Coexistence of Heterogeneous Traffic in CSMA/CA Networks

Tamir Husain
Christopher A. Wood

Agenda

1. Problem statement
2. Markov model iterations
3. Simulation overview
4. Experimental results and observations
5. Q&A

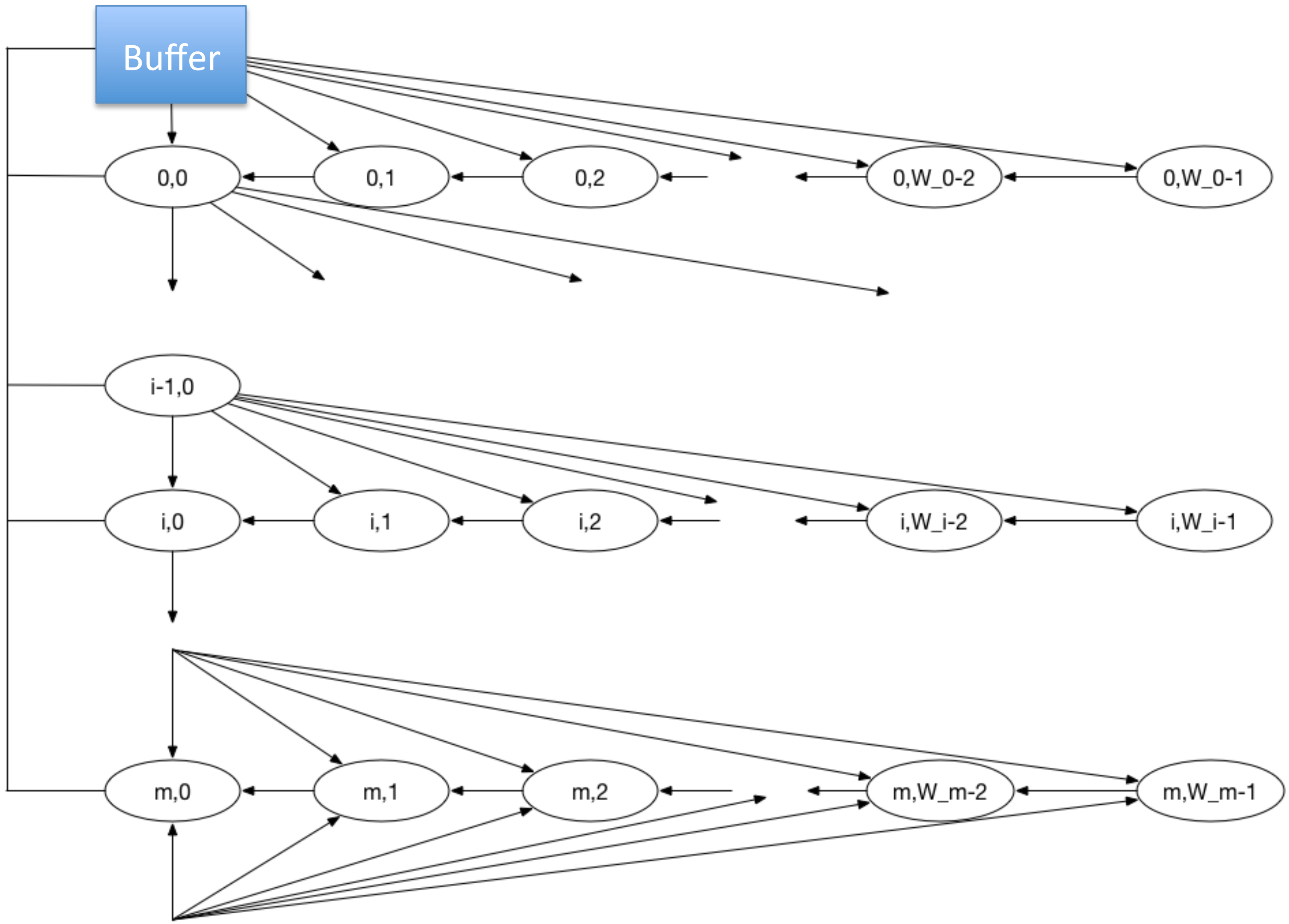
Traffic Trends

- Application traffic is becoming increasingly heterogeneous:
  - Real-time video streaming (e.g., Netflix)
  - Web browsing
  - File downloads
  - VoIP (e.g., Skype)
  - Video games
  - …
- Emerging traffic types vary in packet size, interarrival times, bursty-ness, etc.
- Many applications send and receive traffic over wireless PHY mediums – WiFi

802.11 Distributed Control Function

- A simplistic random access scheme based on the CSMA/CA protocol.
- Failed packets are retried according to a binary exponential backoff rule.
  - At each packet transmission, the backoff is uniformly in the range (0, w-1)
  - The window w is set to W_{min} to begin, and upon every failure, the backoff counter window is doubled.
  - The maximum backoff time is bounded by $W_{max} = 2^m W_{min}$.
- Selections of W_{min} and W_{max} depend on the physical layer specifications in the 802.11 standard

The DCF Model with Saturated Traffic



Question 1

How does the performance of the DCF vary with increasingly heterogeneous traffic?

Question 2

Is universal random backoff the best technique technique to handle collisions? Would a deterministic avoidance scheme be better suited for heterogeneous traffic? Or maybe a hybrid approach?

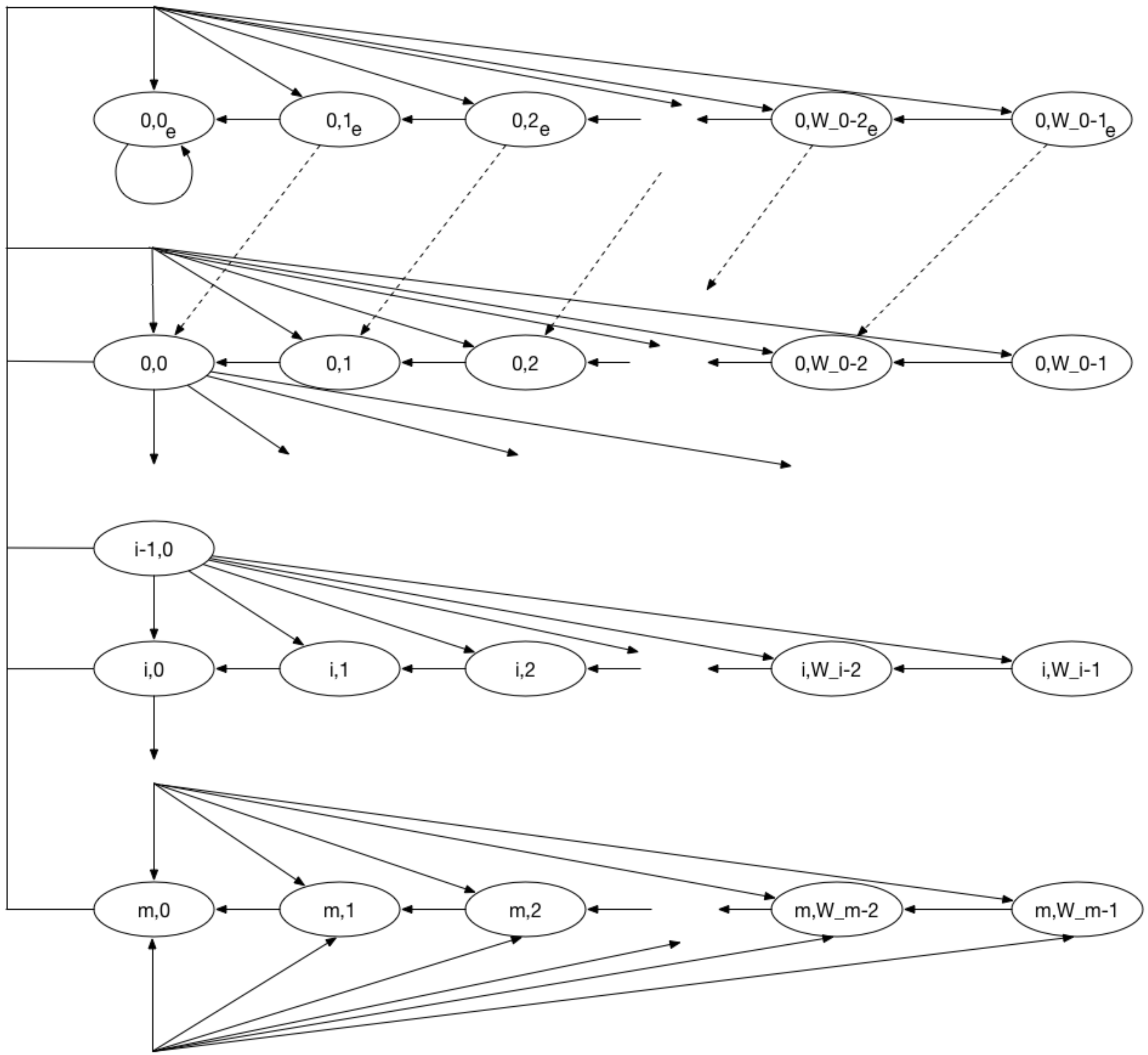
Question 3

Should backoffs be tailored to the type of traffic?

