Modelling data networks

Richard G. Clegg (richard@richardclegg.org)

1 Introduction

There are many difficulties to modelling the internet, for a well-known and excellent summary see [6]

- The internet is big (and growing).
- The internet is heterogeneous to a large degree.
- No central maps exist of the internet.
- The internet is not always easy to measure.
- The internet is rapidly changing.
- It is extremely important to be able to model the internet.

The internet cannot possibly be modelled, yet we must model the internet. How can this be resolved?

- How you model the network depends critically on the problem you are solving.
- What are you trying to show with your model?
- Metrics: what are we trying to measure?
  1. Throughput?
  2. Goodput?
  3. System efficiency?
- Validation: what real data can be used to check the model?
- Sensitivity: what happens if your assumptions change?
  1. What if the demand on the system is slightly different?
  2. What happens if delays and bandwidths are changed?
  3. What happens if users stay longer or download more?

1.1 Example model – peer-to-peer network

Modelling Task: Test the possible improvements expected if we try a locality aware peer selection policy on a global bittorrent network.

What must our model include?

1. The distribution of nodes (peers) on the overlay network (not the whole network).
2. The delay and throughput between these peers (must depend on distance to some extent).
3. How users arrive and depart.
4. What users choose to download.

Note that this might already be a vast modelling task with hundreds of thousands or even millions of nodes.

- Research existing P2P models, do any fit? Don't reinvent the wheel.
- Real data: What real-life measurements exist to validate against?
- If we are modelling a new peer selection we must be sure our model covers existing peer selection well.
- Metrics: what must we measure in our model?
  1. Overall throughput/goodput?
  2. Distribution of time taken for peers to make their download?
  3. Total resources used in system?
- Validation: Instrumented P2P clients exist – how do they compare to our simulation.
- Sensitivity: Different distribution of users? Different delays and throughputs?

1.2 Example model – TCP throughput

Modelling Task: Test a possible improvement to the TCP model which aims to improve fairness and throughput when flows share a link.

What must our model include?

1. Individual packet model with existing TCP protocol as accurately as possible.
2. A reasonable estimate of how long each connection lasts and the rate at which new connections.

3. A model of the probability of round trip time for the parts of the connection not on the link being modelled.

4. A model of the probability of packet loss on the link (due to buffer overflow?)

- Can existing network models help (ns-2 could be an obvious choice)?

- What if the existing protocol shares a link with flows using the old protocol.

- Metrics:

  1. Throughput and goodput.
  2. Fairness between flows.

- Sensitivity, what if we change these parameters:

  1. Number of flows using existing and new protocol.
  2. Bandwidth of link.
  3. Round trip time of flows.
  4. Probability of packet loss.

- Validation: Does our model agree with real measurements?

1.3 Example model – Buffer provisioning model

Modelling task: Given a router with a buffer, how does the buffer size in packets affect the probability of packet loss?

What must our model include?

1. A model of the incoming packets to the buffer.

2. The rate at which packets leave the buffer.

3. Possibly distribution of packet lengths in bytes.

4. Possibly the feedback (TCP) between packet loss and arrival rate.

- Research: what is known about the statistics of internet traffic?

- What is the distribution of inter-arrival times and packet lengths?

- Metrics:

  1. Packet loss.
  2. Packet delay.

- Sensitivity: What if we change the following parameters:

  1. The total arrival rate.
  2. The bandwidth of the outgoing link.

- Validation: Real traffic traces (CAIDA has a collection).

1.4 Avoiding modelling when possible

Fermi Problems are named after physicist Enrico Fermi – quick and dirty estimation problems: typical example "How many dentists are in Chicago?"

- We might save ourselves modelling if we can show that our system can easily cope.
- For example estimate amount of data to download web site:
  1. How many users approx?
  2. How many visits do they make a day?
  3. How much do they each download on average?
- If the system can cope with each of these estimates being "worst case" then no further modelling may be needed.
- See "Consider a spherical cow?" [7]

2 Modelling Overview

2.1 Modelling areas

Now let us focus on several specific areas of interest to modellers.

