Networking 2013 1569711317

The price of evolution in incremental network design: The case of mesh networks

Saeideh Bakhshi
College of Computing, Georgia Tech
saeideh@gatech.edu

Constantine Dovrolis
College of Computing, Georgia Tech
constantine@gatech.edu

***Abstract*—In practice most service-provider and enterprise networks are designed incrementally over time. This ongoing and heuristic design process is driven by changes in the underlying objectives and constraints (the "environment"). We first formulate the incremental network design approach as the constrained minimization of a certain modification cost, and compare that with the classical design approach in which the objective is to minimize the total network cost. We evaluate the cost overhead and the evolvability of incremental designs under two network expansion models (random and gradual), evaluating incrementally designed networks in terms of cost, performance (propagation delay) and robustness. Even though incremental design has some cost overhead, this overhead does not increase as the network grows. In the case of mesh networks, the incremental design process leads to networks with larger link density, lower average delay and improved robustness.**

I. Introduction

Complex technological systems, such as transportation and communication networks, manufacturing processes, microprocessors and computer operating systems, are rarely designed "from scratch." Instead, they are often subject to an evolutionary process in which existing designs are incrementally modified every time there is a change in the desired functionality or in the underlying constraints and objectives (the "design environment"). There are numerous examples and they span every engineering discipline. For instance, consider a wide-area communication network that expands over time to reach new locations, increasing its capacity depending on the offered load, and occasionally providing new services.

Even though the optimized design of communication networks has been studied in depth for several decades, the literature rarely considers that the design process is often incremental. Instead, it is typically assumed that the network is built *tabula rasa*. The corresponding problems are typically formulated as optimizations with multiple constraints, none of which is imposed by an earlier network however. In the relatively few previous studies that consider evolving networks (see section VIII), the focus has been on design algorithms and the corresponding optimization problems, rather than to compare incremental and optimized designs.

In this paper, we attempt a specific instance of the previous comparison in the context of communication network topology design. We ask the following question: *how does an incrementally designed network topology compare to an optimized topology, when both networks interconnect the same set of locations under the same reliability and performance constraints?* We are interested in comparisons that relate to cost, performance and topological robustness. We further limit (admittedly in a simplified manner) how the design environment changes with time: the set of interconnected network locations gradually expands at each time step. In an earlier publication, we focused on ring networks, where some of the previous questions can be answered analytically [2]. Here, we focus on general mesh networks, and the analysis is mostly based on numerical results.

This network topology design problem allows us to examine some fundamental questions about incremental versus optimized designs. First, how can we mathematically formulate the incremental design problem, and how is that formulation different from more traditional optimized topology design formulations? How costly is it to modify an existing design, relative to the cost of re-designing the network from scratch? What is the *"price of evolution"*, i.e., the cost overhead of an incremental design relative to the corresponding optimized design? When is it better to abandon incremental changes on an existing topology and start fresh? Does the incremental design process perform better when the design environment varies in a gradual manner as opposed to randomly? How does an incremental network design compare with the corresponding optimized design in terms of performance (propagation delay in this context)? How do the two networks differ in terms of topological properties, and in particular in terms of robustness to failure?

Section II presents a formulation for the optimized and incremental design problems. Section III provides a summary of the earlier results for *ring networks*, as a reference point. Section IV describes the optimized and evolved mesh network design algorithms. Section V compares the incremental and optimized design processes in terms of cost and performance under a single-node expansion model. Section VI examines a faster expansion model in which multiple nodes are added simultaneously. Section VII compares the topological robustness of the optimized and evolved networks in terms of a node centrality metric. We review the related work in section VIII and conclude in section IX.

This research was supported by the National Science Foundation under Grant No.0831848.

II. Framework and Metrics

In this section, we present more formally the optimized and incremental design problems. Even though these formulations are quite general, in the rest of the paper we apply them in the context of topology design for communication networks. As in any design problem, there is a desired function (e.g., construct a communication network to interconnect a given set of locations), some design elements (e.g., routers, wide-area links), constraints (e.g., related to performance and robustness) as well as an objective (e.g., to minimize the total cost of the required design elements). The design process aims to use the appropriate elements so that we achieve the desired function, while satisfying the constraints and meeting the given objective. It is often assumed that this process is conducted only once in an otherwise static environment. Instead, we consider the case that the design takes place in a *dynamic environment*. In the context of communication networks, the network may expand to new locations, the cost of design elements may fluctuate, or the constraints may become more stringent over time. We consider a discrete-time model and we refer to the k'th time epoch as the *k'th environment*. At a given environment k, we assume that all inputs of the design problem are known and constant.

How can we design a communication network in such a dynamic environment? We identify two fundamentally different approaches. In the *optimized approach* we aim *to minimize in every environment the total cost of the network* subject to the given constraints. In the *incremental approach* we aim instead to *minimize the modification cost relative to the network of the previous environment*, again subject to the given constraints. We refer to the former network as *optimized* and to the latter as *evolved*.

More rigorously, let $\mathcal{N}(k)$ be the set of *acceptable networks* at environment k, i.e., networks that provide the desired function and meet the given constraints at environment k. The cost of a particular network $N \in \mathcal{N}(k)$ is $C(N)$. $C(N)$ is the sum of the costs of all design elements in N. We assume that there are no other costs associated with N; for instance, there is no monetary cost to compute the design or to interconnect its elements.

