Overlay Networks with Linear Capacity Constraints

Ying Zhu, Baochun Li [⋆]

Department of Electrical and Computer Engineering, University of Toronto

Abstract. Previous work have assumed an independent model for overlay networks: a graph with independent link capacities. We introduce a model of overlays (LCC-overlay) which incorporates correlated link capacities by formulating shared bottlenecks as linear capacity constraints. We define metrics to measure overlay quality. We show that LCC-overlay is perfectly accurate and hence enjoys much better quality than the inaccurate independent overlay. We discover that even the restricted node-based LCC yields significantly better quality. We study two problems in the context of LCC-graphs: widest-path and maximum-flow. We also outline a distributed algorithm to efficiently construct an LCC-overlay.

1 Introduction

The proliferation of research on overlay networks stems from their versatility, ease of deployment, and applicability in useful network services such as application-layer multicast [1, 2], media streaming and content distribution [3]. Previous studies have uniformly taken the view of an overlay network as merely a weighted network graph; the nodes are end systems, the links are unicast connections, and the links are weighted by unicast delay and bandwidth. Overlay networks are therefore treated exactly as a flat single-level network, in which the overlay links are independent. In particular, link capacities are independent of each other. This model is inaccurate as the overlay network encompasses two levels: a virtual network of end systems residing on top of an underlying IP network. An overlay link maps to a path, determined by the routing protocols, in the underlying network. When two or more overlay links map to paths that share an underlying link, the sum of the capacities of the overlay links are constrained by the capacity of the shared link, i.e., these overlay links are *correlated* in capacity. This obvious but crucial observation leads us to conclude that an accurate model of overlay networks must include *link correlations*.

In this paper, we propose the model of overlay network with linear capacity constraints (LCC). An LCC-overlay is a network graph in which the capacities of overlay links are represented by variables and link correlations are formulated as linear constraints of link capacities (i.e., LCC). The LCC-overlay model is a succinct way to accurately represent the true network topology with all its link correlations, requiring only the addition of a set of linear capacity constraints to the simple overlay graph.

We address the following questions. How do we qualitatively measure the quality of an overlay? Why do we prefer LCC-overlays instead of a simple network graph with independent links? Our analysis and simulations reveal the necessity of LCC-overlay in assuring the quality of overlay networks and we introduce two qualitative metrics — accuracy and efficiency — to measure overlay quality. We also study a

[⋆] The authors' e-mail addresses are {*yz, bli*}*@eecg.toronto.edu*.

restricted class of LCC, node-based LCC, that is more efficient and of a distributed nature. Surprisingly, we find that even with such restricted and incomplete LCC, the accuracy and efficiency are much better than overlays with no LCC, and they are close to overlays with complete LCC. We propose a distributed algorithm for constructing an LCC-overlay based on node-based LCC. We further study two network flow problems, widest-path (i.e., maximum-bandwidth single-path unicast) and maximum-flow (i.e., maximum-bandwidth multiple-path unicast), with the addition of LCC. Traditional algorithms cannot be used to solve them in a network graph with LCC. We show that widest-path with LCC is NP-complete. We formulate the problem of maximum-flow with LCC as a linear program and propose an efficient algorithm for solving it.

The remainder of the paper is organized as follows. Sec. 2 will introduce the concept of overlays with LCC; provide formal definitions of the LCC-overlay and the quality metrics; and show the necessity of LCC-overlay in ensuring high overlay quality, through analysis and simulations. In Sec. 3, we present the problem of widest-path with LCC and show that it is NP-complete. In Sec. 4, the problem of maximum-flow with LCC is presented and formulated using linear programming; an efficient algorithm for solving it is proposed. Then, in Sec. 5, we outline an algorithm for constructing an LCC-overlay. Sec. 6 describes the related work and Sec. 7 concludes the paper.

2 Overlay with linear capacity constraints

In this section, we will define an overlay with linear capacity constraints (LCC), and two metrics for measuring overlay quality — accuracy and efficiency. We will moreover demonstrate through analysis and simulation that LCC are necessary for ensuring high quality of overlay networks.

As a result of the two-level hierarchical structure, overlay links are virtual links that correspond to paths in the lower-level network. We define *link correlation* as follows: Overlay links are correlated if they map to underlying paths that share one or more physical links. Link correlation is a fundamental property of overlay networks. Yet, in the current prevailing independent overlay model of a graph in which each link is weighted by its unicast capacity, the underlying assumption is that overlay links have independent capacities. Suppose two overlay links both map to a bottleneck physical link of capacity c, then each has the unicast bandwidth c; however, when data flows on these overlay links simultaneously, each has a capacity of only $c/2$. Thus, the independent overlay may be egregiously inaccurate in representing the network in reality.

We propose an overlay model, *LCC-overlay*, that accurately represents the real network topology, by using linear capacity constraints to succinctly formulate link correlations. Essentially, it is a regular overlay graph, but the link capacities are variables, and a set of LCC express the constraints imposed by shared bottlenecks. The formal definition will be presented in Sec. 2.2.

2.1 Worst-case analysis of overlays with no LCC

For the purpose of illustration, we examine a simple example of a two-level network, as seen in Fig. 1(a). The mapping of overlay links to physical paths is the obvious one in the graph. We adopt a simplified overlay construction algorithm, denoted by *OC*, that is nevertheless representative of such algorithms proposed in previous work. In *OC*, every node selects d neighbors to which it has links with the highest bandwidth.[1] With $d = 3$, the overlay graph for our example network is shown in Fig. 1(b); it is not hard to see that the results we reach below hold for $d = 2, 1$. The highest-bandwidth multicast tree for this overlay graph, denoted T_{OC}, is given in Fig. 1(c). Although the *predicted* bandwidth of T_{OC} in the overlay is 3, the actual *achievable* bandwidth of T_{OC} is only 1 because all three tree links share the physical link (r_2, r_3) of capacity 3.

