

# Automated Service Composition and Routing in Networks with Data-path Services

Xin Huang\*, Shashank Shanbhag<sup>†</sup>, Tilman Wolf<sup>†</sup>

<sup>†</sup>Department of Electrical and Computer Engineering  
University of Massachusetts, Amherst, MA 01002

Email: {sshanbha,wolf}@ecs.umass.edu

\* Deutsche Telekom R&D Lab, USA,  
Email: xin.huang@telekom.com

**Abstract**—Advanced packet processing in the data path of networks is common in the current Internet and is expanding to next-generation networks. Dynamic composition of such “network services” allows networks to establish connections with customized communication characteristics to satisfy various application requirements. In these networks, there are two problems that need to be solved: finding the optimal composition of services to satisfy the communication requirements of a given connection (“service composition”) and finding the optimal path for a given composition of services such that the required services are executed in sequence along the path (“service routing”). So far, these two problems have been studied separately. In our work, we combine both into a “service composition and routing” approach that can achieve better solutions than the current, separate approach. We present a novel decision making system that can determine the optimal or near-optimal solutions for the service composition and routing problem. We develop a synthetic benchmark for service requests to help comprehensively evaluate the performance of our systems. The results show that considering service composition and service routing jointly results in connections that have lower end-to-end delay while still meeting all service requirements.

## I. INTRODUCTION

The original design of the Internet requires only very simple packet forwarding functionality in the data path of routers. However, the broad use of the Internet as our society’s data communication infrastructure has broadened the functional requirements. As a result, more and more functionality was added into the network. Examples are Network Address Translation (NAT), queue management, deep packet inspection, Virtual Private Networks (VPN), etc. All these functionalities require advanced packet processing in the data plane.

The trend of adding advanced packet processing along the data path of the network continues in designs for the next-generation Internet. Virtualization and customization demand programmable packet processing as an essential part of the future Internet routers. To manage the complexities of customization in the data path, packet processing functions can be broken down into fundamental components that can be composed on demand. We refer to these fundamental packet processing blocks as “network services.” The network can compose a custom sequence of services during a connection setup. This sequence of processing services is then set up in the network and traffic sent along this connection is processed

accordingly.

There are two major technical challenges emerging in the context of networks with data-path services: service composition and service routing. Service composition refers to the problem of finding a (preferably optimal) composition of services to satisfy the communication requirements of a given request. Service routing refers to the problem of finding a (preferably optimal) path for a given composition of services such that the required services will be executed in order along the path. In a network with data path services, these problems cannot be considered independently since separate optimization may not lead to a global optimum. Instead, the overall problem is to find the optimal path for a given request, which combines both service composition and service routing. Thus, we need to consider a combined “service composition and routing” problem.

In this paper, we present a novel decision making system which can determine the optimal or near-optimal solution for the service composition and routing problem. To comprehensively evaluate the performance of the system, we develop a synthetic benchmark. Our specific contributions are:

- A framework for combining service composition and service routing in a single system.
- The design of a decision making system that can solve the service composition and routing problem.
- A synthetic benchmark that allows a comprehensive evaluation of the performance of different solutions for the service composition and routing problem.
- The results of an extensive evaluation of our solution for different network scenarios generated by our benchmark.

We believe that these contributions are an important step towards making data-path processing services an integral part of future networks.

The remainder of the paper is organized as follows. Section II discusses related works. Section III presents the main concepts and a formal definition of the problem. Section IV discusses the solution to the service composition and routing problem and describes decision making system. Section V presents a synthetic benchmark used to evaluate and compare the different solutions for the problem. Section VI discusses the experimental results to verify and evaluate our system. Section VII concludes this paper.

## II. RELATED WORK

Data-path packet processing in the network is not a new concept. There are many existing examples that have been implemented in the current Internet, such as NAT [1], VPN [2], RED [3], IDS(e.g., snort [4]), etc. The idea of putting advanced packet processing on the data plane of the networks have also been proposed as an inherent features of the next-generation Internet (e.g., Cisco Service-Oriented Network Architecture [5], SILO project [6] [7], and our service-centric network architecture [8]).

Composition of protocols and services has been studied in the context of the existing Internet as well as next-generation Internet. Configurable protocol stacks [9] and protocol heaps [10] have been proposed as a solution to statically compose novel protocol combinations. More dynamic approaches has been proposed in [8] and [7], where composition can be performed on a per-flow basis. The latter uses composition rules and constraints to determine valid compositions [11]. In our work, we attempt to determine valid compositions by input/output format characteristics of the data instead of explicitly enumerating all possible mutual constraints between pairs of protocol features.

Routing in networks with data-path services becomes more complicated than the shortest path problem since it need to consider both cost and service mapping. Choi et al. first developed a centralized algorithm to solve the service routing problem [12]. Two different centralized algorithms are developed later to consider resource constraints [13], [14]. A decentralized algorithm and the corresponding routing protocol is also designed to solve the service routing problem in large-scale networks [15].

In research related to automated service composition, various methods have been proposed in the area of web application composition. Most of these methods fall under the category of AI Planning [16]–[18], and Theorem Proving [19], [20]. In our previous work [21], we designed a system to solve the service composition problem in networks with data-path services. However, the paper only considers the service composition problem. In this paper, we concentrate on service composition and routing as one unit.

