SE331: Introduction to Computer Networks **Recitation 1**
Semester 1 5785 **4,6 November 2024**
Lecturer: Michael J. May **Kinneret College**

Overview, Network Calculations

This recitation will involve three parts. First, we'll do a brief overview of how an end to end network communication event works. That will help place some of the concepts from lecture today in context. Second, we'll do some theoretical network calculation. Third, we'll try some actual network communication and try to measure the actual sending time.

1 The Big Picture

We'll first consider some of the "big picture" of network communication.

1.1 Level 1: Overview

Let's assume you want to open up your web browser to `www.google.com` and do a search (see Figure 1). You open your browser and within a second or two you see a web page from Google with a search box. What happened during that second? Let's break it down.

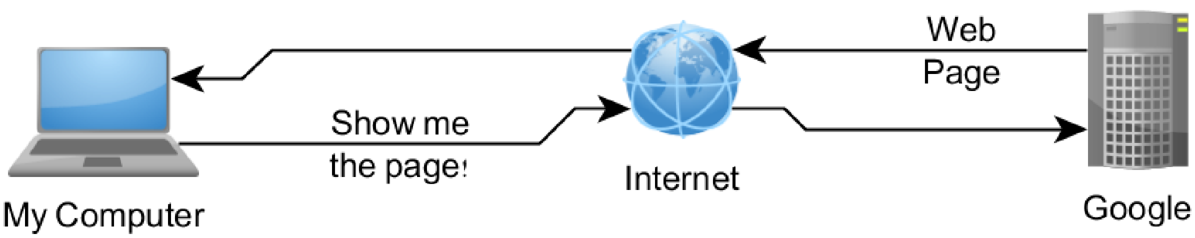

Figure 1: Highest level view of communication with Google

1.2 Level 2: Elaborate the Server Side

If we break it down a bit, we'll see that when you opened the web page, the computer sent a request out to "the internet" and it reached Google via some router, a stop along the way (see Figure 2). There are likely many different routers on the way to Google, but we show just one here for conciseness.

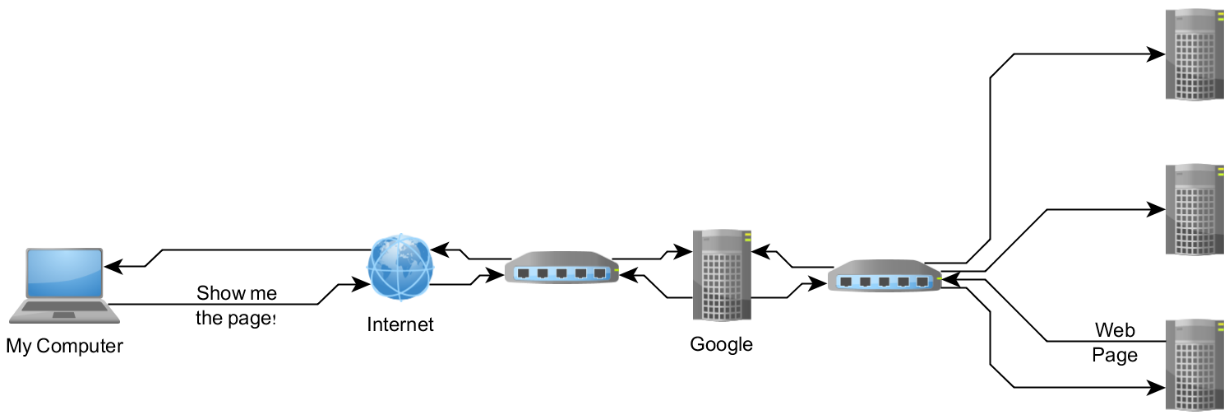

Figure 2: Second level view of communication with Google

Within Google there is more than one computer. The computer we talked to was a web facing one. There are other ones inside the company that hold web pages, indexes, advertisements, etc. The server we talked to communicated with those other computers to answer our request. The communication likely took place within Google over a local area network, a network within a single organization.

