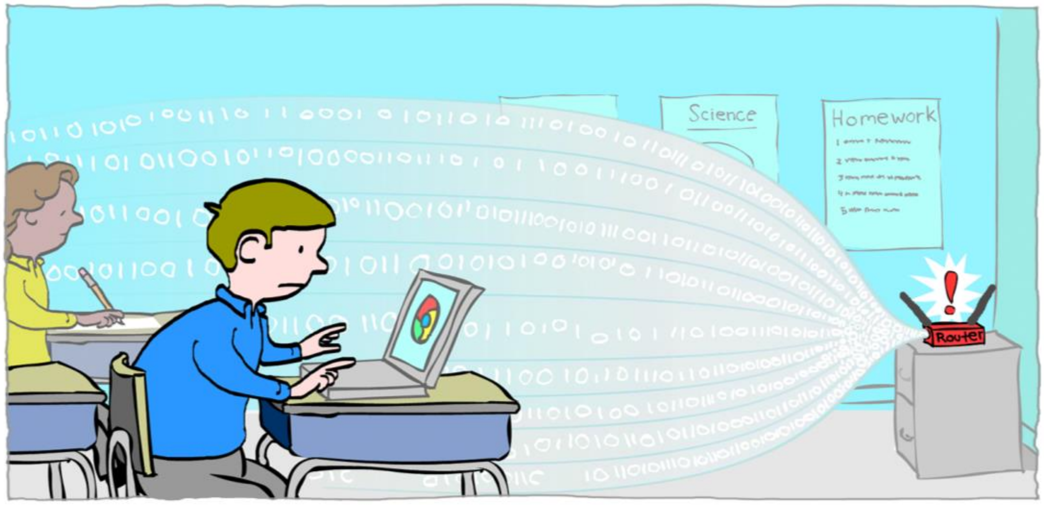
HIDDEN COST OF CHROMEBOOK DEPLOYMENTS:
UP TO 40 TIMES LONGER TO GET THINGS DONE

What took 38 seconds for a Microsoft® Windows® device
took over 28 minutes for a Chromebook™



in our tests while performing school-related media tasks
on a network loaded with the traffic of six Microsoft Windows devices vs. the traffic of six Chromebooks

Any organization, whether in education or business, wants to save time and money—and technology can allow students, teachers, and businesspeople to do more with less. There is always a cost, however. For some cash-strapped organizations, Chromebook-based solutions may seem like an attractive option to save money and provide a decent end-user experience.

Choosing Chromebooks can bring hidden costs to these organizations. Many customers do not fully understand how these devices work or the cost of deploying them. Chromebooks rely heavily on wireless network access for most of their functions, including accessing the cloud for working on saved files. Our previous study shows that Chromebooks can generate nearly 700 times more network traffic than Microsoft® Windows® devices.[1] Schools and businesses need to consider these hidden costs and implications of developing and maintaining a heavily populated wireless infrastructure as part of any Chromebook implementation plan.

What happens when multiple Chromebook users access the network at the same time? This is a common situation in both education and business. Our analysis shows that the additive effect of the number of users attempting to access the cloud over the wireless network, the speed of the wireless router, and the speed (bandwidth) of the Internet connection resulted in tasks taking as much as 40 times longer on the Chromebook we tested versus the Windows device we tested. Windows devices are productive, even without access to the cloud and require far less bandwidth than Chromebooks, helping end users complete tasks more quickly. Expanding the

[1] See the full Principled Technologies report at www.principledtechnologies.com/Microsoft/Chromebook_PC_network_traffic_0613.pdf.

MARCH 2014
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by Microsoft

number of Windows devices in an organization can result in significant cost savings from increased productivity and avoided network infrastructure expenses.

In the Principled Technologies labs, we tested a Windows-based ASUS™ Transformer Book T100 2-in-1 and an Acer® C720 Chromebook in various school-related assignments. We completed a series of assignments on the Chromebook, recorded the traffic it generated, and repeated the assignments while five computers played back the Chromebook traffic—simulating the traffic of five more students—to see how the Chromebook traffic affects the time necessary to complete these class assignments. We repeated this process using the Windows-based ASUS™ Transformer Book T100 2-in-1. Although our testing involved documents and media that students commonly use for school, these use cases can also apply to businesses and other organizations.

When the two systems shared the network with other simulated users of the same system, we observed that the Chromebook took much longer to complete assignments. For example, with six students on the network at the same time, editing, saving, and accessing three photos took 43.7 times longer on the Chromebook than completing the same assignment on the Windows 2-in-1 device; the Chromebook took over 28 minutes to complete the task while the Windows device took only 38 seconds.

CONNECTIVITY DOESN'T ALWAYS EQUAL PRODUCTIVITY

Although the Windows device outperformed the Chromebook in all of our tests, the environment in which we observed the biggest difference between the two systems was when we provided them with the bandwidth recommended bandwidth for schools: about 0.25 Mbps per user (for a total of approximately 1.5 Mbps for the six traffic streams). Figure 1 highlights the different outcomes from the two test systems.

Under our recommended bandwidth, did the system…	Windows device	Chromebook
Convert a DOCX file to a PDF in less than 30 seconds?	✓	✗
Edit and save three photos in less than a minute?	✓	✗
Play a one-minute video with no delay?	✓	✗
Play a 22.3MB song with no delay?	✓	✗

Figure 1: In all scenarios, the Chromebook took longer than the Windows 2-in-1 device to perform the assignments while each system shared a network with five simulated users.