As far as we know, this paper is the first work to consider the service routing and service composition jointly and to solve the problem using a planner based decision-making system.

## III. NETWORK SERVICES AND PROBLEM DEFINITION

Before discussing our decision making system, we first illustrate the concept of network services and then give a formal definition of the service composition and routing problem in the context of next-generation Internet.

### A. Network Services

As defined in our previous work [8], “network service” refers to any type of advanced packet processing on the data-path of the network and the related control-plane processing.

Figure 1 illustrates the concept of networks with data-path services in the context of next-generation Internet. Network

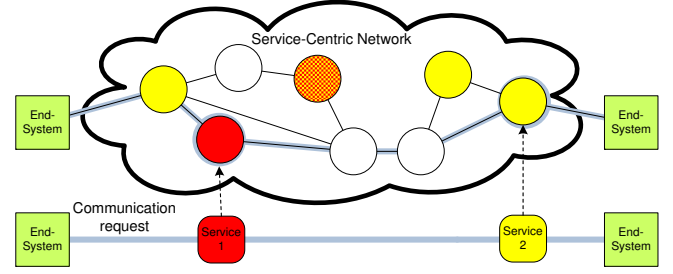


Fig. 1. Next-Generation Network with Data-Path Services.

services are deployed and enabled on nodes that are distributed in the data-plane of the network. The nodes capable of providing services are called “service nodes.” Different service nodes may provide different types of services (as indicated with colors in Figure 1).

A connection that uses this network provides a service request. By specifying different service requests, connections can customize network functionality to satisfy various application requirements.

### B. Control Plane Challenges

The control plane of a network that provides services needs to support the setup of custom service requests. This setup involves several decisions:

- Choice of services: The service request may not fully specify all necessary services. Instead, it may only list the characteristics of traffic at the source and destination. The network then needs to decide which services to instantiate to achieve this translation.
- Choice of service nodes: Since there are typically multiple nodes that can perform the same service, the network needs to select the node that performs service processing for a particular connection.
- Choice of network links: To connect the end-nodes and the nodes that perform service processing, the network needs to set up routes.

This set of decisions leads to the overall problem that we address in this paper:

*How can we achieve optimal or near-optimal setup of connections that involve data-path services?*

Achieving optimality requires consideration of all aspects of the problem listed above.

We make the following assumptions in our work:

- We assume that the network supports the establishment of connections with fixed paths. This is required since traffic needs to traverse a particular set of service nodes as determined by our solution.
- We assume that costs of communication and service processing can be represented using a single metric (e.g., delay).

To illustrate this problem, consider Figure 2. End-system A sends a communication request with communication requirements of sending a video stream in divx format to end-system B which only supports flv format. The end-system

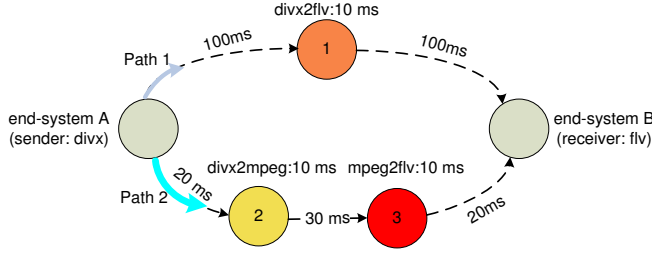


Fig. 2. An Example for the Service Composition and Routing.

sends a connection request to the network, which specifies all the communication requirements:  $A:video:divx \rightarrow B:video:flv$ . The network determines a service composition satisfying the communication requirements, determines the least-cost path for the composition, and sets it up for the connection. The data is then transferred along the connection until it is torn down.

Multiple service compositions and paths may satisfy the same communication requirements. The goal of our work is to find the optimal service composition which will lead to minimum routing cost (i.e., lowest end-to-end delay, which includes both link delay and service processing delay). This is called the “*service composition and routing problem*.” We provide a formal definition of the problem below and present our solution in Section IV.

### C. Motivation for Combined Service Composition and Routing

Before turning to our solution, we present a brief example illustrating why a combination of composition and routing can yield better results than considering these problems separately. An intuitive solution to the problem based on published work is to divide service composition and service routing into two subproblems and solve each separately. The drawback of this approach can be illustrated by the simple example in Figure 2.

The figure shows a network with data-path services. The network provides video transcoding services. The network services provided in this network are video transcoding functions. For example, node1 provides video transcoding from *divx* format to *flv* format, with the processing cost of 10 ms. Node A wants to send a video stream in *divx* format to node B. Node B, however, only supports *flv* video. There are two different paths that can accommodate the service request:

**Path1:** sender A  $\rightarrow$  node1  $\rightarrow$  receiver B, with transcoding service *divx2flv* executed on node1. This solution uses only one service step along the path. The total cost for it is 210ms.

**Path2:** sender A  $\rightarrow$  node2  $\rightarrow$  node3  $\rightarrow$  receiver B, with service *divx2mpeg* executed on node2 and service *mpeg2flv* executed on node3. This solution uses two services steps for the video transcoding but with a total cost of 90ms.