1.3 Level 3: Elaborate the Client Side

Taking it one step further down, let's elaborate on the client side (Figure 3). If we're opening the web page from Kinneret's computer labs, our computer isn't connected directly to the internet. Instead, it's connected via a local area network in Kinneret (the one in the computer lab). The local area network is connected within the building and there is a connection from the exterior router in Kinneret to the internet. We added in that on the left side.

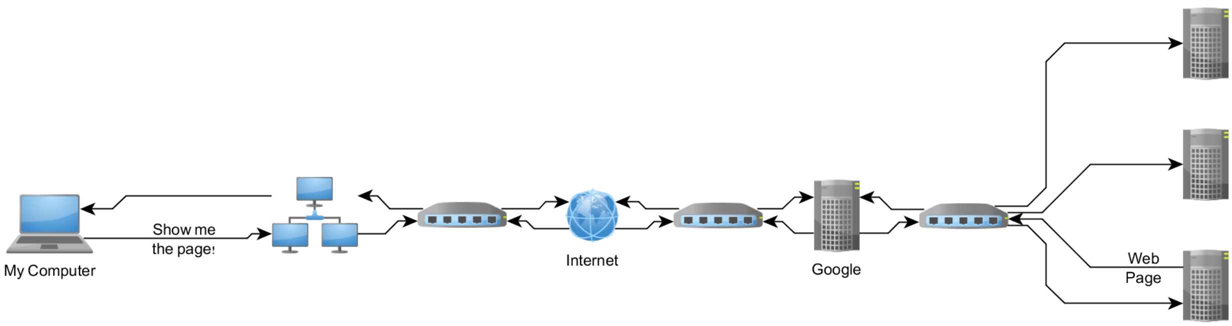

Figure 3: Third level view of communication with Google

Summary So Far We can then take a broader look at the story of the request:

1. My computer sent a request meant for Google that is routed locally in Kinneret along a local area network until it reaches an external router (a gateway).
2. Along the way (while it's being routed), the traffic is destined for google.com, but it routed locally between routers and switches in Kinneret on the way.
3. The routing in the Internet is still a black box, but we can elaborate on it later.
4. The routing within Google is not a single step - the message gets to the correct server (or one of the correct servers) via some internal routing steps.
5. The request is taken care of via a few internal messages and routing decisions.
6. The response has to go the entire journey back.

1.4 Level 4: Adding Domains, Zones, and DNS

We can elaborate a bit more on the concepts of routing in the Internet (Figure 4). It's obvious that we can't expect that every single computer in the world will be known to every single other computer in the world. We need hierarchies. We can then say that the million or so computers owned and operated by Google are part of a single area. The area is often called a domain or network. Doing that aggregation lets us reduce the need for the routing tables to refer to just areas or zones. The routing decisions to reach a single computer within the domain or area is done locally, one we are already in the area/zone.