ABOUT OUR TESTING

We chose to test two similarly priced devices that have basic features in common. For our Windows system, we used an ASUS Transformer Book T100 2-in-1 running 32-bit Windows 8.1. The ASUS T100 features the 1.33GHz quad-core Intel® Atom™ Processor Z3740, 2 GB of RAM, and 64 GB of solid-state storage. Although the 2-in-1 device has the capability to function as a tablet and a notebook, we used the docked keyboard to use the system as a notebook during our tests. The Chromebook we used in our testing was the Acer C720 running Chrome OS™ (33.0.1750.152) and featuring a 1.4GHz Intel Celeron® Processor 2955U and 2 GB of RAM. Because Chromebook users primarily store files on the cloud, only 16 GB of local solid-state storage is available. When a Chromebook user wants to work with documents or watch a video, the device connects to the Internet to deliver the content. With the Windows device, we kept documents and media on the local storage, which Windows can access without an Internet connection.

We show the results of five tests involving the type of work students perform in the classroom. We conducted these scenarios in our labs and measured the amount of time each system took to perform the task in the scenarios. Prior to testing, we copied all test files to each system's default storage area. For the Windows device, this was the system's hard disk drive. For the Chromebook, this was the cloud-based Google Drive™.

We created two test conditions by changing the amount of bandwidth available to each user. First, we conducted the tests with our network's bandwidth capped at 1.5 Mbps to simulate the recommended speed for classrooms in 2014.[2] We tested again without capping, or regulating, the bandwidth to see how the times to complete assignments changed for both the Windows device and the Chromebook.

Note: The results we present in this report are from testing we performed using the enterprise-grade Cisco Aironet 1142N access point. We also tested the two systems with a consumer-grade router; using that router, test times were slightly longer for both systems and the differential between the systems was similar to those we present here.

Recommended bandwidth network (up to 1.5 Mbps)

- One Windows device with five simulated additional users versus one Chromebook and five simulated additional users

Unlimited bandwidth network (up to 100 Mbps)

- One Windows device with five simulated additional users versus one Chromebook and five simulated additional users

What's the holdup?

Adding multiple devices to a wireless network affects Internet performance because there's limited bandwidth going *to* a network. If the local traffic trying to reach the Internet is *more than* the bandwidth can support, the excess traffic will either wait for available bandwidth or be re-sent after waiting. This keeps traffic flowing but results in downtime for users, translating to longer times for task completion.

[2] See the report "A Modernized E-Rate Program" at www.cisco.com/web/strategy/education/us_education/e-rate.html.

Appendix A provides configuration information for the two systems, Appendix B presents the individual steps taken in our testing, and Appendix C shows our detailed results.

IN THE CLASSROOM

Converting DOCX to PDF

Students frequently create reports that contain both text and images. In this scenario, Chris and Shane both need to convert their U.S. History reports to PDFs to share with the rest of the class. The reports are 195KB Microsoft Word documents in the most common DOCX file format. They will convert their reports from DOCX files to PDFs using the productivity applications of their respective devices; Chris will use Google Docs on the Chromebook and Shane will use Microsoft Word on the Windows device.

The Chromebook took over twice as long to convert the document when sharing the network with five other students using Chromebooks than the Windows device took.

Figure 2 shows that in our testing, when the bandwidth was limited, the Chromebook took over twice as long to complete this task when sharing the network with five other students using Chromebooks than the Windows device sharing the network with five other Windows traffic streams. The Chromebooks had to share the available bandwidth because the device required connectivity to complete this task. When we tested the devices with five other users on the network with unlimited bandwidth, the time to complete the conversion task on the Chromebook was still longer than on the Windows device, taking 62 percent more time than the Windows device. The Windows systems performed equally well on the recommended and the unlimited networks, as all functions were performed on the device.

Because Chris has to save his work to the cloud, the Chromebook can leave Chris with less time to do other things when connected to a network along with other students. Shane saves his work locally on his the hard drive of his Windows device and he can quickly move on to reading over his work or starting another task.

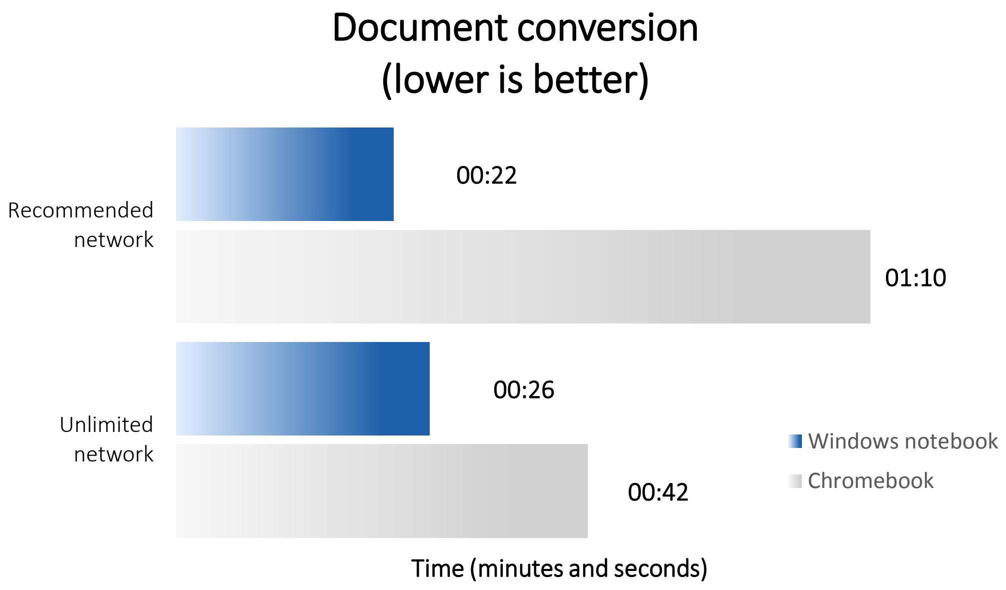


Figure 2: The amount of time the two systems spent converting a 195KB DOCX file to PDF.

