Analysis of Dynamic (Network) Processes

Eli Upfal

Brown University
Dynamic Processes:

- Input is **continuously** injected to the system;
- **Irrevocable** decisions made **online** based on current input, before observing future input items;
- Process runs **forever**.

Static (batch) processes:

- All data is **available** at the beginning of the computation.
- Process **terminates** with an output.
Examples of Dynamic Processes

- **Contention-resolution protocols** - Ethernet, Aloha protocol.
- **Routing protocols** - Internet, parallel computation.
- **Virtual-memory management** - Paging, Prefetching.
- **Load balancing protocols** - time sharing, distributed scheduling.
Analysis of Dynamic Processes

How do we analyze a nonterminating process?

What performance measures are we interested in?

1. **Stability**: The (expected) number of requests/jobs in the systems is bounded;

2. What fraction of requests is actually executed (on average)?

2. **Efficiency**: How long does it take (on average) to process a request?
Related Questions Are Studied in ...

- Queuing theory;
- Theory of infinite stochastic processes;
- Dynamic systems;
- .............
Input Model

- Standard **worst-case** analysis - meaningless in most cases

- **Competitive (online) analysis**

- **Restricted input:**
  - **Deterministic bounds** on input stream - Adversarial Queueing Theory
  - Assuming specific **probability distribution** on input
  - Assuming some **statistical properties** on input distribution (bounds on moments - stochastic adversarial analysis)
Contestation Resolution Protocols

Ethernet Protocol:

- One request can be satisfied at a given time.
- If more than one request is submitted, no request is satisfied.
- A sender can detect if its request is satisfied.
- A satisfied request occupies the resource for one step.
Stochastic Analysis

Tools:

- Combinatorial argument;
- Reduction to queueing theory;
- Stationarity of a Markov chains;
- Renewal process;
- General drift criteria.
Stochastic Performance Measures

\( \lambda \): Arrival rate - expected number of new requests in a unit time interval.

\( N(t) \): Number of requests in the system at time \( t \).

\( W(t) \): The waiting + execution time of a request that entered the system at time \( t \).

**Stability**: We say that the system is stable if the expected number of requests in the system is bounded with respect to time,

\[
\lim_{t \to \infty} E[N(t)] < \infty.
\]
Little’s Formula

\[ N = \lim_{t \to \infty} \frac{1}{t} \sum_{i=1}^{t} N(i) \]

Assume \( \alpha(t) \) arrivals during \([0, t] \), in times \( \tau_1, \ldots, \tau_{\alpha(t)} \). Then

\[ W = \lim_{t \to \infty} \frac{\sum_{i=1}^{\alpha(t)} W(\tau_i)}{\alpha(t)}. \]

**Little’s Equation:** If \( N \) is bounded then the system is stable and

\[ N = \lambda W. \]
"Proof"

The expected number of new arrivals the interval \([0, t]\) is \(\lambda t\).

The expected number of departures is \(\frac{N}{W} t\).

In the limit

\[
\lambda t = \frac{N}{W} t
\]

or

\[
\lambda W = N.
\]

[non-lattice, with probability 1]
Simple Combinatorial Argument: Packet Routing on a Ring

Ring of $n$ stations.

At each step, each station generates one packet with probability $\frac{\lambda}{n}$.

The destinations of the packets are uniformly distributed among the $n$ stations.

Packets are sent clockwise on the ring.

A station can send one packet per step, a packet can traverse one edge per step.
Farthest-first priority - packets with farther destinations have higher priority;

For what values of $\lambda$ is the system stable?

What is the expected routing time of a packet?
Analysis

**Theorem.** Routing with farthest-first priority is stable for any $\lambda < 2$.

**Definition.** Packet $p$ was delayed $t$ steps in crossing edge $e$ if

1. $p$ traversed edge $e$ at step $t + k$; and

2. $e$ is the $k$th edge in the route of $p$. 
Lemma. If a packet $p$ was delayed $t$ steps in crossing edge $e$, then there is an interval of $t$ steps such that some packet crossed edge $e$ in each step of this interval.