1. Topology modelling — how are the nodes in the internet connected to each other?
   - See the internet as nodes and edges (graph theory).
   - Consider numbers of hops between nodes.
2. User/flow arrival modelling — how does traffic arrive on the internet?
   - See arrivals as a stochastic process (probability/statistics)
   - How long do connections last?
3. Application level protocols — what traffic do applications place on the internet?
   - For example peer-to-peer networks use an overlay (graph theory again?)
   - A web page might make connections to many different places.
4. Traffic statistics — what does the traffic along a link look like in statistical terms?
   - See internet traffic as a stochastic process (queuing theory).
   - How does TCP congestion control alter this?

5. Transport/network protocols — how do TCP/IP protocols affect the traffic?

    - See internet traffic as a feedback process (control theory).
    - How do these protocols interact with the rest of the network?

6. Other things to model:

    - Reliability modelling — what happens when links or nodes fail?
    - Overlay networks — P2P increasingly important.

2.2 Topology modelling

- Two levels of topology are usually considered "router level" and "autonomous system" (AS) level.
- Router level topology is still the least well-known — often ISPs take trouble to protect this information for security reasons.
- Topology metrics — these quantities are all rigorously defined and can be found in the literature:

    1. Graph diameter (longest possible "shortest path" between nodes).
    2. Node degree distribution (what proportion of nodes have k neighbours).
    3. Assortivity/disassortivity (do well-connected nodes connect with each other?) – sometimes called "rich club".
    4. Clustering (triangle count) – are the neighbours of a node also neighbours of each other.
    5. Clique size – largest group where everyone is everyone's neighbour (a clique in graph theory).

The node-degree distribution in AS networks is particularly well-studied. Let $P(k)$ be the proportion of nodes with degree k (having k neigbours). To a good approximation

$$P(k) \sim k^{-\alpha},$$

where α is a constant.

- Power law topology of the AS graph shown by Faloutsos et al [5].
- This graph has some interesting properties — some extremely highly connected nodes, what happens if they fail?
- Same type of graph as:

    1. Links on websites, wikipedia and many other similar online systems.
    2. Academic citations in papers.
    3. Human sexual contacts.

Albert–Barabasi [2] "Preferential attachment" model

Constructive model start with a small "core" network. When a new node arrives, attach it to an old node with the following probability

$$\mathbb{P}\left[\text{Attaching to node } i\right] = \frac{d(i)}{\sum_{j \in \text{all nodes}} d(j)},$$

where $d(i)$ is the degree of node i.

- This model "grows" a network with a powerlaw.
- Many similar models have been created which are more general.
- Current best model may be Positive Feedback Preference [9]. which adds a small "faster than exactly proportional" term.

Why not save work by using existing models to generate your topology?

1. Waxman model — considered old-fashioned now. Generates random connections between nodes on a plane.
2. GT-ITM model (Georgia Tech) — several models including a tiered model which models WAN, MAN and LAN separately.
3. Inet — AS level generator produces a random network with similar characteristics but possible issues with clustering and clique size.
4. BRITE — produces router and AS level topologies including power law, includes Waxman model, Barabassi–Albert model and generalisation.
5. Positive Feedback Preference — refinement to Barabassi Albert for AS level topology.
6. Igen — Router level topology generator which concentrates on geographical aspects.

2.3 User/flow arrival modelling

- As a first approximation the arrival of users can be modelled as a Poisson process.
- You might want to consider periodic effects:
    1. Daily – with people's sleep cycles.
    2. Weekly – weekends different.
    3. Yearly – year-on-year growth in traffic.
- Perhaps simpler just to simulate some peak hour and some estimate of growth?

2.4 Application level protocols

- If you are modelling a specific application there will be details associated with this.
- Common applications (www, ftp, p2p) will have existing research — read what is done before setting out on your own.
- If no studies are done what could you compare your application to?
- Could your application be viewed as:
  1. A series of ftp-like transfers of data.
  2. UDP bursts at a given rate for given periods of time
  3. A p2p application which might use existing p2p research methods.
- An important thing to simulate is the length of transfers and for many applications this is heavy-tailed [1].