Editing, saving, and accessing photos

Rosa and Luke are photographers for their school paper. For an upcoming issue, they need to perform basic editing tasks, such as auto adjusting and auto fixing, on three photographs ranging in size from 4.5MB to 6.3MB, and then save them to turn in later. Rosa uses a Chromebook while Luke uses a Windows device, and they use one of the basic picture-editing tools built for their respective systems. The Chromebook has a built-in photo editor. When a student opens a photo on the Chromebook, the edit option becomes available. The Windows device uses Microsoft Photo Gallery (available free of charge from windows.microsoft.com/en-us/windows-live/photo-gallery#photogallery=overview), although the built-in Photos app also has basic local editing functionality.

The Chromebook took over 40 times longer to finish this task when sharing the network with five other students than on the Windows device.

Figure 3 shows that in our testing on a recommended network, the Chromebook took 43.7 times longer to finish this task when sharing the network with five other students than the Windows device took. When we provided unlimited bandwidth to the network, the Chromebook still took 1.2 times longer than the Windows device to complete the photo-editing task. The editing task on the Chromebook uses a local app to auto-fix the photos, which then saves changes over the network to the cloud storage. The Windows device performs both assignments using only the hard drive for storage, so no network usage is required.

Rosa's photos save to the cloud, and the cloud is where she will access them later, so her Chromebook needs a wireless connection. Sharing the network with other Chromebook users can slow her down. Luke's Windows device saves the files locally. After editing the first photo, Luke can quickly move to the next one.

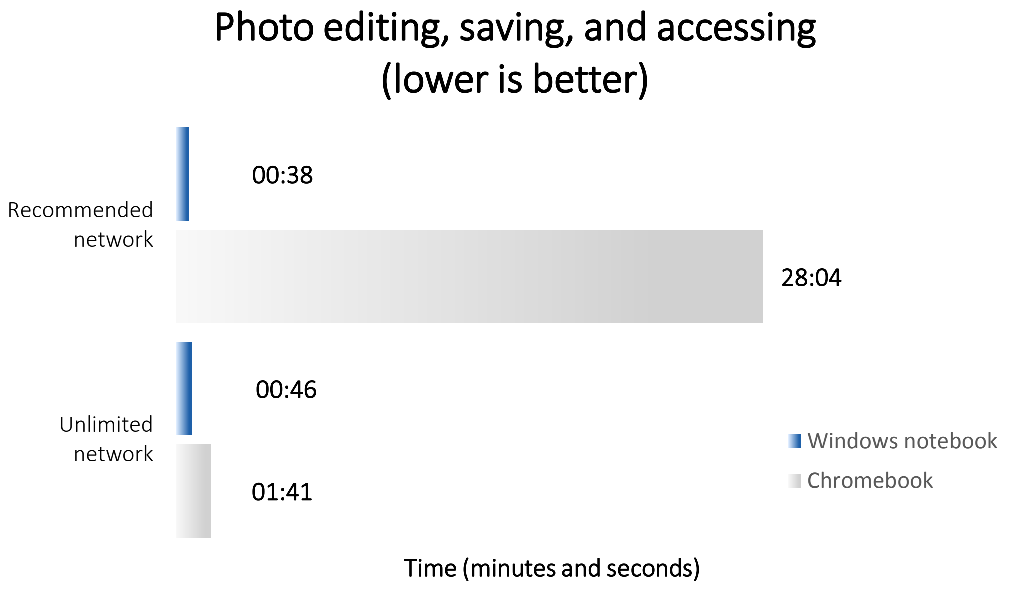


Figure 3: The amount of time the two systems spent editing, saving, and accessing photos.

Converting and playing videos

Miguel and Sierra have been working on videos for their senior art project and want to show them to friends and classmates. Miguel uses the Google app CloudConvert on his Chromebook to convert a 1-minute video file (170 MB in size) from MP4 to MOV format. Sierra converts the same-sized file from MP4 to MOV using Smart Converter (found in the Windows Store) on her Windows device. Miguel and Sierra play the converted videos in the default video-viewing software on each system. The Chromebook has a built-in video viewer and the Windows system has a default video app in the Start screen that allows users to watch videos.

Figure 4 shows that in our testing, with recommended bandwidth, when the Chromebook converted the 170MB video on a network with five other students using Chromebooks, it took 35.7 times longer than the Windows device took to convert the same-sized video on a network with five other students using Windows. After we uncapped, or deregulated, the network connection, the video conversion task on the Chromebook took 5.5 times longer than on the Windows device.

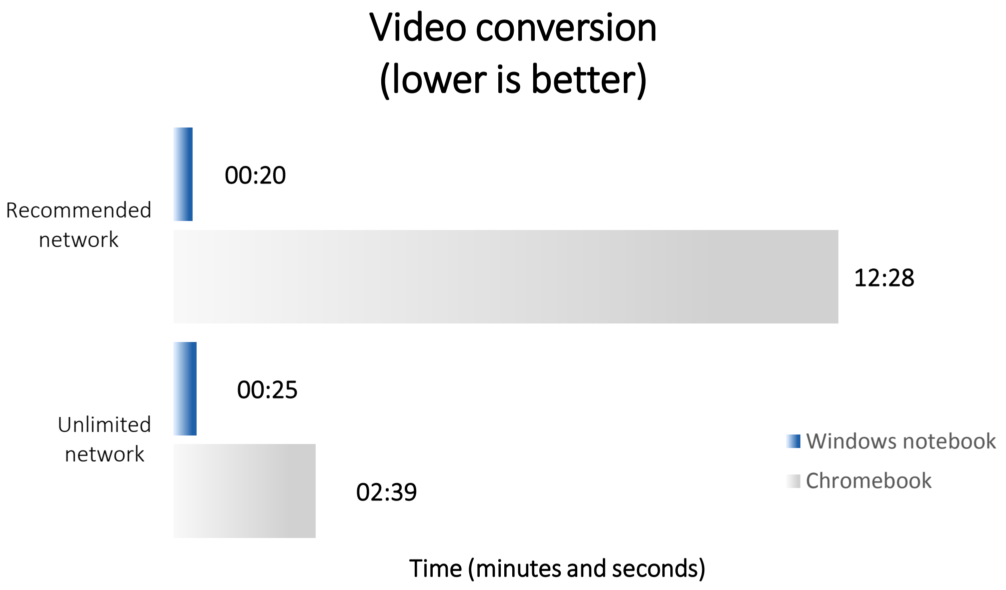


Figure 4: The amount of time the two systems spent converting the videos.