When using a divided approach, we obtain these differing solutions: **Separate Service Composition and Routing (SSCR):** This approach divides the problem into two subproblems and solves them using the following two steps: 1) Solve the service composition problem: convert video stream of *divx* into *flv* format with the minimum cost (service steps). The optimal solution to this problem has one service step: *divx2flv*.

2) Use the output of service composition as the input to service routing, and find the shortest path (least-cost in delay) for this service composition. The output for this step is Path1. **Combined Service Composition and Routing (CSCR):** This approach solves the service composition and routing problem as one problem: find the shortest path (least-cost in delay) from node A to node B, while converting video format from *divx* into *flv* along the path. By considering both Path1 and Path2 and comparing their cost, the output for this solution should be Path2.

In this simple example, the combined solution finds the optimal solution of the service composition and routing problem while the other approach does not. This indicates that it is important to solve this problem as a combined problem since the optimal solution to the subproblems may not lead to the overall optimal solution.

### D. Service Composition and Routing Problem

The service composition and routing problem can be stated as: Given a network, a set of network services, their availability and distribution in the network and a description of communication requirements, find a service composition and the corresponding path that achieves the communication requirements with the minimum cost. In this paper, cost is measured in terms of link delay and service processing delay. The formal definition for the service composition and routing problem is as follows:

The network is represented by a weighted graph,  $G = (V, E)$ , where nodes  $V$  correspond to routers and end systems and edges  $E$  correspond to links. Each edge  $e_{i,j}$  that connects nodes  $v_i$  and  $v_j$  is labeled with a weight  $w_{i,j}$  that represents the communication cost (e.g., delay). Assume  $S = \{S_1, S_2, \dots, S_n\}$  is the set of services available in the network,  $G$ . Each node  $v_m$  is labeled with the set of services that it can perform  $u_m = \{S_k | \text{service } S_k \text{ is available on } v_m\}$  and the service processing cost  $c_{m,k}$  (e.g., processing delay) of each service. A *connection request* is represented as  $R = (s, t, C)$ , where  $s$  is the source node,  $t$  is the destination node, and  $C$  is the specification of communication requirements. As described later in Section IV-B, the communication requirements will be represented as a pair of data formats at the source and the destination.

Given a network  $G$ , the service set  $S$ , and a request  $R$ , we need to find a path for the request such that the source and destination nodes are connected and the communication requirements are satisfied. The path is defined as  $P = (E^P, M^P)$  with a sequence of edges,  $E^P$ , and services mapped to processing nodes,  $M^P$ :  $P = ((e_{s,v_{i_1}}, \dots, e_{v_{i_l},t}), (S_{k_1} \rightarrow v_{j_1}, \dots, S_{k_l} \rightarrow v_{j_l}))$ , such that  $S_{k_x} \in S$  for  $1 \leq x \leq l$  and  $R \Leftrightarrow (S_{k_1} \gg S_{k_2} \gg \dots \gg S_{k_l})$  which means that  $S_{k_1}$  has to be applied before  $S_{k_2}$  and so on. To determine the quality of a path, we define the total cost  $C(P)$  of accommodating the connection request,  $R$  as the sum of link cost and service processing cost:  $C(P) = \left( \sum_{\{(i,j) | e_{i,j} \in E^P\}} w_{i,j} \right) + \left( \sum_{\{(k_x, m_x) | S_{k_x} \rightarrow v_{m_x} \in M^P\}} c_{m_x, k_x} \right)$ . In many cases, it is de-

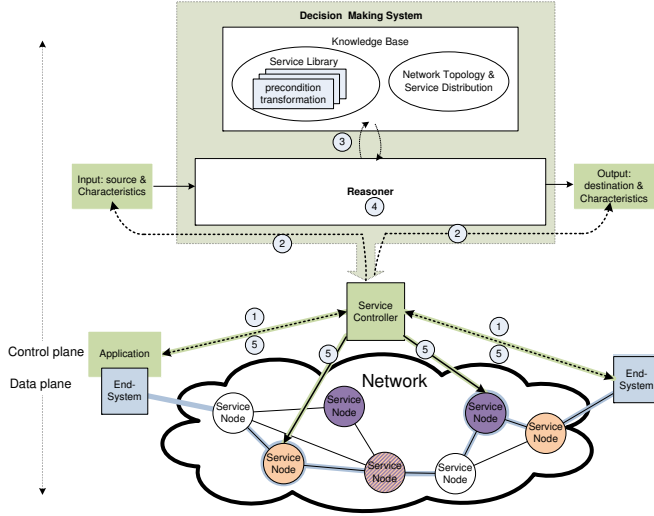


Fig. 3. Decision Making System.

sirable to find the *optimal* connection setup. We view this optimality in terms of the least-cost path of a single connection request.

#### IV. DECISION MAKING SYSTEM

We introduce a novel decision making system to solve the service composition and routing problem. First, we will give an overview of the system design. Then, we will elaborate the design of the important components.

##### A. Design Overview

Solving the service composition and routing problem in a robust and distributed manner requires the dependency relationship between services. Instead, we focus on a systematic description of the input and output characteristics of the services. This helps deduce the composition dependencies between services.

