Rethinking TCP-Friendly

Matt Mathis
Pittsburgh Supercomputing Center (PSC)

ICCRG at IETF 73
17-Nov-2008

http://staff.psc.edu/mathis/unfriendly
http://staff.psc.edu/mathis/papers/ICCRG73.pdf

PSC
PITTSBURGH SUPERCOMPUTING CENTER

Outline

- Background, motivation and goals
- Problems and partial solutions
- A plan forward
- Open mic

The existing Internet “fairness” paradigm

- Routers send independent signals to all flows
- All flows have equivalent response to signals
- This response is defined by AIMD
- Modeled by $$Rate = \frac{MSS}{RTT}\frac{0.7}{\sqrt{p}}$$ [Mathis97]
- Defining TCP-friendly Rate Control (TFRC)

But there are “fairness” problems

- Non-responsive (UDP?) flows
- Applications that open many connections
- Flows with extremely different RTTs
  - TCP matches window size (short term window fair)
- Insufficient Active Queue Management (AQM)
  - RFC 2309
- Short term fair is not at all long term fair
- Defense from DOS attacks
- Many many more

ISP reaction

- Implement traffic controls at the access routers
  - Throttle aggressive users and applications
  - Protect “normal” users and applications
- Over provision core routers
- Nearly universal for home ISPs(?)

- Consider taking this to the limit.....

An alternate universe

- Routers control traffic (“allocate capacity”)
  - Isolate greedy flows
    - Protect small flows from greedy flows
  - Think:
    - Fair Queuing (well not really...)
    - Approximate Fair Dropping (AFD)
    - RE-ECN
- TCP's goal is to keep the network busy
  - It is ok to be greedy (up to a point)
- Cool new property: Neither router behavior nor end-system behavior has to be standardized
  - ISPs can enforce their own “fairness” model
  - Allows TCPs to overcome adverse environments

For illustration: Round Robin Fair Queuing

- Put each flow into a separate queue
  - Each has it's on drop discipline
- Schedule packets from each queue in sequence
- Strong flow isolation
  - Small flows send 0 or 1 packet on each turn
  - Greedy flows experience all the drops

- However RRFQ is not appropriate
  - Too much state per flow
  - Anti-optimal economic incentives

Flow Isolation is key

- Small flows are protected from greedy flows
  - Small means less than “fair share”
    - For whatever definition of “fair share” the ISP uses
  - Small flows don't see congestion signals sent to others
  - But may see 2nd order effects (e.g. jitter)

- This property has a useful colliery:
  - If the ISP can guarantee the threshold for small
  - The ISP can guarantee an SLA for small as well
    - Think of the instrumentation opportunities...

Seems extreme, however we are mostly there

- Most(?) ISPs already control their customers
  - Quite likely much more than half of all Internet users
- Many users dabble in unfriendly services
  - Has the "performance escalation" war started?
  - Other communities have abandoned us: LHC, eVLBI

- Our choice is really between:
  - Insist that TCP-friendly fairness is good enough
    - Pretend that we are in control
  - Embrace the changes
    - Take a real leadership position

A small survey

- Assume: home LAN connected to ISP
  - Is your ISP managing or shaping your traffic?
  - Does your ISP protect you from greedy neighbors?
  - Are you protected from other greedy users at home?
    - Is your network “standard”?
- Similar questions for the office
  - Is somebody managing or shaping your traffic?
  - Are you protected from greedy neighbors
    - Outside the enterprise?
    - Within the enterprise?

The transition problem

- Need to upgraded the infrastructure as we “ease into” more aggressive transport protocols
- Must progressively choreograph:
  - Addressing some open research questions (next section)
  - Encouraging network upgrades
  - Allowing more aggressive transport protocols

- The really really tough problem is expressing this in standards track documents

Difficulties and partial solutions

- At this time, MOSTLY just inventory problems
- Rat hole discovery, not exploration
  - We will stop without going too deep

Too many flows in the Internet core

- Can't keep per flow state
- Use traffic control at or near access links
  - Limit the load in the core
- Core is already over provisioned
  - Can do in steady state since edge costs dominate(?)
- Suspect that “Tiny Queue” result implies:
  - Flows don't significantly interact in the core
  - Flow isolation and fairness are not issues
- What about during a crisis?
  - Is it good enough to bound the unfairness?
  - Can we? Is it already?
  - Does RE-ECN help?