As Figure 5 shows, when the Chromebook played the video with five other Chromebooks on a network with recommended bandwidth, it took over 32 minutes to open and play the 1-minute video in its entirety due to pauses. That is 27.6 times longer than the Windows device took to open and play the same-sized video on that network with five other Windows users. After we moved to a fast network connection, the Chromebook took over 6 minutes to complete the video playback task.

Playing the video file on the Chromebook when sharing the network with five other Chromebook users took over 25 times longer than on the Windows device.

Watching a video on the Chromebook is similar to watching a video online because files on the Chromebook are stored in Google Drive. The Chromebook also requires a wireless network connection to convert MP4 files to the MOV format because the video files are stored in the cloud. Because the Windows system stores video files locally on the hard drive of the device, no network connection is required for converting or viewing.

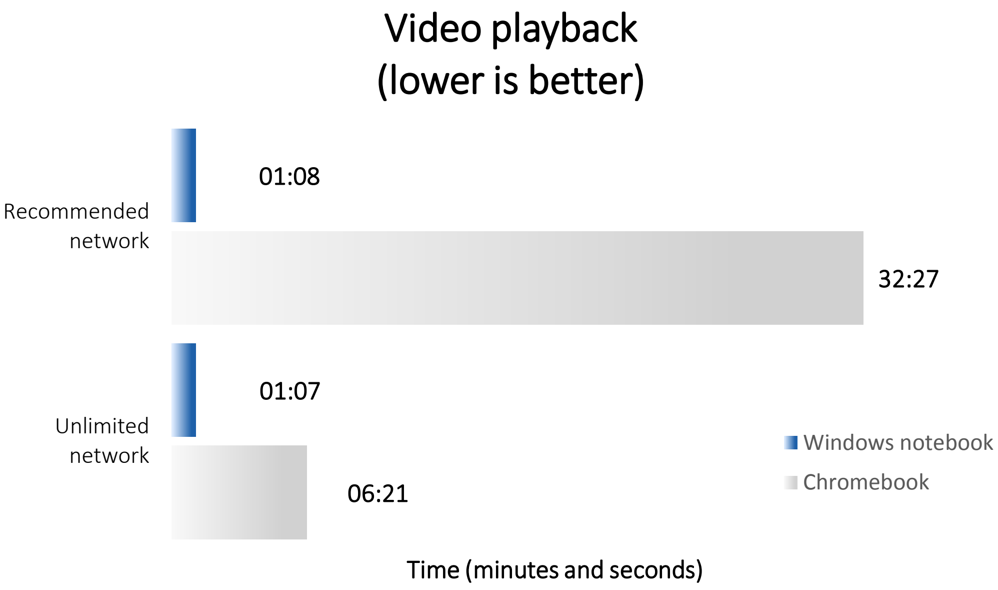


Figure 5: The amount of time the two systems spent navigating to, opening, and playing the one-minute video.

Miguel experiences many long pauses and faltering playback on the Chromebook, making the video difficult to watch. Sierra, however, plays the video file from the hard drive of her Windows device and doesn't have to wait on the wireless connection at all.

Playing music

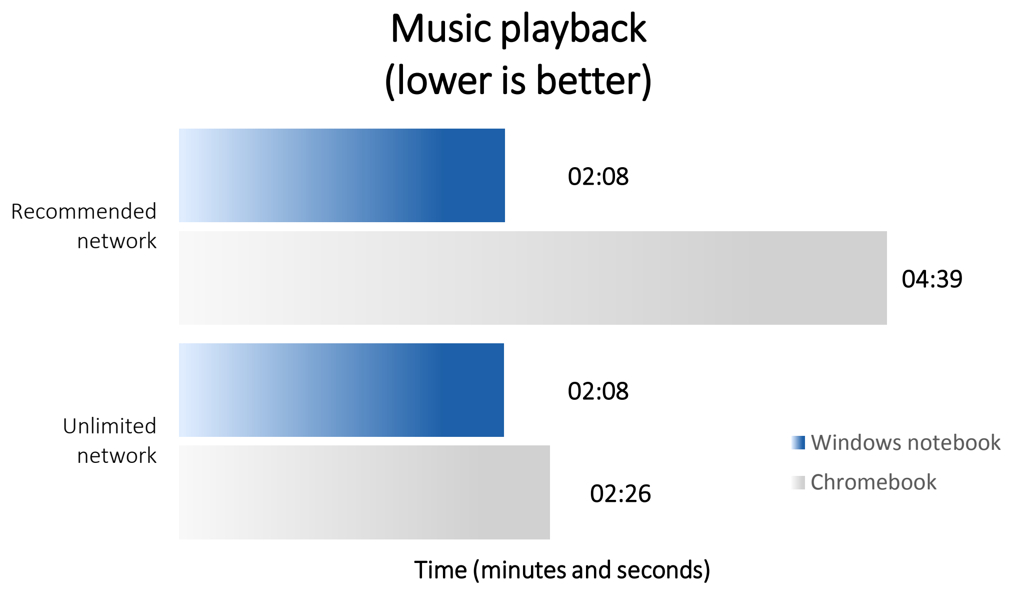
Jenny and Frankie are members of their school bands and both love music. For projects in their music theory classes, they have to play one 22.3MB WAV file. Jenny, using the Chromebook, and Frankie, using the Windows device, need to play the song for their teachers and will use the default music-playing software on their respective devices.

