caveats of our approach.

7.1 Methodology

We first give a precise formulation of the problem, laying out the key questions we seek to address via analysis. We also describe the simulation set-up for corroborating and extending our analytical arguments.

7.1.1 Problem Statement

Let $G = (V, E)$ be an unweighted graph, representing the Internet AS-level graph, with $|V| = n$. Let $d_v$ denote the total degree of a vertex $v$ in $G$. We are given three key aspects pertaining to the graph $G$: the degree distribution of the graph, the routing algorithm used by the nodes in the graph to communicate with each other and the traffic demand matrix determining the amount of traffic between pairs of nodes in the graphs. We give precise definitions of these three aspects, in turn, below.

We will mostly be concerned with graphs having a power law degree distribution, defined below.

**Definition 1** We say that an unweighted graph $G$ has a power law degree distribution with exponent $\alpha$, if for all integers $d$, the number of nodes $v$ with $d_v \geq d$ is proportional to $d^{-\alpha}$.

Similarly, graphs with exponential degree distribution are those in which the number of nodes $v$ with $d_v \geq d$ is proportional to $e^{-\beta d}$, for all integers $d$. Henceforth, we will refer to such graphs as power law graphs and exponential graphs respectively.

Let $S$ denote a routing scheme on the graph with $S_{u,v}$ representing the path for routing traffic between nodes $u$ and $v$. We consider two different routing schemes:

1. **Shortest Path Routing:** In this scheme, the route between nodes $u$ and $v$ is given by the shortest path between the two nodes in the graph $G$. This reflects route selection in BGP where every AS prefers paths with the least number of ASes. When there are multiple shortest paths, we consider the maximum degree of nodes along the paths and pick the one with the highest maximum degree. This tie-breaking rule is reflective of the typical policies employed in the Internet—higher degree nodes are typically much larger and much more well-provisioned providers than lower degree nodes and are in general used as the primary connection by

---

2There is some disagreement about whether a power law correctly models the degree distribution of the AS-level graph. However, it is widely agreed that the distribution is heavy-tailed. While our main results (specifically, simulation results) focus on power law distributions, we believe that they hold equally well for other such heavy-tailed distributions (e.g. Weibull).
stub networks. In Section 7.3.3, we consider alternate tie-breaking schemes such as random choice and favoring lower degree nodes, and show that the tie-breaking rule does not affect our results.

2. **Policy Routing:** In this scheme, traffic between nodes \( u \) and \( v \) is routed according to BGP-policy. We classify edges as peering edges or customer-provider edges (that is, one of the ASes is a provider of the other). These commercial relations between ISPs are known to give rise to “valley-free” routing, in which each path contains a sequence of customer to provider edges, followed by at most one peering edge, followed by provider to customer edges. The key difference between policy routing and shortest path routing, then, is that in the former case, we are restricted to choosing between multiple shortest path routes which are all compliant with valley-free routing. Our goal in studying these two forms of routing is to understand if commercial policies between ISPs significantly change our observations regarding network congestion.

A traffic vector \( \tau \) is a vector containing \( \binom{n}{2} \) non-negative terms, with the term corresponding to \((u,v)\) signifying the amount of traffic between the nodes \( u \) and \( v \). The congestion on an edge \( e \) due to traffic vector \( \tau \) and routing scheme \( S \) is given by the sum of the total amount of traffic that uses the edge \( e \):

\[
C_{\tau,S}(e) = \sum_{(u,v) : e \in S_{u,v}} \tau(u,v).
\]

We define the edge congestion due to traffic vector \( \tau \) and routing scheme \( S \) to be the maximum congestion on any edge in the graph:

\[
\text{EDGE-CONGESTION}_{\tau,S}(G) = \max_{e \in E} C_{\tau,S}(e).
\]

Our goal is to quantify the congestion in a graph with power law degree distribution, for shortest path and policy routing schemes, due to different traffic vectors. Specifically, we consider the following three traffic vectors:

1. **Any-2-any:** This corresponds to the all 1s traffic vector, with a unit traffic between every pair of ASes. While very simplistic, this model is amenable to analysis and provides a baseline for comparison against more complex traffic vectors.

2. **Leaf-2-leaf:** In order to define this model, we classify nodes in the graph as *stubs* and *carriers*. Stubs are nodes that do not have any customers. In other words, consider directing all customer-provider edges in the graph from the customer to the provider. Peering edges are considered to be bi-directed edges. Then, vertices with no incoming edges (corresponding to ASes with no customers) are called stubs or leaves in the graph. In this model, there is a unit of traffic between every pair of stubs in the graph. No other AS sources or sinks traffic on its own.

3. **Clout:** This model approximates the fact that “well-placed” sources, that is, sources which have a high degree and are connected to high degree neighbors, are likely to transmit larger
amounts of traffic than other sources. Accordingly, in this case, \( \tau(u, v) = f(d_u, c_u) \), where \( u \) and \( v \) are stubs, \( c_u \) is the average degree of the neighbors of \( u \) and \( f \) is an increasing function. As in the previous case, there is no traffic between nodes that are not stubs. In what follows, we only use the function \( \tau(u, v) = f(d_u, c_u) = d_u c_u \) for stubs \( u, v \). This is the most sophisticated of the traffic vectors we consider.