A variable X has a heavy-tailed distribution if

$$\mathbb{P}[X > x] \sim x^{-\beta},$$

where $\beta \in (0, 2)$ and $\sim$ again means asymptotically proportional to as $x \to \infty$.

- Obviously an example of a power law.
- A distribution where *extreme values* are still quite common.
- Examples: Heights of trees, frequency of words, populations of towns.
- Best known example, Pareto distribution $\mathbb{P}[X > x] = (x/x_m)^{-\beta}$ where $x_m > 0$ is the smallest value X can have.
- The following internet distributions have heavy tails:
  1. Files on any particular computer.
  2. Files transferred via ftp.
  3. Bytes transferred by single TCP connections.
  4. Files downloaded by the WWW.
- This is more than just a statistical curiousity.
- Consider what this distribution would do to queuing performance (no longer Poisson).
- Non mathematicians are starting to take an interest in heavy tails (reference to "the long tail".

2.5 Traffic statistics

Long-Range Dependence (LRD) is considered to be an important characteristic of internet traffic.

- In 1993 LRD was found in a time series of bytes/unit time measured on an Ethernet LAN [Leland et al '93].
- This finding has been repeated a number of times by a large number of authors (however recent evidence suggests this may not happen in the core).
- A higher Hurst parameter often increases delays in a network. Packet loss also suffers.
- If buffer provisioning is done using the assumption of Poisson traffic then the network will probably be under-specified.
- The Hurst parameter is "a dominant characteristic for a number of packet traffic engineering problems".

Let $\{X_1, X_2, X_3, \ldots\}$ be a weakly stationary time series.
The Autocorrelation Function (ACF) is defined as

$$\rho(k) = \frac{\mathrm{E}\left[(X_t - \mu)(X_{t+k} - \mu)\right]}{\sigma^2},$$

where μ is the mean and σ^2 is the variance.

The ACF measures the correlation between X_t and X_{t+k} and is normalised so $\rho(k) \in [-1, 1]$. Note symmetry $\rho(k) = \rho(-k)$.

A process exhibits LRD if $\sum_{k=0}^{\infty} \rho(k)$ diverges (is not finite).

Definition of Hurst Parameter

The following functional form for the ACF is often assumed

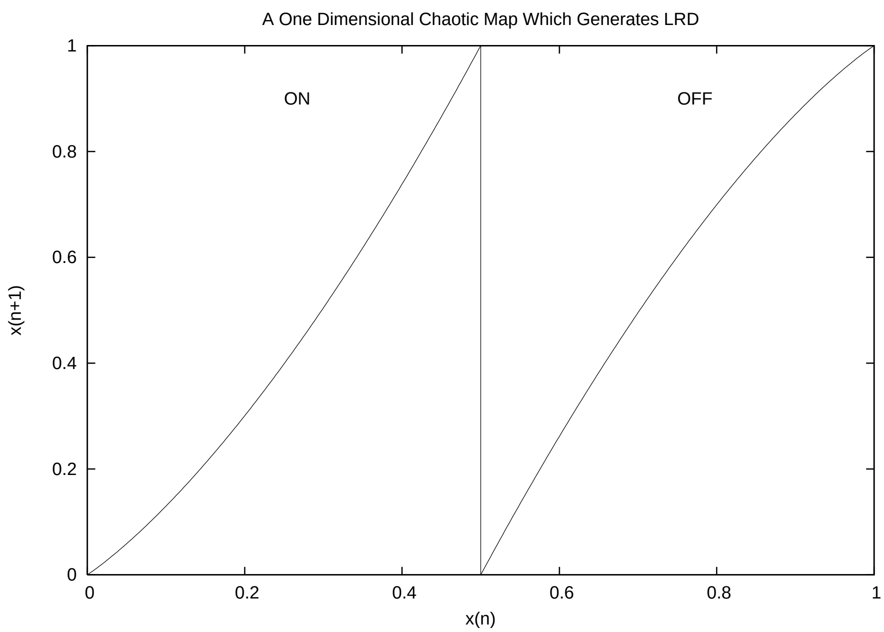
$$\rho(k) \sim |k|^{-2(1-H)},$$

where $\sim$ means asymptotically proportional to and $H \in (1/2, 1)$ is the Hurst Parameter.