As Figure 6 shows, when the Chromebook played the WAV audio file on a network with five students using Chromebooks using the recommended bandwidth, it took over 4.5 minutes. That translates to 117 percent longer than the Windows device time of 2 minutes and 8 seconds. Over an uncapped or deregulated network, the Chromebook still took 14 percent longer than the Windows device took to do the same task.

Like video files, songs are stored on the Google Drive cloud, and require access to a wireless network to listen to the files, similar to listening via sites like Pandora® or Spotify®. On the Windows device, the songs are stored locally on the hard drive, so playing a song does not require a network connection.

Playing the music file on the Chromebook when sharing the network with five other Chromebook users took more than twice as long as on the Windows device.

Figure 6: The amount of time the two systems spent playing the WAV file.



The file that Jenny needs to access is stored on the cloud. Sharing the wireless network means the song may pause and resume many times before it ends, which would make the experience of listening to it aggravating instead of enjoyable. Frankie, on the other hand, plays the file from his Windows device and encounters none of the delay that Jenny experiences with her Chromebook.

WHAT TO CONSIDER

When purchasing computers for use in schools or businesses, cost is usually an important factor. Systems that require users to access the Internet to carry out everyday tasks can be problematic for a variety of reasons, not least of all the large amount of network traffic created by numerous machines in use at any given time. While Chromebooks may seem like a more affordable option than Windows devices, they require paying for much higher bandwidth. To keep pace with this bandwidth, the number of wireless access points in a school or business will need to increase. Implementing and maintaining the new routers can create unforeseen costs in hardware and IT staffing.

A largely unconsidered hidden cost of Chromebooks comes in decreased time for teaching and student productivity. We found that when working with media files on the recommended bandwidth for schools—in particular, editing, saving, and accessing photos—the Chromebook took up to 43.7 times longer when sharing the network with five other Chromebook users than the Windows device (the Windows device shared the network with five other Windows users and was under the same bandwidth limitations). That kind of delay can impede the learning efforts of even the most determined students.

A largely unconsidered hidden cost of Chromebooks: Students can spend much more class time waiting for their systems to finish tasks.

Although the purchase cost of the Chromebook is attractive to cash-strapped schools, a classroom full of students using these devices under the bandwidth restrictions typically in place in schools can experience significant performance problems. Students can end up spending their time waiting and becoming frustrated and distracted rather than focusing on the work of learning. The results in this report come from our testing on an enterprise-grade router, which provided the best possible wireless environment for each system. Testing on a consumer-grade router, which is more likely to appear in schools, produced similar differences between the Windows device and the Chromebook.

Schools and businesses thinking about adopting Chromebooks should factor into their decision the cost of dramatically upgrading their connections to meet the needs of the bandwidth-hungry devices. When you consider the productivity time lost by Chromebook users and the potential expense of increasing networks to mitigate this lost time, Chromebooks become less appealing.

APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 7 provides detailed configuration information for the test systems.

System	Chromebook	Windows 8 device
Vendor and model	Acer® C720 Chromebook	ASUS Transformer Book T100
Processor	Intel Celeron® Processor 2955U (Haswell microarchitecture)	Intel Atom Z3740
Processor frequency (GHz)	1.4	1.33
Processor cores	2	4
Memory	2GB DDR3L SDRAM	2GB DDR3 SDRAM
Storage	16GB SSD	64GB SSD
Battery type	3-cell Li-Polymer	2-cell Polymer
Battery capacity	45 Wh	31 Wh
Display	11.6” (1,366 × 768)	10.1” (1366 × 768)
Wireless	802.11 a/b/g/n	802.11 a/b/g/n
Bluetooth®	Bluetooth 4.0 compatible	Bluetooth 4.0 compatible
USB ports	1 × USB 3.0 1 × USB 2.0	1 × USB 3.0
System weight	2.76 lbs.	1.2 lbs. (tablet) 2.3 lbs. (w/dock)
OS	Chrome OS version 33.0.1750.152	Windows 8.1 32-bit
Firmware	Google_Peppy.4389.81.0	T100TA.220

Figure 7: System configuration information for the test systems.

About the ASUS Transformer Book T100

The ASUS Transformer Book T100 is a 2-in-1 device with detachable 10-inch tablet. Designed for everyday use, the device runs on a quad-core Intel Atom processor, sometimes referred to as the Bay Trail processor. It comes standard with Microsoft Windows 8.1 operating system and 2 GB of memory. The system uses 64 GB of solid-state drive (SSD) storage and a 2-cell polymer battery. For more information, see www.asus.com/in-search-of-incredible/us-en/asus-transformer-book-t100/.

APPENDIX B – HOW WE TESTED

Test design and overview

We designed our scenarios for the Chromebook and Windows device to be representative of the kinds of assignments that students would be likely to perform in real-world classroom environments. We selected the following scenarios: PDF conversion, photo conversion and editing, video conversion and playback, audio playback, and note taking.

Port mirroring on our network switch was a key element of our configuration, allowing a server running TShark to capture the network packets passing through the wireless access point to our systems under test, the Chromebook and the Windows device. This allowed us to perform all of the tests one initial time on each device, while recording and saving the traffic each device sent and received during all of the scenarios. We edited the data in the packets to be usable by a pool of client laptops for replay, and since the packets would no longer be part of actual connections with real remote Internet servers, we rewrote them to be sent to a local sink server. Rather than having a full classroom's worth of physical users performing the assignments simultaneously for each test run, we replayed the captured egress packets from laptops running Tcpreplay software in Ubuntu Linux. This allowed us to emulate a classroom environment with five simulated users sending traffic as Chromebooks or Windows devices, with a person using the real device under test in tandem with the simulated traffic. The local sink server also served to replay the ingress traffic to each of the laptops to simulate traffic from the WAN. Both the laptops' and the sink server's firewalls were set up to discard incoming network packets so that they would not create spurious network traffic with reply packets due to no actual network connections being present. Figure 8 shows how we configured our test network for packet sniffing.

