-
-
-
-

Awesomeness

Designing the Internet - An early exercise that went really well

You have selected **0** posts.

select all

cancel selecting

1

of

4

mcantor

select +replies select
15d

Designing the Internet

feat. Ivan Pakkitz

Learning Objectives

The idea behind this exercise is to get students thinking not only about HOW the internet works but WHY it works the way it does. It's important to include in your framing that this kind of thing isn't necessary to do good work as a web developer, but it IS necessary to CONSISTENTLY do GREAT work, and to understand the limits of our own knowledge and area of expertise.

The information in this exercise is vital to the core value proposition of the WDI. We want to nurture confidence, empowerment and expertise, specifically in the domain of web development.

The difference between a confident expert and a "good enough" coder is in the edge cases: 80% of the time, "good enough" coders and experts are almost indistinguishable. The difference is in the 20% of the time when a task requires expert confidence, expert knowledge, and expert initiative.

In those 20% of situations, a "good enough" coder will throw their hands in the air and say "It can't be done." It is a full-stop situation. By contrast, an expert will at the very least be able to understand what holes in their knowledge are preventing them from proceeding, and will be able to evaluate the risks and benefits of going down the rabbit hole required for moving forward.

Therefore, the information in this exercise reaches one (1) layer of abstraction deeper into the problem domain than required to reach "good enough" competence, giving students the tools to deepen their understanding of WHY things work the way they do, and what to do when "good enough" competence isn't enough for a novel problem.

This exercise is intended to be a "biggest bang for your buck" overview of what questions to ask when you need to become a just-in-time internet expert.

- Describe what problems IP (Internet Protocol) solves
- Describe what problems TCP (Transmission Control Protocol) solves and how it is implemented on top of IP
- Define a packet as a discrete chunk of information to be transferred over an IP-based protocol
- Define metadata and distinguish "headers" from "body".
- Distinguish between "connectionless" protocols (IP and UDP) and "connection-based" protocols (TCP)
- Distinguish between a LAN (Local Area Network) and a WAN (Wide Area Network)
- Describe how routers essentially create a "mini-internet" (LAN) with its own IP range
- Describe how routers "hide" devices on their LAN from the rest of the world
- Describe ports and their usage in directing data
- **Bonus**: Describe why port-forwarding is required for effective peer-to-peer networking such as with torrents and gaming and how it relates to the fact that routers are the middleman between devices on a LAN and the rest of the internet
- **Bonus**: Compare and contrast UDP and TCP and describe their common use cases

Framing

While framing the exercise, the instructor should draw a doofy-looking cartoon character on the board. Details and drawing skills are irrelevant as long as the cartoon character has something resembling a postal worker's hat or messenger bag with a lightning bolt on it, like so:

- No laptops!
- Pair up.
- Imagine the internet does not exist, and you & your pair partner are about to be on opposite sides of the country.
- You want to be able to communicate with each other, and you can only do so by employing the services of one Ivan Pakkitz.
- Ivan is a postal worker who happens to be able to travel at the speed of light (thus the lightning bolt on his hat/bag). He's happy to relay thousands of messages per second between you and your friend. He can only transport one at a time and it has to fit on an index card, but as far as you can tell, it will basically be delivered instantly.
- Unfortunately, Ivan is incredibly lazy and inattentive, so there are four downsides to using his service:

(Instructor should draw a symbol that represents each problem)