In contrast, under the LCC-overlay model, capacities of overlay links are variables and link correlations are captured by *linear capacity constraints*. For instance, the four links $(A, C), (A, D), (B, C), (B, D)$ are correlated, hence the sum of their capacities is constrained by the capacity of shared physical link (r_2, r_3), i.e., $x_{AC} + x_{AD} + x_{BC} + x_{BD} \leq c(r_2, r_3)$. The linear capacity constraints for the overlay graph in Fig. 1(b) are given below in matrix form:

$$\begin{pmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,1\,1\,0 \\ 0\,0\,0\,0\,0\,1 \end{pmatrix} \begin{pmatrix} x_{AB} \\ x_{AC} \\ x_{AD} \\ x_{BC} \\ x_{BD} \\ x_{CD} \end{pmatrix} \leq \begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix} \qquad (1)$$

The overlay graph together with the linear capacity constraints (LCC) form an LCC-overlay. For the LCC-overlay in our example, the highest-bandwidth multicast tree is shown in Fig. 1(d), obtained by a greedy algorithm that is a variation of the one for regular graphs, modified to take LCC into consideration. In this case, the predicted tree bandwidth is equal to the achievable bandwidth; both are 2.

Taking a cue from the above simple example, we arrive at the following.

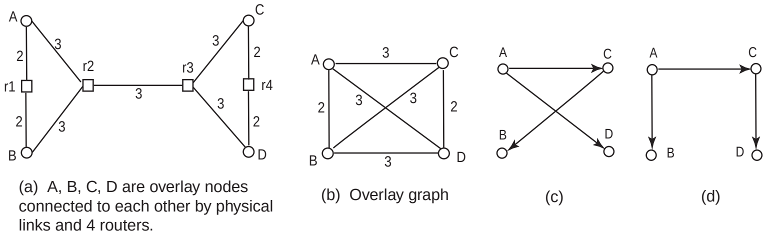
Proposition: For any fixed number of overlay nodes n, there exists a lower-level network G such that the bandwidth of an optimal multicast tree in any overlay graph constructed by *OC* residing over G is asymptotically $1/(n-1)$ of the bandwidth of an optimal multicast tree obtained in the LCC-overlay.

Proof: Consider a generalized graph $G = (R \cup S, E)$ of the one in Fig. 1(a), with n overlay nodes, shown in Fig. 2(a). Any overlay graph constructed by *OC* will contain the middle $(\beta + \epsilon)$-link for every overlay link between the partitions, see Fig. 2(b). An optimal multicast tree in the *OC* graph must include only the $(\beta + \epsilon)$-links, because otherwise its predicted bandwidth would be suboptimal. However, its achievable bandwidth is only $(\beta + \epsilon)/(n-1)$ since all $n - 1$ tree links traverse the same $(\beta + \epsilon)$-link in the middle. In the LCC-overlay, the optimal tree has bandwidth β, as shown in Fig. 2(c). With ϵ approaching 0, the *OC* tree asymptotically achieves $1/(n-1)$ of β. □

2.2 Formal definitions of LCC-overlay and quality of overlay

From the above analysis, we observe that the extreme poor performance of the overlay with no LCC (No-LCC overlay) is a consequence of its *inaccuracy* in representing the true network topology. The LCC-overlay, on the other hand, represents the network with perfect accuracy, and hence achieves the optimal bandwidth. Two questions now arise naturally. (1) How do we quantitatively measure the quality of overlay networks? (2) How does the quality (i.e., accuracy, performance) of LCC-overlays compare with that of No-LCC overlays in realistic networks?

[1] Though fictitious, this is only a slightly simpler variation of the neighbor selection rule in [4].



(a) A, B, C, D are overlay nodes connected to each other by physical links and 4 routers.

(b) Overlay graph

(c)

(d)

Fig. 1. A simple example of the detrimental effect that the independent model of overlay has on the overlay quality.

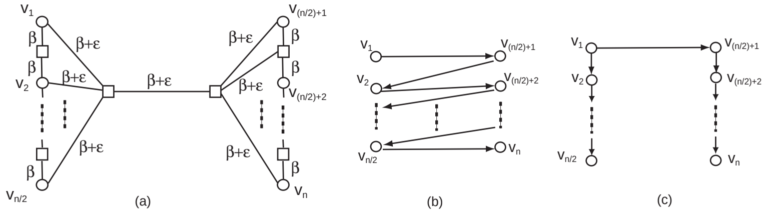


Fig. 2. A worst-case example of the poor quality of an overlay with no LCC.

Before we directly address these questions, we must first formally define the LCC-overlay and the metrics to measure overlay quality. We also will make more precise the notions of predicted and achievable bandwidth of overlay flows.

The two-level hierarchy of an overlay network can be formulated as consisting of: a low-level (IP) graph $G = (V, E)$, each link $e \in E$ has a capacity of $c(e) \geq 0$; a high-level (overlay) graph $\widehat{G} = (\widehat{V}, \widehat{E})$, where $\widehat{V} \subset V$; a mapping P of every overlay edge $(\widehat{v}_1, \widehat{v}_2) \in \widehat{E}$ to a low-level path $P(\widehat{v}_1, \widehat{v}_2) \subset G$ from $\widehat{v}_1$ to $\widehat{v}_2$.

The formulation of capacity constraints in the overlay graph $\widehat{G}$ is where LCC-overlay departs from No-LCC overlay. The No-LCC overlay is a pair $(\widehat{G}, \widehat{c})$, where $\widehat{c}$ is a capacity function such that each link $\widehat{e} \in \widehat{E}$ has a capacity $\widehat{c}(\widehat{e}) \geq 0$. The LCC-overlay is defined as follows.

Definition 1 (LCC-overlay): The *LCC-overlay* is a triplet $(\widehat{G}, C, b)$, where the capacity of each link $\widehat{e}$ in $\widehat{G}$ is a variable $x_{\widehat{e}}$; and (C, b) represent a set of m linear capacity constraints $Cx \leq b$: C is a 0-1 coefficient matrix of size $m \times |\widehat{E}|$, x is the $|\widehat{E}| \times 1$ vector of link capacity variables, $b \in \mathrm{R}^m$ is the capacity vector. Each row i in (C, b) is a constraint of the form $\sum_{\widehat{e}:C(i,\widehat{e})=1} x_{\widehat{e}} \leq b(i)$.