The system (Figure 3) is composed of: the knowledge base and the reasoner. The knowledge base contains the descriptions about the network topology, service availability and distribution, and the service characteristics. The reasoner is a piece of software that can help figure out the optimal service composition and routing according to the input and output requirements and the information in the knowledge base.

The whole framework can be illustrated by the following scenario: Assume the server of an online IPTV service is transferring a HDTV-1080p movie to an end-system which only supports and displays H.264 (176X208) videos. The applications send the connection requests (and input format description) to the service controller at the source. The server contacts the destination and determines the output format of the connection (Step 1). In Step 2, the input/output descriptions are passed to reasoner. In our scenario, the input characteristics is: HDTV 1080p video, and the output characteristics is: encrypted H.264 (176X208) video. The reasoner consults the knowledge base (step 3) and decides the composition and routing for the connection (step 4). The selected path is marked

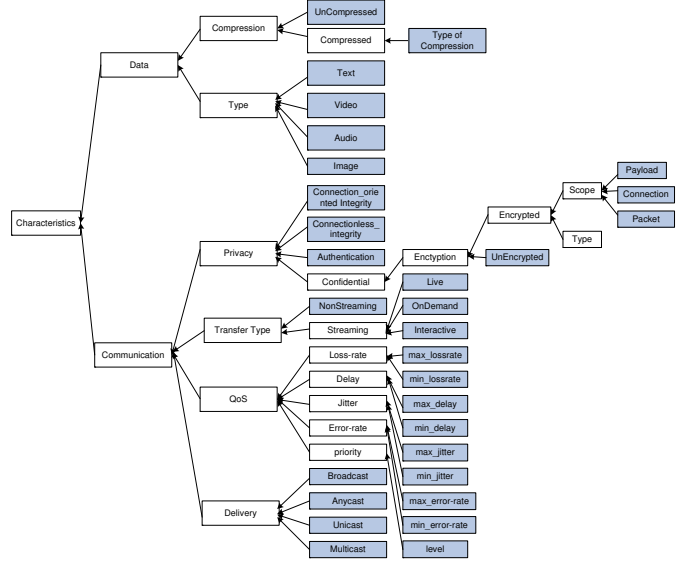


Fig. 4. Semantics for Representing Characteristics.

in blue, and two services, transcoding (marked in orange) and encryption (marked in purple) are executed along the path. At last (step 5), the service controller contacts the nodes along the path and sets up the whole path for the connection.

##### B. Data and Communication Characteristics

One issue that occurs in the design of the automated policy enforcement system is: what characteristics are important enough to be examined when describing the state a policy or a service is mapped to? After an extensive study of the services, applications, and communication paradigms in the current Internet, we designed a semantic tree structure (Figure 4) to represent state in terms of data and communication characteristics. The tree structure includes class hierarchies inherent in data and communication characteristics and forms a comprehensive method of representing state. Using this representation, a wide range of policies can be defined and implemented. For example, an encryption policy will need the traffic to be unencrypted and of type "Data." Various semantic web markup languages are available that simplify the task of describing the semantic tree structure since these languages use a similar structure of classes, subclasses, individuals and the relationships between them. Therefore, the semantic tree structure is a general and comprehensive representation of state with respect to network traffic.

##### C. Combined Service Composition and Routing

Separate service composition and routing (SSCR) (Figure 5) breaks down the overall problem into two parts: service composition and service routing. First, according to the services available in the network, a planner is used to find the optimal service composition for the request (i.e., a sequence of services that can satisfy the communication requirements with the minimum number of steps) [21]. Then, the optimal routing (a path which can traverse the required services in order with the minimum delay) is calculated using the service routing algorithm presented in [15].



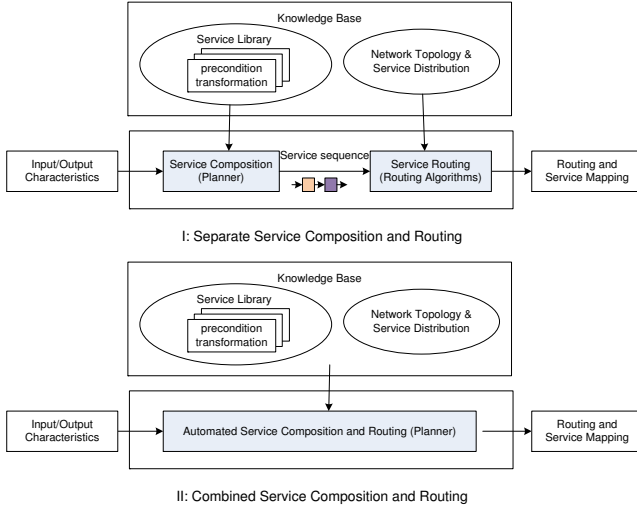


Fig. 5. SSCR vs. CSCR

A fundamental limitation of the SSCR is that it handles the two subproblems separately - *Separate optimization may not lead to a global optimum*. To address this, we present a solution that can figure out the optimal path for the service composition and routing problem by considering the two problems as a whole. CSCR (Figure 5) takes everything into consideration, including the input/output characteristics, service library, network topology, and service distribution, and combines all these factors to calculate the routing.