Proof. Any packet that delayed $p$ must also cross edge $e$.

Each packet can delay $p$ only once.

All the packets that delayed $p$ are moving in front of $p$, thus crossing $e$ in the interval $[k, \ldots, t - 1]$.
The Wide-Channel Model

In the wide-channel model packets are never delayed.

A packet crosses edge $e$ at time $k$ if $e$ is the $k$th edge on its route.

**Lemma.** If $t$ packets crossed edge $e$ in the interval $[k, k + t]$ then there is an interval of length $x \geq t$ such that $x$ packet crossed edge $e$ in that interval in the wide-channel model.
Proof. Let $[\tau', \tau' + x]$ be the maximal interval that includes $[k, k + t]$ in which a packet crossed edge $e$ in each step.

If a packet was delayed $\ell$ steps in crossing edge $e$, then $\ell$ packets precede that packet in crossing $e$.

The first packet to cross $e$ in this sequence was not delayed, then also in the wide-channel model it crosses $e$ at time $\tau'$.

The $i$th packet in the sequence is delayed no more than $i - 1$ steps $\Rightarrow$ in the wide-channel model it traversed $e$ in the interval $\tau', \tau + i - 1$.

$\Rightarrow$ All the $x$ packet pass $e$ in the wide-channel model in the interval $[\tau', \tau' + x]$.
Lemma. There is a constant $\alpha > 0$ such that for a sufficiently large $t$ the probability that a given packet is delayed at least $t$ steps is bounded by $e^{-\alpha t}$.

Proof. If a packet was delayed $t$ steps then there is an edge $e$ such that the packet was delayed $t$ steps in crossing that edge.

Thus, there is an interval of $\tau \geq t$ steps such that $\tau$ packets crossed edge $e$ in the wide channel in that interval.
The expected number of packets that cross edge $e$ in the wide channel at a given step is bounded by:

$$
\sum_{i=0}^{n-1} \frac{\lambda(n-i)}{n} < \lambda \frac{n(n-1)}{2n} = \delta < 1.
$$

($i$ the distance from $e$)

For sufficiently large $t$, the probability of $t$ packets crossing edge $e$ in an interval of length $t$ is bounded by

$$e^{-\alpha t}.$$
Theorem. For any $\lambda < 2$ the system is stable and the expected time till a packet is delivered is $O(N)$.

Proof. The route length is bounded by $N$.

The expected delay is bounded by

$$\sum_{t \geq 1} e^{-\epsilon t} = O(1).$$

Applying Little’s equation, the expected number of packets in the network is bounded $\Rightarrow$ the system is stable. □
**Theorem.** With probability $1 - \frac{1}{n}$ a given queue at a given step has no more than 3 elements.

**Proof.** With probability $1 - \frac{1}{n}$ the given step is within a segment of no more than $C \log n$ consecutive packets crossing $e$.

The queue is bounded by the total number of new elements generated by the node feeding the edge. The probability that more than 2 new elements were generated in this interval is bounded by

$$\left(\frac{C \log n}{2}\right)^2 \lambda 2 < \frac{1}{n}$$
Analysis: Upper bound on $\lambda$

**Theorem.** *The system cannot be stable for $\lambda > 2$.*

**Proof.** The average route of a packet is $n/2$.

There are no more than $n$ packet transitions per step.

To keep the system stable a process can insert a new packet into the system every $n/2$ steps on average. $\square$
Modeling Communication Networks as Networks of Queues

Difficulties:

- What is the distribution of incoming stream to an internal node?
- The inter-arrival times in internal queues are highly correlated.
- Service times of the same packet in different queues are not independent.
M/M/1 Queue

Single (FIFO) queue

Poisson arrival with rate $\lambda$

Service time is exponential with expectation $\mu$

Stable iff $\rho = \frac{\lambda}{\mu} < 1$.

$$\lim_{t \to \infty} P(n(t) = k) = \rho^k (1 - \rho)$$

$$\lim_{t \to \infty} E[N(t)] = \frac{\rho}{1 - \rho}$$
Jackson’s Theorem
(Product Form Queueing Networks)

FIFO queues;

New customers arrive at queue $i$ according to a Poisson process with rate $r_i$ (some $r_i$’s may be 0).

