



www.internet2.edu

Internet2 QBone Scavenger Service (QBSS) is a network mechanism that lets users and applications take advantage of otherwise unused network capacity, without substantially reducing the performance of the default best-effort service class.

QBone Scavenger Service

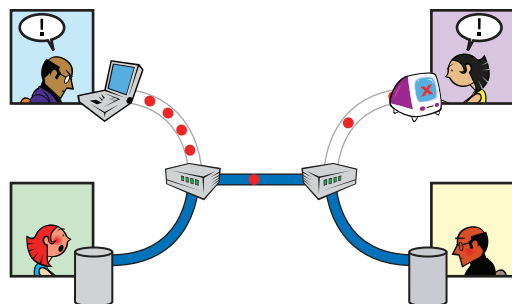
<http://qbone.internet2.edu/qbss/>

Bulk data transfer—for example, of data from radio astronomy or high-energy physics—is a critical application for high-performance networks. Because of the huge quantities of data involved, bulk transfers may cause congestion and interfere with other applications. Happily, bulk data transfers are not delay-sensitive, and people often make an effort to run such jobs when traffic on bottleneck links is light—for example, in the middle of the night. QBSS provides a way to “run the job at night” 24 hours a day.

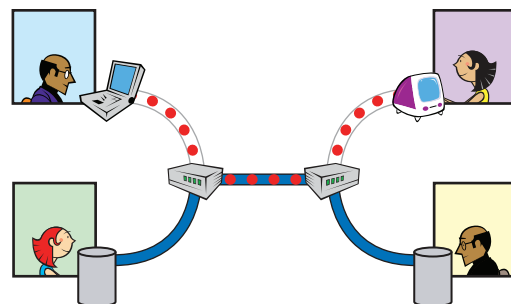
Implementing Scavenger Service creates a parallel virtual network with very scarce capacity. This capacity, however, is elastic: it can expand into the capacity of the normal

best-effort class of service whenever the network has spare cycles. The expansion happens with a very high time granularity, so all capacity not used by the default class of service is available for packets marked for Scavenger Service (DSCP 001000 binary). Routers achieve this dynamic allocation of capacity by using any of a variety of commonly available queueing disciplines, such as weighted round-robin, modified deficit round-robin, or weighted fair queueing.

QBSS is designed to be voluntarily and incrementally deployable. As of November 2002, QBSS patches are available for Apache, BB-FTP, and WU-FTP. Abilene has deployed QBSS, and about 1% of Abilene traffic is marked for Scavenger Service; JANET and GÉANT have also committed to deploying QBSS. See the web address above for the patches, a formal service definition, configuration examples, test results, and other information.



Without QBSS, bulk data transfers hog the network



With QBSS, bulk data transfers stay out of the way of delay-sensitive applications