- Think of LRD as meaning that data from the distant past continue to effect the present.
- LRD was first spotted by a hydrologist (Hurst) looking at the flooding of the Nile river.
- For this reason Mandelbrot called it "the Joseph effect".
- Stock prices (once normalised) also show LRD.

- LRD can also be seen in the temperature of the earth (once the trend is removed).
- Models include Markov chains, Fractional Brownian Motion (variant on Brownian motion), Chaotic maps and many others [4].

Iterated map model for LRD.



$$x_{n+1} = \begin{cases} x_n + \frac{1-d}{d^{m_1}} x_n^{m_1} & 0 < x_n < d, \\ x_n - \frac{d}{(1-d)^{m_2}} (1 - x_n)^{m_2} & d < x_n < 1, \end{cases}$$

where $x_n, d \in (0,1)$, $m_1, m_2 \in (3/2, 2)$. Produces ON and OFF series — packets and not packets with Hurst $H = min(m_1, m_2) - 1$.

2.6 Transport and network level protocols

- It might be important if we are considering a packet level model to model specific details of the TCP/IP protocols.
- Usually this will involve simulating the window size (additive increase multiplicative decrease) of the TCP protocol.
- Remember that a detailed simulation to this level will extremely limit the number of nodes which can be simulated.
- A mathematical model will be demonstrated in the next section.
- In addition, the ns-2 model will be shown which is a packet level simulation of TCP/IP.

2.7 Other things to model

- Of course depending on the nature of your modelling, there may well be other aspects of the network to be modelled.
- Some examples might be:
  1. Reliability of nodes and links.
  2. An overlay network.
  3. Possible hostile attacks to the network.
- In all cases, an important starting point is to find out what research already exists in the area.
- Are any real-life data sets available which could inform your modelling? Could you gather such data?

3 Mathematical modelling

- To create a simulation model we need to be able to write down equations for the system.
- The more work we can do "on paper" the easier the computational burden.
- This will be illustrated with two mathematical models related to networks.
- The first model is a buffer model using Markov chains.
- The second model is a model of TCP/IP to estimate throughput.
- These models can be used as a basis for computer simulation.

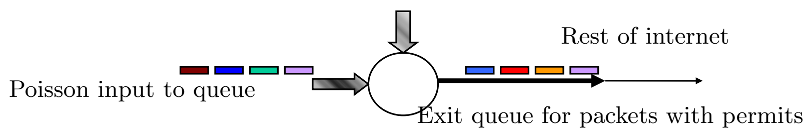
3.1 A Markov model for the leaky-bucket

- A "leaky bucket" is a mechanism for managing buffers and to smooth downstream flow.
- What is described here is sometimes known as a "token bucket".
- A queue holds a stock of "permit" generated at a rate r (one permit every $1/r$ seconds) up to a maximum of W.
- A packet cannot leave the queue without a permit – each packet takes one permit.
- The idea is that a short burst of traffic can be accommodated but a longer burst is smoothed to ensure that downstream can cope.
- Assume that packets arrive as a Poisson process at rate λ.

- A Markov model will be used [3, page 515].