Service at queue $i$ is exponential with rate $\mu_i$.

Markovian Routing: When a customer is served at queue $i$ it proceeds with probability $P_{i,j}$ to queue $j$, and with probability $1 - \sum_j P_{i,j}$ it leaves the system.

All random choices are independent.
Flow Equations

Let $\lambda_i$ be the arrival rate (of new and existing customers) to queue $i$; then

$$\lambda_i = r_i + \sum_{j=1}^{K} \lambda_j P_{j,i} \quad i = 1, \ldots, k.$$ 

$n_i(t)$ = the number of customers at queue $i$ at time $t$.

$n(t) = (n_1(t), \ldots, n_K(t))$ = the state of the system at time $t$. 
Theorem. Assume that

1. \( \lambda_i = r_i + \sum_{j=1}^{K} \lambda_j P_{j,i} \) \( i = 1, ..., K. \)
   has a unique solution.

2. \( \rho_i = \frac{\lambda_i}{\mu_i} < 1, \) for \( i = 1, ..., K. \)

Then the system converges to a steady state \( (n_1, ..., n_K) \), such that

\[
\lim_{t \to \infty} P_t(n) = P(n) = P_1(n_1) \times \cdots \times P_K(n_K)
\]

and

\[
P_j(n_j) = \rho_j^{n_j}(1 - \rho_j), \quad n_j \geq 0.
\]
Jackson Theorem

If the system

\[ \lambda_i = r_i + \sum_{j=1}^{K} \lambda_j P_{j,i} \quad i = 1, \ldots, k \]

has a unique solution with

\[ \rho_i = \frac{\lambda_i}{\mu_i} < 1, \quad \text{for} \quad i = 1, \ldots, K. \]

Then in the limit the system operates as a collection of independent \( M/M/1 \) queues, though the arrivals into each of the queues is not necessarily Poisson.
Applications: Packet Routing

Consider an $N$-node hypercube or an $N$ input/output butterfly network.

Assume $N = 2^n$.

Packet arrival distribution at each (input) node is Poisson with expectation $\lambda$.

Packet have random, independent destinations.

Time to traverse an edge (for the packet at the head of the edge’s queue) has exponential distribution with expectation $\mu$.

Bit-wise routing (fixing bits in a given order) is Markovian.

FIFO queues.
Routing on the Butterfly

Let $\lambda_i$ be the injection rate to an edge at stage $i$ (fixing bit $i$).

$\lambda_1 = 2^{1/2} \lambda_0 = \lambda$

$\lambda_i = 2^{1/2} \lambda_{i-1} = \lambda$

If $\rho = \frac{\lambda}{\mu} < 1$:

- Expected queue size (in the limit) = $\frac{\rho}{1-\rho}$
- Expected routing time (in the limit) = $n \mu (1 + \frac{\rho}{1-\rho})$
Routing on a Hypercube

Expected number of edges a packet traverses is $n/2$.

The injection rate to an edge is bounded by $\lambda$.

Expected routing time (in the limit) is bounded by $\mu \frac{n}{2} \left(1 + \frac{\rho}{1-\rho}\right)$. 
Deterministic Service Time

**Theorem.** Let $Q$ be a layered network of FIFO queues with unit service time, Markovian routing, and Poisson arrivals of new customers (from outside the network) to queues.

$N(t) = \text{be the number of packets in } Q \text{ at time } t.$

$\tilde{Q} = \text{an identical network with service time exponentially distributed with unit expectation.}$

$\tilde{N}(t) = \text{be the number of packets in } \tilde{Q} \text{ at time } t.$

Then,

$N(t) \leq_s \tilde{N}(t).$
The **butterfly** is a layered network

**Bit-wise** routing on the hypercube defines a layered network.

The above analysis bounds the performance of these networks with **unit** service time.
Using Queueing Theory II: The Black-Box Approach

Transform a static algorithm + analysis to a dynamic algorithm + analysis.