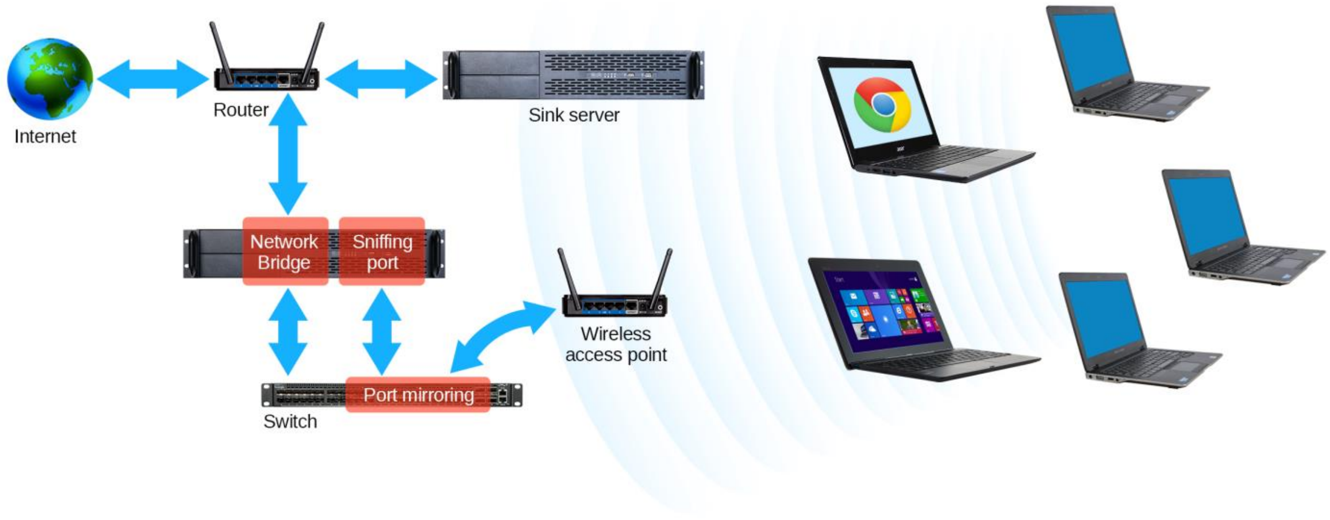


Figure 8: Network configuration for packet sniffing.

In the report cited in the About our testing section, Cisco recommends that schools provide a minimum bandwidth of 1.5 Mbps. To simulate an organization with the recommended minimum of 1.5 Mbps between the Internet and the canary device, we conducted our tests using a Cisco Aironet 1142N access point, we limited the bandwidth between the client laptops and the sink server. This bandwidth level falls in the per-student bandwidth target range for schools in 2014.[3] We repeated our testing without a cap on the bandwidth to see how performance differed.

The recording server was running Red Hat Enterprise Linux (RHEL) 6.4, the sink server was running Ubuntu Server 13.10, and the client laptops were running Ubuntu Desktop 13.10.

We created these methodologies to test both the Windows device and the Chromebook the same, despite the use of different software on the two platforms. There will be some differences in the exact steps required for each device to achieve the same goal in each scenario. We set static IP addresses on all the interfaces prior to the following steps, with the scheme shown in Figure 9.

Device/Interface	IP address
Router	192.168.0.1
Wireless Access Point	192.168.0.51
Recording Server/network bridge	192.168.0.150
Sink Server /wired interface	192.168.0.201
Laptop(1-5)/wireless interface	192.168.0.101-192.168.0.105
Windows device	192.168.0.151
Chromebook	192.168.0.152

Figure 9: The IP scheme used in designing our network configuration.

Installing Tcpreplay

We performed the following steps on the recording server, the sink server, and each of the client laptops.

1. Log in to the machine and access the terminal if using a GUI.
2. Depending on the OS, do the following:
   a. On the recording server (RHEL 6.4), type `sudo yum install tcpreplay` and press Enter, pressing `Y` and then Enter if prompted.
   b. On the sink server (Ubuntu Server 13.10) and each of the client laptops (Ubuntu Desktop 13.10), type `sudo apt-get install tcpreplay` and press Enter, pressing `Y` and then Enter if prompted.

Preparing the recording server to capture network traffic

We performed the following steps on the recording server only.

1. Log in to the machine.
2. Run `ethtool -K emX tso off gso off gro off and ethtool -K emY tso off gso off gro off` to disable TCP segmentation offload, generic segmentation offload, and generic receive offload on the two interfaces (replace emX and emY with your specific interfaces) from which you will create a network bridge.
3. To create the network bridge that will be used for throttling bandwidth, perform the following steps:
   a. In /etc/sysconfig/network-scripts, create a new file called ifcfg-br0, and provide it with the following:

```
DEVICE=br0
TYPE=Bridge
```

[3] *Ibid.*

```
ONBOOT=yes
NM_CONTROLLED=no
IPADDR=192.168.0.150
PREFIX=24
```

Replace the IPADDR field with whatever IP address you want your router-facing interface to have.

b. Save the file, writing changes to disk.
c. Edit the ifcfg-emX and ifcfg-emY files for the two interfaces you want to use for the network bridge, ensuring that `ONBOOT=YES` and `BOOTPROTO=none`, and adding the line `BRIDGE=br0` to both.
d. Send the commands `ifdown emX; ifdown emY; ifup emX; ifup emY; ifup br0`

Capturing the initial traffic

While running our test scenarios with just the one device under test (the Windows device or the Chromebook), we recorded the network traffic sent and received by that device, using the dumpcap command included with the Wireshark package.

