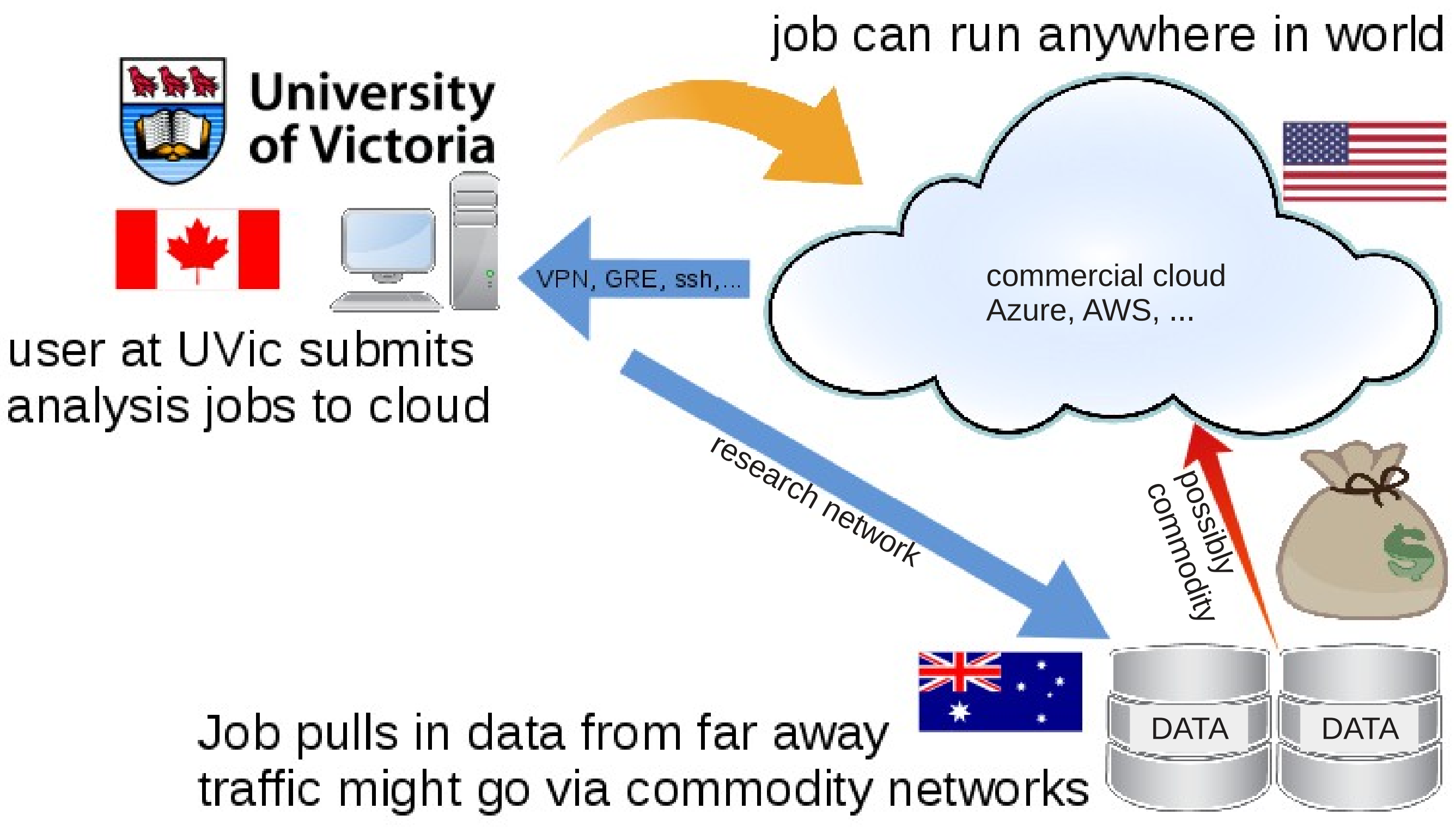
Enabling Research Network Connectivity to Clouds with Virtual Router Technology

C.R. Leavett-Brown, K. Casteels, M. Paterson, R. Seuster, R. Sobie (University of Victoria)

Problem

Jobs might face a problem with connectivity on commercial clouds (red arrow):

- high fees for traffic to up-or download of data via commodity networks
- connectivity problems due to DNS names not resolving or blacklisting



job can run anywhere in world

user at UVic submits analysis jobs to cloud

Job pulls in data from far away traffic might go via commodity networks

Can software tunnels back into home institute help (blue arrows)?