Example: Given an efficient algorithm for static routing on a bounded buffer network, we obtain an efficient algorithm for dynamic routing on that bounded buffer network.
Butterfly Routing with Acknowledgments

Static Result:

\[ N = 2^n \]-node back-to-back butterfly network, with bounded buffers in the network switches.

Up to \( n \) packets in each input, each output is the destination of up to \( n \) packets.

There is a randomized algorithm that routes all packets to their destinations in \( cn \) parallel steps with probability \( 1 - e^{-\alpha n} \), for some \( c > 1 \) and \( \alpha > 0 \).
Dynamic Result

Each input queue considers the network a black box server.

If there are up to \( n \) packets from each input node in the network, the service time is \( cn \) with probability \( 1 - e^{-\alpha n} \).

We need to:

- Guarantee that during the dynamic execution there are no more than \( n \) packets from each input in the network - use acknowledgments.
- Take care of the low-probability bad executions.
An input has up to $n$ packets in the network.

When an input node receives a delivery acknowledgment it can send a new packet.

A packet survives in the network for up to $cn$ steps, if it is not delivered within that time it is deleted.

If an input does not receive an acknowledgment within $2cn$ steps of sending a packet, it sends the same packet again (acknowledgments are routed on the same network).
Assume for simplicity that each input has $n$ queues, one message from each (non-empty) queue is in the network.

The probability that the head of a queue is not delivered in $i2c \log n$ steps is $e^{-i\alpha n}$.

The analysis is reduced to the properties of a single queue.
Modeling as a Markov Chain

Model the process as an aperiodic, irreducible, Markov chain.

A Markov chain has countably infinite state space.

Use drift criteria to show that

1. The process is stable/unstable.

2. For a stable process - bound the expected time in the process.
Contention Resolution Protocols

- Only one request can be satisfied at a given time.
- If more than one request is submitted, no request is satisfied.
- A sender can detect if its request is satisfied.
- A satisfied request occupies the resource for one step.
- A sender’s protocol uses only that station’s history of success and failure requests - no knowledge about other senders, or even the number of other active senders.
The Backoff Protocols

Each sender follows the following protocol.

1. The backoff counter \( b \) is initially set to 0.

2. **While** the sender’s queue is not empty **do**
   (a) With probability \( \frac{1}{f(b)} \) submit a request.
   (b) **If** the request satisfied \( b \leftarrow 0 \), **else** \( b \leftarrow b + 1 \).

   Exponential backoff: \( f(b) = 2^b \)
   (The Ethernet uses \( 2^{\min[b,10]} \))

   Linear backoff: \( f(b) = \alpha b + 1 \).

   Polynomial backoff: \( f(b) = b^\alpha \).
Markov Chain Representation

\[ X_i(t) = \text{number of requests at sender } i \text{ at time } t. \]

\[ b_i(t) = \text{the value of the back-off counter of sender } i \text{ at time } t. \]

The process

\[(X(t), b(t)) = (X_1(t), \ldots, X_n(t), b_1(t), \ldots, b_n(t))\]

is a discrete-time, discrete-space, Markov chain.

The chain is aperiodic and irreducible with countably infinite state space.
Either no state is positive recurrent, in that case the chain is not ergodic, and

\[
\lim_{t \to \infty} E[N(t)] = E\left[\frac{1}{t} \sum_{i=1}^{t} \sum_{i=1}^{n} X_i(t)\right] \infty
\]

Or, the chain is ergodic (positive recurrent), it has a stationary distribution, but \(E[N(t)]\) may still be unbounded.

[Let \(\pi_i\) be the stationary distribution of \(X_i(t)\), with \(\pi_i(j) = \frac{c}{j^{1.3}}\). \(E[N(t)]\) is not bounded.]