Research Approach

- Create Markov models that represent varying types of traffic:
  - Media streams
  - Web browsing
  - File downloads
- Compute metrics for each of these models individually, i.e., *with a constant conditional collision probability*
- Compute metrics when different Markov models of these types are superimposed
- Analyze the data

Supporting Unsaturated Traffic

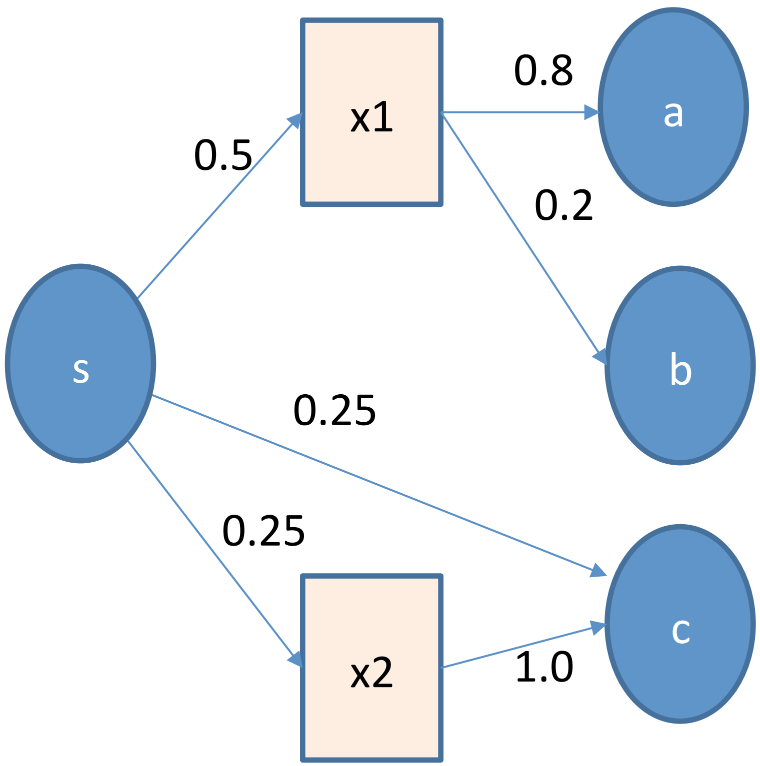


Creating a more complex DCF

- Adding features adds new dimensions to the 2D DCF
  - Packet size, interarrival, postbackoff, etc…
- Create small Markov models
  - Treat each as a black box
  - Connect them through instantaneous transitions
- Compressible states
  - Temporary logical states
  - Probabilities distributed to real states in preprocessing

Compressible States

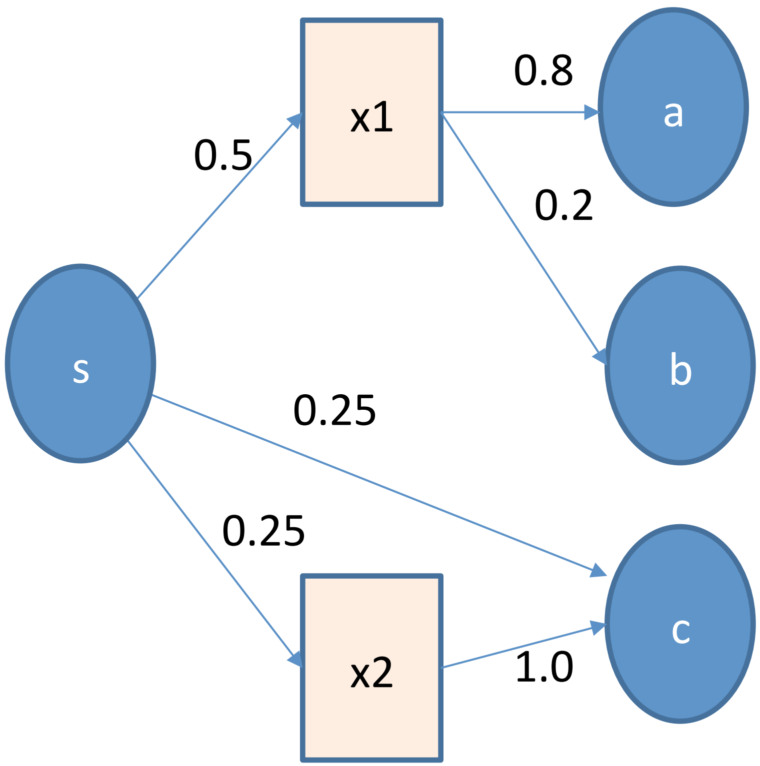
Uncompressed Model



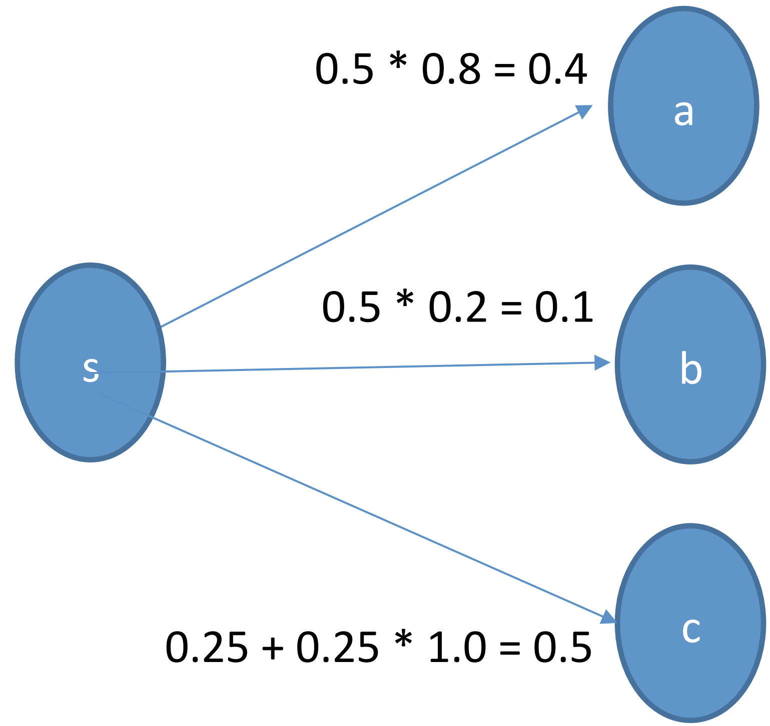
- s, a, b, c: Traditional states
- x1, x2: Compressible States
  - "execution time" = 0