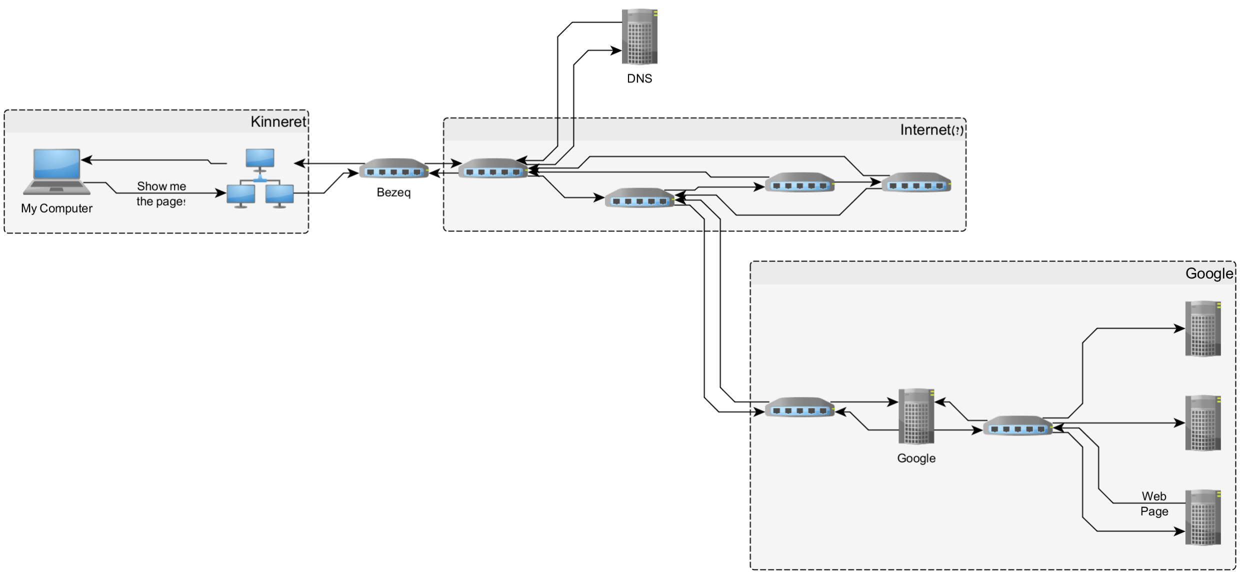

Figure 4: Fourth level view of communication with Google

Once we have areas or zones, we can make things easier for users. Instead of needing to enter the actual address of the computer in Google that we want to talk to, we can just write "www.google.com" and use a directory to look up what is the exact address of the computer we want in Google. One system that implements such a directory is the Domain Name Service (DNS). It converts textual names (e.g. kinneret.ac.il or google.com) to IP addresses that can be used for communication on the Internet.

1.5 Level 5: Considering Local Area Networks and Protocols

We can dig a bit deeper in to how the local area networking works (Figure 5). If we think about it, we don't want to worry about what networking technology is used by a local area network (LAN) that we want to communicate with. We'd like to be able to just send data from one computer to another, regardless of whether the computer on the other end has the same local area networking technology as us.

To take a simple example, we'd like it to be the case that if you're on a wired network (as on the left) you can talk with a computer on a wireless network (on the right) without a problem.

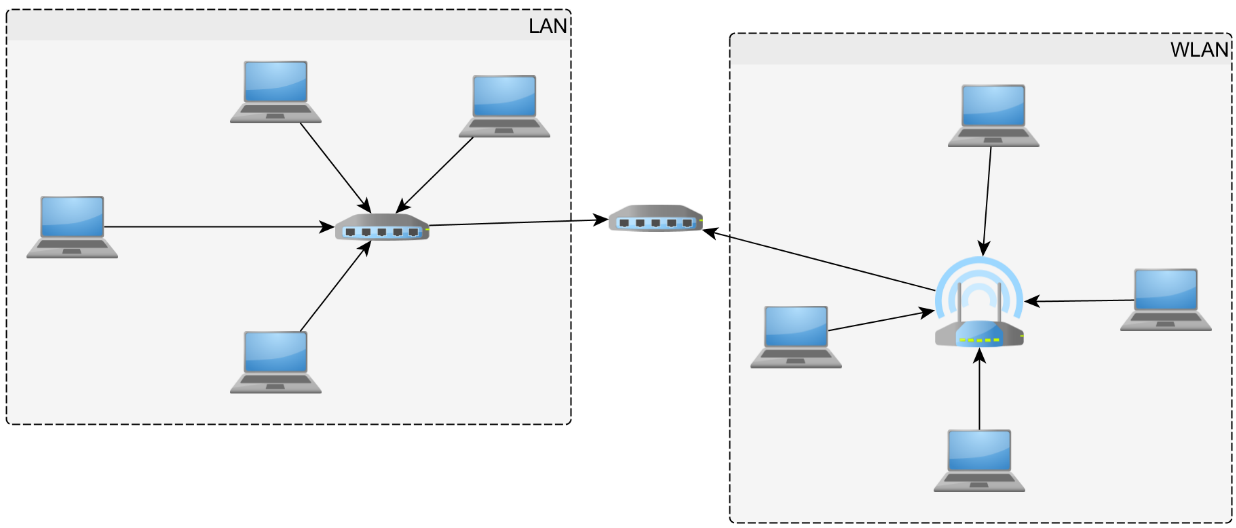

Figure 5: Fifth level view of communication, including a consideration of local area network protocols and routers

We need to establish a shared language or protocol that works on all local area networking technologies. We'll get to that later in the semester.