Admittedly, our choice of the models for Internet routing, as well those for Internet traffic matrices, are somewhat unrealistic. We do try to model real phenomena, such as popularity of some ASes, but make no claims to the realism of our models. However, we still use them in our analysis for reasons of simplicity and for lack of realistic Internet-wide traffic traces.

7.1.2 Simulation Set-up

Our simulations serve two purposes: (1) to corroborate our theoretical results, and, (2) to characterize the congestion in more realistic network models than those considered in our analysis.

Our simulations are run on two different sets of graphs. The first set of graphs contains maps of the Internet AS topology collected at 6 month intervals between Nov. 1997 and April 2002, available at [65]. The number of nodes in any graph in this set is at most 13000, the maximum corresponding to the April 2002 set. The second set of graphs contains synthetic power law graphs generated by Inet-3.0 [100]. In this set, we generate graphs of sizes varying from \( n = 4000 \) to \( 50000 \) nodes.

As mentioned earlier, in order to implement the leaf-2-leaf and clout models of communication, we need to identify stubs in the network (note that these might have a degree greater than 1). Additionally, in order to implement policy routing, we need to classify edges as peering or non-peering edges. To do this for the real AS graphs, we employ the relationship inference algorithms of Gao [37] to label the edges of the graphs as peering or customer-provider edges. These algorithms use global BGP tables [20] to infer relationships between nodes. Then, we use these relationships to identify stubs, as nodes that are not providers of any other node. Henceforth, we shall refer to the real AS graphs as accurately labeled real graphs (ALRs).

Labeling edges and identifying stubs in the synthetic graphs of Inet is more tricky since we do not have the corresponding BGP information. We will refer to synthetic graphs, labeled using the algorithms described below, as heuristically labeled synthetic graphs (HLSs). We use different algorithms for classifying nodes (this is key to implementing leaf-to-leaf communication) and edges (this is key to implementing policy routing in synthetic graphs). We discuss these next.

**Stub identification.** We identify stubs in synthetic graphs as follows: For any edge \( e = (v_1, v_2) \), we assign \( v_1 \) to be the provider of \( v_2 \) whenever \( \text{degree}(v_1) \geq \text{degree}(v_2) \). Notice that we do not explicitly identify peering edges (although edges between nodes of identical degree will be bidirectional). We then identify stubs in graphs labeled as above.

\(^3\)In all our simulations, for any metric of interest, for each \( n \), we generate 5 different graphs of \( n \) nodes (by varying the random seed used by the Inet graph generator) and report the average of the metric on the 5 graphs.
We test the accuracy of this stub-identification algorithm on real AS graphs by comparing the labels produced by our algorithm to the true labels of ALRs, and compute the fraction of false positives and false negatives\(^4\) in these. The accuracy results are shown in Figure 7.1. Note that our simple algorithm has a very low error rate. Notice that the inference algorithms of Gao [37] have some error intrinsically and hence some of the labels on the ALRs might actually be inaccurate.

**Edge classification.** Simply considering all edges in the graph to be customer-provider edges, as done above, is not useful for the purposes of edge classification. Specifically, it results in a significant error on the maximum congestion in real graphs employing policy routing (results omitted for brevity). In order to improve the accuracy of labeling edges, we resort to machine learning algorithms. However, coming up with a good machine learning algorithm for the classification is a challenging task, because there is a huge bias toward customer-provider edges in the graphs (roughly 95% of the edges are customer-provider edges). We use the 3-Nearest Neighbor [62] algorithm for classifying edges as peering or non-peering: each edge in the unlabeled graph is classified as a peering edge if among the three edges most similar to it in the labeled graph, at least two are peering edges. Similarity between edges is judged based on the degrees of their respective end points and neighboring vertices. We measure the accuracy of the procedure by applying it to real graphs and then comparing the classification with true labels.

Our machine learning algorithm gives only 20% accuracy on peering edges and about 95% accuracy on customer-provider edges. However, for the purposes of computing the worst congestion in the graph, this low accuracy of labeling is, in fact, enough. Indeed, as shown in Figure 7.1(b), labeling real graphs using our algorithm results in an error of less than 10% in the worst congestion

\(^4\)False positives are nodes that are identified as stubs by the algorithm, but are not stubs in the ALR. False negatives are stubs in the ALR that are not identified as stubs by the algorithm.
while employing policy routing) in comparison with the congestion computed on ALRs. More importantly, the trends in congestion growth are identical in the two cases.

Other topologies. In addition to power law graphs, we also study congestion in power law trees and exponential topologies. A comparison of the former with power law graphs gives an insight into the significance of density of edges in the graph. The latter model is interesting because most generative models for power law topologies result in exponential distributions in the “fringe” cases [31]. Our power law tree topologies evolve according to the Preferential Connectivity model [18]. To generate exponential topologies, we modify Inet-3.0 to generate an exponential degree distribution first and then add edges in Inet’s usual way. For a given $n$, the exponent $\beta$ for the exponential graphs on $n$ nodes is chosen such that the total number of edges in the exponential graph is very close to that of the corresponding power law graph on $n$ nodes$^5$. Note that due to a lack of real data for exponential graphs, we do not have a good way of labeling edges and nodes in them. Therefore, we do not perform experiments with policy routing or the leaf-2-leaf and clout traffic models for them.