Compressible States

Uncompressed Model

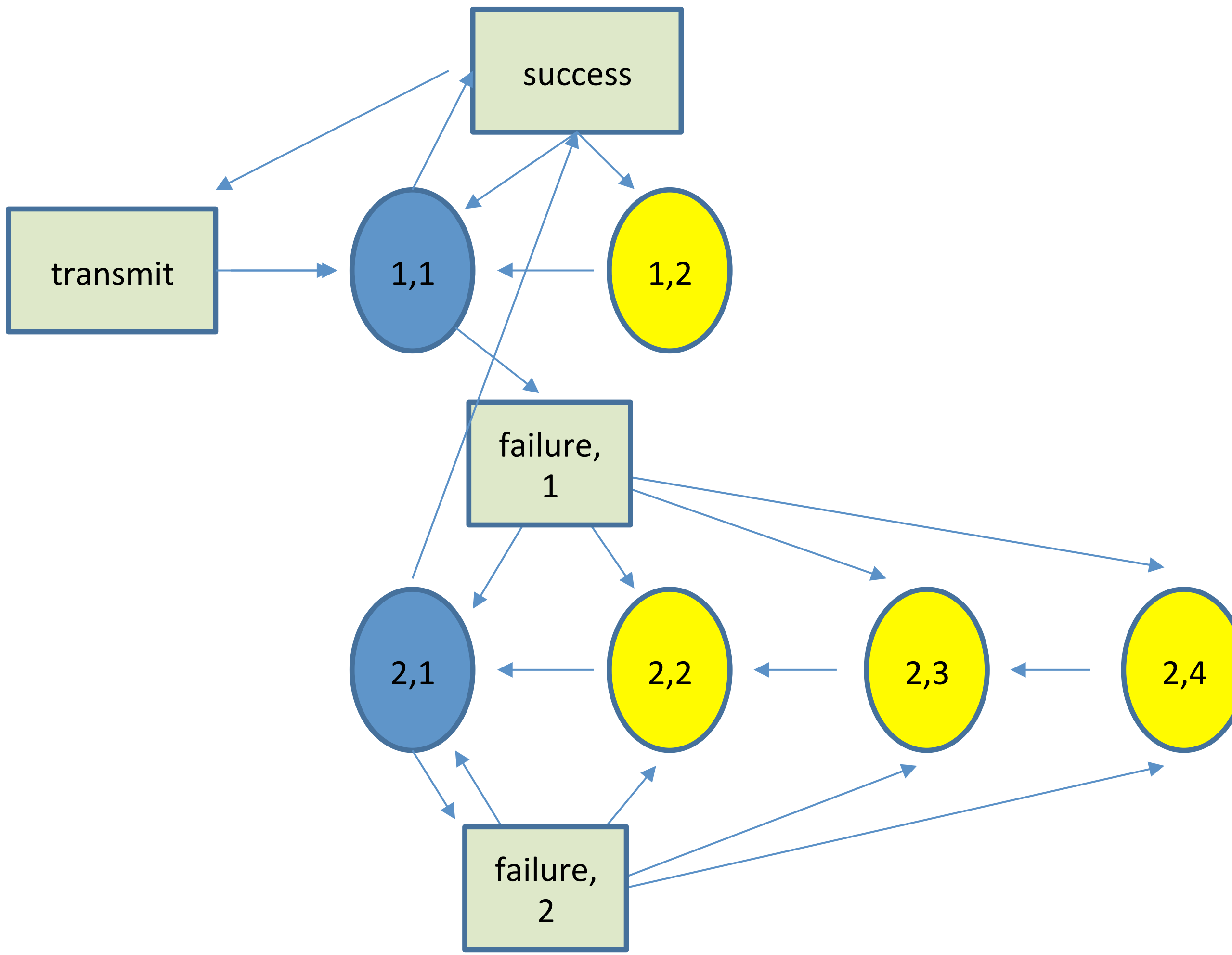


Compressed Model



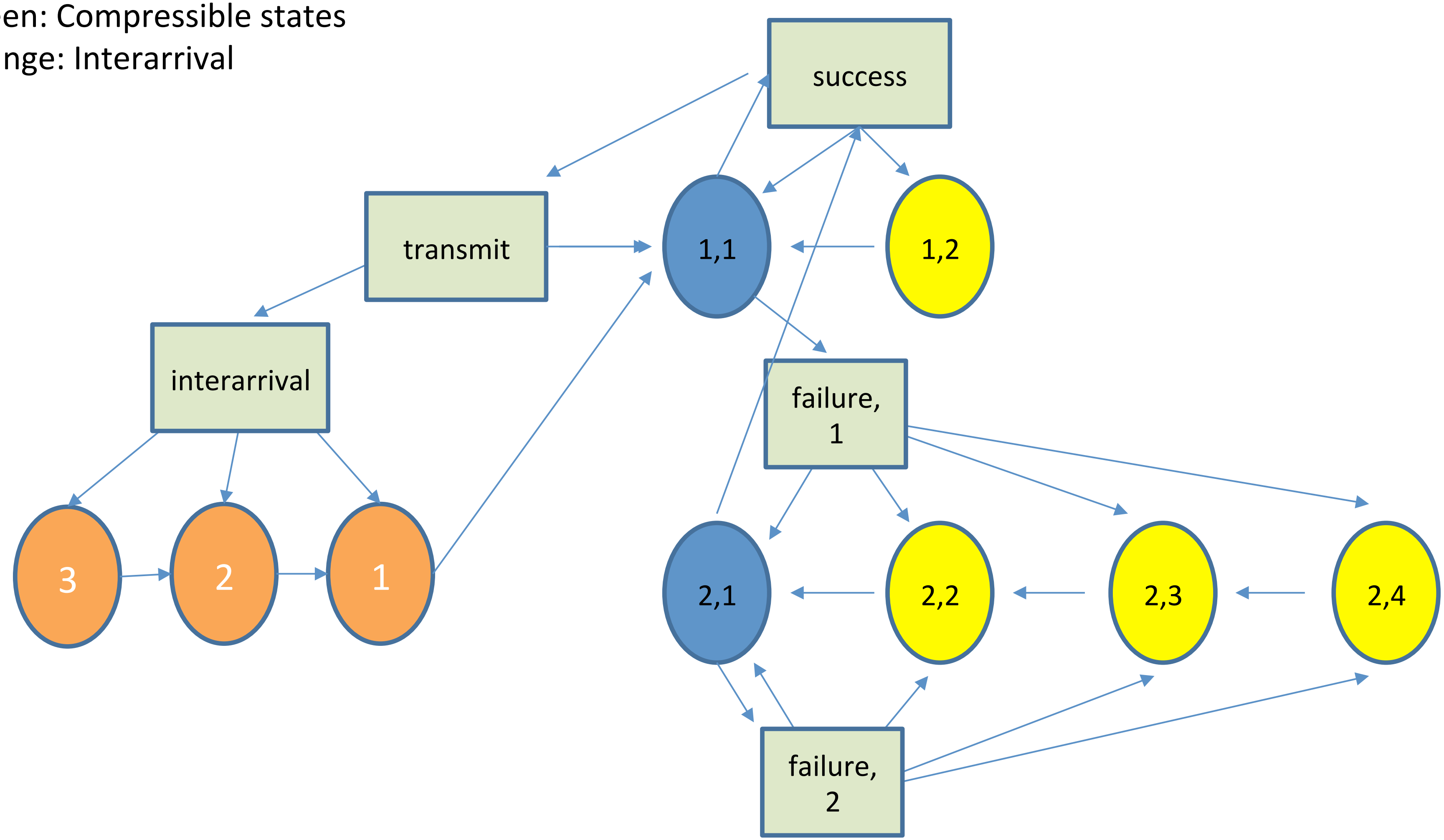
Compressible DCF

- Blue: Original DCF Transmit
- Yellow: Original DCF Backoff
- Green: Compressible states



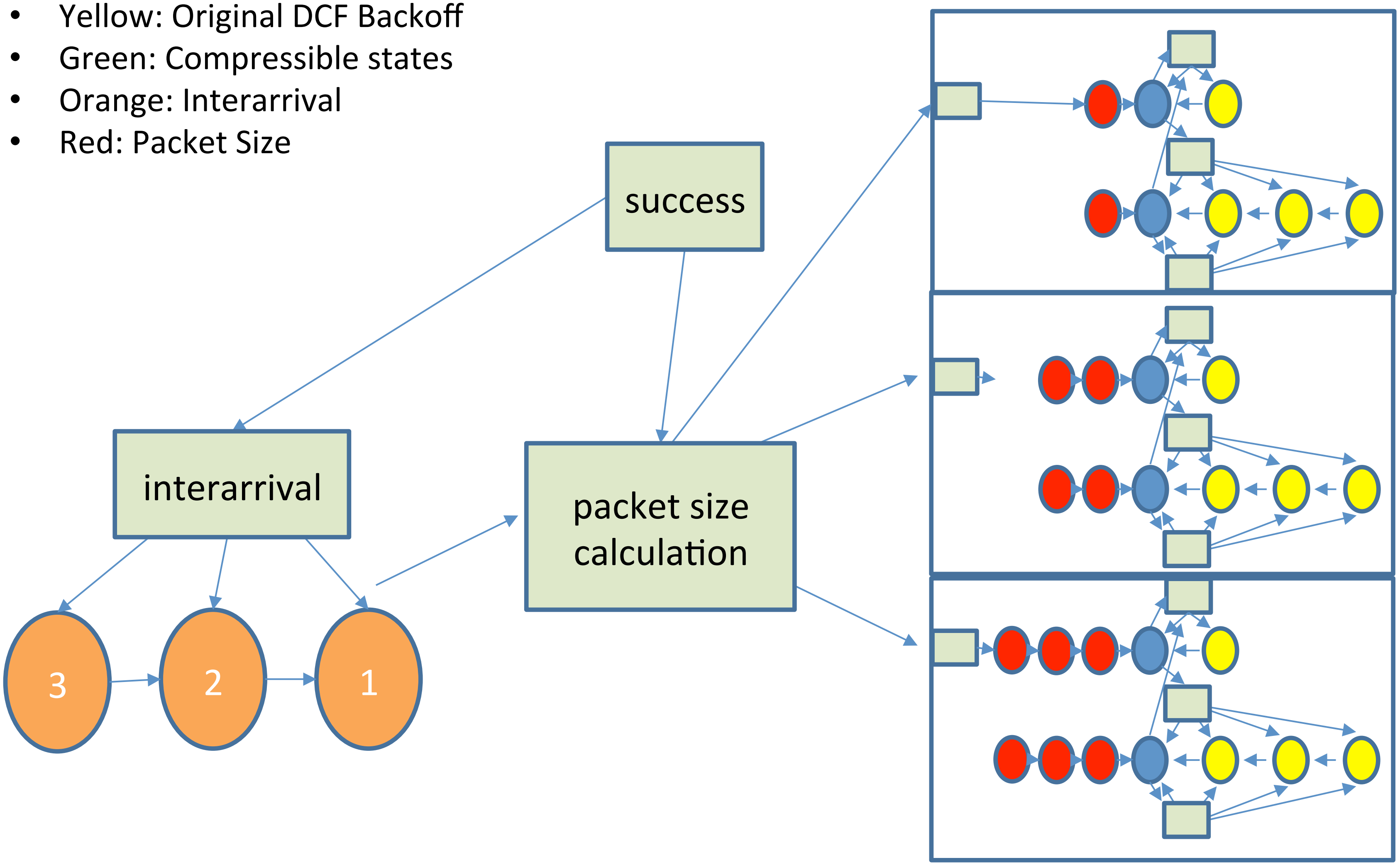
Adding Interarrival

- Blue: Original DCF Transmit
- Yellow: Original DCF Backoff
- Green: Compressible states
- Orange: Interarrival

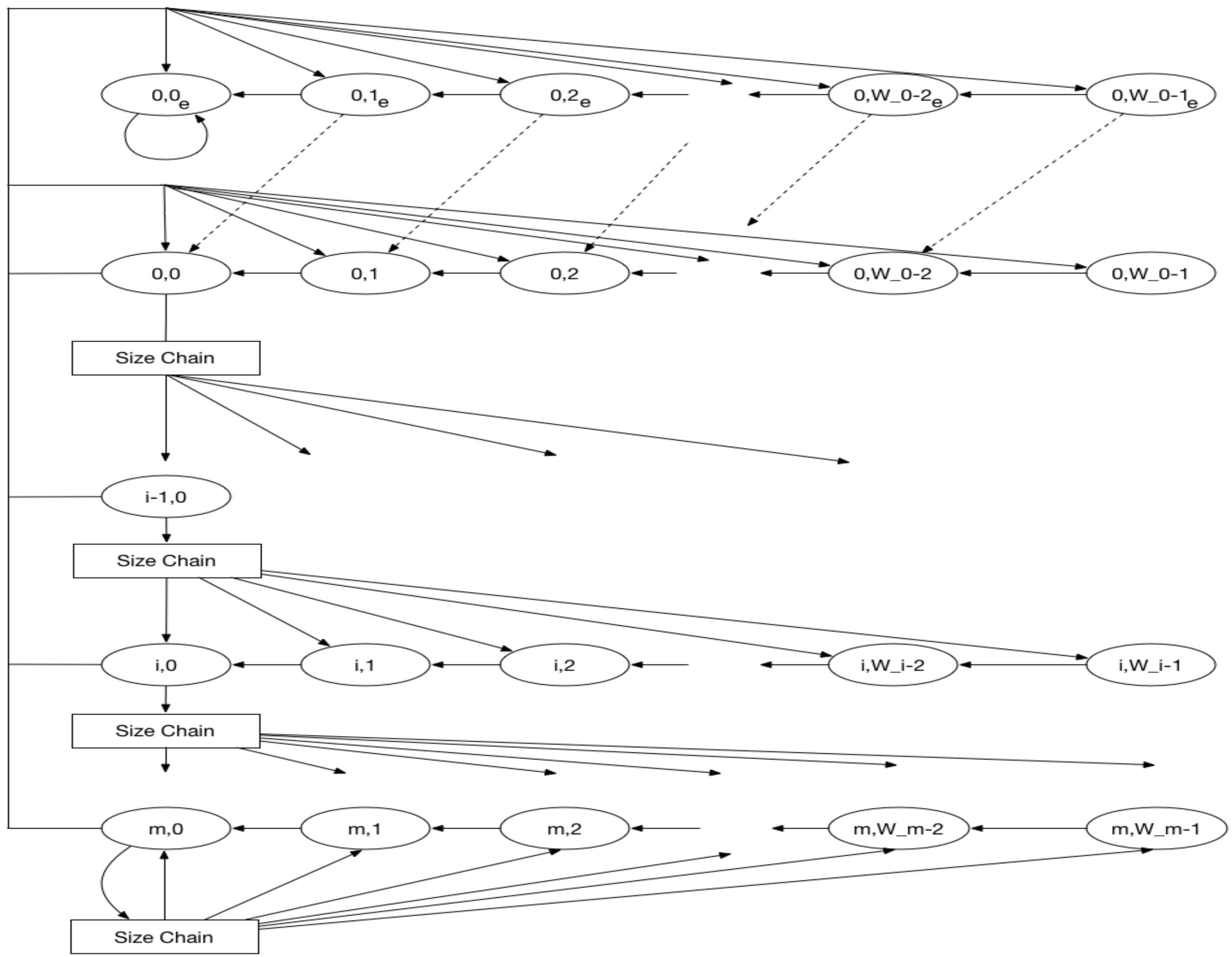


Adding Packetsize Chains (a new dimension)