A flow f from s to t in $\widehat{G}$, is an assignment of bandwidth to every link in $\widehat{E}$ subject to capacity constraints and flow conservation; the flow rate, $|f|$, is the total outgoing bandwidth of s. We denote the *achievable flow* of $f \subset \widehat{G}$ in the low-level G by $\sigma_G(f)$ and the *achievable bandwidth* of f by $|\sigma_G(f)|$. We now describe the procedure for obtaining these.

Let f be a flow from node A to node C in the No-LCC overlay shown in Fig. 1(b), with $f(A, C) = 3, f(A, B) = 2, f(B, C) = 3$, hence $|f| = 3$. The low-level graph $G = (V, E)$ is shown in Fig. 1(a). Suppose low-level link (r_1, r_2) is in $P(A, C) \cap P(B, C)$, then the true capacity of overlay links (A, C) and (B, C) in f is a fair share of the bottleneck capacity, denoted by $\gamma_f(A, C) = \gamma_f(B, C) = c(r_1, r_2)/2$. For link (A, B), $P(A, B) = \{(A, r_1), (r_1, B)\}$, thus $\gamma_f(A, B) = f(A, B)$. Using the true capacities of these three links with respect to f, a maximum flow from A to C can be obtained. This is the achievable flow of f, $\sigma_G(f)$, in which a flow of 1.5 is assigned to all three links, and $|\sigma_G(f)| = 1.5$ is the achievable bandwidth of f.

In general, given G and a flow $f \subset \widehat{G}$, the procedure of determining $\sigma_G(f)$ is shown in Fig. 3.

```
for each $e \in E$
   use max-min fairness to allocate $c(e)$ among $\{\widehat{e} : e \in P(\widehat{e})$ and $f(\widehat{e}) > 0\}$,
   let each allocation be denoted by $\gamma_f^e(\widehat{e})$
   for each $\widehat{e} \in \widehat{E}$
     if $f(\widehat{e}) > 0$     $\gamma_f(\widehat{e}) \leftarrow \min\{\gamma_f^e(\widehat{e}) : e \in P(\widehat{e})\}$
     else            $\gamma_f(\widehat{e}) \leftarrow 0$
   $\sigma_G(f) \leftarrow$ maximum-flow in $(\widehat{G}, \gamma_f)$,    $|\sigma_G(f)| \leftarrow$ bandwidth of $\sigma_G(f)$
```

Fig. 3. The procedure of determining $\sigma_G(f)$.

We introduce two metrics for measuring overlay quality: *accuracy* and *efficiency*. With respect to a maximum flow f in the overlay, accuracy is the predicted flow rate over its achievable bandwidth; it measures the degree to which the overlay over-estimates a maximum flow. Efficiency is the achievable bandwidth of f divided by the low-level maximum flow bandwidth; it measures how good an overlay maximum flow performs in comparison with the low-level optimum (which cannot be attained in overlays). The formal definitions are as follows.

Definition 2 (Accuracy): Accuracy of a maximum-flow f in overlay network $\widehat{G}$ residing over G, is $\alpha^f_{\widehat{G}} = |$ maximum-flow $f \subset \widehat{G}$ $|$ / $|\sigma_G(f)|$.

Definition 3 (Efficiency): Efficiency of a maximum-flow f in overlay network $\widehat{G}$ residing over G, is $\varepsilon^f_{\widehat{G}} = |\sigma_G(f)|$ / $|$ maximum-flow $\bar{f} \subset G$ $|$.

The overall accuracy and efficiency of an overlay are better measured by taking the average of accuracy and efficiency over all possible maximum-flows.

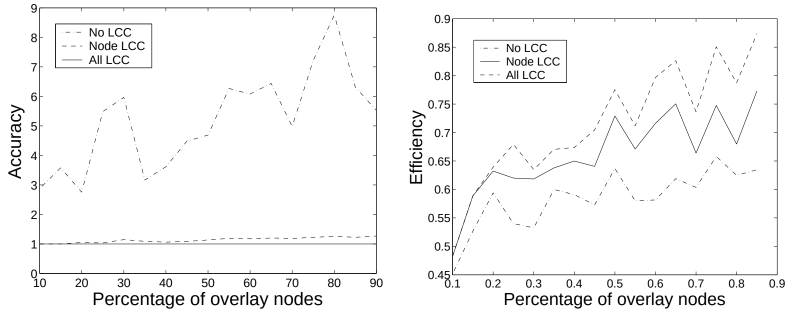
Definition 4 (Accuracy and Efficiency of Overlay): Accuracy of an overlay $\widehat{G}$ is the *mean* of $\{\alpha^f_{\widehat{G}} :$ s-t maximum-flow $f, \forall s, t\}$. Efficiency of an overlay $\widehat{G}$ is the *mean* of $\{\varepsilon^f_{\widehat{G}} :$ s-t maximum-flow $f, \forall s, t\}$.

2.3 Comparing the quality of No-LCC overlay and LCC-overlay in realistic Internet-like topologies

In practical terms, to discover a complete set of LCC incurs high cost, and also requires centralized operations. Motivated by this, we consider a restricted class of LCC that is naturally distributed: *node-based LCC*. A node-based LCC contains only capacity variables of links that are adjacent to a single node. Therefore, we simulate three types of overlays: No-LCC, All-LCC, and Node-LCC. Through simulations with realistic network topologies, we compare the quality of all three types of overlays, using the accuracy and efficiency metrics defined above. We use an Internet topology generator, BRITE [5], which is based on power-law degree distributions. [2]

First, we compare the accuracy and efficiency of the three overlays with various overlay sizes relative to the low-level network size. We fix the number of low-level

[2] A seminal paper [6] revealed that degree distribution in the Internet is a power-law.