That’s why we define stability in terms of \(E[N(t)]\) and not just as positive recurrent chain.
Consider an aperiodic, irreducible Markov chain with countably infinite state space. Assume that there exists a non-negative function $f$ and a constant $\delta > 0$, such that for any state $x$

$$E[f(X_{t+1}) - f(X_t) \mid X_t = x] \geq \delta,$$

or

$$E[f(X_{i+1}) \mid X_t = x] \geq f(x) + \delta.$$
This condition is not sufficient to prove that the chain is transient:

Let

\[ X_{i+1} = \begin{cases} 
0 & \text{with probability } \frac{1}{2} \\
2X_i + 2 & \text{with probability } \frac{1}{2}
\end{cases} \]

For this chain

\[ E[f(X_{i+1}) - f(X_i) \mid X_i = x] = 1, \]

but the chain is \textit{ergodic}. \[\text{44} \]
**Theorem.** Consider an aperiodic and irreducible Markov chain. Assume that there is a non-constant function $f$ that has an absolute upper bound $C$, and such that

$$E[f(X_{t+1}) \mid X_t = x] \geq f(x).$$

Then the chain is transient.

**Proof.** $f(X_i) - C$ is a non-positive submartingale with respect to $X_i$, thus converges with probability 1 to a limit. Let $x$ and $y$ be two states such that $f(x) \neq f(y)$. If states $x$ and $y$ are recurrent, the function $f$ assumes the values $f(x)$ and $f(y)$ infinitely often with probability one, and cannot converge to a limit. □
**Theorem.** The exponential backoff protocol is unstable for any $N > 2$, when the arrival rate at each station is $\frac{\lambda}{N}$, for $\lambda \geq \lambda_0 \approx 0.567$.

**Proof.**

$$(X(t), b(t)) = (X_1(t), \ldots, X_n(t), b_1(t), \ldots, b_n(t)),$$

$$X_i(t) = \# \text{ of requests at sender } i \text{ and time } t$$

$$b_i(t) = \text{value of the counter of sender } i \text{ at time } t.$$
Lemma. Let

\[ f(X(t), b(t)) = (2n-1) \sum_{i=1}^{n} X_i(t) + \sum_{i=1}^{n} 2^{b_i(t)} - n, \]

If \( \lambda \geq \lambda_0 \), then \( E[f(X(t), b(t))] = \Omega(t) \).
Corollary 1. If $\lambda \geq \lambda_0$ then the expected wait of a new request arriving at time $t$ is $\Omega(t)$.

Proof. Given that a new request arrives at time $t$, it arrives at sender $i$ with probability $1/n$. The expected wait of that request is $X_i(t) + 2^{b_i(t)}$.

The expected wait of a new request at time $t$ is

$$\frac{1}{n} \sum_{i=1}^{n} (X_i(t) + 2^{b_i(t)}) \geq \frac{1}{2n^2} f(X(t), b(t)) = \Omega(t).$$

Thus, $\lim_{t \to \infty} E[W(t)]$ is unbounded.
Theorem. [Foster’s Criteria] An aperiodic and irreducible Markov chain on a countable state space \( \Omega \) is positive recurrent (ergodic) if there is a function \( f : \Omega \to R^+ \), a finite set \( C \subset \Omega \), and a constant \( \beta > 1 \) such that

\[
\begin{align*}
E[f(X_{t+1} - f(X_t) \mid X_t = x] & \leq -\beta \quad x \notin C \\
E[f(X_{t+1} \mid X_t = x] & \leq \infty \quad x \in C
\end{align*}
\] (1)
"Proof"

\[ V = \max_{x \in C} f(x) \]

\[ C' = \{ x \mid f(x) \leq V \} \]

Assume that the chain is at \( y \notin C' \). The expected time to return to \( C' \) is

\[ O(f(y)) < \infty. \]