In optimized design, the objective at each environment k is to identify an acceptable network $N_{opt}(k)$ from the set $\mathcal{N}(k)$ that has the minimum cost $C_{opt}(k)$,

$$C_{opt}(k) \equiv C(N_{opt}(k)), \quad N_{opt}(k) \equiv \arg \min_{N \in \mathcal{N}(k)} C(N). \quad (1)$$

If the optimized network is not unique, we break ties with secondary objectives (for instance, minimize the total number of links). Designing such networks is computationally intractable (NP-hard), and so they are often solved heuristically, approximating the previous optimization objective. For this reason, we do not refer to $N_{opt}(k)$ as *optimal* but as *optimized*. The former would be the actual solution to the previous problem if we could compute it; the latter is the best solution we can compute given a certain design heuristic. We present our optimized design algorithms in section IV.

In the incremental design approach, on the other hand, we design the new network $N_{evo}(k)$ based on the network $N_{evo}(k-1)$ from the previous environment $k-1$. The objective of the incremental design process is to identify an acceptable network $N(k) \in \mathcal{N}(k)$ that minimizes the *modification cost* $C_{mod}(N_{evo}(k-1); N(k))$ between networks $N_{evo}(k-1)$ and $N(k)$. For simplicity, we denote the previous modification cost as $C_{mod}(k)$.

To define the modification cost precisely we first need to answer the question: *what should we do with design elements that are present in $N_{evo}(k-1)$ but not in $N(k)$?* We identify three options. First, we keep them active in $N(k)$ even though they are not necessary - this is the *Ownership* option. Second, they can be removed from $N(k)$ (even though they could be re-used in a future environment) - this is the *Leasing* option. The third option is that there is a *Surplus* $S(k)$ of design elements that have been purchased prior to environment k but are not needed in $N(k)$. Any design elements in the surplus can be moved later back to the network at zero cost. The ownership option increases the cost of the evolved network relative to the surplus option, while the leasing option increases the modification cost relative to the surplus option. In the rest of this paper we focus on the surplus option, which is more common in practice and conceptually interesting; the ownership and leasing options are studied in the extended version of this paper.www.cc.gatech.edu/~sbakhshi/evodesign_extended.pdf

In the presence of a surplus, *the modification cost $C_{mod}(k)$ is defined as the cost of new design elements that are needed in $N(k)$ but are not present in $N_{evo}(k-1)$ or at the surplus* $S(k-1)$. Formally, $C_{mod}(k)$ is the cost of the design elements in the set

$$N(k) \setminus [N_{evo}(k-1) \cup S(k-1)] \quad (2)$$

slightly abusing the notation $N(k)$ to also refer to the set of design elements in the network $N(k)$. Similarly, the surplus at environment k includes the design elements that are present in $N_{evo}(k-1)$ but are not present in $N(k)$,

$$S(k) = [N_{evo}(k-1) \cup S(k-1)] \setminus N(k). \quad (3)$$

Let $C_{srp}(k)$ be the total cost of all design elements in the surplus at environment k.

With the previous definitions, we can now formulate the incremental design process as a minimization of the modification cost across all acceptable networks:

$$C_{evo}(k) \equiv C(N_{evo}(k)), \quad N_{evo}(k) \equiv \arg \min_{N \in \mathcal{N}(k)} C_{mod}(k) \quad (4)$$

The evolved network $N_{evo}(k)$ may not be unique in general. Ties are broken by considering a secondary objective: if two networks minimize the modification cost, select the network with the minimum total cost.In our computational experiments, two modification costs are rarely equal because link costs are based on distance and they are real numbers. As in the case of optimized design, we compute the solution of the incremental design problem with a heuristic, described in section IV.

The cost of the evolved network at environment k consists of three terms: a) the cost of all design elements at the evolved network and at the surplus at enviroment $k-1$, b) plus the cost of any newly purchased elements at environment k, c) minus the cost of any remaining elements at the surplus at environment k. Mathematically,

$$C_{evo}(k) = C_{evo}(k-1) + C_{srp}(k-1) + C_{mod}(k) - C_{srp}(k) \quad (5)$$

for $k \geq 1$. We assume that the initial evolved network and its cost $C_{evo}(0)$ is known, and that the initial surplus is empty ($C_{srp}(0) = 0$). Expanding (5), we can write the cost of the evolved network as:

$$C_{evo}(k) = C_{evo}(0) + \sum_{i=1}^{k} C_{mod}(i) - C_{srp}(k). \quad (6)$$

Thus, the cost of the evolved network at environment k is the cost of the initial network plus the cost of all design elements that were purchased in the last k environments, minus anything that remains in the surplus at time k.

A. Metrics

We now introduce three metrics to compare a sequence of optimized and evolved networks.

First, the *cost overhead* $v(k)$ of the evolved design $N_{evo}(k)$ relative to the corresponding optimized design $N_{opt}(k)$ at environment k is:

$$v(k) = \frac{C_{evo}(k)}{C_{opt}(k)} - 1 \geq 0 \quad (7)$$

where the inequality is expected from the definition of $C_{opt}(k)$. What is more important however is whether the cost overhead of the incremental design process increases with k, i.e., whether the evolved networks become increasingly more expensive compared to the corresponding optimized networks. If that is the case, the incremental design process would diverge over the long-term towards extremely inefficient designs. The reader should note that the cost overhead metric is different than the well-known *approximation ratio* in online algorithms.

Second, the *evolvability* $e(k)$ is defined as:

$$e(k) = 1 - \frac{C_{mod}(k)}{C_{opt}(k)} \leq 1. \quad (8)$$

The evolvability represents the cost of modifying the evolved network from environment $k-1$ to k, relative to the cost of redesigning the network "from scratch" at time k. High evolvability, close to 1, means that it is much less expensive to modify the existing network than to re-design a new network. On the other hand, when the evolvability becomes negative it is beneficial to stop the incremental design process and design a new optimized network.