- better control over network paths to data storage elements, although path back onto campus not completely under control

What is the overhead of this traffic through these tunnels ? Are there any security concerns ?

The 3 Software Solutions

Several software products can provide tunnels to home institute to provide required research network connectivity:

OpenVPN

The open source version of OpenVPN[1] is a full-featured SSL VPN solution that accommodates a wide range of configurations, including remote access, site-to-site VPN, Wi-Fi security, and enterprise-scale remote access solutions with load balancing, fail-over and fine-grained access-controls.
Pros:
fairly easy installation, clear instruction, good examples for configuration
Cons:
compilation required to get latest version,
firewall might need to be opened for TCP or UPD port

GRE

Generic Routing Encapsulation[2] is a tunnel protocol developed by Cisco systems, that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an Internet Protocol network and is supported by many operating systems
Pros:
no installation required, GRE supported by linux kernel since many versions,
plenty of examples available, however some outdated
can connect directly to physical router instead of linux box as endpoint
Cons:
firewall might need to be opened for protocol 47 (e.g. in openstack)

sshuttle[3]

It is a python script layer over ssh to dynamically create port forwarding to relay all TCP and (optionally) DNS traffic over ssh tunnels
Pros:
installation very easy
ssh port usually open in firewalls
uses established and well known ssh program and protocol
Cons:
some installation required both on server and client (sshuttle will install some software on client when tunnel is being opened)

Testbed

We evaluated the up- and download performance between two servers at our university, one real server and one virtual machine both connected with 10G network. MTU was left at 1500, no tuning of kernel parameters was done. Artificial latencies were introduced by the 'netem' [4] package from linux kernel: it emulates fixed or variable network delays or package loss, etc.
We emulated latencies of 0ms, 2ms, 5ms, 10ms, 20ms, 30ms, 40ms, 50ms, 100ms and 200ms corresponding to data transfers found typically on campus (0-2ms), in close vicinity (5-20ms), in the same global region (30-50ms) or across continents (100-200ms).

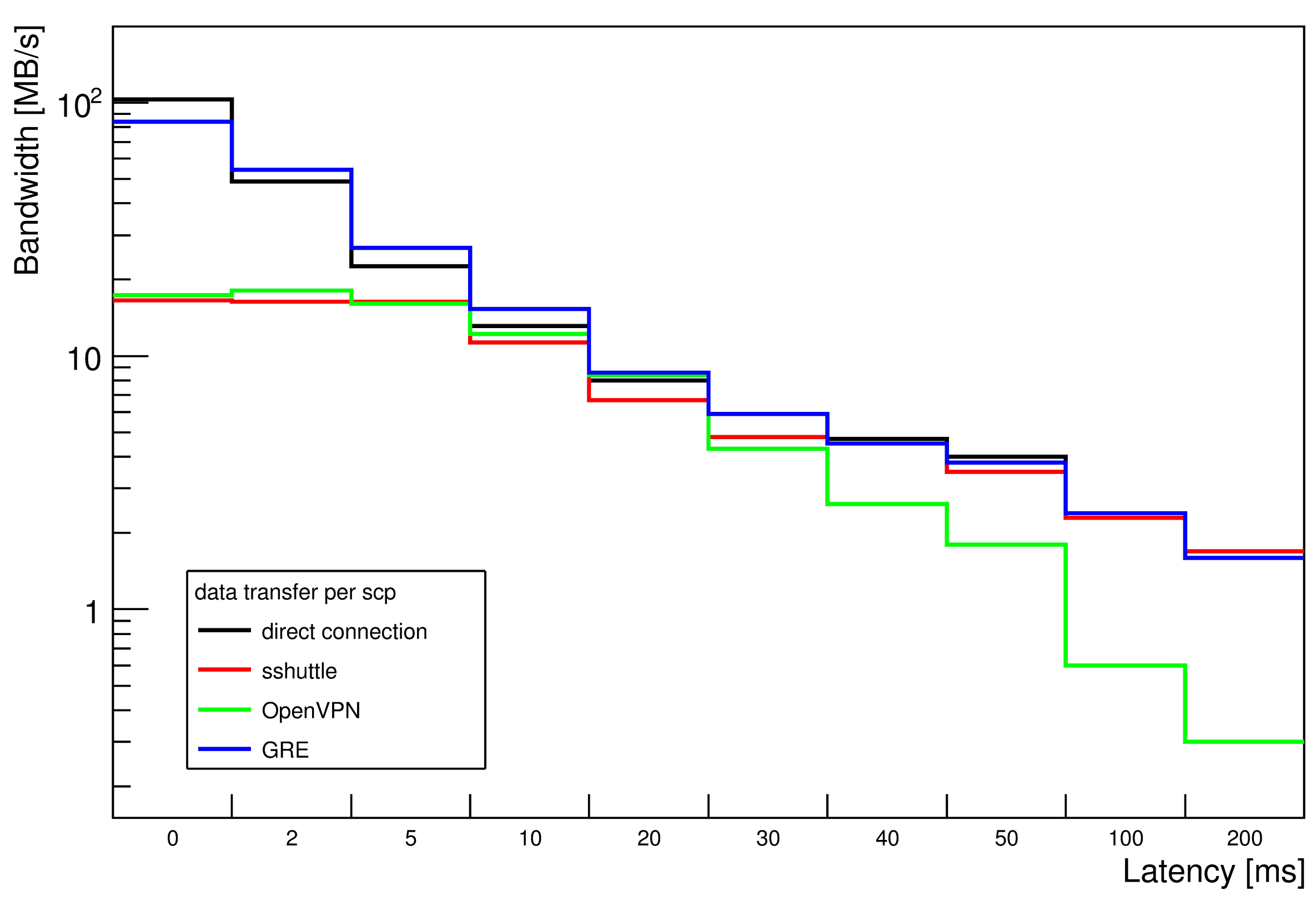
We used two programs for data transfer:
'scp': well established, widely used and known to perform well
'davix-get/davix-put': open source software developed at CERN, provides client for http, WebDAV and Amazon S3, supports X509 proxy certificate extensions commonly used in HEP community, used X509 for these tests