The set \( C' \), and thus the whole chain, are positive recurrent.
Theorem. [Foster’s Criteria] Given an aperiodic and irreducible Markov chain on a countable state space, and a function \( f : \Omega \to [1, \infty) \), the following two conditions are equivalent:

1. The chain is ergodic with stationary distribution \( \pi \), and \( E_\pi[f(x)] \) is finite;

2. There is a finite set \( C \) such that

\[
\sup_{x \in C} E\left[ \sum_{i=1}^{T_{\text{return}}} f(X_i) \mid X_0 = x \right] < \infty,
\]

where \( T_{\text{return}} \geq 1 \) is time of first return to \( C \).
**Theorem.**  The polynomial backoff protocol is stable for any $N > 2$, any $\alpha > 1$, and any $\lambda_i$ such that $\sum_{i=1}^{N} \lambda_i < 1$.

**Proof.** The function

$$f(X(t), b(t)) = \sum_{i=1}^{n} X_i(t) + \sum_{i=1}^{n} (b_i(t)+1)^{\alpha + \frac{1}{2} - n}$$

satisfies the conditions of the above theorem for a polynomial backoff protocol where senders use $f(b) = (b + 1)^{-\alpha}$, with any $\alpha > 1$. □
Modeling as a Renewal Process

Consider a stochastic (continuous time) process that alternates between two states: on and off.

\[ F_{on} = \text{the distribution of the length of an on segment;} \]

\[ F_{off} = \text{the distribution of the length of an off segment;} \]

We assume that the lengths of disjoint segments are independent random variables.

The process has a renewal whenever it switches to an on state.
Two Theorems

**Theorem.** Assume $X_{on} \sim F_{on}$ and $X_{off} \sim F_{off}$,

$$\lim_{t \to \infty} Pr(\text{Process is on}) = \frac{E[X_{on}]}{E[X_{on}] + E[X_{off}]}.$$

**Theorem.** Assume that the process is on at time $t$, let $Z_{on}(t)$ be the length of segment till time $t$ in which the process was on. If the process is not a lattice then,

$$\lim_{t \to \infty} Pr(Z(t) \leq z) = \frac{1}{E[X_{on}]} \int_{0}^{z} (1-F_{on}(y))dy.$$
Example: Load Balancing

- A network of $n$ processors / servers
- Jobs are generated in the processors
- Processors execute one job per step
- The processors can send any number of jobs to a direct neighbor
- No central control (decisions are local)
• Models networks where execution cost is significantly higher than transfer cost
Input Model

• At every step, an adversary places $n$ generators arbitrarily on nodes
  – The adversary may know the whole history - process is not Markovian!

• At the beginning of each time step, every generator creates a new job with probability $\lambda < 1$

• Expected number of new jobs per step = $\lambda n$
Protocol

In each step

- Choose a random matching in the graph
  - This can be done locally (in a distributed way).

- Equalize (modulo 1) the loads between the matched nodes

- Each node executes the first job in its queue
Analysis

- Split time into epochs of fixed length, $\tau = O(\log n)$

- An epoch is successful if, when it ends, the total load of the system is either
  - below threshold $\Lambda = O(n \log n)$, or
  - decreased by at least $\Delta = O(n \log n)$

Lemma.

An epoch is successful with probability at least

$$1 - \frac{1}{n^c},$$

conditioned on all past events.
Alternate Renewal Process

**Good** state - the total load is bounded by
\[ \Lambda = O(n \log n) \]

**Bad** state - the total load is greater than \( \Lambda \)
Length of Good Segment

If the system is in a good state at the beginning of an epoch, and the epoch is successful, then the system is in a good state at the end of the epoch.

\[ T(\text{good}) = \# \text{ of epochs in a good segment}. \]

\[ Pr(T(\text{good}) \geq i) \geq (1 - \frac{1}{n^c})^{i-1}. \]

\[ E[\text{length of a good state segment}] = O(n^c). \]
Length of Bad Segment

Total new load in a bad epoch is $\tau n = O(n \log n)$.

If the system is in a bad state, a successful epoch reduces the load by $\Delta = O(n \log n)$.