Third, the *surplus overhead* $r(k)$ is defined as:

$$r(k) = \frac{C_{srp}(k)}{C_{evo}(k)} \geq 0. \quad (9)$$

The surplus overhead quantifies the cost of design elements that have been previously purchased but are now left unused, relative to the current cost of the evolved network. An incremental design process that leads to a gradually increasing surplus overhead would be inefficient in terms of its cumulative cost over time.

B. Expansion models

We consider only one way in which the environment can change with time: *expansion*. Specifically, the set of locations that the network has to interconnect at any time k is increasing with k. This is probably the most natural way the environment can change with time in the context of communication networks.

In the simplest form of expansion, the network size increases by only one node at each environment; we refer to this as *single-node expansion*.

We also consider a *multi-node expansion* scenario in which the network size increases once by a multiplicative factor ρ, which we refer to as *expansion factor*. Specifically, if the network size increases from n nodes to $n+m$ nodes, the expansion factor ρ is

$$\rho = \frac{n+m}{n} \geq 1. \quad (10)$$

We compare two expansion models: *random* and *gradual*. In both models the set of all possible locations $\mathcal{L}$ is the same. The locations are randomly placed on a bounded region of the Euclidean plane. The two expansion models differ in how they select the new locations that the network will expand to. In random expansion, the new locations are selected randomly from $\mathcal{L}$. In gradual expansion, we select iteratively each of the new locations from $\mathcal{L}$ so that it is the closest location to either one of the existing nodes in network or to any new location we have added to the network.

We choose to study these two expansion models because they represent two qualitatively different ways in which the environment changes with time. In random expansion, the new locations can be anywhere and so it may be costly for the incremental design to adjust the previous network with only minor modifications. In gradual expansion, the environment changes in "smaller steps" because the new locations are as close as possible to nodes of the existing network. In other words, random expansion represents a more challenging dynamic environment in which "anything can happen" while the gradual expansion model represents a more predictable environment.

III. Ring Networks

In this section, we summarize the results of the first part of this work [2], mostly for completeness and for comparison with the mesh network results. Ring networks are widely used mostly in metropolitan-area networks, where the delay constraints are less stringent, as they are robust to single-node failures (two node-disjoint paths exist between any pair of nodes) and they are typically less costly than mesh networks.

In the incremental ring design process, the minimum modification cost under single-node expansion can be computed as follows. Suppose that the existing ring $N_{evo}(k-1)$ has size

n and we add a single extra node z at time k. The minimum modification cost will result if we connect z to two adjacent nodes x and y of $N_{evo}(k-1)$, such that

$$C_{mod}(k) = min_{(x,y)\in N_{evo}(k-1)}(||z-x|| + ||z-y||) \quad (11)$$

also removing the link (x, y).This process is illustrated in Figure 1(a).

In the case of multi-node expansion, we use an iterative heuristic that aims to minimize the modification cost. Suppose that the existing ring $N_{evo}(k-1)$ has size n and we add a set Z of more than one new nodes at time k. In each iteration, we select the node z from Z that minimizes the expression (11), connect z to the existing ring as in the case of single-node expansion, and then move z from Z to the set of nodes in $N_{evo}(k-1)$. This process is illustrated in Figure 1(b). The

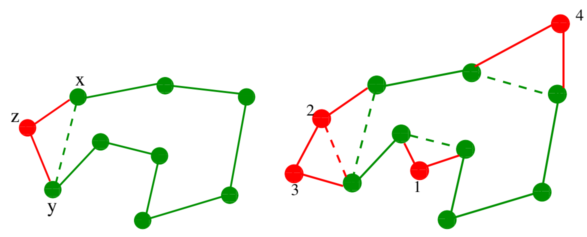


Fig. 1. Connecting new nodes to an existing ring: (a) single-node expansion, (b) multi-node expansion (we show the order in which nodes are connected).

detailed derivations for the case of ring networks can be found in [2]. Here, we only summarize the key results for various ring expansion types in Table-I. The notation $x \sim f(n)$ means that x tends to become proportional to f(n) as n increases. The results for multi-node expansion assume that the number of new nodes m is much lower than the number of existing nodes n (i.e., ρ is close to 1).

IV. Mesh Network Design

We now summarize the algorithms that we use to design optimized and evolved *mesh networks.* A detailed description (pseudocode) of the two algorithms is included in the extended version of this paper.www.cc.gatech.edu/~sbakhshi/evodesign_extended.pdf

In terms of reliability, the design constraint is that every pair of nodes should be connected through at least *two node-disjoint paths*, the primary and the secondary. The primary path is the path with the minimum propagation delay, i.e., the shortest path when the link cost is equal to the link's propagation delay.We assume that the propagation delay of a link is proportional to its straight-line length on the Euclidean plane. The secondary path is also the shortest path, but after we have removed the nodes that participate in the primary path (except the source and destination nodes). In terms of performance, the constraint is that the propagation delay of both paths should be less than a given threshold D. We say that a network is *acceptable* if it meets the reliability and performance constraints for every pair of nodes. Note that depending on the node distances and the threshold D an acceptable network may not exist.

Given a set of locations that we need to interconnect, we should only consider the cost of links (edges). The routers (nodes) introduce the same cost in optimized and evolved networks, and they can be ignored. We assume that the cost of a link is proportional to its length. That cost is typically much larger than the cost of the router interfaces at the two terminating points of the link, it usually does not depend on the capacity of the link, and it increases roughly linearly with distance (at least for transcontinental links).