Our framework design enables us to reduce the service composition and routing problem to a planning problem. Both services and links can be regarded as **actions** and can be fully described by:

- **Preconditions:** logical conditions that need to be satisfied in order for the actions to be executed. For example, a video transcoding service that transcodes from MPEG to H.264 can only be executed when the packet is in MPEG format and it is at a node capable of providing the service.
- **Transformation:** The function it performs and the transformation it causes. For example, it may refer to a link transferring packets from one node to another, a service causing changes in the data and communication formats, or an increase in the delay.

In general, the service composition and routing problem can be represented as the following planning problem: Given a set of actions  $A$ , a set of all possible states  $S$ , an initial state  $I \subset S$ , and a goal state  $G \subset S$  that the planner attempts to achieve, the task is to achieve the goal state  $G$  from the initial state  $I$  by performing a series of actions. In the context of networks with data-path services,  $A$  is the set of network services and links.  $S$  is equivalent to the set of all possible states combining data semantics and the packet location.  $I$  is equivalent to the initial packet location and its semantics.  $G$  is the targeted destination and semantics. The output is a series of actions (links and services) that form an optimal route for the service composition and routing problem.

The planner used in the proposed automated service com-

position and routing system is LPG [22], a fully automated domain-independent planner. It is based on local search and planning graphs and can solve both plan generation and plan adaptation problems. LPG uses the PDDL (Planning Domain Definition Language) for describing domains and problems. PDDL is an action-centric planning domain and problem description language with a LISP like syntax that simplifies the description of the semantics of actions, their preconditions and postconditions.

## V. SYNTHETIC SERVICE BENCHMARK

We sense that the next-generation networking architectures will make use of at least some, if not all of the features proposed in this paper. Algorithms and techniques for automated service composition need a benchmark test environment which is representative, flexible and provides a standard for validation and testing new and existing composition algorithms.

To the best of our knowledge, there does not exist any benchmarking tools for automated service composition systems in the networking domain. To address this need, we describe briefly a Synthetic Service benchmark that makes use of our semantic tree representation (Refer Fig. 4) of various data and communication requirement characteristics.

### A. Synthetic Service Graph

Let  $S$  be a set of atomic and composite services available in the system with every service represented by  $S_j$ , their preconditions,  $Pr_j$  and postconditions,  $Po_j$ . Composite services are single services that can also be represented by combining multiple atomic services. Both  $Pr_j$  and  $Po_j$  are represented by a tuple of  $n$  dimensions,  $(p_{n-1}, \dots, p_1, p_0)$  representing the leaf nodes in the semantic tree representation (Refer Fig. 4) and each  $p_i$  where,  $n-1 \geq i \geq 0$  can take the set of values  $V_i$  for that dimension. The number of dimensions  $n$ , define the service graph, with each node of the graph representing a tuple and the edges representing the services that change one or more dimensions in the tuple. The connectivity of the determines the relationship between the services.

Let  $H_{N,E}$  be the service graph for  $n$  dimensions where  $N$  and  $E$  are the number of nodes and edges which also depend on the set of values  $V_i$  that each dimension  $i$  can take. For example,  $n$  dimensions where each dimension takes on binary values results in a service graph with  $2^n$  nodes. The values are not limited to binary values and can take on continuous values as well.  $E$  is essentially decided by two parameters: the connectivity model and the hamming parameter. The hamming parameter decides the number of nodes of the service graph required to be traversed to reach the destination from the source. For example, a hamming parameter of  $h = 2$  implies that any combination of source and destination nodes in the service graph can be reached by either using a single composite service or a combination of two atomic services.

We define three connectivity models for our benchmark: **Full Connectivity:** service graphs are essentially small-world

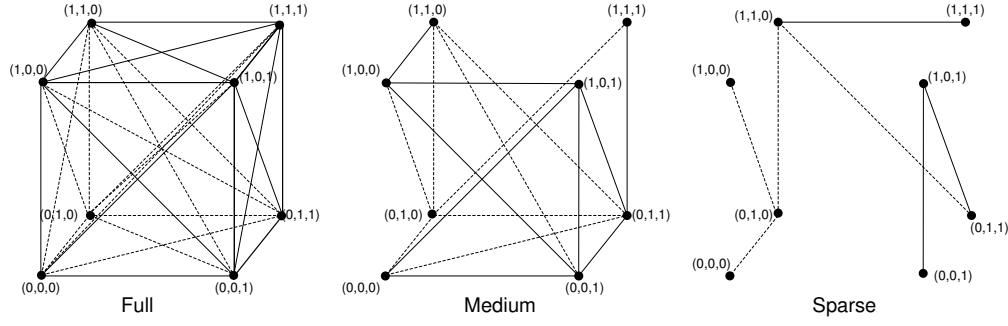


Fig. 6. Synthetic Service Graphs. ( $n=3$ ,  $h=1$ ,  $r=0.56$ )

networks characterized by high connectivity, where the network diameter specifies the longest shortest distance between the nodes. Thus, the destination node in the graph can be reached using a small number of services, either composite or atomic. **Medium Connectivity:** service graphs are an example of random networks of  $N$  nodes, where each pair of nodes are connected with a probability,  $p$ . We define a parameter,  $r$ , which decides the number of edges,  $E_r \subset E$  that can be included in the service graph. However, the edges are added such that the service graph is connected. **Sparse Connectivity:** service graphs are similar to Medium Connectivity service graphs. The difference being the presence of services (edges) in the service graph, the removal of which renders the graph disconnected. This feature represents tuple dimensions that can only be achieved by one service.