- Blue: Original DCF Transmit
- Yellow: Original DCF Backoff
- Green: Compressible states
- Orange: Interarrival
- Red: Packet Size



Variable Packet Size



Varying Packet Types

- Packet types differ in size and frequency of arrival
  - I-Frame packets are larger but less frequent. B/P-Frame packets are more numerous but smaller
  - Web traffic may download a small HTML file followed by many large images.
  - Each packet type has a “fixed” packet size, decided at the beginning of the lifetime of the packet
    - Fixed for the lifetime of the duration, but “calculated” through the markov model
- Packet type transitions may be deterministic

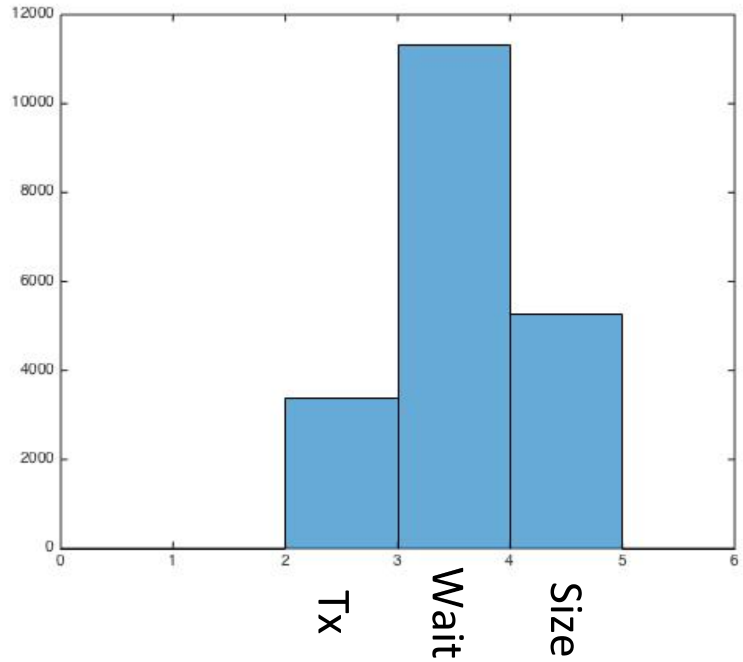
Traffic Models

- We focus on three types of traffic
  - Multimedia
  - Web browsing
  - File downloads
- Each traffic can be characterized according to
  - Packet size
  - Interarrival time and packet queue saturation
  - Type of packet

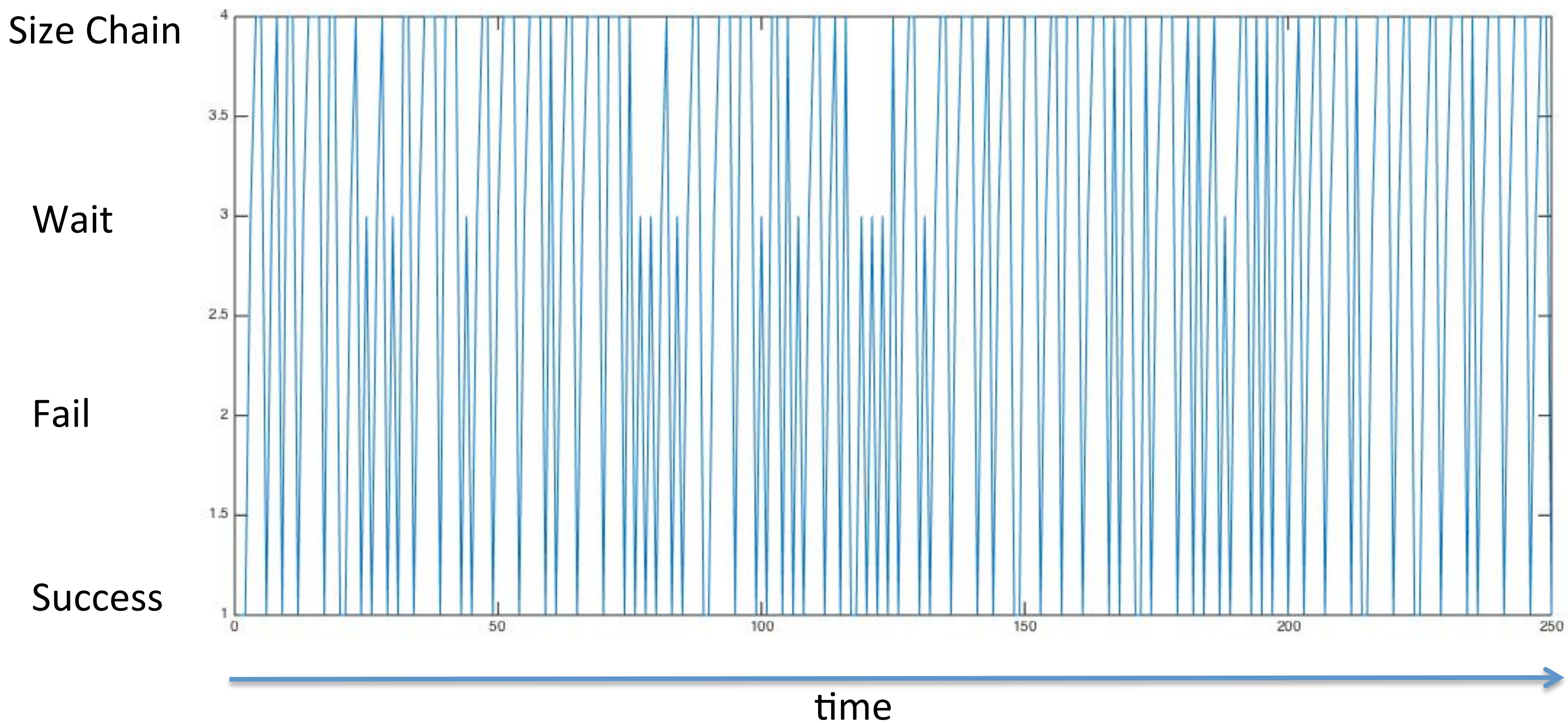
Consequence: our model needs to *parameterize* each of these characteristics to model each type of traffic

Web Traffic

- Primary characteristics
  - Randomly sized packets
  - Random interarrival time
- Parameters to tune:
  - Packet size (random)
  - Interarrival time (random)