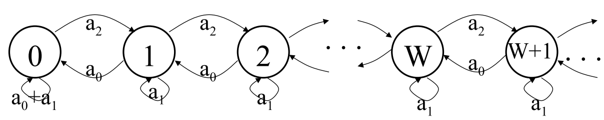
Use a discrete time Markov chain where we stay in each state for time $1/r$ seconds (the time taken to generate one permit). Let a_k be the probability that k packets arrive in one time period. Since arrivals are Poisson,

$$a_k = \frac{e^{-\lambda/r}(\lambda/r)^k}{k!}.$$



- In one time period (length $1/r$ secs) one token is generated (unless W exist) and some may be used sending packets.
- States $i \in \{0, 1, \dots, W\}$ represent no packets waiting and $W - i$ permits available. States $i \in \{W + 1, W + 2, \dots\}$ represent 0 tokens and $i - W$ packets waiting.
- If k packets arrive we move from state i to state $i + k - 1$ (except from state 0).
- Transition probabilities from i to j, $p_{i,j}$ given by

$$p_{i,j} = \begin{cases} a_0 + a_1 & i = j = 0 \\ a_{j-i+1} & j \geq i - 1 \\ 0 & \text{otherwise} \end{cases}$$



Let π_i be the equilibrium probability of state i. Now, we can calculate the probability flows in and out of each state.

For state one

$$\pi_0 = a_0\pi_1 + (a_0 + a_1)\pi_0$$
$$\pi_1 = (1 - a_0 - a_1)\pi_0/a_0.$$

For state $i > 0$ then $\pi_i = \sum_{j=0}^{i+1} a_{i-j+1}\pi_j$. Therefore,

$$\pi_1 = a_2\pi_0 + a_1\pi_1 + a_0\pi_2$$
$$\pi_2 = \frac{\pi_0}{a_0}\left(\frac{(1 - a_0 - a_1)(1 - a_1)}{a_0} - a_2\right).$$

In a similar way, we can get π_i in terms of $\pi_0, \pi_1, \ldots, \pi_{i-1}$.

- We could use $\sum_{i=0}^{\infty} \pi_i = 1$ to get result but this is difficult.
- Note that permits are generated every step except in state 0 when no packets arrived (W permits exist and none used up).
- This means permits arrive at rate $(1 - \pi_0 a_0)r$.
- Rate of tokens arriving must equal λ unless the queue grows forever (each packet gets a permit).
- Therefore $\pi_0 = (r - \lambda)/(ra_0)$.
- Given this we can then get π_1, π_2 and so on.

To complete the model we want to calculate T average delay of a packet.

- If we are in states $\{0, 1, \ldots, W\}$ packet exits immediately with no delay.
- If we are in states $i \in \{W+1, W+2, \ldots\}$ then we must wait for $i - W$ tokens $(i - W)/r$ seconds to get a token.
- The proportion of the time spent in state i is π_i.
- The final expression for the delay is

$$T = \frac{1}{r} \sum_{j=W+1}^{\infty} \pi_j (j - W).$$

- For more analysis of this model see [3, page 515].

3.2 A model of TCP throughput

This work is taken from Padhye et al 1998 [8].

Figure 1 shows in diagramtic form some of the quantities we need to define to study triple duplicate (TD) ACK packet loss.

Assume that there is a constant probability p of a packet being lost. We split our mathematical model into periods which occur between losses. W_i is the window size at the end of the ith period. Define Y_i as the number of packets send in the ith period and A_i as the time that the ith period took. It in the long term we can say that the bandwidth B in pkts/sec is given by:

$$B = \lim_{n \to \infty} \sum_{i=0}^{n} \frac{Y_i}{A_i} = \frac{\mathrm{E}[Y]}{\mathrm{E}[A]} \qquad (1)$$

which gives the obvious idea that the bandwidth is expectation of the number of packets sent over the time taken to send them.

