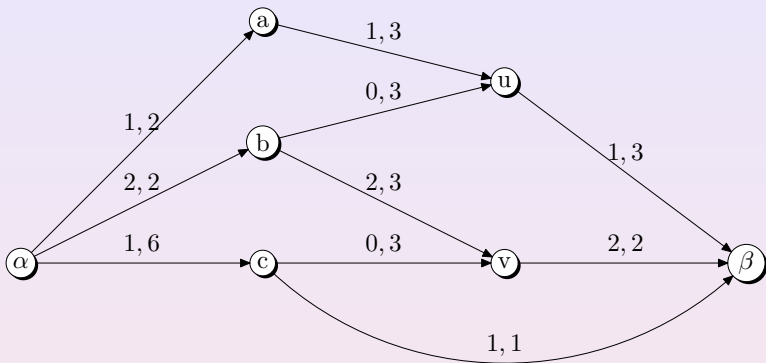


Modelling data networks



Richard G. Clegg (richard@richardclegg.org)— 8th November 2007

Available online at <http://www.richardclegg.org/lectures> accompanying printed notes provide full bibliography.

(Prepared using \LaTeX and beamer.)

Difficulties in modelling the Internet

- See [Floyd & Paxson 2001].
- The internet is big (and growing).
- The internet is heterogenous 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?

Steps to modelling

- 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?

Important questions for modelling

- 1 How much of the network do we model?
 - Whole internet (then we can't even model every computer – every AS?)
 - A few typical nodes?
 - A sub net?
 - A single queue and buffer?
- 2 What level of modelling is appropriate?
 - Mathematical – solution “instant” (or quick)
 - Detailed simulation
 - Combined approach (equations abstract away some details with approximations)
- 3 How far down the network stack need we go?

Model example one – 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.

Approach to model one – peer-to-peer network

- 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?

Model example two – TCP protocol model

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?)

Approach to model two – TCP protocol model

- 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?

Model example three – Buffer overflow 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.

Approach to model three – Buffer overflow model

- 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).

Areas of modelling interest(1)

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.

Areas of modelling interest(2)

- 1 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?
- 2 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?
- 3 Other things to model:
 - Reliability modelling — what happens when links or nodes fail?
 - Overlay networks — P2P increasingly important.

Internet topology

- 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).

AS level topology

Power law networks

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

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

where α is a constant.

- Power law topology of the AS graph shown by [Faloutsos x3].
- 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.

Mathematics to generate AS topology

Albert–Barabasi [Barabasi 99] “Preferential attachment” model

Constructive Start with a small “core” network. When a new node arrives, attach it to an old node with the following probability

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

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 [Zhou 2004] Positive Feedback Preference which adds a small “faster than exactly proportional” term.

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?

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.

What is a Heavy-Tailed distribution?

Heavy-Tailed distribution

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 $x \rightarrow \infty$.

- 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.

Heavy tails and the internet

- 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 curiosity.
- 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”).

Long-Range Dependence (LRD) and the Internet

- 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 underspecified.
- The Hurst parameter is “a dominant characteristic for a number of packet traffic engineering problems”.

Long-Range Dependence (LRD)

Let $\{X_1, X_2, X_3, \dots\}$ be a weakly stationary time series.

The Autocorrelation Function (ACF)

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

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

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

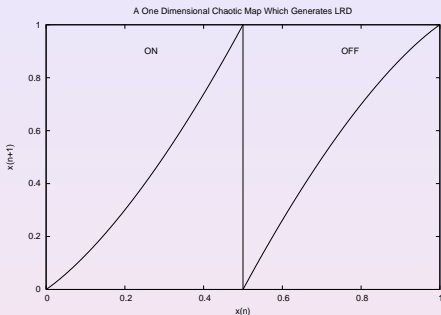
where \sim means asymptotically proportional to and $H \in (1/2, 1)$ is the Hurst Parameter.

More about LRD

- 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 [Clegg 2007].

Iterated chaotic map model for LRD

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 — packets and not packets with Hurst $H = \min(m_1, m_2) - 1$.

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.

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?

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.

