Dynamic Routing on Networks with Fixed-Size Buffers

William Aiello	AT&T Labs
Eyal Kushilevitz	Technion
Rafail Ostrovsky	Telcordia Tech.
Adi Rosén	Technion

Simple Packet Network Model

- directed graph $G = (V, E)$, $|V| = n$, $|E| = m$;
- d - longest simple path
- synchronous model
- unit capacity edges – 1 packet / time step $\times$ edge
- a packet (s, t, π)–injected at s and destined for t,
- and routed along the (possibly implicitly) prescribed paths (π)
- "store and forward" routing – buffers at the tail of edges

Protocol Analysis

Ideally, protocols are:

- online
- local control

Goal: analyze and compare efficacy of various protocols on various topologies

Previous Approaches–Stability Analysis

Network Model

- *A priori*, buffers are assumed to be infinitely large
- No packets are dropped

Protocols

- **Scheduling algorithm:** Method for deciding whether to forward a packet from an output buffer, and if so, which packet
- **Greedy scheduling:** Always send a packet if there is one to send. If there is more than one, decide which one to send

Stability Analysis–cont'd

Traffic Model

- **Queuing Theory**: Random arrival process (typically poison) and random destination (typically uniform)
- **Adversarial Queuing Theory**: Abritrary traffic with the following restiction.

> For each edge and during each inteval of time, the number of packets injected during that time interval (that pass through that edge) cannot exceed a certain bound proprtional to the size of that time interval.

Metric

- Queue size: Maximum number of packets ever in a buffer
- Queuing Theory: Expected Queue Size
- Adversarial Queuing Theory: Worst-Case Queue-Size

Stable Protocols

- A protocol on a given network is *stable* if the queue size does not grow with time
- In order that there be *any* stable protocols, the traffic model cannot inherently overload the network
- For "reasonable" traffic models, the queue size of stable protocols is an increasing function of network parameters
- In AQT, the queue size is at least $\Omega(d)$

Provisioning Buffer Memory

- With an *a priori* bound on the network size, and
- With an accurate traffic model

The results of stability analysis can be used to bound the memory size needed by routers so that no packet is ever dropped

Traffic modeling

- What happens when traffic does not follow the model ?

Scalability

- What happens when the network continues to grow?

Empirically in the Internet the size of the buffers and the traffic are such that packets are routinely dropped.

Our Approach: Competitive Network Throughput

Network Model

- Buffers of preallocated size B - The size B is a parameter of our model and is independent of network parameters.

Protocols

- We require both a scheduling algorithm and a contention resolution algorithm
- **Contention Resolution algorithms:** Methods for deciding which packets in the input ports to transfer to the output ports and which to drop
  - **Geedy Contention Resolution:** Do not drop packets unless the output buffer is full
  - **Preemptive Contention Resolution:** May drop packets already in the buffer

Our Approach–cont'd

Traffic Model

- Completely arbitrary traffic–no restrictions

Metric

- Competitive ratio of the throughput
- Must fundamentally deal with effects of dropped packets on throughput in analysis
- No online algorithm can be competitive for the measure of the number of packets *dropped*

Goal: Use the Competitive Network Throughput model and metric to compare and contrast the performance of various protocols on various network topologies

Some Details: Throughput-Competitiveness

Compare the online local protocol to the (utopian) offline clairvoyant algorithm.

- Let $ADV^t(\sigma)$ be the number of packets delivered by the adversary by time t on traffic σ.
- Let $P^t(\sigma)$ be the number of packets delivered by P by time t on traffic σ.

A protocol P is c-throughput-competitive if:
$\forall\sigma \ \ \forall t \qquad P^t(\sigma) \geq (1/c)OPT^t(\sigma) - \alpha.$
where α is a constant, independent of the traffic σ.

- Some input traffic sequences inherently have low throughput
- The online algorithm is only penalized when the offline algorithm has high throughput and it has low throughput

Well-Known Greedy Protocols

Consider the following prioritizaion schemes for

- Scheduling, and
- Preemptive Contention Resolution

NTG - Nearest To Go
FFO - Furthest From Origin

FTG - Furthest To Go
NTO - Nearest To Origin

LIS - Longest In System
SIS - Shortest In System

FIFO - First In First Out

- We typically use the same scheme for both scheduling and contention resolution

Main Results

- *All greedy* protocols *are* competitive on *all DAGS*:

  For graph G, $O(f(G))$-competitive

- *Some greedy* protocols *are not* competitive on *networks that contain a cycle*:

  FTG, NTO, SIS, FIFO

- *Some greedy* protocols *are* competitive *on arbitrary networks*:

  NTG, FFO, LIS

- For the topology of the line:

  NTG - $O(n^{2/3})$-competitive

  FTG; LIS - $\Theta(n)$-competitive

  Any greedy protocol is $\Omega(n^{1/2})$-competitive.