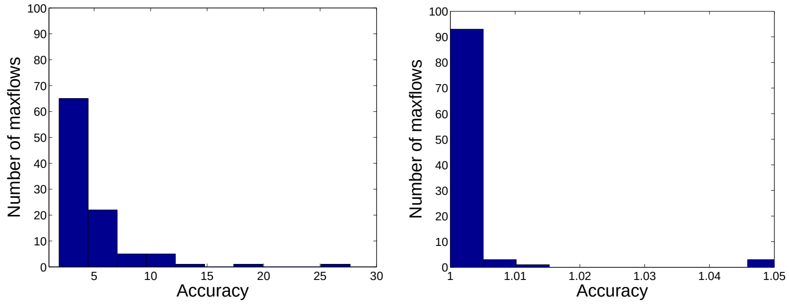


(a) Accuracy of No-, All- and Node-LCC overlays, versus % overlay nodes.

(b) Efficiency of No-, All- and Node-LCC overlays, versus % overlay nodes.

Fig. 4. Overlay quality versus ratio of overlay size to low-level size.

nodes to 100 and vary the number of overlay nodes from 10 to 90; the data are averaged over numerous maximum flows with randomly selected source and destination nodes. In Figure 4(a), accuracy is plotted against ratio of overlay over low-level size. The All-LCC overlay always achieves its predicted maximum flows (accuracy of 1) because it has all the bottleneck information. As the number of overlay nodes increases, the accuracy of Node-LCC only deviates negligibly from 1. No-LCC fares much worse, with much higher values for the accuracy metric, which indicate that it over-estimates in predicting maximum flow values and the achievable bandwidths are substantially lower than predicted.

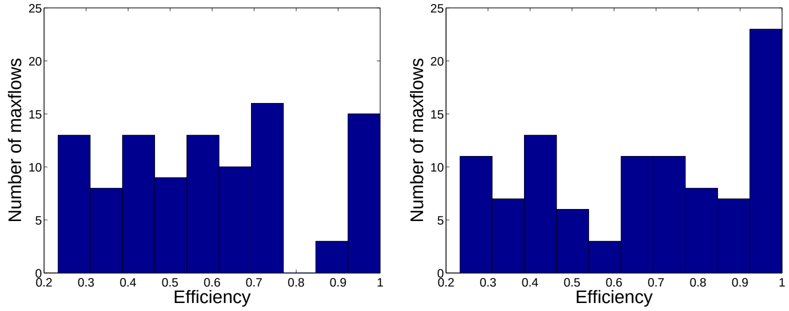


(a) Accuracy distribution for No-LCC, for fixed % overlay nodes.

(b) Accuracy distribution for Node-LCC, for fixed % overlay nodes.

Fig. 5. Accuracy distributions for No-LCC (a) and Node-LCC (b), with the fixed ratio 30% of overlay to low-level size.

Figure 4(b) shows efficiency versus overlay-to-low-level ratio for the three overlays. All-LCC has the highest efficiency, as expected, since it has the optimal overlay efficiency, i.e., higher efficiency cannot be achieved by only using overlay links. The surprise here is how closely the Node-LCC efficiency curve follows that of All-LCC

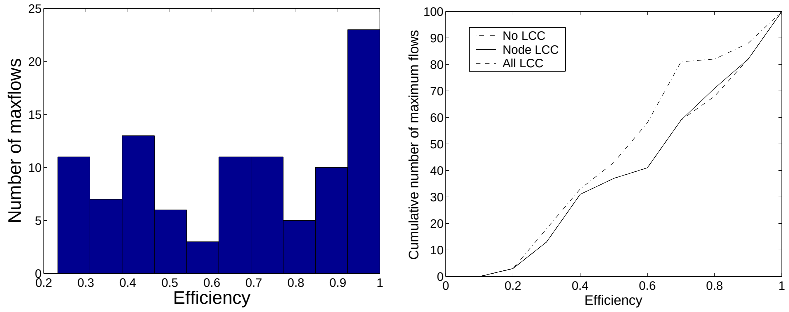


(a) Efficiency distribution for No-LCC, for fixed % overlay nodes.

(b) Efficiency distribution for Node-LCC, for fixed % overlay nodes.

Fig. 6. Efficiency distributions of No-LCC (a) and Node-LCC (b) for fixed ratio 30% of overlay to low-level size.

for all realistic overlay ratios (less than 65%). No-LCC has much lower efficiency than both All-LCC and Node-LCC. It should be noted that No-LCC efficiency is not as poor as its accuracy, relatively to the two LCC. This can be explained by the fact that No-LCC heavily over-estimates (indicated by its poor accuracy) link capacities, and thus overloads low-level links to their full capacity and thereby benefiting the efficiency. But overloading some low-level links results in other links being under-utilized, because it was not foreseen that they were needed. This is why No-LCC is still significantly less efficient than Node-LCC.



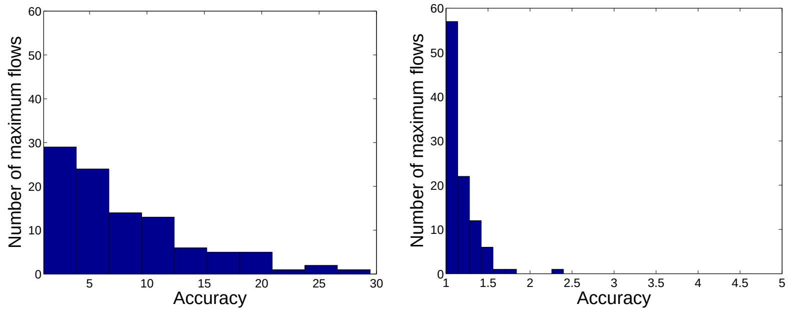
(a) Efficiency distribution for All-LCC, for fixed % overlay nodes.

(b) Cumulative distribution of efficiency for No-, All- and Node-LCC.

Fig. 7. Efficiency distribution of All-LCC (a) and the cumulative distributions for the three overlays (b), for the fixed ratio 30% of overlay to low-level size.