Figure 6 shows the three connectivity models for  $n = 3$  dimensions. The hamming parameter chosen for Full connectivity service graph is  $h = 1$ . For the medium connectivity service graph, the  $r$  parameter takes the value 0.56.

## VI. EVALUATION

In this section, we evaluate the performance of CSCR under different synthetic and practical network scenarios, and compare the results with those of SSCR. The simulation was carried on a 64-bit Quad-Core Intel Xeon 5300 with 2x4MB shared cache and 2GB memory.

### A. Synthetic Benchmark Scenarios

All of our synthetic scenarios are based on a network topology with 96 service nodes organized into 12 ASs such that each AS has 8 nodes. We simulate the decision making system using both CSCR and SSCR, and compare them under three different service scenarios. As shown in Figure 7, in the three synthetic scenarios, data and communication characteristics are presented in 6-dimensions with each dimension taking on binary values. This can also be designed to include continuous values. The full connectivity service graph represents networks with numerous redundant services, while the sparse connectivity graph is representative of networks with just enough services to satisfy the communication requirements.

For each scenario, network services defined by the synthetic benchmark are randomly distributed inside the network. And a total of 1000 connection requests are generated randomly. Two versions of the decision making system are implemented:

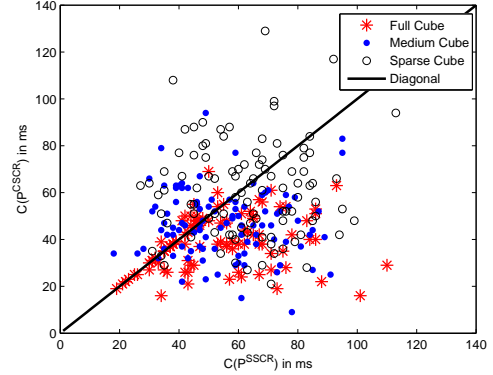


Fig. 8. Performance: CSCR vs. SSCR.

CSCR and SSCR. The same flow requests are sent to the two systems and total cost of their final routing decisions are compared. In our experiments, the total cost includes both link delays and service processing delays in milliseconds.

Figure 8 shows the path cost for SSCR  $C(P^{SSCR})$  on the x-axis and the path cost for CSCR  $C(P^{CSCR})$  on the y-axis. Each data point represents one connection request. Data points below the diagonal for a connection implies that CSCR has a better performance compared to SSCR. In case of the Full Connectivity Synthetic scenario, CSCR performs better than SSCR most of the time. However, in the Sparse Connectivity Synthetic scenario, CSCR does no better than SSCR. The reason for the disparity in performance is due to the nature of the synthetic graphs themselves. Full connectivity implies large number of redundant services and multiple routes through the network, which translates to an increase in the probability of LPG finding the optimal or a near-optimal path.

The same trend can be seen more clearly from Figure 9, which represents the cumulative distribution function(CDF) for the relative cost. The relative cost here is defined as the cost of the best path found by SSCR to that of CSCR. When the relative cost is larger than 1, it represents the situation where CSCR finds a lower cost path. The larger the value of  $x$ , the better the performance of CSCR. Figure 9 also compares the performance of the two techniques. For the full connectivity benchmark scenario, CSCR outperforms SSCR. However, in case of sparse connectivity, for about half of the cases, SSCR does better than CSCR.

There is a fair amount of disparity in the plan generation

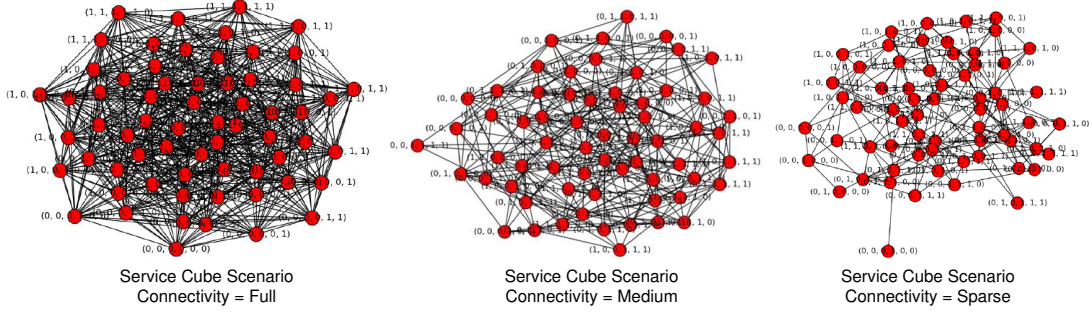


Fig. 7. Synthetic Benchmark Scenarios. ( $n=6$ ,  $h=2$ ,  $r=0.5$ ,  $|V_i|=2$ )