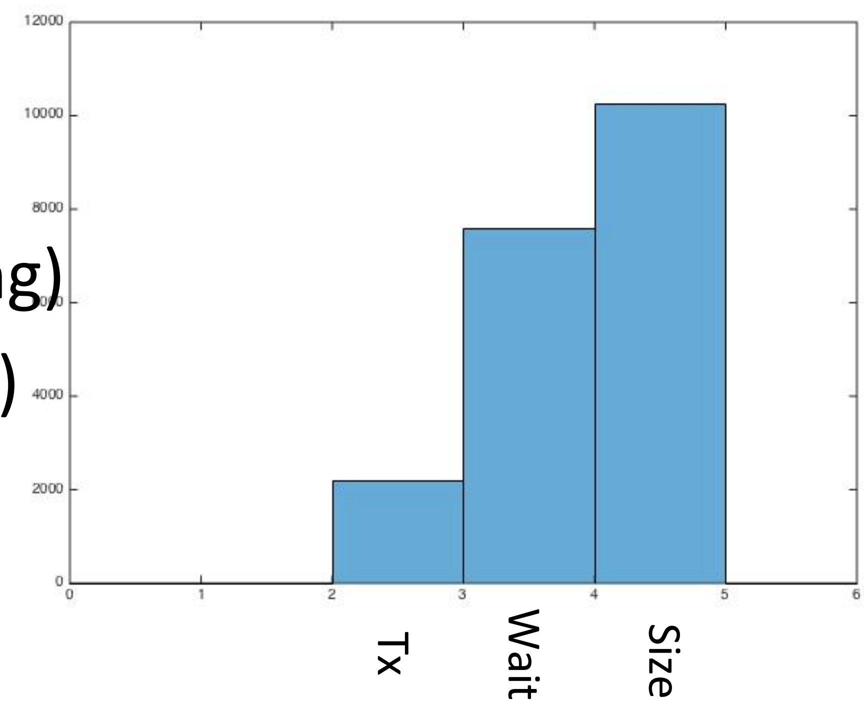


Web Traffic Illustrated

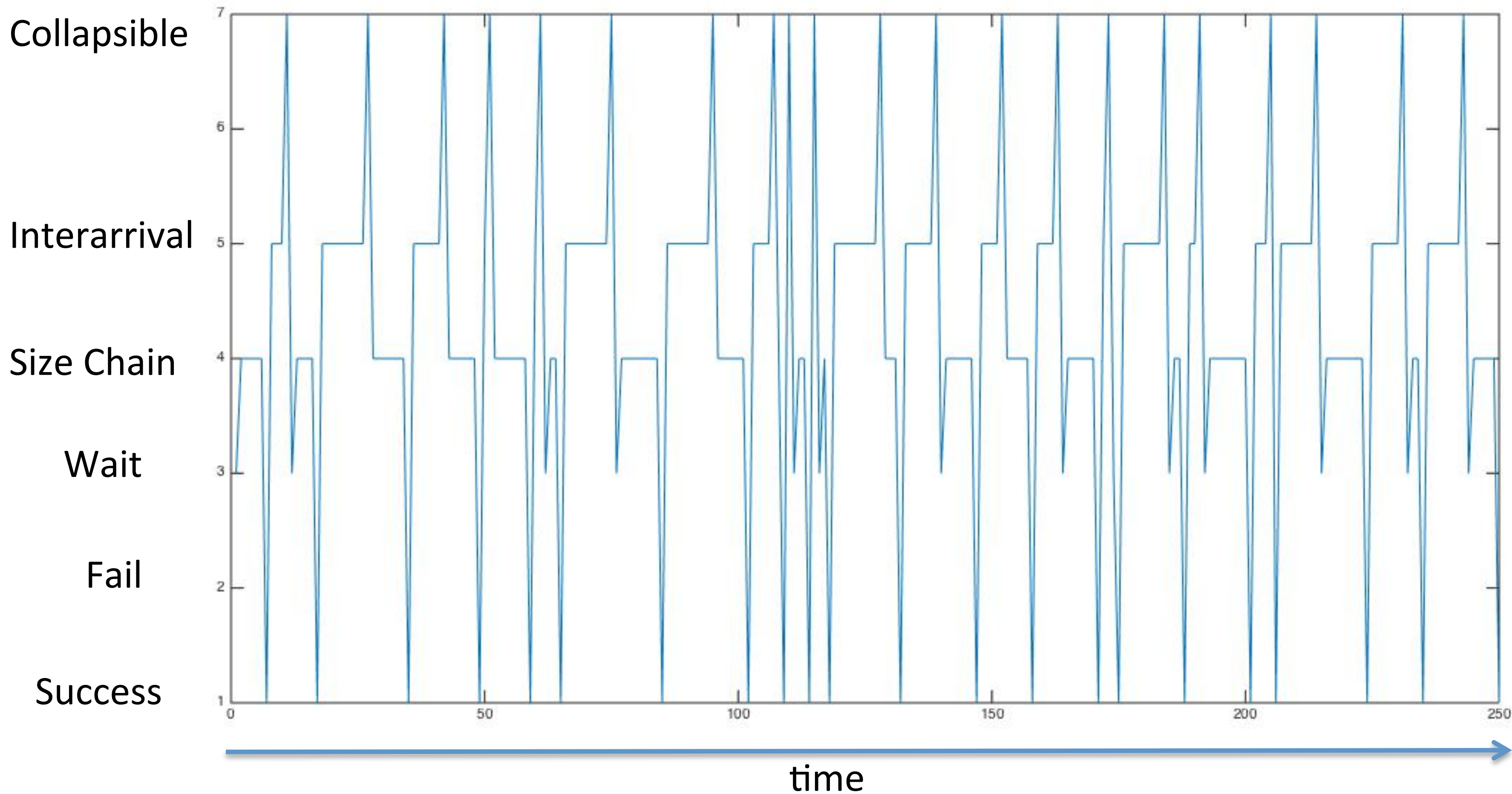


File Downloads

- Primary characteristics:
  - Long periods of activity and long inter-burst time
  - Usually large packet sizes
- Parameters to tune:
  - Packet size (high)
  - Queue arrival time (long)
  - Interarrival time (short)

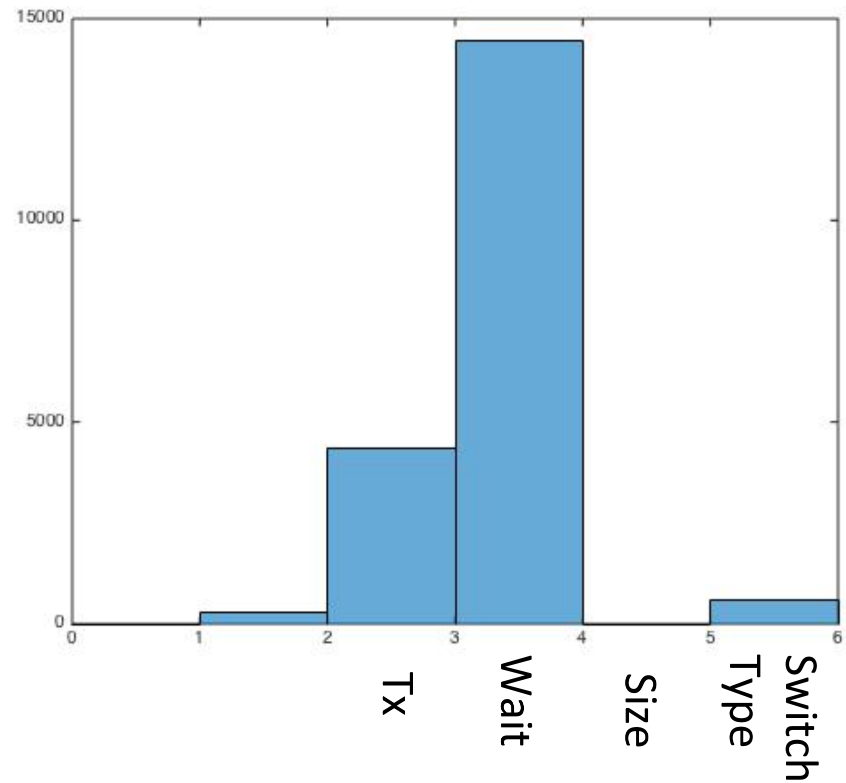


File Downloads Illustrated



Multimedia Traffic

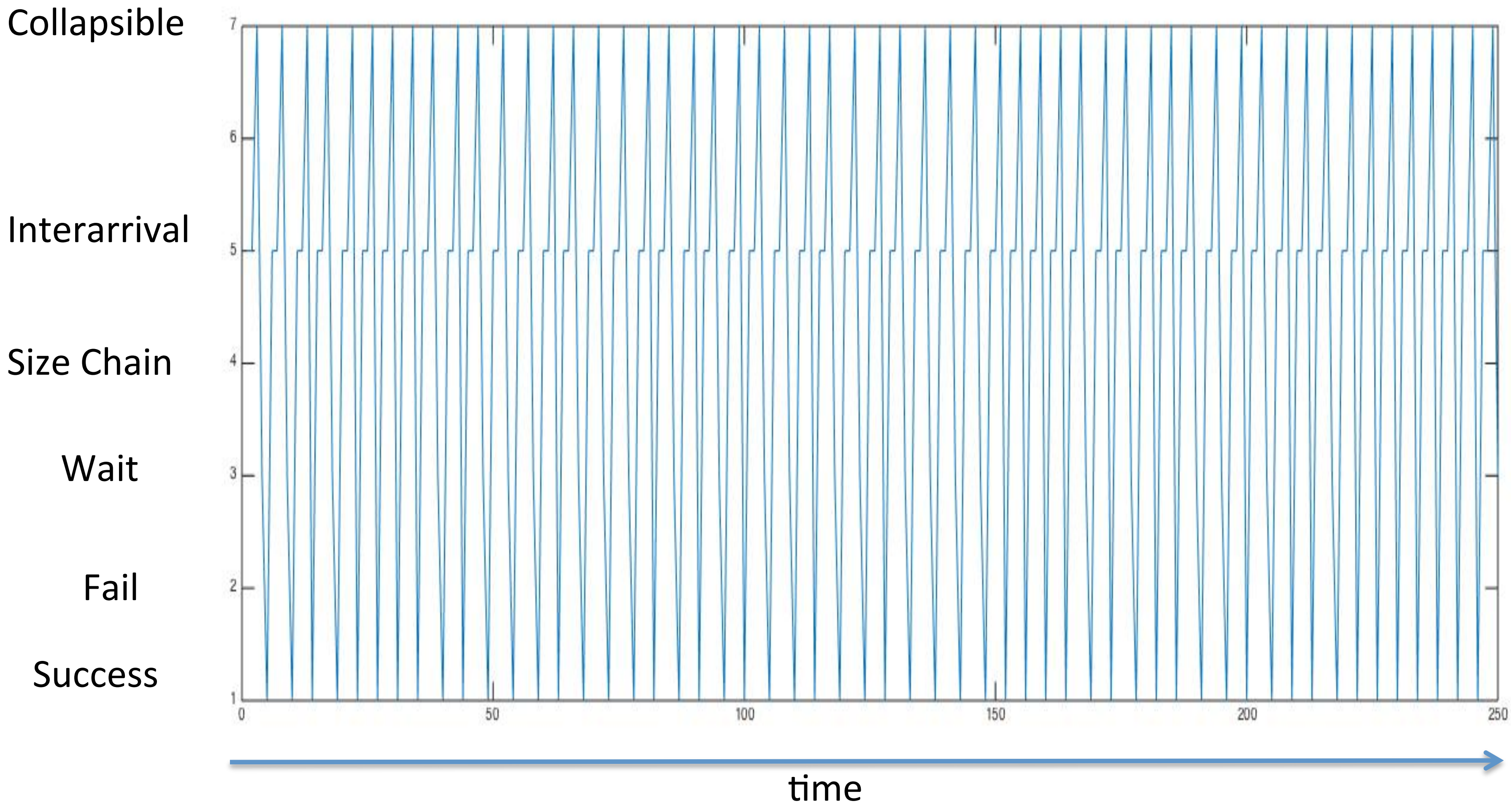
- Primary characteristics:
  - Streams of different types of packets, each of a different length
- Parameters to tune:
  - Packet size
  - Packet type