Next, we evaluate the accuracy and efficiency of maximum flows with a fixed overlay-to-low-level ratio of 30%. The distributions of accuracy over 100 maximum flows for No-LCC and Node-LCC are given in Fig. 5(a) and (b), respectively. As above, effectively all Node-LCC maximum flows have perfect accuracy, while No-LCC is remarkably inaccurate.

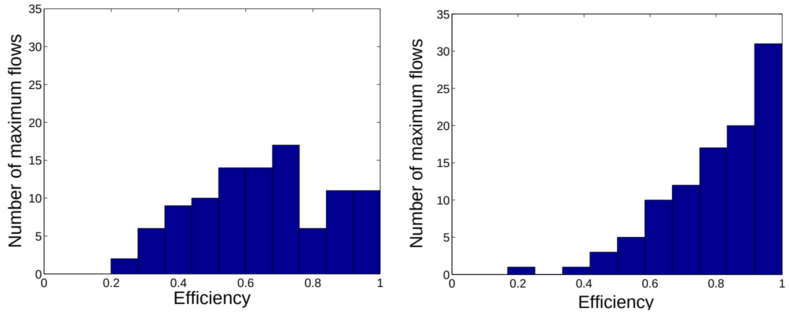
The distributions of efficiency are more interesting. In No-LCC, shown in Fig. 6(a), only a small fraction of maximum flows are efficient. It is quite different for All-LCC, seen in Fig. 7(a), where a majority of maximum flows have high efficiency. The Node-LCC distribution in Fig. 6(b) looks almost the same as All-LCC. The coinciding of Node-LCC efficiency with All-LCC efficiency is confirmed in their cumulative distributions in Fig. 7(b), where the two curves are almost the same.



(a) Accuracy distribution for No-LCC, for network size 500.

(b) Accuracy distribution for Node-LCC, for network size 500.

Fig. 8. Accuracy distributions for No-LCC (a) and Node-LCC (b), for network size 500.



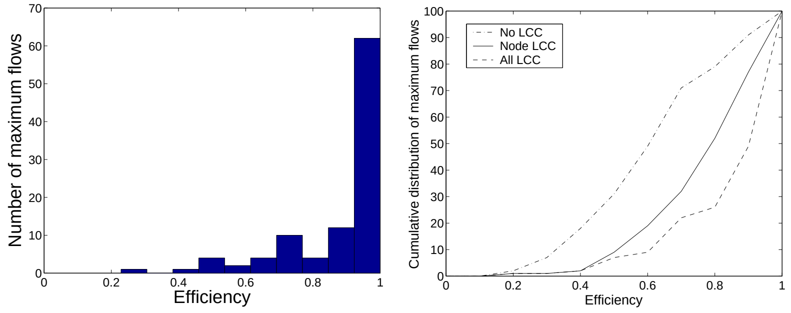
(a) Efficiency distribution for No-LCC, for network size 500.

(b) Efficiency distribution for Node-LCC, for network size 500.

Fig. 9. Efficiency distributions for No-LCC (a) and Node-LCC (b) for network size 500.

We examine the impact of larger network sizes on accuracy and efficiency, by increasing the network size to 500 nodes and keeping the percentage of overlay nodes at 30%. Figure 8 shows the accuracy distributions of No-LCC and Node-LCC. No-LCC accuracy is much worse than for the previous smaller network size. However, the increased network size causes only a tiny change in Node-LCC accuracy, which is still almost perfect. The efficiency distribution for All-LCC, given in Fig. 10(a), shows extremely high efficiency for almost all the maximum flows sampled. All-LCC efficiency has significantly improved for increased network size. The reason, we conjecture, is that the low-level maximum flows have to travel longer paths in the larger network, thus they are more similar to the paths that overlay flows map to, which means that both overlay and low-level maximum flows encounter much of the same bottlenecks. The same reasoning explains the improved efficiency for Node-LCC in this larger network; Fig. 9(b) shows its efficiency distribution. As can be seen in Fig. 9(a), No-LCC efficiency is more inferior compared to Node-LCC than in the smaller network.

The cumulative distribution graph in Fig. 10(b) illustrates that the gap in efficiency between Node-LCC and All-LCC is smaller than the gap between Node-LCC and No-LCC. In Node-LCC, most of the maximum flows have high efficiency. Moreover, Node-LCC is (like All-LCC) more efficient for the larger network size than for the smaller one. We conclude that increasing network size causes significant deterioration in No-LCC quality, but actually improves significantly the quality of All-LCC and Node-LCC.



(a) Efficiency distribution for All-LCC, for network size 500.

(b) Cumulative distribution of efficiency for No-, All- and Node-LCC, for network size 500.

Fig. 10. Efficiency distribution for All-LCC (a) and cumulative distributions for the three overlays (b), for network size 500.

3 Widest-Path with LCC is NP-complete

The LCC-overlay is an entirely different type of network graph than traditional network graphs. Existing algorithms for network flow problems may not work in the LCC-graph. In this section, we consider the problem of widest-path with LCC, i.e., finding a highest-bandwidth path from source to destination. Widest-path can be solved by a variation of Dijkstra's shortest-path algorithm, however, this algorithm does not in general find a widest path in an LCC-graph.

We are given an LCC-graph $\{G = (V, E), C, b\}$, as defined above in Sec. 2.2. The width of a path $p = \langle e_1, e_2, \ldots, e_k \rangle \subset G$, $w(p)$, is defined as: maximize x_{e_1} subject to $x_{e_j} = 0, \forall e_j \notin p, Cx \leq b$, and $x_{e_1} = x_{e_2} = \ldots = x_{e_k}$. This can be computed by assigning 1 to $x_{e_i}, \forall e_i \in p$, and 0 to the remaining variables; and obtain $\min\{b_j/x_j : j \text{ s.t. } x_j = 1\}$. We define Widest-Path with Linear Capacity Constraints (WPC) as a decision problem: INSTANCE: An LCC-graph (G, C, b), where

$G = (V, E)$ and (C, b) are a set of LCC, specified s and t, a positive integer $K \leq \max\{b_i\}$. QUESTION: Is there a directed path p from s to t whose width is no less than K?