We need a new definition for “Internet safe”

- “TCP-friendly” was in part a workaround for a more abstract “Internet safe” test
- How do you assure no congestion collapse?
- How do you assure “stability”?
- How do you “crisis proof” a protocol?

Enterprise networks

- Are there special deployment problems in enterprise networks?
  - LAN traffic competing with wide area traffic
  - Less history with fairness or capacity allocation issues
  - Many enterprises with old gear
    - Especially universities....

- Can use a policy workaround
  - Forbid new transport algorithms
  - Network admins will want to have protocol signatures

What flow granularity?

- Tussle between
  - Per connection
  - Per host
  - Per access port (proxy for per user)
  - Per aggregate (e.g. network or AS)
- We need to somehow support them all
  - At lest mixing them can't have pathological behaviors
- Unfortunately we will loose “short-term-window-fairness” provided by uniform AIMD
  - It helps with cascaded bottlenecks using different flow granularities
- RE-ECN might help

Loosing “short-term-window-fairness”

- Was provided by TCP-friendly
- Probably optimal choice for default fairness in uncontrolled networks
- Important when flows have multiple connections
- Important when flow granularities differ

- It might come back if we transition all the way to a new non-AIMD model for TCP-friendly
  - Yes I have an idea for an AIMD replacement
    - No, I am not going to talk about it
    - It does not play nice with AIMD
    - But introduces some new extremely useful properties
    - Planed for IETF 74

Fixing IETF documents

- 58 RFCs use “TCP friendly” or “TFRC”
  - About 8 probably contain defining language
  - About 4 are index documents
  - The rest are “just” references
  - Others may reflect the concept but w/o the words
- We will need to (eventually) inspect all of them
  - And update many

Our (TSV area's) reputation

- We have been rather closed minded
- Killed protocols that were not TCP-friendly

Our Global Village

- We have built a global village
  - No traffic control devices
  - Implicit yield signs at every intersection
  - Carefully trained users that (mostly) share nicely
- This paradigm was ideal In our infancy
  - However we have outgrown it
- With traffic control
  - Restructure protocols to be more aggressive
  - Raises efficiency
    - Higher loads
    - Less idle capacity
  - ISPs can tune capacity allocation to fit their business

My proposed (short-term) plan

- Some attitude adjustments
- Start on three documents

Attitude adjustments

- Don't sweat the small stuff
  - If a new algorithm looks AIMD friendly and
  - Nobody can find any significant unfriendly example and
  - It looks ok under crisis conditions (resource starvation)
  - Then let it in
    - we have more important things to worry about
    - We need more experience w/ algorithmic diversity anyhow
- Assume “fairness problems” belong to the net
  - When there are problems, look harder at the network
  - And less at the protocol
  - What should be responsible for capacity allocation?

New Documents

- A “vision statement”
  - Think draft architectural statement
    - But we are not the IAB
- A new test for “is it safe for the Internet?”
  - How to replace “TCP-friendliness” testing
  - Congestion collapse testing
  - Stability critera
  - Bounds on wasted transmissions and inefficiency
- Interim guidelines for testing non-AIMD-friendly
  - Start with environments needing specialized solutions

Interim guidelines for testing non-AIMD-friendly

- Limited deployment in specialized environments
  - Not loaded by default
    - requires expertise and/or action to install
  - Strong warnings in the documentation
  - Off by default (requires per application action)
  - Has published (e.g. packet trace) signatures
    - So net-admins can Identify it
  - DSCP set to “Scavenger” by default

Open discussion

- I am most interested in problem discovery
  - What have I overlooked?
- Please avoid the “Fairness” rat hole
  - We need to attack that separately
    - I believe it is an orthogonal problem