Multimedia Traffic

- MPEG-4 packet types include I/B/P Frames
  - GOP (group of pictures) ordering determines type
    - I B B P B B P B B P…etc
  - I-Frames are key frames, full images, no references
    - 1 I-Frame per GOP. Around 2/second
  - P-Frames use data from previous frames
    - Comes in repeated groups of BBP. Around 10/sec
  - B-Frames are bi-directional. Heavily compressed
    - Around 20/second

Multimedia Illustrated

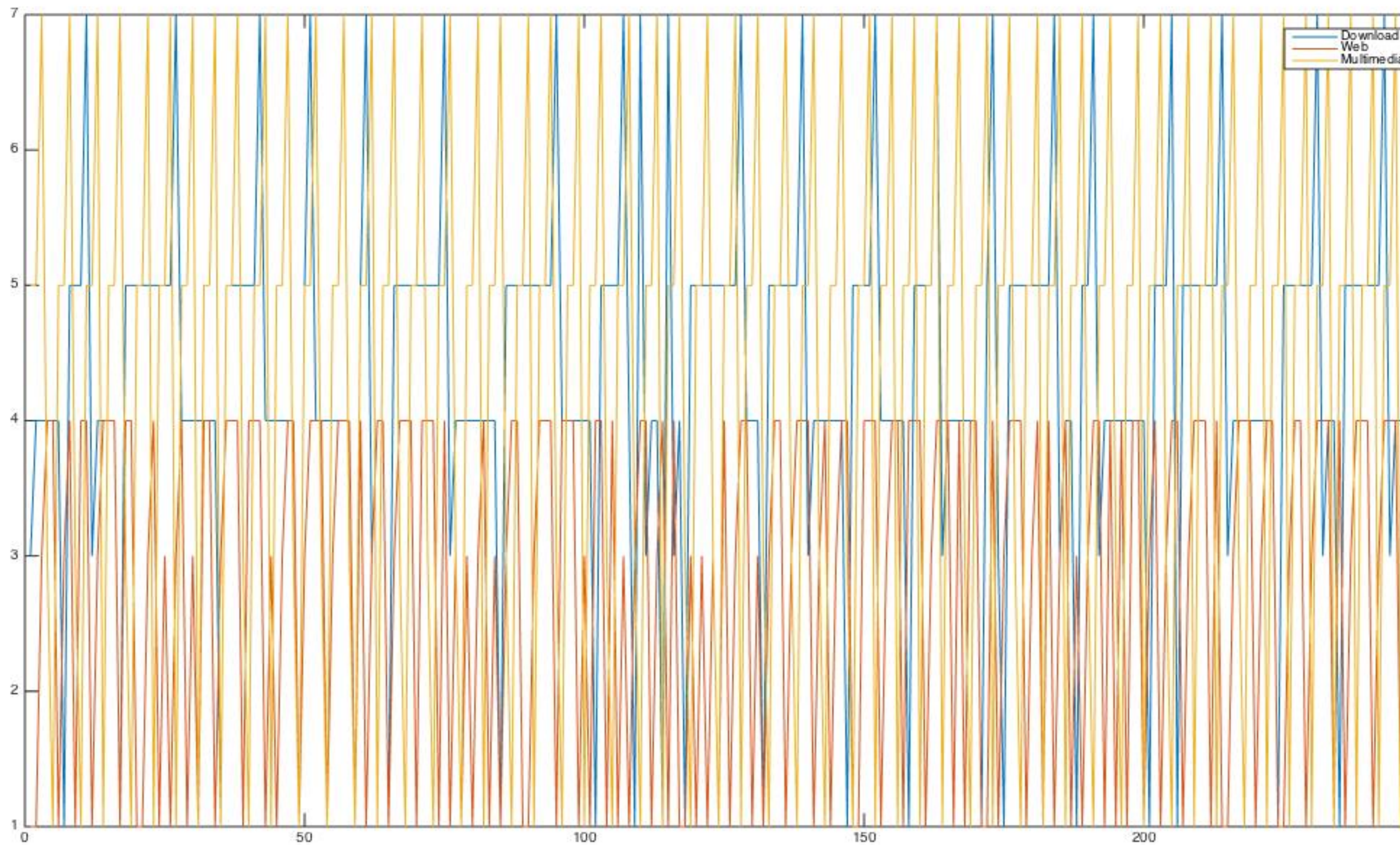


All Together

Web Browsing

File Downloads

Multimedia Streaming



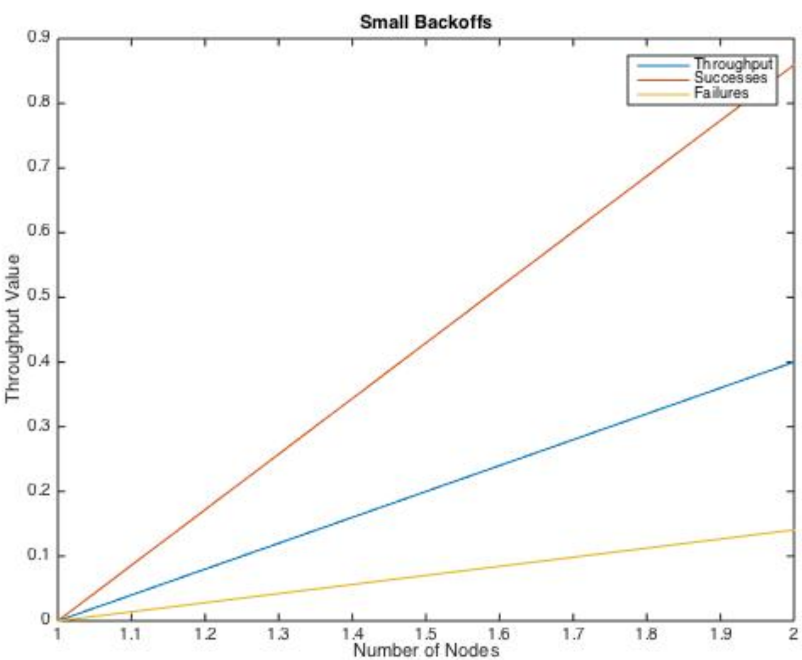
time

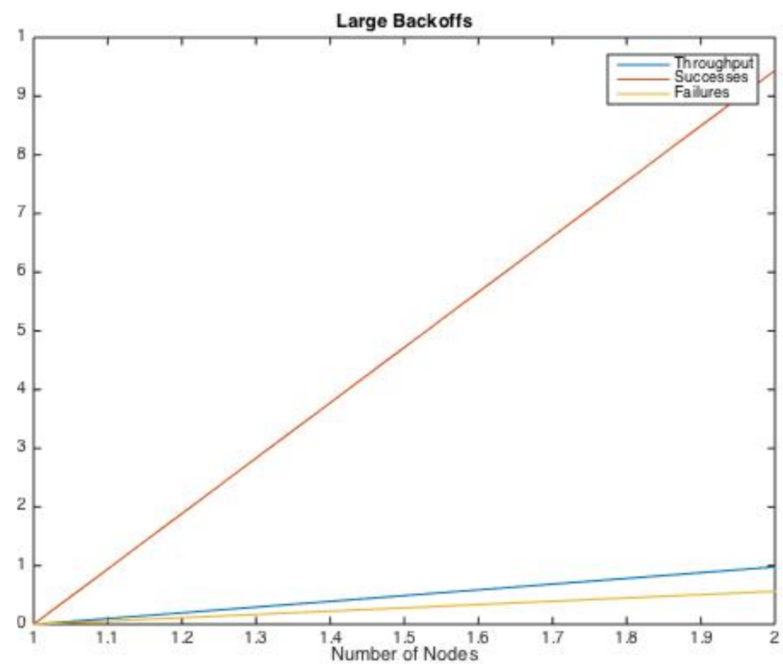
Experimental Metrics

- Average and individual node:
  - **Throughput (cumulative and variance)**
  - Idle time (backoff, post-backoff)
  - Packet transmit probability
  - Packet loss probability

The Case for Small Backoffs

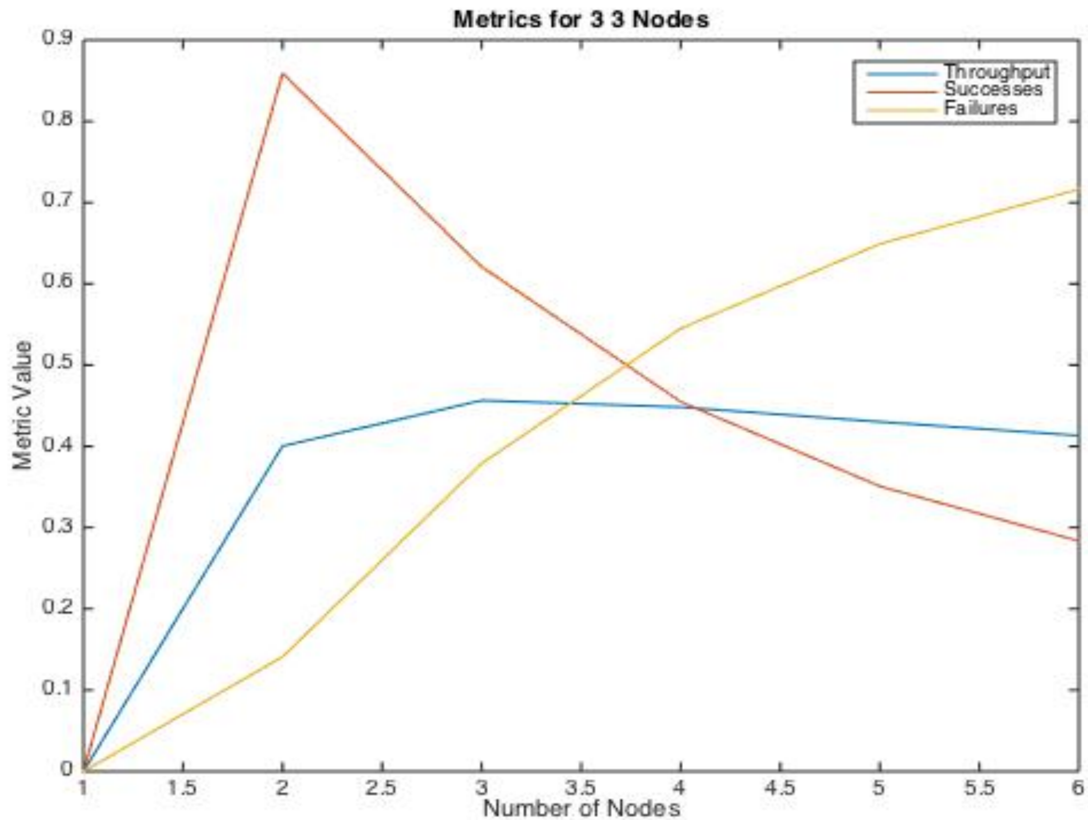
Simulations with large backoff values make metrics less "visible"