7.2 Analytical Results

In this section, we show that the expected maximum edge congestion in a power law graph, specifically the congestion on the edge between the two highest degree nodes, grows as $\Omega(n^{1+\frac{1}{\alpha}})$ with $n$, when we route a unit flow between all pairs of vertices over the shortest path between them. We consider the Preferential Connectivity Generative Model of Barabasi et al. [18]. This model uses a fixed constant parameter $k$. The model starts with a complete graph on $k + 1$ nodes. This set of nodes is called the core of the graph. The graphs grows in time-steps. Let the graph at time $i$ be denoted $G^i$. At time step $i + 1$, one node is added to the network. This node picks $k$ nodes at random from $G^i$ and connects to them. Each vertex $v$ has a probability $\frac{d^i_v}{D^i}$ of getting picked, where $d^i_v$ is the degree of the vertex at time $i$, and $D^i$ is the total degree of all nodes at time $i$. At the end of $n$ steps, with $k = 3$, this process is known to generate a power law degree distribution. Also, it is easy to see that in the resulting power law graph (for that matter in any power law graph), the maximum node degree is $n^{1/\alpha}$.

In order to show a lower bound on the congestion of a power law graph, our plan is roughly as follows. We consider the edge between the two highest degree nodes in the core—$s_1$ and $s_2$. Call this edge $e^*$. For every vertex $v$ in the graph, we consider the shortest path tree $T_v$ rooted at vertex $v$. We show that in expectation, $\Omega(n)$ such trees contain the edge $e^*$. Moreover, in these trees, the expected number of nodes in the subtree rooted at edge $e^*$ is at least $\Omega(n^{1+\frac{1}{\alpha}})$. This gives us the lower bound in the following way: the routes taken by each connection are precisely those defined by the above shortest path trees; thus the congestion on any edge is the sum of congestions on the edge in these shortest path trees. Now, as described above, in $\Omega(n)$ shortest path trees, the congestion on

---

$^5$We employ heuristic hill-climbing to estimate the value of the exponent $\beta$ that minimizes error in the number of edges.
edge $e^*$ is at least $\Omega(n^{1/2})$. Therefore, the total congestion on edge $e^*$ is at least $\Omega(n^{1+1/2})$. Note that $e^*$ is not necessarily the most congested edge in the graph, so the maximum congestion could be even worse than $\Omega(n^{1+1/2})$. We get the following theorem:

**Theorem 1** The expected value of the maximum edge congestion in a power law graph with exponent $\alpha$ grows as $\Omega(n^{1+1/2})$ with $n$, when we route a unit flow between all pairs of vertices over the shortest path between them.

In the following, the distance between two nodes refers to the number of hops in the shortest path between the two nodes. We make a few technical assumptions. We assume that $1 < \alpha < 2$, and $s_1$ and $s_2$ are the highest degree nodes in the graph. For reasonably “small” numbers $h$, we assume that for any node $v$ in the graph, the number of nodes within distance $h$ of $v$ is less than the number of nodes within distance $h$ of $s_1$. In other words, $s_1$ is centrally placed in the graph. Here, “small” refers to distance around $s_1$ that contains lesser than half the nodes. These assumptions are justified by experimental evidence and some prior analysis [36] of the preferential connectivity generative model.

We begin with a technical lemma.

**Lemma 1** Let $r$ be the maximum integer for which at least $n^{r+1}$ vertices lie at a distance $r+1$ or beyond from $s_1$. Then, $\Omega(n)$ nodes lie within distance $r-1$ of every node in the core of the graph. In particular, for any node in the core, $\Omega(n)$ nodes lie at a distance exactly $r-1$ from it.

*Proof:* We prove that at least $\Omega(n)$ nodes lie within a distance $r-2$ of $s_1$. Then, since all vertices in the core are neighbors of $s_1$, these $\Omega(n)$ nodes lie within a distance $r-1$ of any vertex in the core of the graph. We begin by showing that at least $\Omega(n)$ nodes lie within a distance $r$ of $s_1$, and then extend this to nodes at distance $r-1$ and $r-2$. Let level $i$ denote the set of nodes at distance exactly $i$ from $s_1$.

Remove from the graph all vertices that are at level $r+2$ or higher. The remaining graph has at least $n^r$ vertices, by the definition of $r$. Now, assume that there are at least $n^{r+1}$ vertices at level $r+1$, otherwise, we already have $> \frac{2n}{n}$ nodes in levels 0 through $r$, implying that $\Omega(n)$ nodes lie within distance $r$ of $s_1$.

Now, let the number of nodes at level $r$ be $x$. All the nodes in level $r+1$ in the residual graph are connected to nodes in level $r$. So, their number is at most the size of the neighbor set of level $r$. Now, in the best possible case, the nodes in level $r$ could be the highest degree nodes in the graph. In this case, the minimum degree of any node in level $r$ is given by $y$ with $ny^{-\alpha} = x$. We get $y = \left(\frac{n}{x}\right)^{1/\alpha}$.

