



P2P Application Requirements for TANA

IETF-72, Dublin
Laird Popkin, CTO, Pando Networks
Co-Chair, P4P Working Group



P2P apps want users to have a great experience. This means P2P working well, other apps working well, and no surprise expenses.

- Outline:
 - P2P applications are different
 - P2P apps can utilize transport level information to make better decisions
 - A few specific suggestions/requests



P2P Apps are different from most other applications

- P2P has multiple data sources to choose from
 - Optimization of source selection by topology, etc., is ALTO.
 - Optimization of source selection by congestion, etc., is TANA.
- P2P uses multiple streams to multiple data sources in parallel
 - Can manage both stream and link congestion
- Can differentiate between “foreground” and “background” transfers to use different sources
 - e.g. RSS vs. video streaming.



P2P Apps would benefit from transport enhancements

More transport level information would allow:

- Better performance of p2p applications
 - Decongested links are faster
- Reduced impact of p2p applications on other applications
 - Same computer
 - Same home or campus
 - Same local loop or neighborhood
- Perhaps avoid transit volume surcharges



P2P Apps would benefit from transport enhancements

The following are a few ideas of specific capabilities that p2p apps would like to have. Not all will fall under TANA.

If there are existing solutions to these issues, I'd like to learn more and present them to the p2p community.



Potential enhancements

- ‘Fairness’ between applications on a computer
 - e.g. a p2p app with 40 streams vs. a web browser with 4 streams.
 - Can’t quantify whether this is important or not. Thoughts?



- Stream congestion
 - If we know which streams are congested, we can shift transfers to other, uncongested streams. Need to know both up and down status.
 - Derived estimation of congestion (e.g. slower throughput) is extremely unreliable, so the current best strategy is to keep pumping data.
 - ECN and Re-ECN look promising.



Potential enhancements

- Link congestion
 - If we know the degree to which a user's internet access is saturated, we can 'dial back' to de-saturate the link. Need to know both up and down status.
 - We want to do this, as performance of congested links is extremely poor, and we don't want to annoy our users (and their roommates and neighbors).
 - Current approaches are extremely inaccurate.
 - "If all streams are slow, the link is congested"
 - "If pings are slow, the link is congested"



Potential enhancements

- Less than Best Effort?
 - There are cases where p2p transfers could be “less than best effort”
 - e.g. downloading game patches overnight, PodCasts, video pre-caching.
 - But we would need a reason to do so (e.g. data doesn't count against usage quotas)



Potential enhancements

- Link capacity/quota (is this in scope for TANA?)
 - If we know that a user has a bandwidth quota, and where they are relative to that quota, we can make informed decisions.
 - E.g. don't use nodes with quotas that they are close to hitting as seeds.
 - Or use “scavenger class” data transport (if it doesn't count)?
 - We want to do this so that we don't cost users money by running over quota.
 - Needs to be a standard that we can code against, not per-ISP or manual.



Thank You