Because the problem of minimum-cost topological design for mesh networks with reliability and delay constraints is NP-Hard [11], we rely on heuristics. Even though there are some approximation bounds for special networks, mostly trees, we are not aware of such bounds and approximation algorithms for general mesh networks under the previous design constraints. Our objective is not to develop new network design algorithms but to compare optimized with evolved designs, and so we use two rather simple algorithms referred to as *OPT* and *EVO*. Both algorithms are probabilistic and iterative, they are quite similar in terms of how they add and remove links, but they differ in their objective function. Additionally, the *EVO* algorithm makes use of *surplus links*.

In the *OPT* algorithm the objective function is to minimize the total network cost, i.e., the sum of all link costs. In the *EVO* algorithm, the objective is to minimize the modification cost relative to the previous network, re-using any required surplus links.If the link between nodes X and Y is in surplus at environment k, those two nodes were connected at an earlier environment but the link between them is not required at environment k and so it is not "lit" in the corresponding network. In *EVO*, the modification cost is the total cost for all *new links* that are required in the evolved network. That cost does not include the cost of existing links in the previous network, or any surplus links that are re-used in the new network. In both algorithms, we use the same stopping criterion. Because the algorithms are probabilistic, each iteration may result in a different acceptable network (if such a network exists). If the new network is *not* better, in terms of the optimization objective of each algorithm, than the best network that has been computed up to that point, we move to the next iteration. The algorithms terminate if we cannot improve the optimization objective of each algorithm for a number (10) of successive iterations.

Each iteration of the OPT algorithm aims to find an acceptable network. At the end we select the network with minimum total cost among all designed acceptable networks. Each iteration has two phases. In the first phase, the algorithm adds links probabilistically, in order of increasing cost, until an acceptable network is computed. In the second phase, the algorithm attempts to remove as many existing links as possible, in order of decreasing cost, as long as the network remains acceptable. That stage is also probabilistic. The previous process starts from an optimized ring that interconnects all given locations, computed using the Concorde TSP solver

Expansion type	C_{mod}	C_{evo}	C_{opt}	$1-e(n)$	$v(n)$
Random Single Node	$\sim \frac{1}{\sqrt{n}}$	$\sim \sqrt{n}$	$\sim \sqrt{n}$	$\sim \frac{1}{n}$	$\sim$ constant
Gradual Single Node	$\sim$ constant	$\sim n$	$\sim n$	$\sim \frac{1}{n}$	$\sim$ constant
Random Multi Node	$\sim m/\sqrt{n}$	-	$\sim \sqrt{n+m}$	$\sim \frac{\rho-1}{\sqrt{\rho}}$	$\sim \frac{\rho-1}{\sqrt{\rho}}$
Gradual Multi Node	$\sim m \times$ constant	-	$\sim n+m$	$\sim 1-\frac{1}{\rho}$	$\sim 1-\frac{1}{\rho}$

TABLE I
SUMMARY OF RING RESULTS

[6].

We found empirically that the two probabilities p_{add} and p_{del} do not have a strong impact on the resulting minimum network cost, as long as they are between 0.8 and 1; we use p_{add}=p_{del}=0.9.

The EVO algorithm is similar to *OPT* in the way it adds and deletes links, but with three important differences. First, *EVO* connects the set of new nodes to the previous network using the iterative process; recall that that algorithm aims to connect each new node to the existing network introducing the lowest modification cost. Second, *EVO* attempts to re-use surplus links as much as possible so that we minimize the cost of new links that must be acquired at each environment. Third, *EVO* has two link deletion phases. It first removes (probabilistically) *new links* that are not necessary in order of decreasing cost. Any link deletions in this phase reduce the modification cost. Then, it removes (again, probabilistically and in order of decreasing cost) existing links that are not necessary – those links become part of the surplus. Any link deletions in this phase do not reduce the modification cost, but they reduce the cost of the evolved network. Note that the latter is a secondary objective, and it is pursued only after we have reduced the modification cost as much as possible. We use the same values for p_{add} and p_{del} as in *OPT*.

V. SINGLE-NODE MESH EXPANSION

We now present computational results for mesh networks under single-node expansion. We focus on comparisons between optimized and evolved networks, as well as between random and gradual expansion.

The network expansion process is performed as follows. We consider a rectangular area of length 3000 and width 1500 (roughly the aspect ratio of the continental US) and 500 potential locations in that area. We start from a randomly chosen location and in each step we grow the network (according to the random or gradual expansion model) by adding one more location to the network. The optimized and evolved networks always interconnect the same set of locations. The largest network consists of 60 locations - this is a realistic scale for the backbone of a nation-wide service provider. The experiments are repeated 20 times, and we report 90% confidence intervals for all results.

The delay bound is set to $D = 1.3 \times d$, where d is the length of the diagonal of the previous rectangle. With this delay bound, the designed networks are sparse (the number of links is typically less than twice the number of nodes), which is also a characteristic of physical-layer backbone connectivity in practice [7]. Further, with this value of D we can always compute an acceptable network using the algorithms of the previous section.

We have also experimented with other delay bounds between $1.2 \times d$ and $2.5 \times d$, without observing significant qualitative differences. As D increases, the resulting networks get sparser and after a certain point they become rings. As D approaches d, on the other hand, it becomes likely that there are no acceptable networks for a given set of locations or the networks become unrealistically dense.