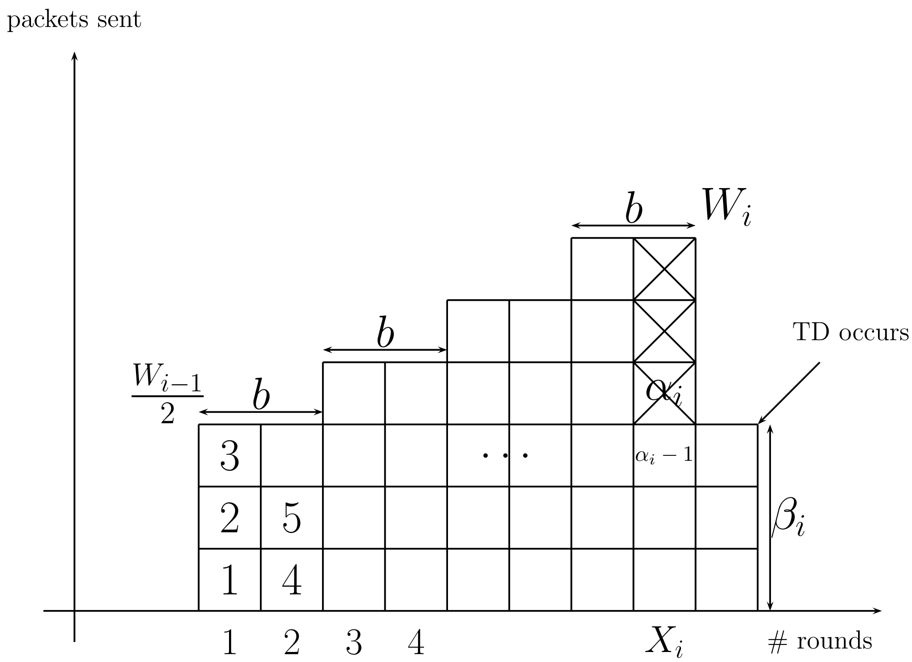


Figure 1: A diagram showing the TCP window size with triple duplicate ACK packet loss. α_i is the first packet to be lost.

Consider figure 1. After a TD loss, the window size is halved (if we ignore the exponential *slow start* part of the algorithm). Therefore the ith period begins with a window size of $W_{i-1}/2$. If we split each period into *rounds* as shown in the diagram then, after every b successful rounds, the window size is incremented by one. Denote by α_i the number of the first packet lost in the ith period (assume we number the packets from one beginning afresh every period). X_i is the number of the round in which this packet is lost.

Since, at the end of round i the window size is W_i then when this first packet is lost, $W_i - 1$ packets are outstanding already (since W_i is the number of unacknowledged packets at the end of the round). Therefore, a total of $Y_i = \alpha_i + W_i - 1$ packets are sent in the $X_i + 1$ rounds. Therefore:

$$\mathrm{E}[Y] = \mathrm{E}[\alpha] + \mathrm{E}[W] - 1. \tag{2}$$

Now, if p is the probability that a packet is lost then the probability that the kth packet is the first lost is given by:

$$\mathbb{P}[\alpha_i = k] = (1-p)^{k-1}p \quad k = 1, 2, \ldots$$

since this implies $k-1$ were received then one was lost. Thus:

$$\mathrm{E}[\alpha] = \sum_{k=1}^{\infty} (1-p)^{k-1} pk = \frac{1}{p} \tag{3}$$

It is instructive to go through this simple derivation

$$\sum_{k=1}^{\infty} (1-p)^{k-1} pk = p \sum_{k=1}^{\infty} q^{k-1}$$

where $q = (1-p)$. We notice that:

$$p \sum_{k=1}^{\infty} q^{k-1} k = p \frac{d}{dq} \left(\sum_{k=1}^{\infty} q^k \right).$$

Now, if $0 < p < 1$ then we know that $\sum_{k=1}^{\infty} pq^{k-1} = 1$ since this is the sum of the probability over all states (that is the probability that packet loss ever occurs). Some trivial rearrangement gives us $\sum_{k=1}^{\infty} q^k = q/(1-q)$. Therefore substituting above:

$$\sum_{k=1}^{\infty} (1-p)^{k-1} pk = p \frac{d}{dq} \left(\frac{q}{1-q} \right) = \frac{p}{(1-q)^2} = \frac{1}{p}$$