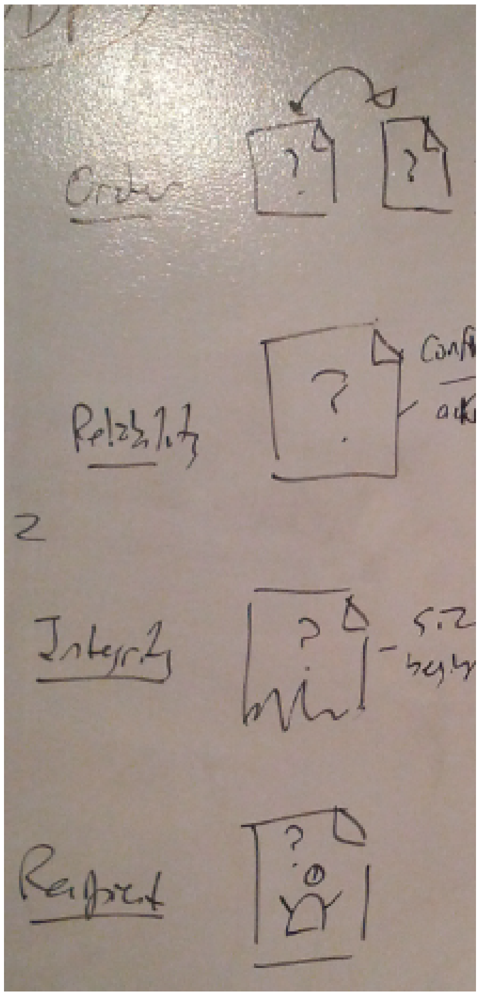
1. Reliability: He cannot guarantee that any particular message will arrive at all (Symbol: a piece of paper with a question mark on it)

2. Order: He cannot guarantee that messages which DO make it will arrive in the same order in which they were sent. (Symbol: Two pieces of paper marked 2 and 1 with a question mark between them)

3. Integrity: He cannot guarantee that messages will arrive in their entirety (Symbol: a torn piece of paper)

   and ...

4. Recipient: He cannot guarantee that messages will be delivered to the correct recipient. (symbol: a head-and-shoulders human icon with a question mark over them)



The Exercise

Students now work with their pair partner for ~30 minutes to agree on a system that mitigates these four problems as much as possible, remembering that after they agree on this system, they'll be across the country from each other and can only communicate through the inimitable Mr. Pakkitz.

The first 15 minutes should be spent talking generally about ideas.

The second 15 minutes should be spent crafting actual messages on actual index cards. This forces people to think about specifics: does each message start with a single-line header and the body underneath it? Are the headers on the back of the card?

Ideally, students will come up with something pretty similar to how IP actually works, involving sending the same message over and over again until you get a message that says the message was received (reliability/integrity), and attaching metadata to the packets themselves (recipient/order).

Review, Lecture, and Discussion

Now the instructor circles Ivan Pakkitz's name and writes his initials, IP.

> You guys just designed the internet! Ivan Pakkitz is IP, IP is Internet Protocol. You went through the same process that experts went through when figuring out how to make all of this stuff work, and you even came to a lot of the same conclusions.

Once the students are done, call on each group and ask them how they solved each of the problems.

Directions to go in from here:

- Did any students get REALLY specific about their "protocol"? Ask them to show you some index cards they made. Which of the decisions were arbitrary? This index card says "Recipient: Max" but this one says "From: Max". Which one is it? You have to make a decision even though neither is clearly superior.

  This is a GREAT point to drive home early in the course, because A TON of confusion points in learning to program come from being subject to the apparently arbitrary decisions of developers who came before us.

  Covering it now creates a reference point for the entire course: "Remember when we designed the internet and had to pick a format for our messages even though we couldn't find something perfect? Every programmer has to do that whenever they write code, including the programmers who wrote the browsers, and interpreters, and templating systems, and frameworks that we are using."

- Talk about how all of this is often interpreted as just plain text. Drive home the idea of "headers" vs. "body" as this will come in handy when talking about HTTP.

- Talk about what routers do. Internet Service Providers want to give each customer a single IP address, but if you have three computers, how do you handle that? Well, the router lets your ISP pretend that you only have one device connected to their network. To your router, your laptop is 192.168.0.1 and your XBox is 192.168.0.5, but to the rest of the world they're whatever your router's IP address is. Your ISP is to your router as your router is to your devices. Every time your router opens a connection, it remembers which device actually requested the connection. This process is called Network Address Translation (NAT).

- Introduce Ivan's new associate, Tony C. Packets (TCP). Tony has done all of this protocol business before, and he has colleagues at every postal hub in the country who will work with Ivan to deliver your messages whole, reliably, in the right order, and to the right person.