The Case for Few Nodes

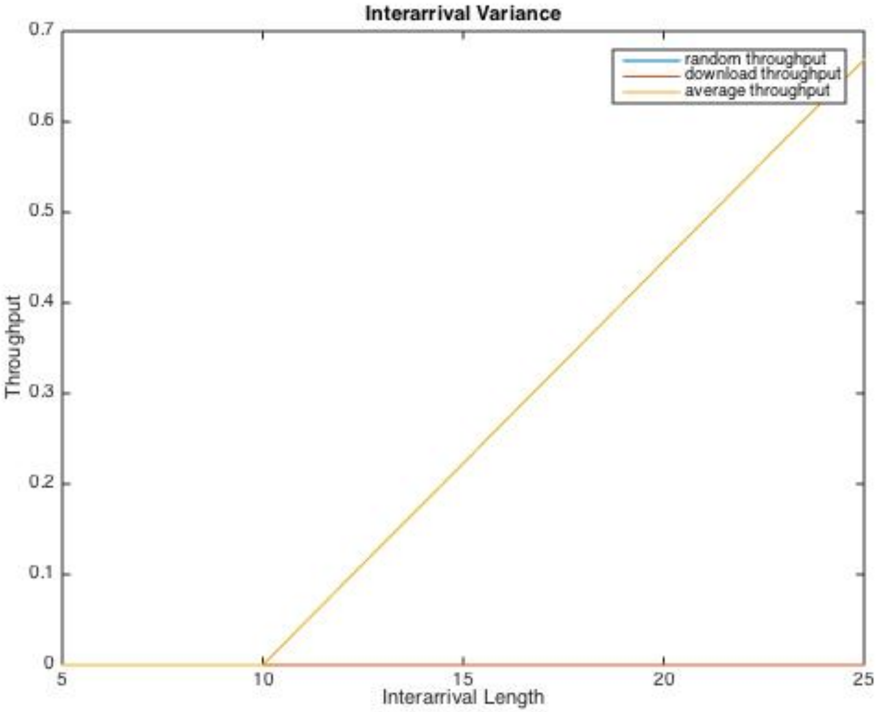
Simulations with many nodes obfuscate the effects of traffic parameters



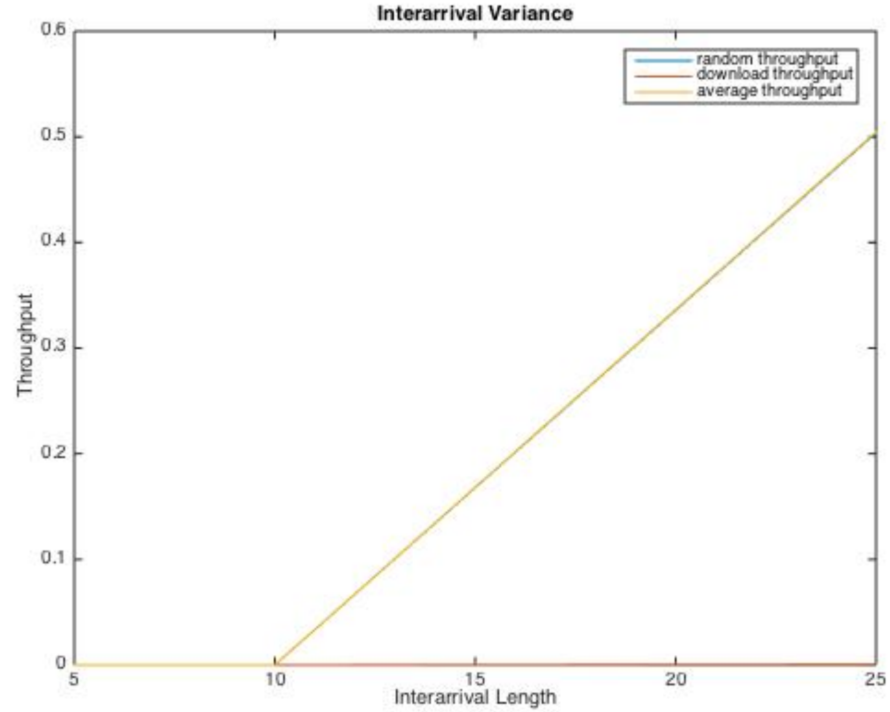
Experimental Setup

- Choose one **fixed** random node
- Choose another node of the target traffic type and vary a **single** parameters,
  - Interarrival time (chain length)
  - Packet size distribution
  - Likelihood of entering an interarrival chain

File Downloads: Varying Interarrival Time

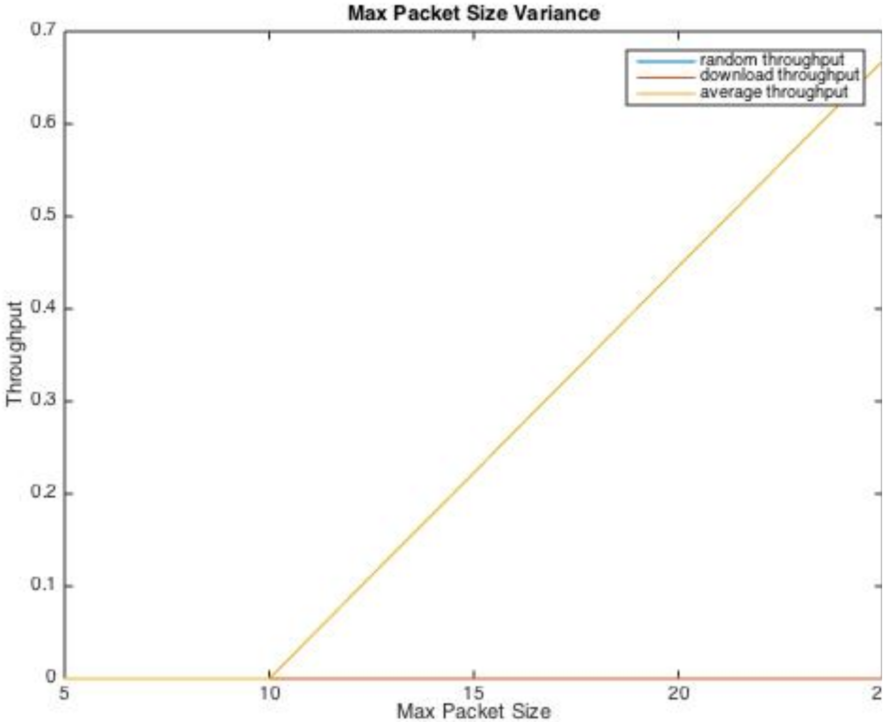


Packet Size = [1]

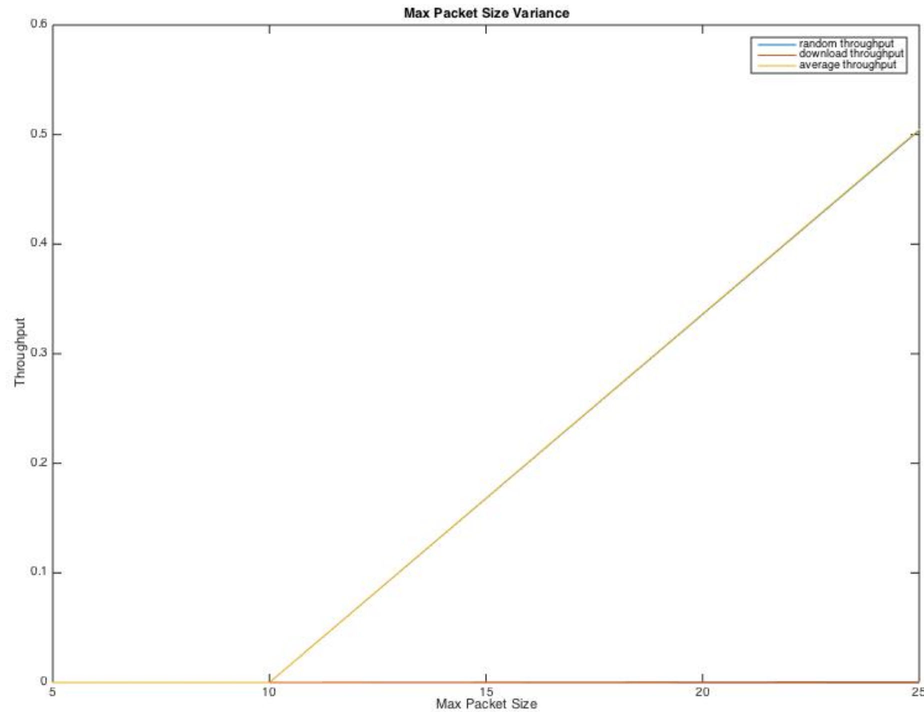


Packet Size = [1,2,3,4,5]

File Downloads: Varying Packet Size



Packet Size = [1]