Combining (2) and (3) we therefore have:

$$\mathrm{E}[Y] = \frac{1-p}{p} + \mathrm{E}[W]. \tag{4}$$

Consider the round trip times of the packets (the time taken for a packet to be sent and an ACK to be received for it). The ith period has X_i complete rounds plus the partial round. We would therefore expect the duration of the round to be given by $A_i = (X_i + 1)RTT$ where RTT is the round trip time (assume that RTT is an independent random variable). Therefore:

$$\mathrm{E}[A] = (\mathrm{E}[X] + 1)RTT. \tag{5}$$

Now, we need to know the value of $E[W]$. Since, every b rounds the window size increases by one and we have X_i rounds then clearly we have:

$$W_i = \frac{W_{i-1}}{2} + \frac{X_i}{b} \quad i = 1, 2, \ldots \tag{6}$$

[We should note that this is an approximation since it allows window sizes to be fractional when,obviously, they must really be integer]. The number of packets transmitted in the ith period is given by:

$$Y_i = \sum_{k=0}^{X_i/b-1} \left(\frac{W_{i-1}}{2} + k \right) b + \beta_i. = \frac{X_i W_{i-1}}{2} + \frac{X_i}{2} \left(\frac{X_i}{b} - 1 \right) + \beta_i.$$

Substitution from (6)

$$Y_i = \frac{X_i}{2}\left(\frac{W_{i-1}}{2} + W_i - 1\right) + \beta_i. \tag{7}$$

From (6), if we make the assumption that X and W are i.i.d. random variables, (note that this assumption is not really necessary, we could model the behaviour of X_i as a Markov chain) we get:

$$\mathrm{E}\left[W\right] = \frac{2}{b}\mathrm{E}\left[X\right]. \tag{8}$$

From (4) and (7) we get:

$$\frac{1-p}{p} + \mathrm{E}\left[W\right] = \frac{E[X]}{2}\left(\frac{\mathrm{E}\left[W\right]}{2} + \mathrm{E}\left[W\right] - 1\right) + \mathrm{E}\left[\beta\right]. \tag{9}$$

Assuming that β_i is uniformly distributed between 1 and W_i we have $\mathrm{E}\left[\beta\right] = W_i/2$. Combining (8) and (9) we get:

$$\mathrm{E}\left[W\right] = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \tag{10}$$

Since p is likely to be small, it is worth noticing that:

$$\mathrm{E}\left[W\right] = \sqrt{\frac{8}{3bp}} + o(1/\sqrt{p})$$

Combining (10) and (8) we get:

$$\mathrm{E}\left[X\right] = \frac{2+b}{6} + \sqrt{\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2}$$

and combining this with (5) we have

$$\mathrm{E}\left[A\right] = RTT\left[\frac{2+b}{6} + \sqrt{\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2} + 1\right].$$

Notice

$$\mathrm{E}\left[X\right] = \sqrt{\frac{2b}{3p}} + o(1/\sqrt{p}).$$

From (1) and (2),

$$\begin{aligned} B(p) &= \frac{(1-p)/p + \mathrm{E}\left[W\right]}{\mathrm{E}\left[A\right]} \\ &= \frac{\frac{1-p}{p} + \frac{2+b}{6} + \sqrt{\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2}}{RTT\left[\frac{2+b}{6} + \sqrt{\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2} + 1\right]} \end{aligned}$$

which we can express as:

$$B(p) = \frac{1}{RTT}\sqrt{\frac{3}{2bp}} + o(1/\sqrt{p}).$$

4 Event-based modelling

- Event-based modelling is a common modelling framework.
- The simulation holds a time-ordered list of "events" which represent the important happenings in the network.
- The events are "executed" in order and may trigger other events.
- For example "TCP packet arrives at node 54 (from node 23)" at time 123.044 may trigger "acknowledgement arrives at node 23 (from node 54)" at time 123.156.
- An example will better illustrate this.
- An alternative to event-based is a time-step model.
- In that type of model the modelling proceeds by small time increments.
- For example, each packet being modelled advances a small amount to its destination.
- This can be much slower and also has the problem of what happens to things which happen part way though a time step.
- However, it can be useful for visualisation.
- As we shall see, a hybrid model can be used.

