Migration of scaled-down topologies

G. Kesidis
CS&E and EE Depts
The Pennsylvania State University
kesidis@engr.psu.edu
June 1, 2010

I. INTRODUCTION

Topological modeling is a mature element of distributed or centralized simulation or emulation of large-scale communication networks [10], [6], [12]. These models may be informed by topological snapshots such as publicly available BGP routeview data. Assuming limited resources, topological scale-down techniques have been explored [9], [7], [8], [1].

Suppose a large physical topology $G = (V, E)$ of a communication network is fixed, where $V$ is the vertex set and $E$ is the edge set. For example, $K \times K$ router will account for either $K$ (output buffer model) or $2K$ vertices in $V$. A dynamic group of unidirectional flows, $F$, each specified by

- an egress and ingress vertex of $V$, and
- a model or replayed trace of packet transmission into the ingress vertex.

A routing algorithm maps each flow to a cycle-free path through $G$ from its ingress to egress. For any given flow-group $F$, we can identify a subset $V_F \subset V$ of congested nodes, where $|V_F| \leq |V|$, i.e., $V_F$ is determined by the routing algorithm and the characteristics of the flows in $F$.

More precisely, $|V_F|$ corresponds to the maximum-sized graph that can be simulated or emulated using the available resources. Each simulated vertex could be a multiplexer of ingress packet flows, a first-in-first-out queue/buffer, the queue’s egress server, and finally flow demultiplexer. So, for each $F$, only the vertex set $V_F$ is simulated/emulated with direct-link set $E_F$ connecting its nodes, consistent with the routes currently chosen for $F$ through $G$. Note that $(V_F, E_F)$ is not exactly a subgraph of $G$ because the edges of $E_F$ would possibly correspond to overlapping cycle-free paths in $G$ (groups of edges of $E$). Let $G_F = (V_F, E_F)$ be the resulting scaled down graph of $G$. Note that only flows in $F$ that interact with at least one element of $G_F$ are actually simulated. The packets of flows that do not use any part of $G_F$ are assumed to experience no loss and constant (propagation) delays.

II. CHANGING ROUTES AND ITS IMPACT ON THE SCALED-DOWN TOPOLOGY

Certain network phenomena of interest caused significant route churn/thrashing, e.g., denial-of-service floods and the aggressive scanning activity of certain worms such as SQL Slammer [4]. So, we’re interested in how the routes may change for a single set of flows, or as the flows arrive and terminate. Given a fixed set of active flows using scaled-down graph $G_1 = (V_1, E_1)$, suppose we wish to migrate the simulation to different graph $G_2 = (V_2, E_2)$, as directed by the routing algorithm and the topological scale-down method in use, e.g., that of Psounis et al. or simply identifying the vertices with maximal traffic “intensity” (mean aggregate arrival rate divided by mean service rate).

At the point of topological transition, assume that the flow set $F$ is fixed. This said, it may be likely that changes in the active flow set $F$ occurred prior to this point, resulting in flow redistribution by the routing algorithm, and thus requiring the topological transition under consideration.

To implement this change, we need to encode two basic operations

This work was supported by the National Science Foundation under grant 0831068.
• remove vertex, and
• create vertex.
These operations are performed until $G_1 \rightarrow G_2$.

### A. Remove vertex

When a vertex (node) $v \in V_1 \setminus V_2$ is removed, all of the packets currently resident in its queue become “in flight” packets and suffer no further queueing delay at $v$. In an event-driving simulation, these changes would occur in the packet records stored in the event queue, in addition to new arrival times at vertices immediately downstream from $v$ depending on the propagation delays of the newly added edges. This operation is performed in an order-preserving manner as if the service rate of the queue had become infinite.

The edges in $E_1$ that connect to $v$ are also removed (i.e., these edges are not in $E_2$). For each flow $f$ that used $v$, its node immediately upstream and immediately downstream are now directly connected by an edge in the scaled-down topology. These new edges may become members of $E_2$ when the process of topological change $G_1 \rightarrow G_2$ is completed.