Theorem: WPC is NP-complete.

Proof: WPC is in NP because a nondeterministic algorithm need only guess a subset of E and check in polynomial time whether these edges form a path p with $w(p) \geq K$.

We transform the Path with Forbidden Pairs (PFP) [7] to WPC. The PFP problem is defined as follows. INSTANCE: Directed graph $G = (V, E)$, specified vertices $s, t \in V$, collection $F = \{(a_1, b_1), \ldots, (a_n, b_n)\}$ of pairs of vertices from V. QUESTION: Is there a directed path from s to t in G that contains at most one vertex from each pair in F?

Let G, s, t, F be any instance of PFP. We must construct a graph $G' = (V', E')$, $s, t \in V'$, a set of linear capacity constraints $Cx \leq b$ for edges in E', and an positive integer $K \leq \max_i\{b_i\}$ such that G' has a directed path from s to t of width no less than K if and only if there exists a directed path from s to t in G that contains at most one vertex from each pair in F.

Any vertex $v \in V$ not in F and any edge $e \in E$ not incident to a vertex in F remain unchanged in V' and E', respectively. For every vertex u in F, we replace it with vertices u', u'' and a directed edge e_u from u' to u'', called u's replacement edge. For every edge $e = (v, u) \in E$ that enters u, an edge $e' = (v, u'$ is added to E'; similarly, for every edge $e = (u, v) \in E$ that exits u, we add $e' = (u'', v)$. Now we form the linear capacity constraints. Each non-replacement edge $e \in E'$ gives rise to a one-variable constraint $x_e \leq 1$. For each pair $(a, b) \in F$, having replacement edges e_a and e_b in G', respectively, we form a two-variable constraint $x_{e_a} + x_{e_b} \leq 1$. Finally we set $K = 1$. Clearly the construction can be accomplished in polynomial time.

Suppose there exists a directed path p from s to t in G containing at most one vertex from each pair in F. A corresponding path p' can be obtained in G' by substituting all p's constituent vertices that appear in F by their replacement edges in G'. All non-replacement edges in p^{prime} are assigned 1. The PFP condition ensures that for each replacement edge e_a, where $(a, b) \in F$, e_b is not in p'; thus $x_{e_a} = 1, x_{e_b} = 0$. It is easy to see that all the one-variable and two-variable constraints are satisfied, and $w(p^{prime}) = 1$, hence a solution of WPC.

Conversely, let p' be an $s - t$ path in G' satisfying all the constraints and having width no less than 1. The width of no less than 1 and every two-variable constraints being satisfied imply that at most one edge from any two-variable constraint appears in p'. Collapsing p' to a path $p \in G$ by shrinking the replacement edges into corresponding vertices, it is obvious that p satisfies the PFP condition. □

Even though the WPC problem is NP-complete, we discovered through simulations that widest paths obtained without considering LCC can usually achieve optimal bandwidth. The reason is that it is highly unlikely for links in a single path to correlate heavily. Therefore traditional widest-path algorithm suffices in realistic overlay topologies. In general, however, the WPC problem — with consideration of all possible pathological cases — is still NP-complete.

4 Maximum Flow with LCC

In this section we study the problem of maximum flow in an LCC graph. The traditional maximum flow algorithms such as Ford-Fulkerson and Push-Relabel cannot solve the maximum flow with LCC problem. We first formulate the problem as a linear program and then propose an algorithm for it based on Lagrangian relaxation and existing algorithms for minimum cost flow.

Maximum Flow with LCC Problem (MFC): Input : $\widehat{G} = (\widehat{V}, \widehat{E}), C, b$. Output: A flow $f \subset \widehat{G}$ satisfying LCC constraints (C, b). Goal : Maximize $|f|$.

Like the maximum flow problem, the MFC problem can be viewed naturally as a linear program. A variable v is used to indicate the total flow out of s and into t. In the flow conservation constraint, A is the node-arc adjacency matrix for $\widehat{G}$,[3] and d is a vector with a 0 for every node, except $d(s) = -1$ and $d(t) = 1$.

$$\begin{array}{rc} \text{Maximize} & v \\ \text{subject to} & Af + dv = 0,\ Cf \leq b,\ f \geq 0 \end{array}$$

The MFC linear program can be solved by general linear programming algorithms, such as the simplex method. However, due to their general nature, they may not be as efficient as algorithms that are tailored to the problem. We propose such an alternative algorithm.

Note that the MFC linear program only differs from the generic maximum flow linear program in having $Cf \leq b$ (LCC) as the inequality constraint instead of $f \leq b$. MFC can be seen as a generalized maximum flow problem; maximum flow is a special case of MFC with the identity matrix as C. With that observation, we modify the linear program slightly to reveal even more clearly the embedded maximum flow structure. We do this by sieving (uncorrelated) link capacity constraints from (C, b): for each link e, add the constraint $f(e) \leq b_l(e)$, where $b_l(e) = \min\{b(j) : C(j, e) = 1\}$, that is, minimize over all constraints in C involving $f(e)$. The additional $f \leq b_l$ constraints do not change the feasible flow region, therefore the new linear program is equivalent to the original one. The objective function is expressed in a different form for convenience.

$$z^* = \text{Minimize} - v \quad \text{subject to} \quad Af + dv = 0,\ f \leq b_l,\ Cf \leq b,\ f \geq 0. \tag{2}$$

It is now evident that MFC is a maximum flow problem with some additional constraints $Cf \leq b$(i.e., the LCC). We apply the decomposition solution strategy of Lagrangian relaxation [8] to the MFC problem, by associating nonnegative Lagrange multipliers $\mu = [\mu_i]_1^m$ with the LCC constraints ($Cf \leq b$), and creating the following Lagrangian subproblem:

$$L(\mu) = \min \quad -v + \mu(Cf - b) \quad \text{subject to} \quad Af + dv = 0,\ f \leq b_l,\ f \geq 0. \tag{3}$$

For any given vector μ of the Lagrangian multipliers, the value $L(\mu)$ of the Lagrangian function is a lower bound on the optimal objective function value $z^* = \min -v$ of the original problem (4). Hence, to obtain the best possible lower bound, we need to solve the Lagrangian multiplier problem

$$L^* = \max_{\mu \geq 0} L(\mu). \tag{4}$$

[3] Rows are nodes; columns are edges; for each directed edge $e = (i \to j)$, $A(i, e) = 1, A(j, e) = -1$, otherwise entries of A are zero.