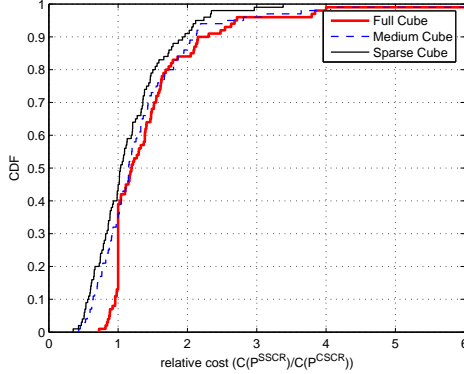


Fig. 9. CDF: CSCR vs. SSCR

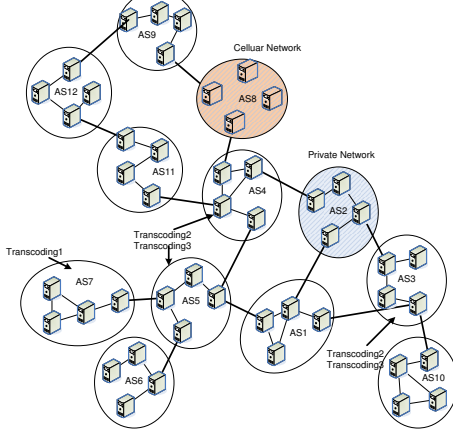


Fig. 10. Practical Network Scenario.

times between CSCR and SSCR. For the experiments, the average plan generation times for CSCR were 4.75ms, 2.00ms and 1.35ms for full, medium and sparse service graphs respectively. SSCR took 0.14ms, 0.09ms and 0.07ms for full, medium and sparse service graphs respectively. This disparity is primarily because CSCR has to account for both the network topology and service cube graphs while generating the plan. The planner in the SSCR only needs to account for the service cube graph while the routing algorithm takes care of the service placement and routing.

### B. Practical Scenarios

In this section, we further illustrate with practical scenarios the benefits of CSCR over SSCR.

The topology of the network is shown in Figure 10, and consists of 48 nodes within 12 ASs. Each AS has 4 service nodes. Among all the ASs, AS2 is a private network, where every connection through is automatically tunneled through a private security channel (i.e., encryption and decryption services will be enforced on the packet traveling through it). AS8 is a cellular network with bandwidth restrictions and therefore all packets through AS8 has to be compressed and later decompressed. The decision making systems take into consideration these policies while formulating the least cost path. Three transcoding services, S1, S2 and S3 have been implemented in the network. S1 converts a video from format X to Z. S2 converts a video from format X to Y. S3 converts a video from format Y to Z. Therefore, the decision making system, given the source format X and destination format Z, can choose between S1 as a single service to meet the requirement or combine S2 and S3 to meet the same. Also, S2 and S3 are redundant all through the network (e.g., in AS3, AS4, AS5, etc.). However, S1 is implemented in a pool of servers limited to AS7. We generated two types of requests connecting every possible source and destination pair in the network: 1) Source node to destination node requests without any data-path services. 2) Source node to destination node requests with transcoding from video format X to format Z.

Figure 11 compares the performance of CSCR with that of SSCR. Observe that CSCR consistently outperforms SSCR in most of the cases. This is observed in the CDF plots of Figure 12 as well. CSCR outperforms SSCR for 80% of the requests for requests that require transcoding. This is because the service composition stage does not account for delays while generating the service sequence but only the number of steps needed to achieve the goal resulting in a huge detour later in the routing stage. CSCR, on the contrary, takes into account both link delays and service processing delays and thus outperforms SSCR.

## VII. CONCLUSION

Service composition and routing is an important problem in networks with data-path services. However, previous work in the literature usually divides the problem and solves the sub-problems separately. In this paper, we presented a decision making system which can consider the problem as a whole and automate the process of finding the optimal or near-optimal path for the service composition and routing prob-

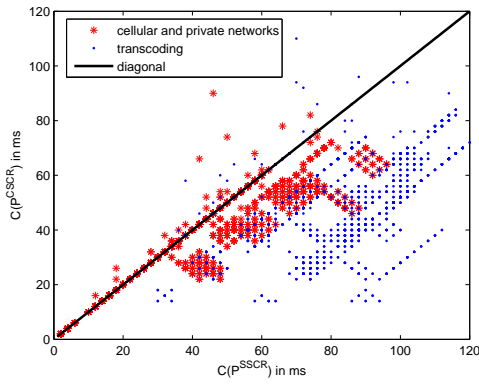


Fig. 11. Performance: CSCR vs. SSCR.