### B. Create vertex

When a vertex $v \in V_2 \setminus V_1$ is created, for each flow $f$ that uses it, the vertices immediately upstream $u \in V_2$ and immediately downstream $d \in V_2$ are now directly connected to $v$ by an edge. Moreover, edges directly connecting $d$ and $u$ need to be removed, unless there are flows in $F$ that use them but not $v$. Note also the possible need to change the destination vertex (and corresponding arrival times) of in flight packets in the event queue on these affected edges.

### III. Fluid simulation considerations

If the system employs fluid simulation methods, e.g., [5], [3], then the insertion of a vertex may simply cause the fluid traveling in the affected edges to begin to accumulate in the vertex’s queue. That is, accumulate if its service capacity is lower than the instantaneous arrival rate at the point of topological transition, where we note the service capacity of the queue is just the bandwidth of the egress edge of the vertex. When removing a queue, a temporary violation in the transmission capacity of the affected edges may occur as a result of instantly “flushing out” the queue’s contents.

### IV. Determining the most congested links in simulation

#### A. Network set-up

Consider a group of $N \geq 2$ lossless, single-server, work-conserving queuing stations. Queuing stations have a mean required service time of $1/\mu_n$ for all $n \in \{1, 2, ..., N\}$. The packet arrival process to the $n^{th}$ station is a superposition of $N + 1$ component arrival processes, and each of these consist of a number ($\geq 0$) of currently active packet-flows. Let $r_{m,n}$ be the average fraction of packets (or bytes) departing the $m^{th}$ station and routed to and immediately arrive at the $n^{th}$ station. Also, let $r_{m,0}$ be the average fraction of packets (or bytes) departing station $m$ to leave the queuing network forever; here we use station index 0 to denote the world outside the network. Clearly, for all $m$,

$$\sum_{n=0}^{N} r_{m,n} = 1.$$  

These variables $r$ can be identified by the current routes and transmission rates (packets/s or bytes/s) of the individual packet flows presently in the system.

Packets from the outside world arrive to the $n^{th}$ station at rate $\Lambda_n$. It is these interactions with the outside world that make the network open.
B. Flow balance equations

Let \( \lambda_n \) be the total arrival rate to the \( n \)th station. These quantities can be found by solving the so-called flow balance equations which are based on the notion of conservation of flow and require that all queues are stable, i.e., \( \mu_n > \lambda_n \) for all \( n \): for all \( n \in \{1, 2, \ldots, N\} \),

\[
\lambda_n = \Lambda_n + \sum_{m=1}^{N} \lambda_m r_{m,n}.
\]  

(1)

Note that the flow balance equations can be written in matrix form:

\[
\Delta^T = \Lambda^T + \Delta^T R
\]

(2)

\[
\Rightarrow \Delta^T (I - R) = \Lambda^T,
\]

(3)

where the \( N \times N \) matrix \( R \) has entry \( r_{m,n} \) in the \( m \)th row and \( n \)th column. Thus, the flow balance equations can be uniquely solved if \( \det(I - R) \neq 0 \), i.e., \( I - R \) is invertible so that

\[
\Delta^T = \Delta^T (I - R)^{-1}.
\]  

(4)

Again, we are assuming that \( \Delta < \mu \). This also requires that \( r_{m,0} > 0 \) for some station \( m \) (i.e., packets can exit the network). Otherwise, \( \overline{R} \) would be a stochastic matrix (all entries nonnegative and all rows sum to 1) so that 1 is an eigenvalue of \( R \) and, therefore, 0 is an eigenvalue of \( I - R \), i.e., \( I - R \) is not invertible [2].

C. Large-scale flow-balance

Suppose we can simulate only \( M = |V| \ll N \) queues in detail. Periodically, or when flow traffic substantially changes (due to packet flow arrivals, departures, or reroutes), our objective is to find the subset of size \( M \) of queues \( i \) which have maximal traffic intensity \( \lambda_i/\mu_i \).

There are a variety of numerical techniques that can be used, e.g., iterative methods based on (2) (i.e., for integer index \( k \), \( \Delta^T(k+1) = \Delta^T + \Delta^T(k)R \), with convergence performance depending on the largest-magnitude eigenvalue of sub-stochastic matrix \( R \)) or techniques for sparse matrices \( R \) [11].

REFERENCES