4.1 A toy event-based model

- Let us consider a toy model which illustrates the concept of event-based simulation.
- The simulation model will be an extremely simple simulation of peer-to-peer networking for transfer of a single file.
- The file is split into *fragments* for the purposes of transmission.
- Our simulation must allow transfer of fragments between nodes.
- It must allow nodes to enter and leave.

The following events are necessary for the toy model.

1. Node arrives (first node arriving is assumed to be the seed).

2. Node leaves.
3. Node requests fragment.
4. Fragment arrives.
5. Fragment request denied.
6. Simulation ends.

The simulation is initialised with a "node arrives" event at time 0 and a "simulation ends" event at a desired maximum time.

The "node arrives" event

- This event has a node number and represents a new node appearing in the network.
- The first time this event happens the node is the "seed" who has the full file.
- Every other node arrives with no file pieces.
- This event triggers another "node arrives" event some time later (Poisson process).
- This event triggers a "node leaves" event for this node some time later (time is a distribution based on research).
- This event triggers a "node requests fragment" event, the time is the delay to the node requested from.

The "node requests fragment" event

- This event has two node numbers for the requesting node and the node to be requested from.
- This event triggers either
    1. A "fragment request denied" event after a time for the delay between nodes (if the requested node is too full).
    2. A "fragment arrives" event after a time for the transmission of the fragment between these nodes.
- It may trigger another "fragment request" event if the node does not have enough partners.

The other events in brief

- "Fragment arrives" may trigger another fragment request to the same node (unless we have all the fragments).
- "Fragment request denied" may trigger another fragment request to a different node.
- "Simulation ends" obviously ends the simulation.
- The simulation may end early if a full copy of the file no longer exists.
- Obviously this simulation needs to be greatly refined to properly replicated bittorrent.
- However, it is a demonstration of how we could build a more complex model.

5 The ns-2 simulation

- ns-2 is a freely available event-driven simulator which simulates packet-level traffic.
- It is available from `http://www.isi.edu/nsnam/ns/`
- The simulator is written in C++ but uses tcl for simulations.
- The scripts used for the rest of this lecture are available at `http://www.richardclegg.org/lectures`

6 Final thoughts

- Select an appropriate level of modelling — if you need to model the whole internet you cannot do packet level modelling. If you need to model intricate protocol details for packets you cannot model the whole internet.
- Check against real data where possible that your modelling assumptions are justified.
- Is your experiment repeatable? Do you get similar results if you try slightly different starting scenarios?
- Remember sensitivity analysis: What happens if the bandwidth is a little less? What if the demand is a little more?
- Can statistical analysis of your results help?
- Remember that what you model today is out of date in a year and hopelessly obsolete in ten years.

7 Bibliography

References

[1] R. J. Adler, R. E. Feldman, and M. S. Taqqu, editors. *A Practical Guide to Heavy Tails*. Birkhäuser, 1998.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[3] D. P. Bertsekas, and R. G. Gallagher Data Networks *Longman Higher Education* 1986

[4] R. G. Clegg. Simulating internet traffic with markov-modulated processes. *Proceedings of UK Performance Engineering Workshop*, 2007. Available online at: `http://www.richardclegg.org/pubs/rgc_ukpew2007.pdf`

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power–law relationships of the Internet topology. *Comput. Commun. Rev.*, 29:251–262, 1999.

[6] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Trans. on Networking*, 9(4):392–403, 2001. `http://www.icir.org/floyd/papers/simulate_2001.pdf`

[7] W. Kaufmann. Consider a Spherical Cow 1985

[8] J. Padhye, V. Firoiu, D. Towsley and J. Kurose Modelling TCP throughput: a simple model and its empirical validation *ACM SIGCOMM Computer Communication Review* 28(4), 1998.

[9] S. Zhou and R. J. Mondragón. Accurately modelling the Internet topology. *Phys. Rev. E*, 70(066108), 2004.