If the system is in a bad state during $i$ epochs, at least $\theta i$ epochs must be unsuccessful (for some constant $\theta > 0$).

$$Pr(T(\text{bad}) \geq i) \leq \binom{i}{\theta(i-1)} \left(\frac{2}{n^c}\right)^{\theta(i-1)} \leq n^{-\theta(c-1)(i-1)}$$
\[ E[\text{length of a bad state segment}] = O(1). \]

\[
\lim_{t \to \infty} Pr(\text{system is in bad state at time } t) = \frac{E[\text{length of a bad state segment}]}{E[\text{length of a good+bad state segment}]} = O\left(\frac{1}{n^c}\right).
\]
Load

In good state the load of the system is $c_1 n \log n$.

If the system has load $i \tau n \log n$ it had to be in bad state for at least $i - 1$ epochs.

$L(t) =$ load of the system at time $t$.

$$\lim_{t \to \infty} E[L(t)] = c_1 n \log +$$

$$n \log n \sum_{i \geq i} Pr(\text{bad for last } i \text{ epochs})$$

$$\lim_{t \to \infty} E[L(t)] = O(n \log n).$$

[Need to work in continuous time.]
Theorem. The system is stable and as time tends to infinity the expected total load in the system is $O(\gamma \cdot n \ln n)$.

1. $\lim_{t \to \infty} \Pr(W(t) \leq O(\gamma \cdot \ln n)) \geq 1 - n^{-c}$.

2. $\lim_{t \to \infty} E[W(t)] = O(\gamma \cdot \ln n)$.

The above bounds hold without the limits if the system starts empty.
Stochastic Adversarial Queueing Model

- Stochastic analysis that is independent of the specific input distribution.
- Minimum restrictions on the set of possible input distributions.
\( N(t, t + w) = \# \) of new jobs in the interval \([t, t + w)\).

\( H_t \) - history till time \( t \).

**Definition.** An adversary is a \((w, \rho)\)-stochastic adversary if for all \( t \),

\[
E[N(t, t + w) \mid H_t] \leq \rho w.
\]

**Definition.** A \((w, \rho)\)-stochastic adversary is properly bounded if for some constants \( p > 2 \) and \( V \), and for all \( t \)

\[
E[(N(t, t + w))^p \mid H_t] \leq V.
\]
More general input distribution:

\[ E[\# \text{ of new jobs in } w \text{ steps}] < n \cdot w \]

\[ E[(\# \text{ of new jobs in } w \text{ steps})^p] < M, \text{ for some } p > 2 \]
Theorem. [Pemantle & Rosenthal - 99] Let $X_1, X_2, \ldots$ be a sequence of nonnegative random variables satisfying the following conditions:

1. There exist positive constants $\alpha$ and $\Theta$ such that for all $x_1, \ldots, x_i$ with $x_i > \Theta$,

$$E[X_{i+1} - X_i | X_1 = x_1, \ldots, X_i = x_i] \leq -\alpha$$

2. There exists a positive constant $\Psi$ and a $p > 2$ such that for all $x_1, \ldots, x_i$,

$$E[|X_{i+1} - X_i|^p | X_1 = x_1, \ldots, X_i = x_i] \leq \Psi.$$
Then there exist $\Psi = \Psi(X_0, \alpha, \Theta, \xi)$ and $t_0$ such that for all $t \geq t_0$,

$$E[X_t|X_0] \leq \Psi + \max(0, X_0 - \Theta).$$

Furthermore, assuming that $p$ is a constant,

$$\Psi = O\left(\Theta + \alpha \left(1 + \frac{\xi}{\alpha^p}\right)^{3p}\right), \quad \text{as } n \to \infty.$$
Load Balancing with Stochastic Adversary

Theorem. For any \((w, \rho)\)-properly bounded stochastic adversary with \(\rho < 1\),

- The load-balancing protocol is stable;
- Limit expected load in the system is \(O(wn \log n)\).
Proof

Define an epoch as $\max[\tau, w]$ time steps.