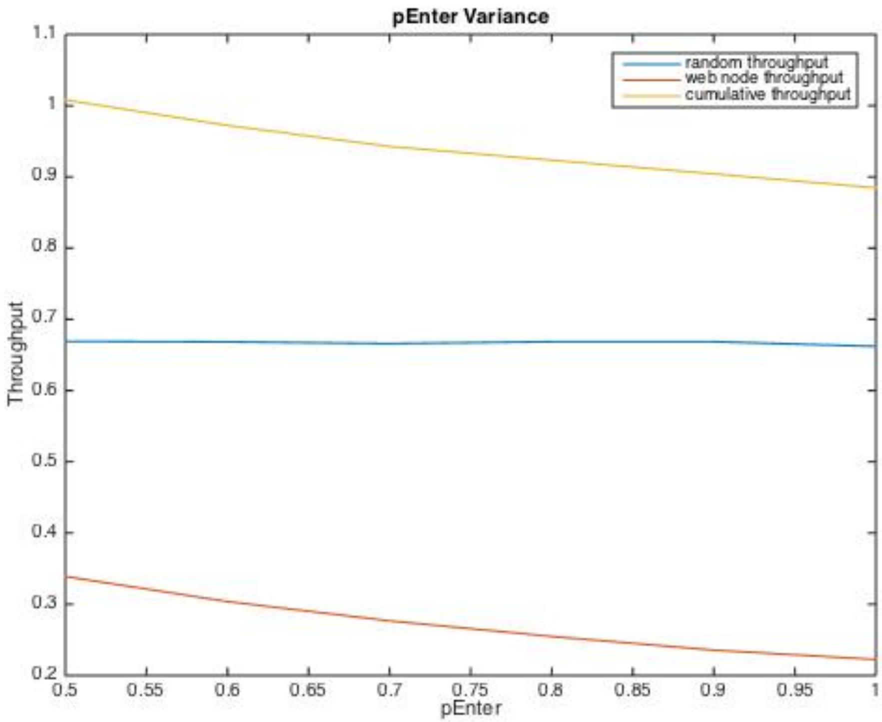


Packet Size = [1,2,3,4,5]

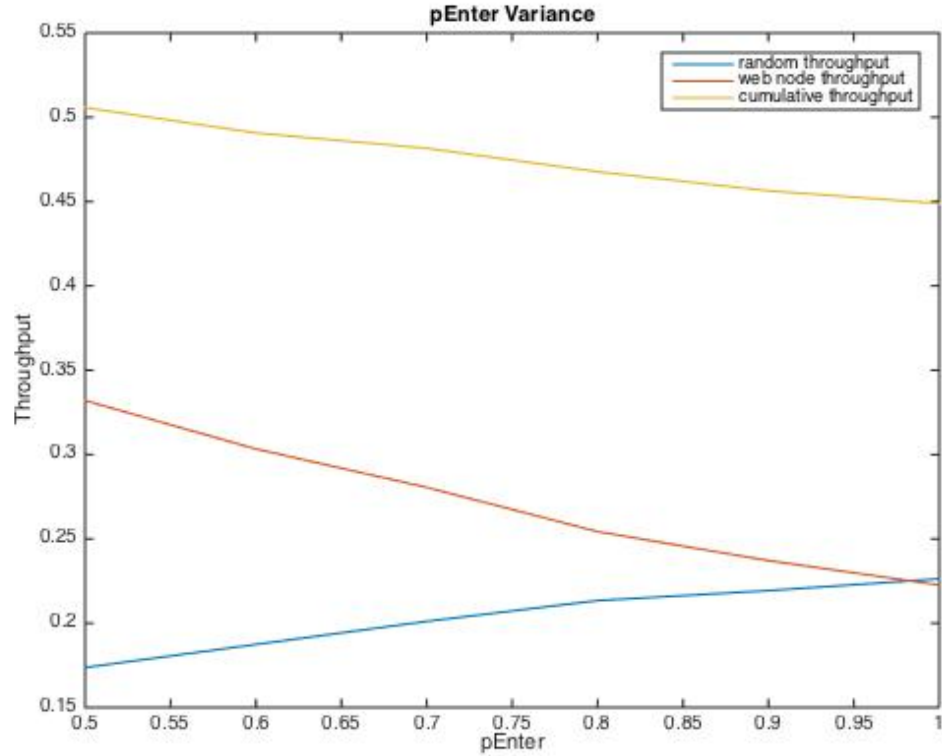
Observations

- **Average throughput** increases as the time between file download bursts of packets increases
- **Average throughput** increases as the size of packets increases
- **Download throughput** is starved due to increasing packet sizes and long interarrival time (longer time between attempts)
- Long times between bursts leave more room for other nodes to transmit

Web Browsing: Probable Traffic Gaps



Packet Size = [1]

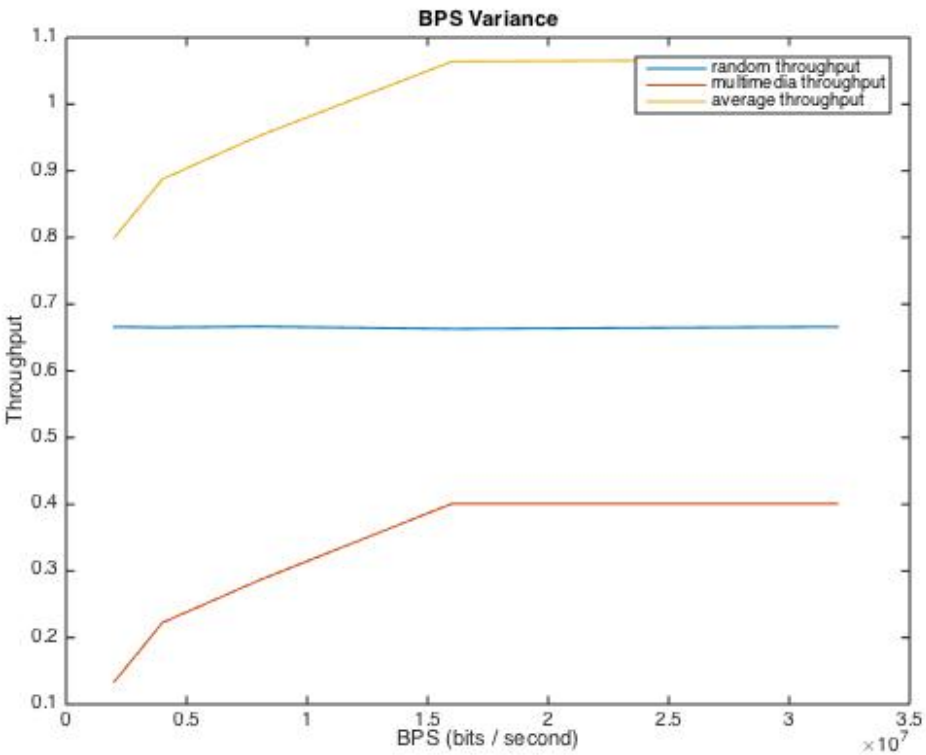


Packet Size = [1,2,3,4,5]

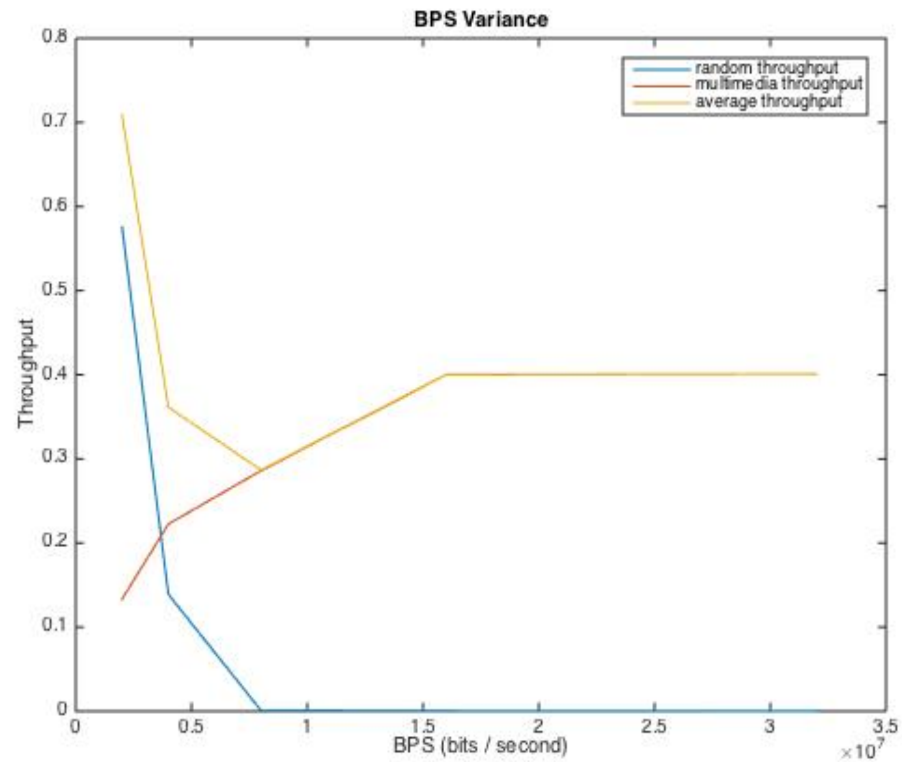
Observations

- For small gaps, random node throughput unaffected
- For large gaps, random node throughput increases
- **Overall:** Sporadic nature of web traffic characteristics favors other nodes in the system

Multimedia: Varying Streaming BPS



Packet Size = [1]



Packet Size = [1,2,3,4,5]

Observations

- Higher streaming constraints starve other nodes in the network
  - Multimedia node throughput increases and random node throughput decreases as a function of streaming quality
- Small packet sizes for random nodes aren't affected by multimedia streaming
  - Probable cause: streams of traffic are *mixed together*

Conclusions

- The 802.11 DCF function behaves differently depending on the type of traffic being transmitted
- Throughput quickly degrades as more than one node is using the channel (higher collision probability)
- Application-agnostic random backoffs are perhaps not the best choice for collision avoidance when dealing with non-multimedia streams
- Random backoffs serve multimedia traffic quite well in the presence of other nodes

Questions?