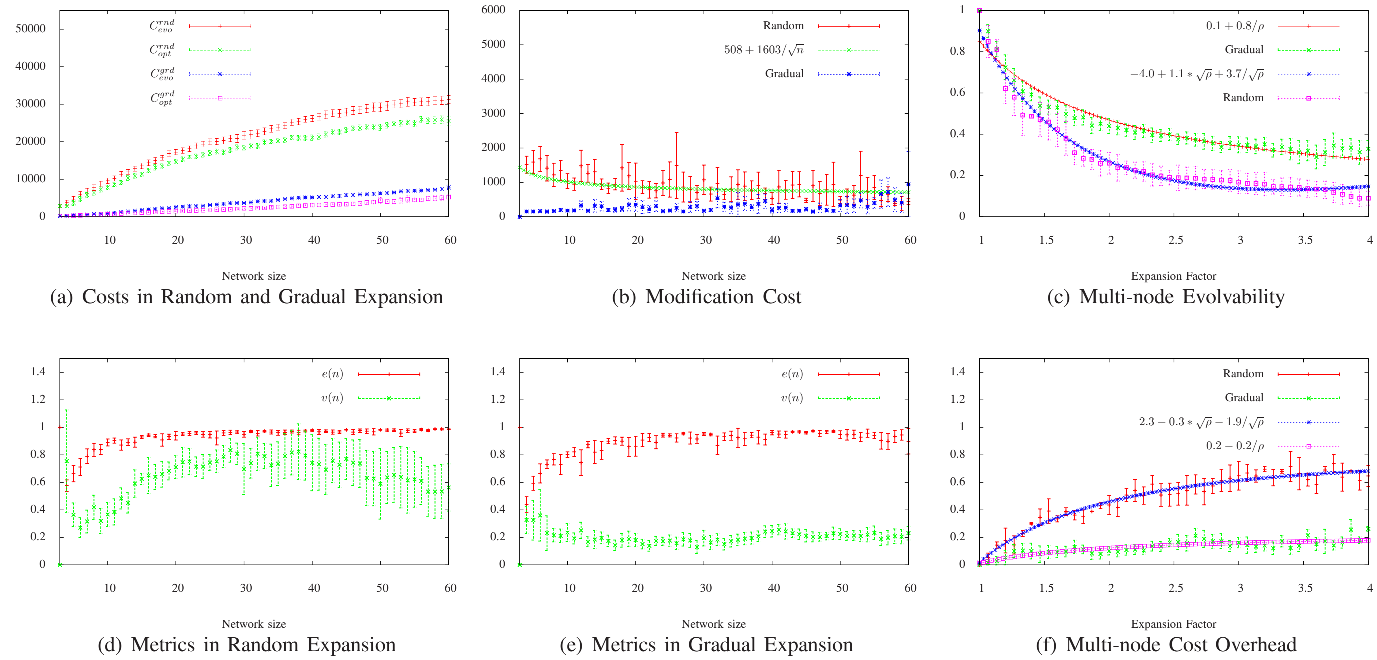
The costs of the optimized and evolved networks for random and gradual are shown in Figure 2(a). We have confirmed that these costs scale as $\sqrt{n}$ in the case of random expansion, and as n in the case of gradual expansion (the regression lines are omitted for clarity). Interestingly, these are the same scaling expressions with the case of ring networks [2].

Even though there is significantly higher variability in mesh networks. Specifically, the modification cost decreases as $1/\sqrt{n}$ under random expansion, and it remains practically constant under gradual expansion (Figure 2(b)). All previous costs are higher in random than in gradual expansion. This is because gradual expansion leads to much shorter links (Figures 3(a) and 3(e)).

In terms of cost overhead, the scaling analysis for rings [2] predicts that $v(n)$ should not increase with the network size, at least for large values of n. Figures 2(d) and 2(e) show the cost overhead for mesh networks under random and gradual expansion, respectively. The variability across different expreriments is large, and so we use the non-parametric Mann-Kendall hypothesis test for trend detection. Indeed, when we focus on the larger values of n, say $n > 20$, the test *cannot reject* the null hypothesis that the cost overhead shows no trend (p-value = 0.36 expansion and 0.34 for gradual expansion). So, we expect that the cost overhead does *not* increase under single-node expansion, even in the case of mesh networks. The cost overhead is significantly lower under gradual expansion than under random expansion; gradual expansion leads to networks that are much closer to optimized, in terms of cost, than random expansion.

Figures 2(d) and 2(e) also show the evolvability under random and gradual expansion. The evolvability increases fast in both cases until it becomes approximately one. So, it is less costly to incrementally modify an existing network that to re-design it from scratch.

The surplus overhead is shown in Figure 3(b). There is a



(a) Costs in Random and Gradual Expansion
(b) Modification Cost
(c) Multi-node Evolvability
(d) Metrics in Random Expansion
(e) Metrics in Gradual Expansion
(f) Multi-node Cost Overhead

Fig. 2. Results for mesh networks under random and gradual single-node and multi-node expansions.

decreasing trend under both random and gradual expansions. The fact that the surplus overhead does *not* increase means that the surplus does not become increasingly more costly relative to the cost of the evolved network. If that was the case, the evolved network would gradually accumulate a "baggage" of unused links with increasing cost relative to the cost of the network itself. The opposite happens: even though the cost of the surplus increases in absolute terms, it decreases relative to the cost of the network. It should be noted, however, that the surplus overhead can be significant in absolute terms, especially in the first few environments. For instance, in the first ten environments the average surplus overhead is often as high as 3-5, meaning that the total cost of surplus links is 3-5 times higher than the cost of the links that are actually "lit." This implies that many of the links that are added early in the expansion process quickly turn out to not be necessary and they are moved to surplus.

A. Performance comparisons

We have also compared the performance of evolved and optimized networks in terms of the path propagation delay metric. *Which networks give lower-delay paths and why?* When there are no failures, each (directed) pair of nodes communicates through its primary path. Figure 3(c) shows the ratio of the average primary path delay between the optimized and evolved networks.

The key observation here is that, *under random expansion, evolved networks have a significantly lower average path delay than the corresponding optimized networks.* After the network size has reached about 20 nodes, the primary path delay in evolved networks is about 50%-70% of the primary path delay in optimized networks.