Now, the size of the neighborhood of level $r$ is at most the total degree of nodes in the level. This is given by

$$\int_y^n \frac{1}{z} z^{-\alpha} a^{-1} dz = \frac{an}{\alpha - 1} \left(y^{1-\alpha - \frac{1}{\alpha}} - n^{1-\alpha - \frac{1}{\alpha}}\right)$$
This quantity is at least \( \frac{n}{10} \) by our assumption above. Thus we get that \( x = \beta n \), where \( \beta = \left( \frac{1}{10} \left( 1 - \frac{1}{\alpha} \right) \right)^{\frac{\alpha - 1}{\alpha - 2}} \). This is a constant fraction of \( n \).

Now, we can apply the same technique to compute the number of nodes at level \( r - 1 \) and then, \( r - 2 \). We get that the number of nodes at level \( r - 2 \) is at least \( \beta^2 \left( \frac{1}{\alpha - 1} \right)^n \), with \( \beta \) as given above.

Let \( r \) be the distance defined by the above lemma. Let \( V_r \) denote the set of nodes that are within distance \( r - 1 \) of every node in the core of the graph (see Figure 7.2). By lemma 1, we have \( |V_r| = \Omega(n) \). Now, the proof of the theorem has two parts. The first shows that many trees \( T_v \) corresponding to \( v \in V_r \) contain the edge \( e^* \).

**Lemma 2** The expected number of shortest path trees \( T_v \), corresponding to nodes \( v \in V_r \), that contain the edge \( e^* \) is \( \Omega(n) \).

**Proof:** Consider the tree \( T_v \) for some node \( v \in V_r \). This is essentially a breadth first tree starting from \( v \). If \( s_1 \) and \( s_2 \) are at the same level in the tree, then the edge \( e^* \) is not contained in the tree. On the other hand, if the nodes are at different depths in this tree, let \( s_1 \) be closer to \( v \) without loss of generality. In this case, one shortest path from \( v \) to \( s_2 \) is via \( s_1 \) and since we break ties in favor of paths with high degree nodes, \( T_v \) will contain this path via \( s_1 \). This implies that \( e^* \) is contained in the tree. Thus, trees containing \( e^* \) correspond to those \( v \) that are not equidistant from \( s_1 \) and \( s_2 \). We now show that there are \( \Omega(n) \) nodes \( v \in V_r \) that are not equidistant from \( s_1 \) and \( s_2 \). This implies the result.
First, observe that if we pick a random node in the graph, then conditioned on the fact that this node lies at a distance \(d - 1\), \(d\) or \(d + 1\) from \(s_2\), there is at most a constant probability that this node lies at distance \(d\) from \(s_2\). This is because using an argument congruent to that in lemma 1, we can show that the number of nodes at distance \(d - 1\) from \(s_2\) is a constant fraction of the number of nodes at distance \(d\).

Now, consider the nodes at distance \(r - 2\) from \(s_1\). These are at least \(\Omega(n)\) in number (lemma 1) and lie in \(V_r\). Given that a node \(v\) is picked from this set, \(v\) is at a distance \(r - 3\), \(r - 2\) or \(r - 1\) from \(s_2\). By the above argument, the probability that this node lies at distance \(r - 2\) from \(s_2\) is at most a constant. Thus \(\Omega(n)\) nodes in this set are not at distance \(r - 2\) from \(s_2\) and we are done. ■

Next we prove that in any tree \(T_v\) \((v \in V_r)\) containing \(e^*\), \(e^*\) has a high congestion.

**Lemma 3** Let \(T_v\) be a shortest path tree, corresponding to \(v \in V_r\), that contains the edge \(e^*\). Then the expected congestion on edge \(e^*\) in this tree is \(\Omega(n^{1/\alpha})\).

**Proof:** Without loss of generality, let \(s_1\) be closer to \(v\) than \(s_2\). We show that the degree of \(s_2\) in \(T_v\) is \(\Omega(n^{1/\alpha})\). This implies the result. Let level \(i\) denote the set of nodes at distance \(i\) from \(v\) in the tree.

Let \(d\) be the distance between \(v\) and \(s_2\). All neighbors of \(s_2\) lie in levels \(\geq d - 1\) in the tree. Note that \(d \leq r - 1\). Therefore by our assumption, the number of nodes lying at levels \(\geq d + 1\) in the tree is at least the number of nodes at distance \(r\) or greater from \(s_1\). This number is at least \(\frac{n}{2}\), by the definition of \(r\). Let \(W\) denote the set of nodes that lie at levels \(\geq d - 1\) in the tree, and that arrived in the graph after step \(\frac{n}{4}\). Note that there are at least \(\frac{n}{4}\) nodes at level \(d + 1\) or higher that are in set \(W\). Therefore, a constant fraction of the nodes in \(W\) lie at levels \(\geq d + 1\) in the tree.

First observe that the probability that a node entering the graph at time step \(t\) attaches to \(s_2\) is roughly \(t^{d - 1}\). This probability increases as the graph becomes larger and larger, as this is related. By removing the first quarter of the nodes entering the graph from consideration, and using the fact that these nodes are less likely to attach to \(s_2\) than nodes arriving later, we conclude that the number of neighbors of \(s_2\) that arrive after step \(n/4\) is at least 3/4th of the total degree of \(s_2\).