Queuing analysis of the leaky bucket model

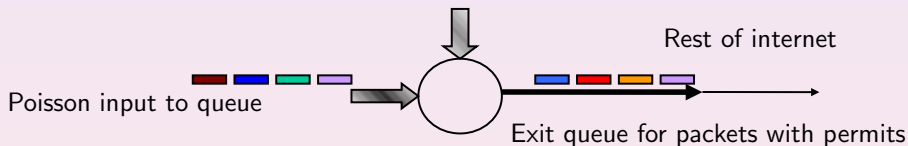
- A “leaky bucket” is a mechanism for managing buffers and to smooth downstream flow.
- What is described here is sometimes known as a “token bucket”.
- A queue holds a stock of “permit” generated at a rate r (one permit every $1/r$ seconds) up to a maximum of W .
- A packet cannot leave the queue without 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 [Bertsekas and Gallager page 515].

Modelling the leaky bucket

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,

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

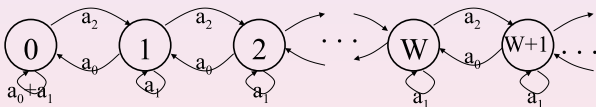
Queue of permits
(arrive every $1/r$ seconds)



A Markov chain model of the situation

- 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

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



Solving the Markov model

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 $\pi_0, \pi_1, \dots, \pi_{i-1}$.

Solving the Markov model (part 2)

- 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.

Completing the model

- Want to calculate T average delay of a packet.
- If we are in states $\{0, 1, \dots, W\}$ packet exits immediately with no delay.
- If we are in states $i \in \{W + 1, W + 2, \dots\}$ 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

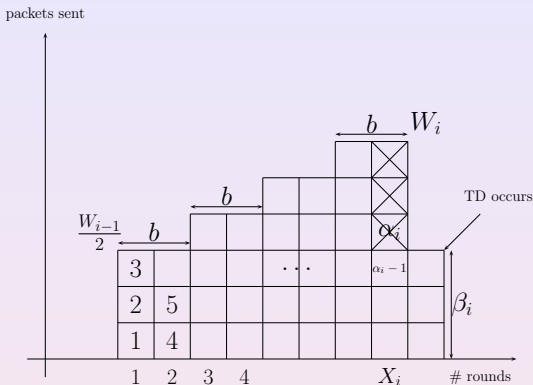
$$T = \frac{1}{r} \sum_{j=W+1}^{\infty} \pi_j (j - W).$$

- For more analysis of this model see Bertsekas and Gallager page 515.

A probabilistic model of TCP throughput

- 1 Calculate bandwidth as function of RTT and p the prob. of packet loss.
- 2 The model is taken from [Padhye et al 1998] (do not worry about details, in notes and full paper) but think about how model works.
- 3 Recall that TCP increases the window size by one when a full round of packets is received.
- 4 TCP reduces the window size by half when a packet loss is detected.
- 5 If three “out of order” (duplicate) ACK packets are received, a packet is assumed lost.
- 6 Only TD loss here. Slow start, timeout loss and other issues are dealt with in the full paper.
- 7 Consider “rounds” to be the protocol running between losses.

Probabilistic model of TCP throughput



W_i window size at end of round i , $\alpha_i = \#$ of first packet lost in round i ,
 $\beta_i = \#$ packets in partial round after loss, $X_i = \#$ full rounds before loss
 $b = \#$ rounds before window increase (usually 1).

Beginning the model

Let Y_i be the number of packets sent in round i . Let p be the iid probability a packet is lost. Let $B(p)$ be the bandwidth in pkts/sec given p . Let A_i be the time that the i th round takes. Let RTT be the round trip time.

$$B(p) = \lim_{i \rightarrow \infty} \frac{E[Y_i]}{E[A_i]} = \frac{E[Y]}{E[A]}$$

$$E[Y] = E[\alpha] + E[W] - 1.$$

Since $E[\alpha] = 1/p$ (full derivation in notes) then

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

Also

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

Completing the model

$$B(p) = \frac{(1-p)/p + E[W]}{\text{RTT}(E[X] + 1)}.$$

The final details of how the model is solved are given in the printed notes. Values for $E[W]$ and $E[X]$ can be substituted in giving the final model

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

- This model can be further extended to cover timeout losses and TCP slow start.
- Note the vital role played by RTT.
- However, the assumption that p is constant and iid is suspect.
- The model stands up well to empirical investigation.

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.

Event-based versus time-step models

- 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 through a time step.
- However, it can be useful for visualisation.
- As we shall see, a hybrid model can be used.

Event-based simulation — a toy 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.

Events for the toy model

- 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.

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>

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.