We also need to build routers (such as the one in the middle of the figure) that speak multiple local area network protocols. We can't expect that a wired LAN and a wireless LAN will use the same low level signaling protocols. They also need different mechanisms to manage the traffic on their media. That conflicts with the requirement for a uniform communication protocol.

The solution is protocol layering in which we put a message of one format inside a message in a different format. The exact nesting technique will depend on the protocol layers that we'll learn about soon.

2 Network Communication Calculations

2.1 Sending Times

Calculate the total time required to transfer a 1.5-MB file in the following cases, with an RTT of 80 ms, a packet size of 1 KB and an initial $2 \times$ RTT of "handshaking" before data is sent.

(a) The bandwidth is 10 Mbps, and data packets can be sent continuously.

(b) The bandwidth is 10 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.

Calculate the effective bandwidth for the sending.

(c) The bandwidth is "infinite", meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.

(d) The bandwidth is infinite, and during the first RTT we can send one packet (2^{1-1}), during the second we can send two packets (2^{2-1}), during the third we can send four (2^{3-1}), and so on. (A justification for such an exponential increase will be given in Chapter 6)

2.2 Sending Times

Calculate the total time required to transfer a 1000-KB file in the following cases, assuming an RTT of 100 ms, a packet size of 1 KB, and no handshaking time.

(a) The bandwidth is 1.5 Mbps, and data packets can be sent continuously.

(b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.

Calculate the effective bandwidth for the sending.

(c) The bandwidth is "infinite," meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.

(d) The bandwidth is infinite, and during the first RTT we can send one packet (2^{1-1}), during the second RTT we can send two packets (2^{2-1}), during the third we can send four (2^{3-1}), and so on.

2.3 Latency and Bandwidth

Consider computers A and B which wish to communicate over a 150Mbps link with a 200ms RTT. A wants to send a 15MB file to B. B wishes to send a 20MB file to A. Assume that the transport protocol used (*e.g.* TCP) requires one $2 \times RTT$ *handshake* before data can be sent in either direction.

(a) (3 points) Assume that A will send first. Calculate how long it will take A to transmit and propagate its file to B. That is, the time from the start of the handshake until the last bit of A's file reaches B.

(b) (7 points) Assume the computers will take advantage of the full duplex communication medium and send their files simultaneously. Both will use the same maximum packet size: 32KB.

The sending proceeds as follows:

- A will initiate the hand shake with B (which will take $2 \times RTT$ as above).
- After the handshake, B will send its first packet ($B1$) to A.
- When A receives the first bit of $B1$, it will start sending its first packet $A1$.

- When B finishes sending $B1$, it will wait for the first bit of $A1$ to arrive to send its second packet $B2$. If the first bit of $A1$ arrives before $B1$ is finished being sent, B will proceed with $B2$ immediately after $B1$ finishes.
- When A finishes sending $A1$, it will wait until the first bit of $B2$ arrives to send its second packet $A2$. If the first bit of $B2$ arrives before $A1$ is finished being sent, A will proceed with $A2$ immediately after $A1$ finishes.

Since A has less data to send, when B has finished receiving all of A's packets, it will send the rest of its file uninterruptedly.

Calculate the total transfer time for both files. That is, the time from the start of the handshake until the last bit of last packet sent arrives.

Calculate your answers using the exact values for KB, MB, and Mbps. Round all of your answers to the nearest 0.001 second.