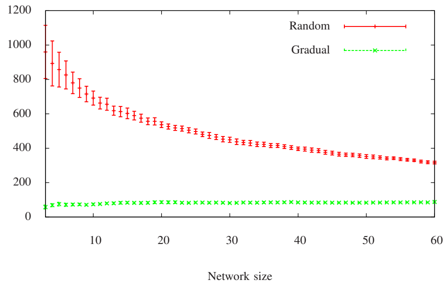
There are two factors that contribute to path delays: the individual link delays and the number of links in each path (the path hop-count). The average link delay is only slightly higher in evolved networks compared to optimized networks (Figure 3(a)). The average primary path hop-count, however, is significantly lower in evolved networks (Figure 3(d)). This is because evolved networks have few nodes with large degree (hubs), as shown in section VII. Those nodes appear in a large fraction of primary paths, providing interconnection "shortcuts" for many pairs of nodes. Optimized networks, on the other hand, do not have hubs (see section VII) and their average primary path hop-count is about twice as large (for $n > 20$) compared to evolved networks.

The previous performance advantage of evolved networks does *not* hold, however, under gradual expansion (Figure 3(a)). In those evolved networks, the effect of the lower average path hop-count is smaller in magnitude, and it is offset by a slightly higher average link delay relative to optimized networks.

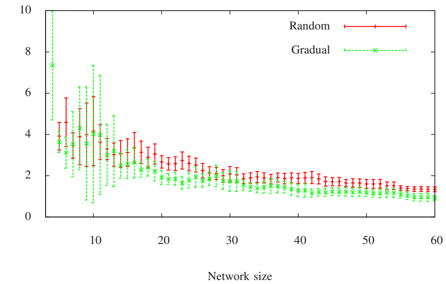
VI. Multi-Node Mesh Expansion

In this section, we consider multi-node expansion in mesh networks. An expansion factor ρ means that the network expands at a single environment from an initial size of n nodes to $\lfloor \rho\, n \rfloor$ nodes. In the following experiments, n=15 nodes.

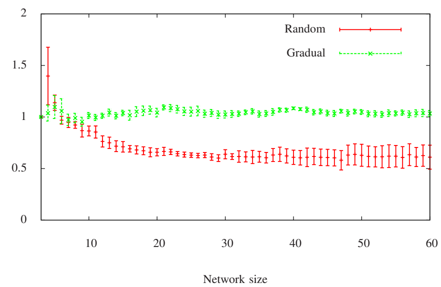
The initial evolved network (before the multi-node expansion) is designed using single-node expansion until it reaches n nodes. We focus on the effect of ρ on the evolvability $e(\rho)$ and cost overhead $v(\rho)$.



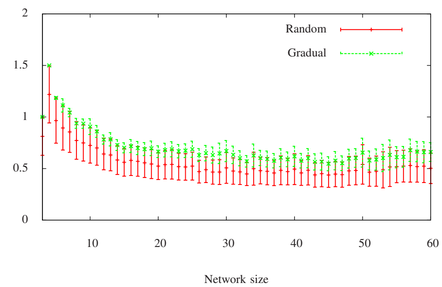
(a) Average Link Delay in EVO



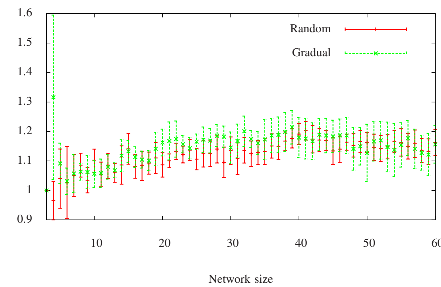
(b) Surplus Overhead



(c) Primary Path Avg Delay (ratio $\frac{EVO}{OPT}$)



(d) Primary Path Avg Hop-Count (ratio $\frac{EVO}{OPT}$)



(e) Average Link Delay (ratio $\frac{EVO}{OPT}$)

Fig. 3. Mesh network properties under single-node expansion.

Figure 2(c) shows the evolvability under random and gradual expansion as ρ increases. The evolvability decreases with ρ, and the decrease is faster under random expansion. The critical expansion factor $\hat{\rho}$ at which the evolvability becomes zero under random expansion is larger than four. Thus, at least based on these computational results, it is less costly to modify the existing network incrementally than to redesign it from scratch if the network size is increased by less than a factor of four. On the other hand, the evolvability under gradual expansion remains positive in the range of network sizes that we could design. It is an open question whether the evolvability can ever be negative under gradual expansion.

Figure 2(f) shows that the cost overhead increases with ρ under both random and gradual expansion. The latter leads to significantly lower values. The increase of the cost overhead is concave under both random and gradual expansion, and it does not exceed 100% when ρ is less than four, at least in these computational results.

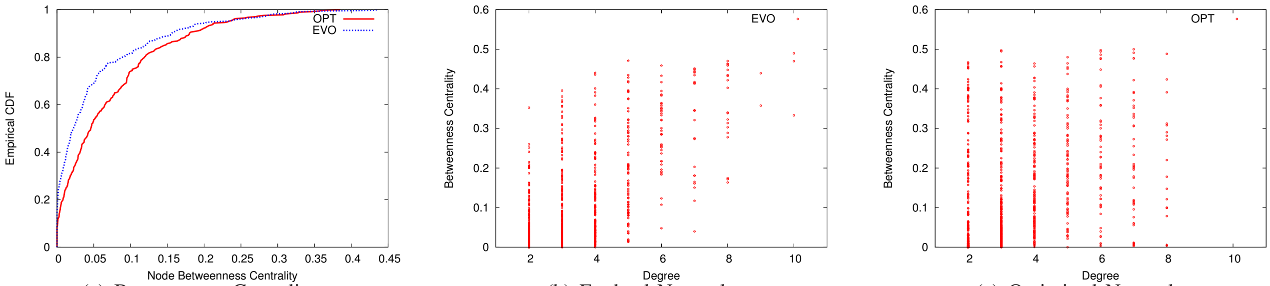
The scaling expressions that were derived for ring networks [2] assuming that ρ is close to one also give accurate regression curves for mesh networks (when $\rho < 3.5$). Recall however that these expressions should not be used to examine the asymptotic behavior of the evolvability or cost overhead as ρ increases.