1. Log in to the recording server.
2. When the test is to begin, type `dumpcap –i emS –w path/to/capturefile/AllTraffic.pcap` replacing emS with the interface that will sniff for packets, and the last argument with the path to which you will save the capture file, which will contain all the traffic seen by the interface.
3. Press Enter, and begin testing.
4. Press Ctrl+C to stop recording and save the capture file.

Splitting the initial traffic captures

We performed the following on both the Windows and Chromebook capture files, which we named AllTraffic.pcap for clarity.

1. Log in to the recording server.
2. Type `tshark –r /path/to/capturefile/AllTraffic.pcap –R "ip.src eq 192.168.0.XXX" –q –w /path/to/capturefile/EgressTraffic.pcap`
   replacing 192.168.0.XXX with the IP address of the device under test, to create a file containing only the packets sent by the device under test.
3. Press Enter, and wait for the process to complete.
4. Type `tshark –r /path/to/capturefile/AllTraffic.pcap –R "ip.dst eq 192.168.0.XXX" –q –w /path/to/capturefile/IngressTraffic.pcap`
   replacing 192.168.0.XXX with the IP address of the device under test, to create a file containing only the packets sent to the device under test.
5. Press Enter, and wait for the process to complete.

Rewriting the captured packets for replay

Before performing these steps, we collected the IP addresses and MAC addresses of the wired interfaces to be used for ingress replay on the sink server, and the wireless interfaces for egress replay on the client laptops. We needed to change the IP addresses and MAC addresses in the packets, as well as remove VLAN tagging.

1. Log in to the recording server.
2. Run `tcprewrite --infile=/path/to/capturefile/EgressTraffic.pcap --outfile=/path/to/capturefile/EgressTrafficX.pcap --enet-dmac=SINKMACADDRESS --enet-smac=LAPTOPMACADDRESS --srcipmap=0.0.0.0/0:LAPTOPIPADDRESS --dstipmap=0.0.0.0/0:SINKIPADDRESS --fixcsum --enet-vlan=del`
   replacing the following:
   a. outfile name with the number of the device to send from
   b. SINKMACADDRESS with the MAC address of the interface on the sink server
   c. LAPTOPMACADDRESS with the MAC address of the wireless interface on the laptop in question
   d. LAPTOPIPADDRESS with the IP address of the wireless interface on the laptop in question
   e. SINKIPADDRESS with the IP address of the interface on the sink server
3. Run `tcprewrite --infile=/path/to/capturefile/IngressTraffic.pcap --outfile=/path/to/capturefile/IngressTrafficX.pcap --enet-smac=SINKMACADDRESS --enet-dmac=LAPTOPMACADDRESS --dstipmap=0.0.0.0/0:LAPTOPIPADDRESS --srcipmap=0.0.0.0/0:SINKIPADDRESS --fixcsum --enet-vlan=del`
   replacing the following:
   a. outfile name with the number of the device to send from
   b. SINKMACADDRESS with the MAC address of the interface on the sink server
   c. LAPTOPMACADDRESS with the MAC address of the wireless interface on the laptop in question
   d. LAPTOPIPADDRESS with the IP address of the wireless interface on the laptop in question
   e. SINKIPADDRESS with the IP address of the interface on the sink server
4. Repeat steps 2 and 3 for each laptop to be used for replay.

Having run all of the above commands, you should have five EgressTrafficX.pcap files, and five IngressTrafficX.pcap files. Through whatever means you prefer, copy the five IngressTrafficX.pcap files to the sink server and copy each EgressTrafficX.pcap file to its corresponding laptop.

Setting up firewall rules

We performed the following on the sink server and on each of the client laptops, but not on the recording server.

1. Log in to the machine and access the terminal if using a GUI.
2. Type `sudo iptables -A INPUT -P tcp -dport ssh -j ACCEPT; sudo iptables -A INPUT -j DROP` and provide a password if prompted. Now the firewalls will still allow ssh traffic while also silently throwing away any non-ssh packets received.

Preparing replay scripts

We wrote a script on each of the laptops and ten scripts for the sink server, as follows:

1. On each of the laptops, create a new text file and fill it with the following text:

```
#!/bin/bash
 tcpreplay --intf1=wlan0 /path/to/EgressTrafficX.pcap
```

replacing wlan0 with the wireless interface name appropriate to the laptop, and replacing the path with the path to the specific laptop's egress capture file. Save each of these files as EgressReplayX.sh, replacing X with the number of the laptop to which each script pertains.

2. For the sink server, create text files for each traffic stream and fill them with the following text:

```
#!/bin/bash
 tcpreplay --intf1=em1 /path/to/IngressTrafficX.pcap
```

replacing em1 with the wired interface name appropriate to the server, and replacing the path with the path to the specific ingress capture file. On the sink server, make scripts with references to the capture files for laptops 1 through 5. Save each of these files as IngressReplayX.sh, replacing X with the number of the laptop to which each script pertains.

Testing procedure

Performing these tests requires terminal access to the sink server, the recording server, and all of the laptops involved in the specific scenario, which is why we allowed ssh traffic in the firewall settings in each system.