Results

Performance for uploading with secure copy 'scp'

The results for the four different connections are presented in Fig. 1. Direct connection means that data was transferred just by a direct scp connection between the two linux servers. For the other three connections scp was going through tunnels opened by GRE, OpenVPN or sshuttle. GRE provides a very high performing connection with close to no overhead compared to the direct connection. sshuttle has a significant overhead for small latencies but reaches a similar performance for latencies above 10ms than the direct connection. OpenVPN falls behind for small and large latencies but keeps up well with the other solutions for latencies around 10-30ms.

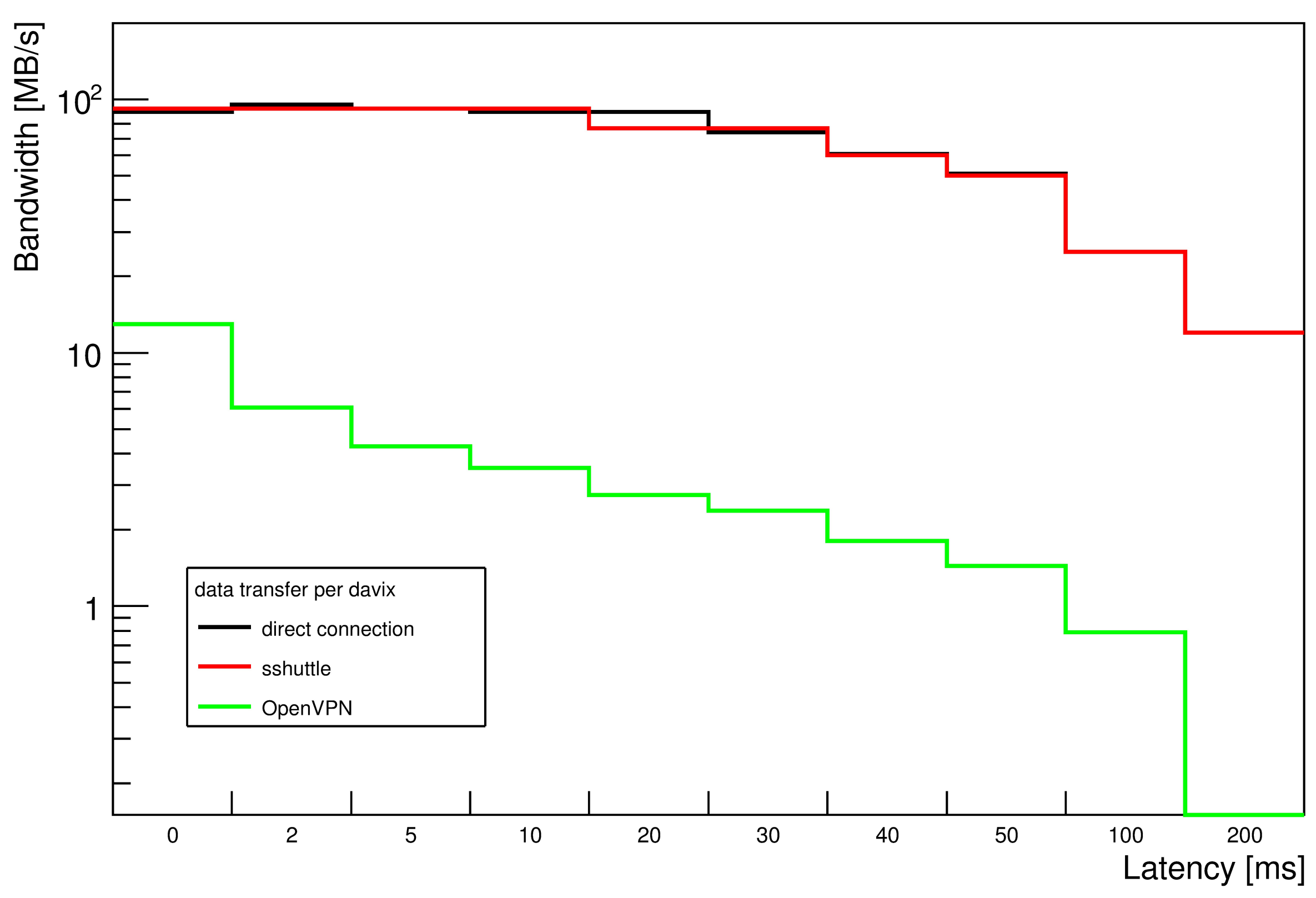
scp: Bandwidth vs Network Latency



Performance for downloading with davix-get

For downloading, we could not establish reliably a GRE tunnel with a decent performance. Our network card required far from optimal settings for GRE to function, so we decided to omit these results in the Fig. 2. Here, the OpenVPN solution falls behind by an order of magnitude in performance compared to the other two solutions, the direct connection and the sshuttle solution. Both provide in this setup a very similar performance. davix-get out-performs scp by far and provides a performance that is independent of the network latency up to latencies around 50ms indicating that it provides its own receiving buffer of about 5MB.

davix: Bandwidth vs Network Latency



Security Concerns

The three solutions used in this study provide different levels of security.