Now all neighbors of \(s_2\) lie at levels \(\geq d - 1\) in the tree. Then, by the observation in the previous paragraph, we have that at least 3/4th of the neighbors of \(s_2\) lie in the set \(W\). Note that when a node in \(W\) entered the graph, the size of the graph varied between \(\frac{n}{4}\) and \(n\) nodes. The probability that this node attached to \(s_2\) varied between \(n^{\frac{1}{\alpha} - 1}\) and \((\frac{n}{4})^{\frac{1}{\alpha} - 1} < 4n^{\frac{1}{\alpha} - 1}\). Thus each node in \(W\) is roughly equally likely to attach to \(s_2\) (within a factor of 4).

Now the degree of \(s_2\) in the tree is at least the number of its neighbors in \(W\) that lie at levels \(\geq d + 1\). Using the fact that a constant fraction of the nodes in \(W\) lie at levels \(\geq d + 1\) in the tree, we get that a constant fraction of the neighbors of \(s_2\) lie at levels \(\geq d + 1\) in the tree, in expectation. The result follows from the fact that the degree of \(s_2\) is \(\Omega(n^{\frac{1}{\alpha}})\). ■
Figure 7.3: (a) Simulation support for the analytical model: Figure (a) shows the fraction of shortest path trees that do not contain the edge $e^*$. Figure (b) plots the ratio of degrees of $s_1$ and $s_2$ in a random shortest path tree to their degrees in the graph.

Figure 7.4: Maximum edge congestion: Plotted as a function of $n$, in Inet-3.0 generated graphs, with $\alpha = 1.23$. The figure also plots four other functions to aid comparison – $n^4$, $n^2$, $n^1.8$, $n^1.4$.

7.2.1 Experimental Support

In this section, we report experimental results to show that the theoretical results obtained above hold not just for the Preferential Connectivity Model, but also for Internet-like AS graphs generated by Inet-3.0. Unfortunately, the graphs generated by Inet-3.0, have different values of $\alpha$ for different $n$. This is consistent with the observed properties of the Internet’s AS graph, that $\alpha$ decreases with time. (We discuss this in further detail in the subsequent section). In order to validate our theoretical results and observe the asymptotic behavior of congestion for a fixed value of $\alpha$, we modify the Inet-3.0 code so that it always uses a fixed value of $\alpha = 1.23$, instead of recalculating it for every value of $n$. Each reported value is an average over multiple runs of the simulation, corresponding to different random seeds used for generating the graphs.

Figure 7.3(a) plots the fraction of nodes that are equidistant from $s_1$ and $s_2$. Note that this fraction always lies below 0.4 and is consistent with our result in Lemma 2 that at least a constant fraction of the trees, $\frac{2}{n^2}$ in this case, contain the edge $e^*$. Figure 7.3(b) compares the degrees of the two highest degree nodes in the graph to their corresponding degrees in the shortest path tree.
corresponding to some random node $v$. We find that the ratio of the two degrees for $s_1$ is consistently above 0.9. Similarly, the ratio of the two degrees for $s_2$ is always above 0.8 and increasing. This is consistent with the findings of Lemma 3.

Finally, we plot the maximum congestion in graphs generated by Inet-3.0, as a function of the number of nodes in the graph, in Figure 7.4. Note that the maximum congestion scales roughly as $n^{1.8}$, which is exactly $n^{1+1/\alpha}$ for $\alpha = 1.23$. This corroborates our finding in Theorem 1.

7.3 Simulation Results

In this section, we present the results from our simulation study over Inet-generated graphs. Henceforth, we shall use the graphs generated by Inet 3.0 as is, that is, we do not alter the way Inet chooses $\alpha$ to depend on $n$. In what follows, we first show results for shortest-path routing, followed by policy-based routing. In both cases, we first present results for the any-2-any communication model, then for the leaf-2-leaf model and finally for the clout model.

7.3.1 Shortest-Path Routing

Figure 7.5(a) shows the maximum congestion in power law graphs generated by Inet-3.0 as a function of the number of nodes. We use the any-2-any model of communication here. From the trend in the graph, it is clear that the maximum congestion in Internet-like power law graphs scales as $n^{1+\Omega(1)}$ or worse. Notice also that the slope of the maximum congestion curve is slightly increasing. This can be explained as follows. As mentioned earlier, Inet-3.0 chooses the exponent of the power law degree distribution as a function of the number of nodes $n$: $\alpha = at + b$, where $t = \frac{1}{s} \log \frac{n}{n_0}$, $a = -0.00324$, $b = 1.223$, $s = 0.0281$ and $n_0 = 3037^6$. Notice that the absolute value

\[^6a, b \text{ and } s \text{ are empirically determined constants. } n_0 \text{ is the number of ASes in the Internet in November 1997.}\]
of $\alpha$ decreases as $n$ increases, and so, as our lower bound of $\Omega(n^{1+1/\alpha})$ suggests, the slope of the function on a log-log plot should steadily increase. In fact around $n = 28000$, $\alpha$ becomes less than 1 and at this point we expect the curve to scale roughly as $O(n^2)$, which is the worst possible rate of growth of congestion.