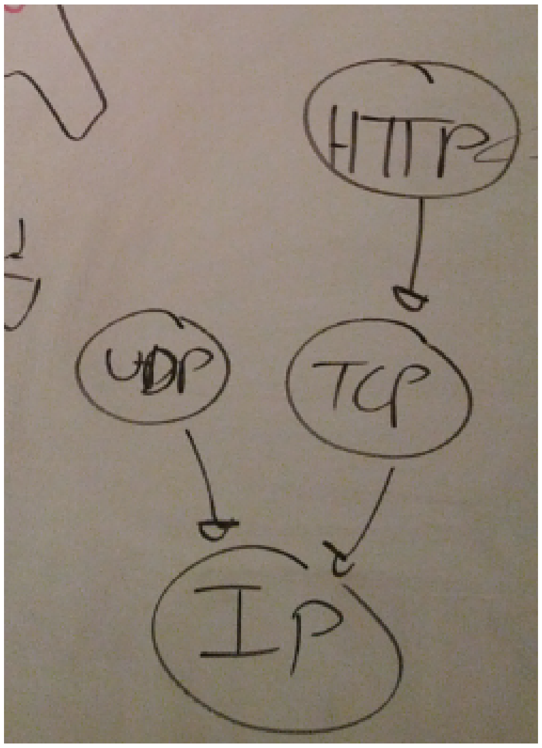
- Tony's system (TCP) requires that you specify a port which you must agree upon with your friend beforehand. One good analogy for describing ports is like ports in a marina: the oil tanker usually arrives in Port 5, the cruise liner usually arrives in Port 50. Web servers usually listen on Port 80, remote shells usually listen on Port 22.

- Tony's colleagues are very strict, and ALWAYS take extra time to double-check things, so it takes much longer: your messages might take a whole second to arrive!

- So Tony's services (TCP) are great for plain text messages like e-mails and IMs and a whole lot of other things, but if you really, really need to go fast and send LOTS of data, like for streaming Netflix and playing video games, you need a connectionless protocol with UDP, which is pretty much exactly the same as IP, but has ports. It doesn't matter if you miss a few Netflix packets; you just get an "image tearing" effect like what everyone sees all the time. Now they know why: it's running on UDP, not TCP.

- Touch on IPv4 vs. IPv6. The people who originally designed the IP address system didn't expect us to give IP addresses to toasters and windmills (http://www.shodanhq.com/) and we ran out of IP addresses. IPv4 only supported 2^32 addresses, about 4 billion. IPv6 supports 2^128 addresses which would be enough for every atom on the surface of the Earth.

- Finally you can introduce HTTP, and illustrate the hierarchy of protocols:



That's all I got!

Reply
3 people liked this. Like it too.

- **Created**

  17 Apr

- **Last post**

  3 days

- **Posts**

  4

- **Views**

  23

- **Users**

  2

- **Likes**

  4
- 2

  2

Reply as new Topic

aldric

select +replies select
14d

Very cool. I'm just sad about the whole 'drawing' thing, because I'm really bad at it, but I suppose I can practice 😄

This is a great way to get people to think about the consequences of their choices, as well as implementations to solve a particular problem.

There's a very high chance that we will be reusing this 😄

Reply

Reply as new Topic
Loading...

Loading...

Invite Friends Star Share Clear pin Flag Reply

Regular

- WatchingYou will be notified of every new post in this topic. The count of unread and new posts will also appear next to the topic's listing.
- TrackingYou will be notified only if someone mentions your @name or replies to your post. The count of unread and new posts will also appear next to the topic's listing.
- RegularYou will be notified only if someone mentions your @name or replies to your post.
- MutedYou will never be notified of anything about this topic, and it will not appear on your unread tab.

You will be notified only if someone mentions your @name or replies to your post.

Suggested Topics

Topic	Category	Posts	Likes	Views	Activity	
Improving instructor talent recruitment new		3		6	1d	2h
“Programming Sucks”	Awesomeness	1	1	11	3d	3d
Upcoming features in ECMA6	Awesomeness	1	3	15	29d	29d
Raw Github Displaying Live HTML	Awesomeness	2	1	15	2 Apr	24d
Better Bash Scripting in 15 Minutes	Awesomeness	1	1	12	14d	14d

There is 1 new topic remaining, or browse other topics in Awesomeness.

share a link to this topic

quote reply