Note that for the our Lagrangian subproblem (4), for any fixed value of Lagrangian multipliers μ, $L(\mu)$ can be found by solving a minimum cost flow problem. A polynomial-time minimum cost flow algorithm is the cost scaling algorithm, with a running time of $O(n^3 \log(nC))$, where n is the number of nodes and C is the upper bound on all the coefficients in the objective function. Since the objective coefficients are 1 or -1, the time complexity in this case is $O(n^3 \log(n))$. We choose the cost scaling algorithm precisely because its running time depends neither on m (number of rows in C), nor on U (upper bound on values in b_l), which may have large values, whereas C is a constant here.

Now that we can solve the Lagrangian subproblem for any specific μ, we can solve the Lagrangian multiplier problem (4) using the subgradient optimization technique. It is an iterative procedure: begin with an initial choice μ^0 of Lagrangian multipliers; the subsequent updated values μ^k are determined by $\mu^{k+1} = [\mu^k + \theta_k(Cx^k - b)]^+$. Here, the notation $[.]^+$ means taking the maximum of 0 and each vector component; x^k is a solution to the Lagrangian subproblem when $\mu = \mu^k$; θ_k is the step length at the kth iteration. The step length is selected to be a popular heuristic, $\theta = \frac{\lambda_k(UB - L(\mu^k))}{\|Cx^k - b\|^2}$, where $0 < \lambda_k < 2$ and UB is any upper bound on the optimal value of (4). [4]

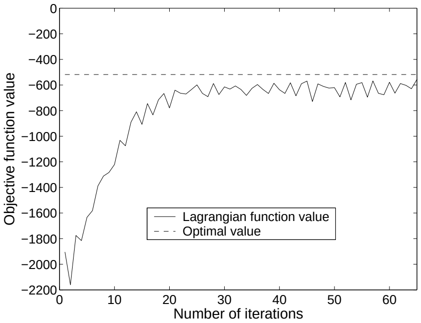


Fig. 11. Shows the convergence of Lagrangian function values (or Lagrangian subproblem solutions) $L(\mu)$ (in Problem 4) to a value near to the true optimal value z^* (in Problem 4), after a relatively small number of iterations.

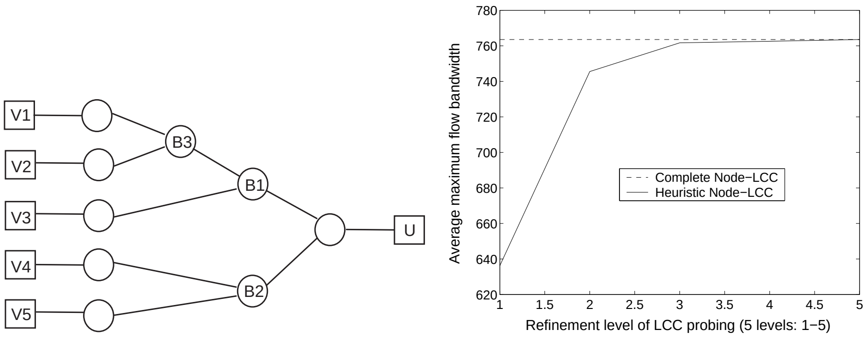
We show in Fig. 11 that for MFC in a simulated network in which 30% of the nodes are overlay nodes, the Lagrangian function values converge to near optimal value in around 65 iterations.

5 Constructing an LCC overlay

In this section, we present a distributed scheme for constructing an LCC overlay. In Sec. 2, we showed that node-based LCC exhibits notably better quality than no-LCC. The advantage of node-based LCC is that they are naturally distributed. In our scheme, an overlay node first determines a conservative set of node-based LCC; it then *successively refines* the LCC.

[4] It should be noted that sometimes there may be a gap between the optimal Lagrangian multiplier objective function value and the optimal value for the original problem, the branch and bound method can be used to overcome the gap. We do not go into the details here.

The input is a set of overlay nodes, each possessing a list of other known nodes; the list may not be complete at first, but it is periodically disseminated and updated. Existing methods make use of unicast probes to estimate link bandwidth. Independent unicast probes cannot yield shared bottleneck information. Therefore, the probing tool we use in our scheme is an efficient and accurate technique for detecting shared bottlenecks (DSB), proposed by Katabi et al. in [9, 10]. This technique is based on the entropy of the inter-arrival times of packets from flows. A set of flows are partitioned into groups of flows, each group of flows share a bottleneck, and the bottleneck capacities are also measured. We refer to this probing tool for detecting shared bottlenecks as DSB. Every time DSB is executed with the input of a set of flows, the output is a collection of groups of flows with their corresponding bottleneck capacities. Prior to determining LCC, a node selects k neighbors; for our simulation, the k highest bandwidth links are selected.



(a) Illustrates the phenomenon of hidden bottlenecks.

(b) The rapid convergence of the accuracy of discovered node-based LCC.

The node-based LCC are obtained in iterations of increasing refinement. In the first stage, the least refined set of LCC is determined. A node executes DSB once with the input of the set of k flows to all its neighbors. The k flows are partitioned into n bottleneck-sharing groups of flows, $g_1, g_2, \ldots, g_n$, with the respective bottleneck capacities $b_1, b_2, \ldots, b_n$. The LCC obtained are thus $C_1 = \{\sum_{e \in g_i} x_e \le b_i\}_{i=1}^{n}$. Since DSB detects only the dominant bottlenecks, some bottlenecks cannot be discovered in the first stage. We give an example of this in Fig. 5(a); assume that node U is using DSB to probe for bottlenecks, and assume that bottleneck $B1$ has a smaller capacity than $B3$. When node U executes DSB with all 5 flows from its neighbors ($V1, \ldots, V5$), only the most dominant bottlenecks $B1$ and $B2$ can be discovered. To determine more refined LCC, node U must execute DSB with the input of only the flows from $V1$ and $V2$. This will be done in the second iteration of LCC refinement.