The figure also shows the maximum congestion in power law trees and exponential graphs. The power law trees we generate, have the exponent $\alpha$ between 1.66 and 1.8, the value increasing with the number of nodes in the tree. These exponents are significantly higher than those of the corresponding power law graphs. Notice that the edge congestion on power law trees grows much faster compared to graphs which is expected since trees have much fewer edges. Our lower bound on the maximum congestion, which holds equally well for trees satisfying power law degree distributions, predicts the slope of the curve for trees to be at least 1.5, which is consistent with the above graph.

On the other hand, we notice that edge congestion in exponential graphs is much smaller compared to power law graphs. In fact, edge congestion in exponential graphs has an approximately linear growth. This could be explained intuitively as follows: Recall that for each $n$, we choose the exponent $\beta$ of the exponential distribution so as to match the total number of edges of the corresponding $n$-node power law graph. Because the power law distribution has a heavier tail compared to the exponential distribution, the latter has more edges incident on low degree nodes. Consequently, low degree vertices in an exponential graph are better connected to other low degree vertices. Edges incident on low degree nodes “absorb” a large amount of congestion leading to lower congestion on edges incident on high degree nodes. As $n$ increases the degree distribution becomes more and more even, resulting in a very slow increase in congestion.

In Figure 7.5(b), we show the congestion across all links in a power law graph. Notice that at higher numbers of nodes, the distribution of congestion becomes more and more uneven. The corresponding set of graphs for the leaf-2-leaf communication model is shown in Figure 7.6. The worst congestion is consistently about 0.8 times the worst congestion for the any-2-any model (not explic-
The results for the clout model are more interesting with the resulting maximum congestion in the graph scaling much worse than before. Indeed, as Figure 7.7(a) shows, the maximum congestion scales worse than $n^5$. This is because the total traffic in the graph also grows roughly as $O(n^4)$. Again, as with the any-2-any model, the smaller absolute values of $\alpha$ in the graphs generated by Inet-3.0 for larger values of $n$ is the reason for the increasing slope of the curve. The graph of the congestion across all edges in this model, shown in Figure 7.7(b), is equally interesting. Compared to Figure 7.6(b) of the leaf-2-leaf model, Figure 7.7(b) looks very different: the unevenness in congestion is much more pronounced in the clout model of communication. In other words, the non-uniform traffic demand distribution only seems to exacerbate the already poor congestion scaling of the Internet-like graphs.

### 7.3.2 Policy-Based Routing

Figure 7.8 shows the maximum edge congestion for the three communication models, when policy based routing is used. For the any-2-any and leaf-2-leaf models, shown in Figure 7.8(a), the maximum edge congestion scales almost identically to that for shortest path routing (compared with Figure 7.5(a) and 7.6(a)). However, somewhat surprisingly, for the clout model (Figure 7.8(b)), congestion under policy based routing scales only as $n^3$ compared to over $n^5$ for shortest-path routing.

Figure 7.9(a) compares maximum congestion obtained for policy routing to that for shortest path routing. Notice that the two curves are almost overlapping, although policy routing seems to be slightly worse when the graph is small and gets better as the graph grows larger. This observation can be explained as follows: policy routing disallows certain paths from being used and could, in general, force connections to be routed over longer paths. This would increase the overall traffic in
the network leading to higher congestion, especially for a smaller graph size. However, as the size of the graph grows, there are more and more shortest paths available. As a result, the constraints placed by policy-based routing might not have any significant impact on the path lengths in the graph. In fact, at higher numbers of nodes, policy routing could provide better congestion properties, albeit only marginally different, than shortest path routing. This is because while shortest path routing always picks paths that go over high degree nodes, a fraction of these paths might not be allowed by policy routing as they could involve more than one peering edge. In this case, policy routing moves traffic away from the hot-spots, thereby, partially alleviating the problem. We believe that this is also the reason for the congestion scaling in the clout model to be better when considering policy-routing, as opposed to shortest-path routing.

In order to verify that the above observation is not just an artifact of our machine learning-based labeling algorithms, we plot the same curves for ALRs in Figure 7.9(b). These display exactly the same trend—policy routing starts out being worse than shortest path, but gets marginally better as $n$ increases. To summarize, policy routing does not worsen the congestion in Internet-like graphs, contrary to what common intuition might suggest. In fact, policy routing might perform marginally
better than shortest path routing.

7.3.3 Shortest Path Routing Variations

As mentioned in Section 7.1.1, in the shortest path routing scheme, whenever there are multiple shortest paths between two nodes, we pick the path that contains higher degree nodes to route the flow between them. It may appear that the poor congestion properties of power law graphs are a result of this tie breaking rule, and an alternate rule that favors low degree nodes may perform better by alleviating the congestion on high degree nodes. In order to confirm that our results are robust across various tie-breaking rules, we performed the experiments with two variants of the tie-breaking rule: favoring paths that contain lower degree nodes, and choosing a random shortest path when there is a choice of more than one.