1. Prepare a line resembling the following example on the sink server: `sudo bash /path/to/IngressReplay1.sh &; sudo bash /path/to/IngressReplay2.sh &; ...` including the five replay scripts for the configuration.
2. On each of the laptops desired to replay traffic, log in to their terminals and prepare the following line on each: `sudo bash /path/to/EgressReplayX.sh` replacing the path here with the path to your egress replay script for the host laptop.
3. On the recording server, if you wish to capture the packets sent and received during the test run, prepare the command `dumpcap –i emS –w /path/to/capturefile.pcap` replacing emS with the interface that will sniff for packets, and the last argument with the path to which you will save the capture file, which will contain all the traffic seen by the interface.
4. At this point, make sure that you have commands ready to go on each laptop you want to perform replays, at least one sink server depending on how many laptops you want to replay, and on the recording server if you want to record the traffic replayed.
5. Start a timer for one minute, while simultaneously pressing Enter on the recording server's dumpcap command if you are planning to record traffic.
6. When the timer goes off, press Enter on each of the laptops, as well as on the sink server that will be running replay scripts, and begin testing.

Scenario 1: DOCX to PDF Conversion

We used the following test document:

- Word document - 201 KB (201,083 bytes)

The Word document conversion test resulted in a PDF that was roughly 693 KB on the Windows device and roughly 5.3 MB on the Chromebook (the file size differed).

Performing this task on the Chromebook

1. Document conversion
    a. Launch the Google Drive app by clicking the icon on the taskbar, and navigate to the directory containing the test files.
    b. Right click on WordDOCX1.docx→Open with→Google Docs.
    c. Click File→Download as→PDF Document (.pdf).
    d. Once it finishes downloading, click the PDF file on the bottom of the browser to open it.
    e. Scroll to the bottom of the document to confirm that it loaded fully.
    f. Close all windows to return the system to an idle state.
    g. Note the time, and wait one minute before continuing.

Performing this task on the Windows device

1. Document conversion
    a. Open WordDOCX1.docx in Microsoft Word.
    b. Click File→Save As.
    c. Choose the location on the local drive where you want to save the file.
    d. In the drop-down menu for Save as type, choose PDF, and click Save.
    e. The new file should automatically open in the default Modern PDF viewer. Once you have determined that it opened correctly, close all windows to return the system to an idle state.
    f. Note the time, and wait one minute before continuing.

Scenario 2: Photo manipulation

We used three JPEG-format images with the following file sizes:

- 6.29 MB (6,604,836 bytes)
- 4.45 MB (4,670,931 bytes)
- 5.28 MB (5,542,087 bytes)

Performing this task on the Chromebook

1. Photo editing
    a. Open the Files app, and navigate to the directory containing the test files.
    b. Double-click on the first JPEG, and wait for it to open.
    c. On the bottom of the picture, click the Edit button.
    d. Uncheck Overwrite original, and click Auto-fix.
    e. When the auto-fix is finished, click the right side of the picture to pull up the next file.
    f. Repeat steps d and e on the next two pictures.
    g. When the last file is finished, close all windows to return the system to an idle state.
    h. Note the time, and wait one minute before continuing.

Performing this task on the Windows device

1. Photo editing
    a. Navigate to the test image files.
    b. Right click the first file, and choose Open with→Photo Gallery.
    c. Click Edit, Organize, Share in the top left corner.
    d. Select the first image file, click on the Edit tab, and click Auto-adjust.
    e. Click Next, and repeat step d for each JPEG file.
    f. Close all windows to return the system to an idle state.
    g. Note the time, and wait one minute before continuing.

Scenario 3: Video manipulation

We used a 1-minute video file with a size of 170 MB (179,150,606 bytes).

Performing this task on the Chromebook

1. Video conversion
    a. Launch the Google Drive app by clicking the icon on the taskbar, and navigate to the directory containing the test files.
    b. Select the MP4 file, click More→Open with→CloudConvert.
    c. To the right of the MP4 file, click select format→mov.
    d. Click Start Conversion.
    e. When the conversion is finished, close all windows to return the system to an idle state.
    f. Note the time, and wait one minute before continuing.
2. Video playback
    a. Open the Files app, and navigate to the directory containing the test files.
    b. Double-click the MP4 video file to launch the video in the default player.
    c. Let the video play.
    d. When the video is finished, close all windows to return the system to an idle state.
    e. Note the time, and wait one minute before continuing.

Performing this task on the Windows device

1. Video conversion
    a. From the Desktop, double-click Smart Converter to open the program.
    b. Click Select, and choose the MP4 file you want to convert.
    c. Click Other→QuickTime.
    d. Click Convert.
    e. When the conversion finishes, close all windows to return the system to an idle state.
    f. Note the time, and wait one minute before continuing.
2. Video playback (default video player)
    a. Navigate to the video file.

b. Double-click the MP4 file to start playing in the default video player.

c. When the video is finished, close all windows to return the system to an idle state.

d. Note the time, and wait one minute before continuing.

Scenario 4: Music playback

We used one music file, 2:02 in duration, with a file size of 22.3 MB (23,407,892 bytes).

Performing this task on the Chromebook

1. Music playback
   a. Open the Files app, and navigate to the directory containing the test files.
   b. Double click the WAV file to launch the song in the default player.
   c. Let the song play.
   d. When the song is finished, close all windows to return the system to an idle state.
   e. Note the time, and wait one minute before continuing.

Performing this task on the Windows device

1. Music playback
   a. Navigate to the music file.
   b. Double-click the WAV file to start playing the song in the default music player.
   c. When the song is finished, close all windows to return the system to an idle state.
   d. Note the time, and wait one minute before continuing.

APPENDIX C – WHAT WE FOUND

Figure 10 shows the detailed results of our testing.

Task	Recommended bandwidth		Unlimited bandwidth	
	Chromebook	Windows device	Chromebook	Windows device
PDF Conversion	70	22	42	26
Photo Editing	1,684	38	101	46
Video Conversion	748	20	159	25
Video Playback	1,947	68	381	67
Music Playback	279	128	146	128

Figure 10: The time to complete each task in seconds.

ABOUT PRINCIPLED TECHNOLOGIES

Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.

Disclaimer of Warranties; Limitation of Liability:
PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.