In order to guarantee that all bottlenecks are found, all possible subsets of flows in each group must be probed separately. However, the brute-force search is exponential in computational complexity and hence infeasible. We maintain a low complexity by randomly dividing each group g into two subsets and execute DSB on each subset. Our simulation results show that this non-exhaustive approach is not only efficient but also able to quickly find LCC that are negligibly close to the complete LCC.

The entire procedure of discovering node-based LCC is summarized as follows:

1. Start with G containing one single group including all k flows.
2. Execute DSB with each group g from G separately.
3. Every group g is partitioned into n sub-groups, from which n LCC derive; add to C (the growing set of LCC) those LCC not redundant with ones already in C.
4. Each sub-group of > 2 flows is randomly divided into two groups, add to G.
5. Repeat step 2 as long as more LCC can be found.

The simulation results for a network of 100 nodes with 30% overlay nodes are given in Fig. 5(b). In our simulation, LCC obtained at successive stages of refinement are used to compute maximum flows, and maximum flows are also computed from the complete node-based LCC. A large number of source and destination pairs are randomly chosen to compute maximum flows and the maximum flow bandwidths are averaged over all such pairs. In Fig. 5(b), the average maximum flow bandwidths for successive stages of LCC refinement are plotted, and compared to the average maximum flow bandwidth computed using complete node-based LCC. After only 5 refinement stages, the DSB LCC are as good as complete node-based LCC. The number of stages required for such accuracy may have something to do with the node degree limit, which is set at 6 in this simulation, because the node degree limit determines the maximum size of the groups given by DSB.

The complexity of the procedure depends on two factors: number of executions of DSB and number of flows probed. A reasonable estimate of the number of packets per DSB flow, based on reported empirical results in [9], is a few hundred packets. In our simulation, to obtain LCC that are 98% accurate of complete node-based LCC, DSB is executed a few times and the number of flows probed is around 10, on average. This translates to a total of a few thousands of probes used. It is worth noting, though, that the probing can be done passively. The overlay can begin data transmission without knowledge of LCC. The data transmission acts as passive probing and is used to determine more and more refined node-based LCC over time. The data dissemination topology can adapt to the discovered LCC.

6 Related Work

To the best of our knowledge, there has not been previous work on overlays with LCC. Prior work have without exception assumed an overlay model of independent link capacities, with no correlation. To alleviate overloading of shared underlying bottlenecks, the typical approach is to limit overlay node degrees. Several projects based on Distributed Hash Tables, e.g., CAN [11] and Chord [12], designed structured overlay networks. Distributed algorithms for general-purpose overlay construction were proposed by Young *et al.* in [13] and by Shen in [4], using heuristics of minimum spanning trees and neighbor selection based on unicast latency and bandwidth. Application-specific proposals have been made for overlay multicast [1], content distribution [3] and multimedia streaming [2]. Also relevant is work by Ratnasamy *et al.* [14]. A distributed binning scheme is designed to build unstructured overlays; the aim is to incorporate more topological awareness. This work differs from ours in focusing exclusively on latency. Due to the additive nature of the latency metric (the bandwidth metric is concave), overlay links are essentially independent of each other in latency. We focus on overlay link capacity correlation.

Common to all these proposals are heuristics that use unicast probing to select overlay routes with low latency or high bandwidth. They view and treat overlay links as independent. However, we propose a new overlay model and hence work upon a premise distinct from previous work.

7 Conclusions

We have introduced a new overlay model, LCC-overlay, that uses linear capacity constraints to efficiently and accurately represent real networks with link correlations. We showed that LCC-overlay has optimal quality, and even the restricted node-based LCC yields good quality, while overlays with no LCC has poor quality which deteriorates as network size increases. We proposed a distributed algorithm for LCC-overlay construction. We also studied the problems of widest-path and maximum-flow with LCC.

References

1. Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A Case for End System Multicast," *IEEE Journal on Selected Areas in Communications*, pp. 1456–1471, October 2002.
2. M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-Bandwidth Multicast in Cooperative Environments," in *Proc. of the 19th ACM Symposium on Operating Systems Principles (SOSP 2003)*, October 2003.
3. J. Byers and J. Considine, "Informed Content Delivery Across Adaptive Overlay Networks," in *Proc. of ACM SIGCOMM*, August 2002.
4. K. Shen, "Structure Management for Scalable Overlay Service Construction," in *Proc. of NSDI*, 2004.
5. A. Medina, A. Lakhina, I. Matta, and J. Byers, *BRITE: Boston University Representative Internet Topology Generator*, http://www.cs.bu.edu/brite.
6. C. Faloutsos, M. Faloutsos, and P. Faloutsos, "On Power-Law Relationships of the Internet Topology," in *Proc. of ACM SIGCOMM*, August 1999.
7. M.S. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
8. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs NJ, 1993.
9. D. Katabi and C. Blake, "Inferring Congestion Sharing and Path Characteristics from Packet Interarrival Times," Tech. Rep., Laboratory of Computer Science, Massachusetts Institute of Technology, 2001.
10. D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *Proc. of ICCCN '01*, 2001.
11. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proc. of ACM SIGCOMM*, August 2001, pp. 149–160.
12. I. Stoica, R. Morris, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. of ACM SIGCOMM*, 2001.
13. A. Young, J. Chen, Z. Ma, A. Krishnamurthy, L. Peterson, and R. Wang, "Overlay Mesh Construction Using Interleaved Spanning Trees," in *Proc. of INFOCOM*, 2004.
14. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-Aware Overlay Construction and Server Selection," in *Proc. of the IEEE INFOCOM*, 2002.