For these experiments, we set $\alpha$ to be a constant value of 1.23 in Inet 3.0 and compare the resulting relations between maximum edge congestion and the number of nodes. As Figure 7.10 depicts, there is no noticeable difference between the three types of tie-breaking methods. The same holds true for leaf-2-leaf and clout models of traffic (results are omitted for brevity). This is because very few vertex pairs have multiple shortest paths between them. We thus conclude that our scheme of breaking ties by favoring paths containing higher degree nodes does not skew our results.

7.4 Improving the Congestion Scaling Properties

Our analytical and simulation results have shown that the maximum congestion in Internet-like power law graphs scales rather poorly in the graph size—$\Omega(n^{1+\Omega(1)})$. Our results show that edges between high degree nodes, which are typically peering edges between backbone carriers in the Internet core, are likely to get congested more quickly over time than other edges. In such a situation, to enhance the scaling properties of the network, it might become necessary to either change the routing algorithm employed by ASes in the Internet (i.e., BGP-style routing) or alter the intercon-
Figure 7.11: Degree vs congestion: Edge Congestion versus the average degree of the nodes incident on the edge (any-2-any model with shortest path routing). The congestion is higher on edges with a high average degree.

connection. Next, we address the latter issue of altering the structure of the Internet graph. Specifically, we focus on mechanisms for increasing the parallelism in edges between neighboring nodes in the Internet AS-graph.

7.4.1 Adding Parallel Network Links

We examine ways in which additional links can be placed in the network, so as to contain the effect of bad scaling of the maximum congestion. Specifically, we consider the model in which each link can be replaced by multiple links (between the same pair of nodes) that can share the traffic load. Ideally, we would like to provide sufficient parallel links between a pair of nodes, so that the total congestion on the corresponding edge divided equally among these parallel links, even in the worst case, grows at about the same rate as the size of the network. The number of parallel links between a pair of nodes may need to change as the network grows to achieve this goal. Notice that this change does alter the degree-structure of the graph, but the alteration is only due to increased connectivity between already adjacent nodes. In other words, this does not require new edges between nodes that were not adjacent before.

In fact, the network, at an AS level, already incorporates this concept of parallel links. For example, the power law structure of the AS graph only considers the adjacency of ASes: the link between Sprint and AT&T, for instance, is modeled by a single edge. However, in the real world the Sprint and AT&T ASes are connected to each other in a large number of places around the world. However, not much is known about the degree of such connectivity in the Internet today.

In order to guide the addition of parallel edges between adjacent nodes, we first observe that there is clear correlation between the average degree and edge congestion. Figure 7.11 plots the congestion of each edge against the average degree of the nodes on which it is incident. We show the results for shortest path routing on an Inet generated graph of 30000 nodes where any-2-any

7For results on alternate methods of alleviating congestion, please refer to [6].
8Note that the routing is still done based on the original degrees of nodes.
communication is used. The figure shows that edges incident on high degree nodes have much higher congestion than those incident on lower degree nodes. This suggests that a good choice for the number of parallel links substituting any edge in the graph, could depend on the degrees of nodes which an edge connects.

We examine several ways of adding parallel links based on the above observation. In particular, we let the number of links between two nodes be some function of the degrees of the two nodes and we consider the following functions: (1) sum of degrees of the two nodes, (2) product of the degrees of the two nodes, (3) maximum of the two degrees and, (4) minimum of the two degrees. We then compute the maximum relative congestion for these functions, that is, the maximum over all edges, of the congestion on the edge divided by the number of parallel links corresponding to each edge. The maximum congestion on Internet graphs for these models of adding new edges is shown in Figure 7.12. Notice that, surprisingly, when parallel links are added according to any of the above four functions the maximum relative congestion in the graph scales linearly. This implies that adding parallelism in the edges of Internet-like graphs according to the above simple functions is enough to ensure that uniform scaling of link capacities (for example, based on Moore’s Law-like technology trends) can maintain uniform levels of congestion in the network and avoid persistent hot-spots.

7.5 On Networking Modeling and Congestion Scaling

There have been several theoretical studies aimed at studying the properties of large-scale, Internet-like graphs, as well as those analyzing congestion scaling properties of general graphs. In this section, we present a brief overview of some of these studies.

Of studies aiming to characterize properties of Internet-like graphs, one class has proposed various models of graph evolution that result in a power law degree distribution. Notable examples include the power law random graph model of Aiello et al. [1], the bi-criteria optimization model of Fabrikant et al. [31] and the Preferential Connectivity model of Barabasi and Albert [18, 11].
Another class of studies in this category [32, 73, 94] is aimed at analyzing the properties of power law graphs. However, most of these are based on inferences drawn from measurements of real data. The primary application of this latter class of studies is to construct realistic generators [60, 100, 94] for Internet-like graphs.

