# Is There a Problem With Peer-to-peer Traffic?

## *Why ISPs and their customers can seem to be in conflict.*

**Toby Moncaster (BT), Bob Briscoe (BT & UCL), Louise Burness (BT)**

Peer-to-peer (P2P) applications, especially BitTorrent, have been one of the great success stories on the Internet. Unfortunately that success brings with it a downside for end-users that don't use P2P. This memo seeks to more precisely understand the nature of this problem and thus hopefully make some progress towards solving it.

### The Problem

P2P applications are bandwidth hungry – the more bandwidth they can find the faster files get downloaded. In order to get the most bandwidth it can, BitTorrent employs several tricks: continually probing for peers with more upload bandwidth, running TCP sessions for long periods so their congestion windows grow and opening multiple TCP flows simultaneously. One of the consequences of this is that interactive traffic either struggles to find sufficient bandwidth or suffers from unpredictable delay and jitter.

Many people argue that operators simply need to upgrade their backhaul links to provide more capacity. Unfortunately, operators (be they residential, enterprise or campus) have no incentive to do this [briscoe] because they can simply use deep packet inspection to control the balance of resources between users. DPI boxes are an order of magnitude more cost effective than capacity upgrades and thus give far greater short-term profits for the operator. DPI also allows operators to invest in capacity upgrades knowing that these won't be disproportionally consumed by P2P applications. DPI boxes use mechanisms varying from the extreme (spoofing TCP reset packets) to simply using DiffServ to give different QoS levels to P2P traffic. Whether you realise it or not there is a good chance that your own ISP will be doing this, in the interests of the majority of its customers.

But there is another, more fundamental reason why DPI is a more attractive solution than simply adding bandwidth. Since the earliest days of telecommunications operators have used statistical multiplexing to maximise the number of customers that can share a backhaul link. This reflects the fact that it is not economically viable to provide each user with an end-to-end dedicated link at the speed they require. Statistical multiplexing assumes that at any one time only a handful of users will be actively using any given link. This allows telecoms companies to share resources between users deep in the core and thus take advantage of significant cost savings. Packet switched networks are the logical extension of this drive for efficiency in the network.

P2P undermines the assumption on which statistical multiplexing is built. Nowadays there are some users who download (and upload) content 24 hours a day, 7 days a week. With the growth in P2P TV this can only get worse. Operators could just give the P2P flows dedicated circuits but that is obviously a daft idea and if the cost of such a network was passed back to P2P users, those users would disappear. Getting rid of so many network users is not good business practice for an operator! We need a different approach to solve the problem.

### The Solution Space

As we have seen, adding bandwidth is not the whole solution, although motivating providers to add more backhaul bandwidth will help. But if bandwidth alone doesn't help, where might the solution lie?

The authors are partly funded by Trilogy, a research project supported by the European Community under its Seventh Framework Programme. The information in this document is provided 'as is', and no guarantee or warranty is given that the information is fit for any particular purpose. It reflects only the views of the author(s). The user makes use of the information at their sole risk and liability.

- **Quality of service**: Providers can prioritise real-time and interactive applications. This is done by some DPI boxes but it would be better if operators didn't just guess which flows are real-time flows. However there is an issue of who decides what QoS level a given packet needs. This requires trust – users need to trust the operator to actually give appropriate QoS and operators have to trust users to only ask for it when it is needed. This could be helped by making the users "pay" appropriately for the higher QoS using volume caps or other policing measures.
- **Educate users better**: Encourage them to only user peer-peer in the middle of the night and other known slack periods (some operators already have tariffs that seek to encourage this, although these fail to take into account the global nature of the Internet). However if all users do this then you just shift the problem to a different time of the day.
- **Encourage peer-peer applications to be more responsive to congestion**: Persuade them to somehow leave a little spare capacity for others where they share a bottleneck link. This would require some form of incentive for the application to do this.
- **Download deadlines:** Create P2P applications that allow users to specify the deadline for their download to complete. This would allow them to download the file at a slower rate if the deadline was some time off.
- **Give better information about network topologies**: Then peer-peer overlays can be more sensitively built, taking into account the actual underlying network topologies. Alternatively you could simply give better information about the true costs of the network (e.g. identify which inter-AS links are cheaper). However few, if any, commercial operators will be prepared to reveal this sort of information to their customers.
- **Design network topologies with peer-to-peer in mind**: Currently backhaul and core networks are built with all sorts of complex extra loops in them. If you know that P2P is a dominant service in your network you could redesign the topology to take account of this. This would be a poor solution as it would be hardwiring in assumptions about the behaviour of network applications.
- **Improve statistical multiplexing**: Scheduling theory has long known that it is more efficient to deal with the shortest jobs first then move on to the longer jobs. Because shorter jobs complete more quickly, they allow more efficient multiplexing in the network. The impact on longer flows is minimal – they still complete in about the same time. However the impact on shorter flows is huge – they now complete in a fraction of the time they otherwise would.

Many of these solutions go some way toward solving the problem. They broadly divide into 2 groups – those that rely on users (or applications) voluntarily altering their behaviour and those that alter the way the network handles different traffic. What is clear is that there is a need for better information sharing between the user, provider and application developer. Currently end-systems and middleboxes rely on educated guesswork to decide where congestion is, to predict the round trip time, to decide if a packet is lost or has merely been re-ordered and to work out if the packet is part of a real time flow. For years we have followed the end-to-end principle which has lead to a situation where path and priority information is only known at the endpoints, whilst topology and congestion information are known mainly in the network. Perhaps now is the time to start looking at solutions that share this information throughout the network.

**Reference**

[Briscoe] We don't have to do fairness ourselves. draft-briscoe-tsvwg-relax-fairness-00