$Y_i =$ be the total load in the system at the begining of segement $i$.

When load $> \Gamma$, a succesful epoch reduces the load by at least $\Delta$.

We prove: There exist positive constants $\Delta'$, such that for all $y_1, \ldots, y_i$ with $y_i > \Gamma$,

$$E[Y_{i+1} - Y_i | Y_1 = y_1, \ldots, Y_i = y_i] \leq -\Delta'$$
There exists a positive constant $\Psi = O(wn \log n)$ and a $p > 2$ such that for all $y_1, \ldots, y_i$, 

$$E[|Y_{i+1} - Y_i|^p | Y_1 = y_1, \ldots, Y_i = y_i] \leq \Psi.$$
Adversarial Queueing Model for Packet Routing

Packets are generated by an adversarial process that controls the arrival times, route, and destinations of packets, subject to a fixed bound $\alpha$ on the amount of traffic generated at a given step for a given edge.

Allows detailed study of stability (and delay) of different queueing disciplines and network topologies.
Stability Results

Definition. An adversary is \((w, \rho)\)-deterministic adversary if for all \(t\),

\[ N(t, t + w) \leq \rho w. \]

Definition. A queuing discipline is greedy if some packet is transmitted whenever the queue is not empty.

Theorem. Any directed acyclic network with any \((w, 1)\)-deterministic adversary is stable.
Theorem. The ring with $(w, 1)$-deterministic adversary input is

1. Unstable for FIFO and LIS (Longer In System);

2. Stable for FTG (Further To Go).
Universal Stability Results

**Theorem.** LIS (Longer In System) and NIS (Newest In System) are stable for any \((w, 1-\epsilon)\)-deterministic adversary and any network, for any \(\epsilon > 0\).

**Theorem.** The ring is stable for any greedy routing discipline and any \((w, 1-\epsilon)\) deterministic adversary, for any \(\epsilon > 0\).
Dynamic Process as a Competitive Game

So far we have assumed that all parties collaborate, or follow the same rules.

In some settings it may be more accurate to model the parties as competing for shared resources.

[In the Internet: a user wants to receive a stream of video sufficiently fast - doesn’t care about the rest.]

What is the steady state when parties try to improve their performance at the expense of other parties?
Game Theory

A mathematical treatment of the behavior of players with conflicting interests.

Players - an independent decision unit.

Each player has a set of possible actions that it can take.

The end state (outcome) is a function of the actions of all players.

Each player has an objective function on the possible end states.

A solution is the set of “rational” outcomes of the game.
Communication Setting

A communication graph with edge capacities (delay functions).

A set of pairs of nodes \{\((t_1, s_1), \ldots, (t_k, s_k)\)\} representing ongoing communication (can add communication parameters such as required speed).

Flow between a given pair can (cannot) be split between different paths.

Each pair can detect the speed of their communication and try to improve by shifting (some) flow to other paths.
Questions

1. Is there an ”interesting” scenario, where the communication needs of all players can be satisfied by a global control, but not in the adversarial model?

2. When does the adversarial process converge? How fast?

3. Assume that there is a cost to shifting communication stream; how often should it be done?
• Routing on the ring: Leighton - 90.

• Modeling as Network of Queues: Stamoulis & Tsitsiklis - 91; Harchol-Balter & Black - 90; Mitzenmacher -94.

• “Black Box” - Broder, Freize, & U - 97; Scheideler & Vocking - 99;

• Markov Chain Approach: Aldous - 87, Hastad, Leighton, & Rogoff - 86; Berenbrink, Friedetzky, & Goldberg - 01;

• Renewal Process: Adversarial Queueing Theory: Borodin, Kleinberg, Raghavan, Sudan, & Williamson - 96; Cruz - 91; Andrews, Awerbuch, Fernandez, Kleinberg,
Leighton, Liu -96; Load Balancing: Anagnostopoulos, Kirsch. U - 03;

- Game Theory: Roughgarden & Tardos - 01;