The problem of characterizing congestion in graphs, and specifically designing routing schemes that minimize congestion, has been studied widely in approximation and online algorithms. The worst congestion in a graph is inversely related to the maximum concurrent flow that can be achieved in the graph while obeying unit edge capacities. The latter is, in turn, related to a quantity called the cut ratio of the graph. Aumann et. al. [16] characterize the relationship between maximum concurrent flow and cut ratio: The maximum concurrent flow that can be achieved in a graph is always within a factor of $O(\log n)$ of the cut ratio, where $n$ is the number of nodes. Okamura et. al. [69] give bounds on the cut ratio for special graphs. Algorithmic approaches to the problem (see [53, 54] for a survey) use a multi-commodity flow relaxation of the problem to find a fractional routing with good congestion properties (wherein demand between pairs of nodes is split across multiple paths). Although fairly good approximation factors have been achieved for the problem, most of the proposed routing schemes are not distributed, involve a lot of book-keeping, or require solving large linear programs, which makes them impractical for wide-area Internet routing.

Perhaps the work that shares similar goals as ours is that of Gksanditis et al. [39]. Using arguments from max-flow min-cut theory, their paper shows that graphs obeying power law degree distribution have good expansion properties in that, they allow routing with $O(n \log^2 n)$ congestion, which is close to the optimal value of $O(n \log n)$ achieved by regular expanders. In a follow-up paper, Mihail et al. [61] prove similar results on the expansion properties of graphs generated using the Preferential Connectivity model. The results presented in this chapter, in contrast with these two contemporary papers, focus specifically on commonly-used routing algorithms such as policy routing and shortest path routing. We show that power law graphs exhibit poor scaling properties with respect to these routing algorithms.

7.6 Analysis Caveats, Summary of Observation and their Implications

In this chapter, we addressed the question of how the worst congestion in Internet-like graphs (specifically at the AS-level) scales with the graph size. The key observations are shown in Table 7.1. Using a combination of analytical arguments and simulation experiments, we showed that maximum congestion scales poorly in Internet-like power law graphs. Our simulation results showed that the non-uniform demand distribution between nodes only exacerbates the congestion scaling. However, we found, surprisingly, that policy routing between adjacent ASes may not worsen the congestion scaling on power law graphs and might, in fact, be marginally better when compared to shortest-path routing.

We note that with the current trend of the growth of the Internet it is possible that some locations
The expected maximum congestion in Internet-like Preferential Attachment power law graphs with unit traffic demands and shortest path routing scales as $n^{1+\Omega(1)}$, where $n$ is the number of nodes in the graph.

When non-uniform traffic matrices are considered, the congestion scaling properties worsen significantly.

Policy routing results in similar, if not marginally better, congestion than shortest path-based routing.

The poor congestion scaling properties of the Internet graph can be fixed using very simple heuristics to alter the topology of the network. For example, adding parallel edges between adjacent nodes in proportion to the minimum of their degrees can result in a linear scaling of congestion.

<table>
<thead>
<tr>
<th>Table 7.1: Congestion Scaling in the Internet: Summary of key observations regarding the scaling properties of the Internet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>in the network might eventually become perpetual hot-spots. Fortunately, however, there is an intuitively simple fix to this problem. Adding parallel links between adjacent nodes (ASes) in the graph according to simple functions of their degrees will help the maximum congestion in the graph scale linearly. In this case, it might not be necessary for the capacity of some links in the graph to grow at a faster rate than the others. In fact, a natural evolution of link capacities according to Moore’s Law may be sufficient to accommodate the growing traffic demand in the network.</td>
</tr>
</tbody>
</table>

Next, we discuss important caveats in our analysis and simulations.

**Analysis Caveats.** We would like to mention that the results presented in this chapter may not hold in general for all power law graphs. Our results (both simulation-based and analytical) are meant for graphs representing Internet connectivity at the AS level. These results, may not apply to power law random graphs [1]. Note also that while the preferential connectivity model is known to yield graphs with a similar degree distribution as the AS-level graph, it is not clear whether the model accurately captures the AS-level connectivity dynamics (e.g., economic considerations for peering). That said, our simulations on measured AS-level graphs show that our key observations hold for the
existing AS graph. Therefore, if the current dynamics of connectivity between ASes continues to hold in the future, we can expect our results to hold for future AS-level graphs too.

Our analysis does not extend to router-level graphs. Analyzing the router-level topology is much harder compared to the AS-level graph due to three reasons: (1) Not much is known about the topology of Internet’s router-level graph. Most existing maps of the Internet’s router-level topology (such as Rocketfuel maps [90, 23]) are considered incomplete; (2) IP-level routing cannot be modeled easily using shortest path routing or simple inter-domain policy-based routing, since this would require knowledge of traffic engineering employed by ASes in the Internet; (3) Finally, some researchers have used power law graphs resulting from probabilistic models (such as power law random graph models [1]) to approximate the router-level connectivity (see for example [94]). However, recent work has shown that such models are error-prone since they do not explicitly consider the technological and economic constraints or trade-offs behind router interconnections [55]. Graphs arising from such trade-offs are referred to as **Heuristically Optimal Topologies**. However, there are no analytically-tractable models for generating such topologies. A thorough analysis of the router-level interconnection is a challenging open problem.

The key results from our study of congestion scaling in the Internet graph may be simply summarized as follows: The congestion along edges in the Internet graph is likely to scale poorly with the growing size of the network. As a result, end-networks cannot employ routing-based mechanisms such as route control to extract good performance from the future network. However, simple heuristics for adding parallel edges between adjacent vertices in the graph can help significantly improve the congestion scaling properties.