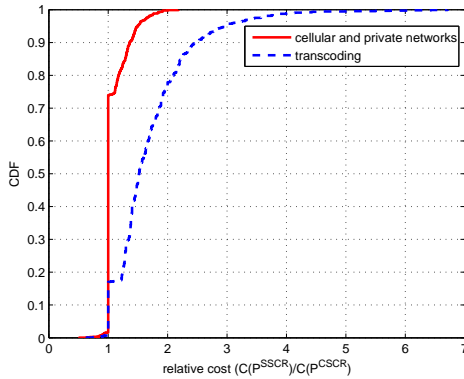


Fig. 12. CDF: CSCR vs. SSCR.

lem. We implemented our framework design based on the LPG planner and the PDDL language. We also presented a synthetic benchmark useful in evaluating service composition algorithms. Through the simulations, we evaluated our system design under both different synthetic network scenarios as well as a practical network scenario. The results shows that the Combined Service Composition and Routing works better than the Separate Service Composition and Routing in most of the network scenarios, and has the potential of obtaining much better service mapping and routing decisions. We believe that this approach is an important step towards make network architectures with advanced data-path services a practical reality.

#### ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0626690.

#### REFERENCES

- [1] K. B. Egevang and P. Francis, "The IP network address translator (NAT)," Network Working Group, RFC 1631, May 1994.
- [2] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and M. A., "A framework for IP based virtual private networks," Network Working Group, RFC 2764, Feb. 2000.
- [3] S. Floyd and V. Jacobson, "Random early detection (RED) gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
- [4] *The Open Source Network Intrusion Detection System, Snort*, 2004, <http://www.snort.org>.

- [5] *Cisco Unified Communications and Service-Oriented Network Architecture*, Cisco Systems, Inc., Jul. 2007.
- [6] I. Baldine, M. Vellala, A. Wang, G. Rouskas, R. Dutta, and D. Stevenson, "A unified software architecture to enable cross-layer design in the future internet," in *Proc. of Sixteenth IEEE International Conference on Computer Communications and Networks (ICCCN)*, Honolulu, HI, Aug. 2007.
- [7] G. N. R. Rudra Dutta, I. Baldine, A. Bragg, and D. Stevenson, "The SILO architecture for services integration, control, and optimization for the future internet," in *Proc. of IEEE International Conference on Communications (ICC)*, Glasgow, Scotland, Jun. 2007, pp. 1899–1904.
- [8] T. Wolf, "Service-centric end-to-end abstractions in next-generation networks," in *Proc. of Fifteenth IEEE International Conference on Computer Communications and Networks (ICCCN)*, Arlington, VA, Oct. 2006, pp. 79–86.
- [9] N. T. Bhatti and R. D. Schlichting, "A system for constructing configurable high-level protocols," in *SIGCOMM '95: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, Cambridge, MA, Aug. 1995, pp. 138–150.
- [10] R. Braden, T. Faber, and M. Handley, "From protocol stack to protocol heap: role-based architecture," *SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 17–22, Jan. 2003.
- [11] M. Vellala, A. Wang, G. N. Rouskas, R. Dutta, I. Baldine, and D. Stevenson, "A composition algorithm for the SILO cross-layer optimization service architecture," in *In Proc. of the Advanced Networks and Telecommunications Systems Conference (ANTS)*, Mumbai, India, Dec. 2007.
- [12] S. Y. Choi, J. S. Turner, and T. Wolf, "Configuring sessions in programmable networks," *Computer Networks*, vol. 41, no. 2, pp. 269–284, Feb. 2003.
- [13] S. Y. Choi and J. S. Turner, "Configuring sessions in programmable networks with capacity constraints," in *Proc. of IEEE International Conference on Communications (ICC)*, Anchorage, AK, May 2003.
- [14] X. Huang, S. Ganapathy, and T. Wolf, "Evaluating algorithms for composable service placement in computer networks," in *Proc. of IEEE International Conference on Communications (ICC)*, Dresden, Germany, June 2009.
- [15] —, "A scalable distributed routing protocol for networks with data-path services," in *Proc. of 16th IEEE International Conference on Network Protocols (ICNP)*, Orlando, FL, Oct. 2008.
- [16] S. McIlraith and T. Son, "Adapting golog for composition of semantic web services," in *Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002)*, Toulouse, France, April 22–25 2002, pp. 482–493.
- [17] S. McIlraith, T. Son, and H. Zeng, "Semantic web services," *IEEE Intelligent Systems. Special Issue on the Semantic Web*, vol. 16, no. 2, pp. 46–53, March/April 2001.
- [18] S. Narayanan and S. McIlraith, "Simulation, verification and automated composition of web services," in *Proceedings of the Eleventh International World Wide Web Conference (WWW-11)*, Honolulu, Hawaii, USA, May 7–11 2002, pp. 77–88.
- [19] J. Rao and P. Kungas, "Application of linear logic to web service composition," in *In The First International Conference on Web Services, Las Vegas*. CSREA Press, 2003, pp. 3–9.
- [20] —, "Logic-based web services composition: From service description to process model," in *In 2nd Intl. Conference on Web Services*. IEEE, 2004, pp. 446–453.
- [21] S. Shanbhag, X. Huang, S. Proddatoori, and T. Wolf, "Automated service composition in next-generation networks," in *Proc. of the International Workshop on Next Generation Network Architecture (NGNA)*, Montreal, Quebec, Canada, June 2009.
- [22] A. Gerevini and I. Serina, "Lpg: a planner based on local search for planning graphs," in *Proc. of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS)*, Toulouse, France, 2002.