OpenVPN: is an established product and protocol which uses server and client certificates. It does not provide encryption, which would likely influence its performance negatively, private IP addresses are certainly possible.
GRE: is an established product and protocol which does not provide encryption. It creates point to point connections where both sides need each other's IP addresses of the other end during setup, private IP addresses are possible, this is one reason why GRE tunnels have been designed.
sshuttle: is a new solution based on an established product and protocol (ssh), which encrypts the traffic, but a ssh key needs to be transferred into the VM to establish this tunnel, this can cause security problems on campus. There are solutions which mitigate this problem.

Conclusion

All three products provide viable solutions for solving cloud connectivity problems, and can be implemented on very short time scales. OpenVPN typically provides the worse performance, can however by a good choice on nearby clouds. sshuttle gives a competitive performance to a direct connection except for very close-by clouds. GRE performs very well, but can require modifications to firewalls. In our setup, the network card prevented us operating GRE with good performance for downloads. Further investigation is required. davix clearly outperforms secure copy, however tuning network parameters in the linux kernel likely mitigate the effect. As next steps we will test these solutions in production environments on research and commercial clouds.

[1] https://openvpn.net/index.php/open-source/overview.html

[2] https://en.wikipedia.org/wiki/Generic_Routing_Encapsulation

[3] https://github.com/apenwarr/sshuttle

[4] https://wiki.linuxfoundation.org/networking/netem