VII. Centrality and Robustness

The generated networks are robust to single node failures because they have two node-disjoint paths (primary and secondary) between each pair of nodes. We could compare the robustness of the designed topologies by considering multiple node or link failures. We rely instead on a network analysis approach that is based on the *node betweenness centrality* metric [9]. Specifically, let us define the Betweenness Centrality of a node (node-BC) as the fraction of primary paths that traverse that node, among all primary paths in the network. The nodes with the highest BC values can be thought of as the network's most critical components; if they are somehow perturbed (without necessarily failing), the impact on the entire network will be more severe. We can compare the robustness of two networks X and Y that have the same number of nodes (and thus the same number of primary paths) using the BC metric. If the average node-BC across all nodes in X is higher than in Y, network X is more susceptible (or less robust) to node perturbations than network Y.

In the following, we compare the robustness of evolved and optimized networks under the *single-node random expansion model.* Figure 4(a) shows the empirical CDF of the BC metric across all nodes in networks of size n=50. These empirical CDFs are constructed from 20 independently generated networks (thus, the sample size is 1000 node-BC values). Note that the nodes of the optimized network have higher betweenness centrality than the nodes of the evolved network, at least when the BC is less than 20%. The average node-BC is 0.15 in optimized networks and 0.10 in evolved networks. So, the expectation is that *an evolved network will be more robust to node perturbations than an optimized network of the same size.*

There is an interesting difference between the two networks however, which is not evident in the previous CDFs. Figure 4(b) shows a scatter plot for the node degree and node-



(a) Betweenness Centrality (b) Evolved Network (c) Optimized Network

Fig. 4. Robustness results under single-node random expansion (n=50).

BC in the 20 evolved networks, while Figure 4(c) shows the corresponding scatter plot for the 20 optimized networks. In the evolved networks, there is a strong positive correlation (Pearson's correlation coefficient: 84%) between node degree and node-BC; there is no significant correlation in optimized networks (Pearson's correlation coefficient: 11%). In other words, *in the evolved networks, the most critical nodes (highest BC values) are also the nodes that have the largest number of connections; this is not the case in optimized networks.* The previous observation implies that even though evolved

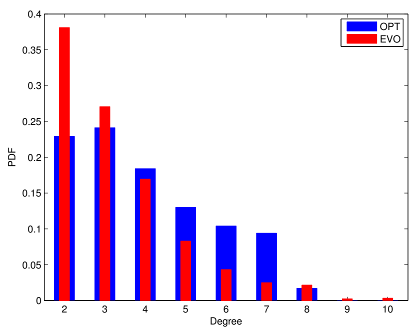


Fig. 5. Degree distribution of evolved and optimized networks.

networks are more robust to *random node perturbations* than optimized networks, *they can be more fragile to perturbations that affect those higher-degree nodes.* This is similar to a well-known finding about scale-free networks generated using the preferential attachment model [1]: those networks are robust to the failure of randomly selected nodes but are fragile to the failure of high-degree nodes, referred to as *hubs.* We cannot examine whether the evolved network topologies are also scale-free because of the computational complexity involved in designing such networks with thousands or millions of nodes. We can ask, however, whether evolved networks tend to have hubs or not.

Specifically, we can identify hubs using the theory of random graphs. In that network model, the node degree follows the binomial distribution $B(n, p)$ where n is the number of nodes and p is the probability that an edge exists between two nodes (i.e., $n\,p$ is the average node degree). For each value of n, we measure the average node degree for the generated evolved and optimized networks. Then, we identify as *hubs* those nodes that are unlikely (less than 5% probability) to appear with the same or higher degree in a random graph of the same network size and average degree. *In evolved networks, the probability that a network includes at least one hub is about 20%.* On the contrary, *optimized networks do not appear to have hubs*; the corresponding probability is less than 2%.

To further understand the topological differences between evolved and optimized networks, we also compare the degree distribution of evolved and optimized networks (see Figure 5). In evolved networks, about 4% of the nodes have a degree of eight or more; the corresponding percentage is 2% in optimized networks and there are no nodes with degree higher than eight. Additionally, the percentage of nodes with only two links (the minimum degree that is necessary to satisfy the reliability constraint) is 38% in evolved and 23% in optimized networks. In other words, *the evolved networks also have a more skewed degree distribution*; the skewness of the degree distribution is 1.2 in optimized and 1.5 in evolved networks.

Why is it that the incremental design process creates topologies that have hubs? Recall that the evolved network at environment k is generated from the corresponding network at environment $k-1$, also re-using "for free" any surplus links of each node. The incremental design process avoids the purchase of new links, so that it can minimize the modification cost. This means that if a node X is already a hub at environment k, or if it is not a hub but it has accumulated a large link surplus by environment k, it can be a hub at environment $k+1$ without introducing any modification cost. Because of the surplus option, however, a hub at environment k may not remain a hub at the next environment if some of its links are no longer needed at time $k+1$. Indeed, the conditional probability that a hub at time k remains a hub at the next environment is about 50%. In other words, *the hubs in evolved networks are not persistent.* This is not the case in preferential-attachment networks, where the hubs tend to be the nodes that have been added early in the network.