Main Results (cont'd)

A comparison of several deterministic greedy protocols. The rows denote whether the protocols are throughput-competitive or AQT stable for *all* networks. [Borodin et al., Andrews et al., Gamarnik] :

	NTG	FFO	FTG	NTO	SIS	LIS	FIFO
stable	No	No	Yes	Yes	Yes	Yes	No
competitive	Yes	Yes	No	No	No	Yes	No

- FTG is stable on any topology as long as traffic allows stability [Gamarnik; Andrews et al.];
- NTG can be unstable even at arbitrary low injection rates [Borodin et al.].

All 4 combinations of stable or unstable with competitive or non-competitive are present!

Main Results (cont'd)

Stability analysis may not be the right means to compare protocols:

- in continuously growing networks,
- with ill-behaved traffic, and
- with buffers that don't scale in size with the network
- the Internet.

REST of THE TALK: two examples

- NTG is Throughput-Competetive on all networkds.

- FTG is NOT Throughput-Competetive on the cycle.

Nearest-To-Go is Throughput Competitive

Lemma: If at time t NTG stores k packets, then by time $t + dB$ it delivers at least $\min\{k, B\}$ packets.

Proof:

ℓ_τ - shortest distance-to-destination in network at time τ.
If $\ell_\tau = 1$, a packet is delivered at time $\tau + 1$.
If $\ell_\tau > 1$, $\ell_{\tau+1} < \ell_\tau$.
Therefore at least one packet is delivered within d time steps.

Number of packets in the system drops below $\min\{k, B\}$ only by packet arrival.

$\square$

Nearest-To-Go (cont.)

definitions:

- weight of packet p -

$$weight(p) = \begin{cases} 1 & \text{if } p \text{ delivered by adversary} \\ 0 & \text{otherwise} \end{cases}$$

- Frame j - $[(j-1)dB+1, jdB)$
- a_j - total weight of packets injected in frame j.
- b_j - number of packets delivered by NTG in frame j.

Nearest-To-Go (cont.)

Lemma: $\forall j,\ b_j + b_{j+1} \geq \min\{a_j, B\}$.

Proof:

case 1: NTG stores at some time in frame j B packets.

$b_j + b_{j+1} \geq B$.

case 2: During all of frame j NTG stores less than B packets.

No packet is dropped during frame j.

A t the end of frame j, NTG has at least $a_j - b_j < B$ packets.

$b_{j+1} \geq a_j - b_j$; $b_j + b_{j+1} \geq a_j$.

□

Lemma: $\forall j,\ a_j \leq 2mdB$.

Proof:

At most $m \cdot dB$ delivered during frame.

At most mB stored in buffers at the end of frame.

□

Nearest-To-Go (cont.)

Theorem: $NTG^t \geq \frac{ADV^t}{4md} - B$.

Proof:

$NTG^t = \sum_{j=1}^{s} b_j \geq$
$\frac{1}{2}\sum_{j=1}^{s-1}(b_j + b_{j+1}) \geq$
$\frac{1}{2}\sum_{j=1}^{s-1} \frac{a_j}{2md} \geq$
$\frac{ADV^t}{4md} - B$

$\square$

FTG not competitive on the cycle

- Unidirectional cycle of n nodes, $[0, n-1]$.

- $\forall t$, $\forall i$, inject at i a packet with destination $(i+2)$ $\texttt{mod}$ n.

- Adversary can deliver (roughly) half of all the packets.

Conclusions

- A model for the analysis of network protocols in a setting that explicitly addresses dropped packets, allows constant-sise buffers and arbitrary traffic.
- A number of results using this model comparing protocols and topologies.
- Some conclusions are in contrast to those of Adversarial Queuing Theory — Stability analysis may not be the right means to compare protocols:
  - in continuously growing networks,
  - with ill-behaved traffic,
  - with buffers that don't scale in size with the network, and
  - for the Internet.
- Many open questions!