VIII. Related Work

The topology design literature is extensive both in computer networking and in theoretical computer science, and it is well covered in a recent book by Pioro and Medhi [11].

The majority of that literature, however, focuses on optimized network design. Those (few) studies that focus on incremental network design (also referred to as "multi-period design"), do not compare with optimized design [8], [5], [4], [10], [14], [15]. Instead, these works mostly propose algorithms for incremental network design under a wide range of different constraints and objectives. None of them is significantly relevant to our study because they do not compare incremental designs with the corresponding optimized designs, and they do not consider different expansion models (e.g., random versus gradual or single-node versus multi-node).

In the context of data-center networks, some authors have recently recognized that it is important to be able to design such networks incrementally. Curtis et al. [3] examined an incremental approach that adds one switch at a time to tree-based topologies. Another incremental data-center design approach is Jellyfish [12], where the authors mention in brief that the path length and capacity of Jellyfish topologies are close to those of optimized topologies with the same number of nodes.

A quite different, but still relevant, study by Tero et al. [13] compared the Tokyo rail system (as an example of an optimally designed transportation network) with a natural network formed by the slime mold Physarum polycephalum. The natural network grew in an incremental manner, without any centralized control or "intelligence." The authors compared the two networks in terms of efficiency, fault tolerance, and cost and found that they are actually quite similar.

IX. Conclusions

We now return to the questions that were asked in the introduction and summarize our main findings. The following conclusions are supported by asymptotic scaling expressions for rings and by computational results in the case of mesh networks.

1. We formulated the incremental network design process as an optimization problem that aims to minimize the modification cost relative to the previous network. We also identified and compared certain expansion models (random versus gradual, and single-node versus multi-node).
2. Even though an evolved network has higher cost than the corresponding optimized network, the cost overhead of the former does not increase as the network grows, at least under single-node expansion.
3. The incremental design process leads to networks with higher performance in terms of the average propagation delay.
4. We found out that under random expansion, there is much higher possibility of seeing a hub. The existence of hubs contributes to improving the performance of evolved network, by decreasing the number of hops on primary path.
5. The incremental design is more robust in terms of a node betweenness centrality metric, compared to optimized networks. This effect is more pronounced under random expansion.
6. Under single-node (random or gradual) expansion, it is less costly to follow the incremental design approach than to re-design the network from scratch. The evolvability under basic expansion approaches one as the network grows.
7. Under multi-node and random expansion, there is a critical value $\hat{\rho}$ of the expansion factor beyond which it is less costly to abandon the existing network and re-design the network from scratch. It is not clear whether this is ever the case under gradual expansion; our computational experiments have never produced negative evolvability in that case.
8. The incremental and optimized design processes lead to significantly different network topologies. The evolved network has a more skewed degree distribution compared to the optimized network, and it includes few nodes (hubs) with much higher degree and betweenness centrality than most other nodes.
9. The surplus overhead of the incremental design process does not increase with time, and so the cumulative cost of the surplus does not diverge relative to the cost of the evolved network.
10. Under gradual expansion, the evolvability is higher and the cost overhead is lower than under random expansion. The model of gradual expansion represents a more "evolution-friendly" dynamic environment than random expansion.

References

[1] R. Albert, H. Jeong, and A. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406:378, 2000.

[2] S. Bakhshi and C. Dovrolis. The price of evolution in incremental network design (the case of ring networks). In *Bio-Inspired Models of Networks, Information, and Computing Systems*, pages 1–15. Springer, 2012.

[3] A. R. Curtis, S. Keshav, and A. Lopez-Ortiz. Legup: Using heterogeneity to reduce the cost of data center network upgrades. In *Proceedings of the 6th International COnference*, Co-NEXT, 2010.

[4] N. Geary, A. Antonopoulos, E. Drakopoulos, and J. O'Reilly. Analysis Of Optimisation Issues In Multi-Period DWDM Network Planning. In *IEEE INFOCOM*, 2001.

[5] S. Gopal and K. Jain. On Network Augmentation. *IEEE Transactions on Reliability*, 35(5):541–543, 1986.

[6] M. Hahsler and K. Hornik. TSP Infrastructure for the Traveling Salesperson Problem. *IEEE/ACM Transactions on Networking*, 23(2), December 2007.

[7] L. Li, D. Alderson, W. Willinger, and J. Doyle. A First-Principles Approach to Understanding the Internet's Router-Level Topology. In *Proceedings of ACM SIGCOMM*, 2004.

[8] A. Meyerson, K. Munagala, and S. Plotkin. Designing Networks Incrementally. *IEEE FOCS*, 2001.

[9] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[10] M. Pickavet and P. Demeester. Long-term Planning of WDM Networks: A Comparison Between Single-Period and Multi-Period Techniques. *Photonic Network Communications*, 1(4):331–346, 1999.

[11] M. Pioro and D. Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks*. The Morgan Kaufmann Series in Networking, 2004.

[12] A. Singla, C. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, April 2012.

[13] A. Tero, S. Takagi, T. Saigusa, K. Ito, D. Bebber, M. Fricker, K. Yumiki, R. Kobayashi, and T. Nakagaki. Rules for Biologically Inspired Adaptive Network Design. *Science*, 327(5964):439–442, 2010.

[14] B. Yaged. Minimum Cost Routing for Dynamic Network Models. *Networks*, 3:193–224, 1973.

[15] N. Zadeh. On Building Minimum Cost Communication Networks Over Time. *Networks*, 4:19–34, 1974.