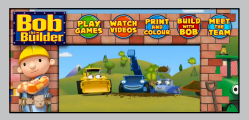
SECURITY (COMP0141): NETWORK BACKGROUND

UCL

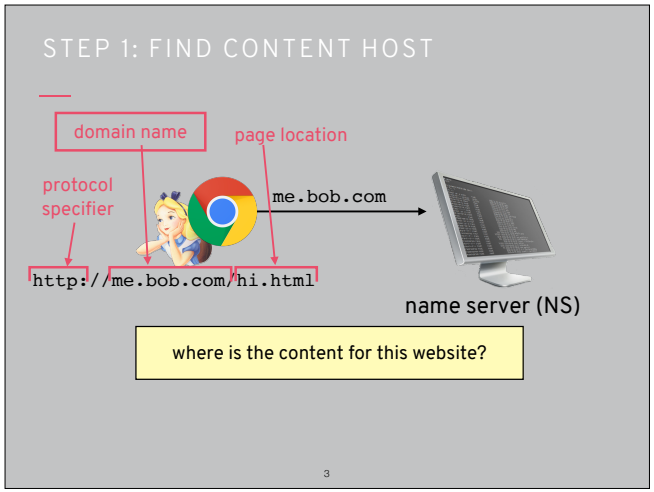
HOW DOES THE INTERNET WORK?

goal: get Alice to that website!

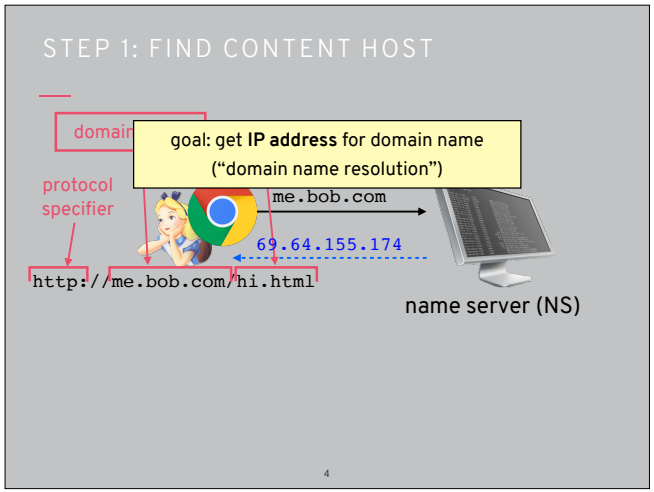


http://me.bob.com/hi.html

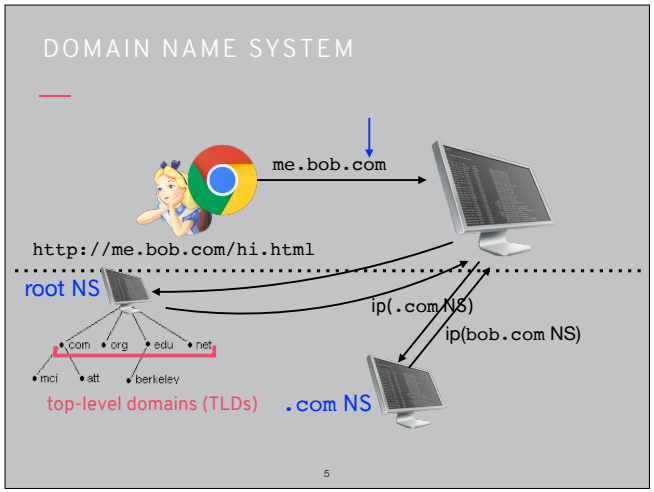
Let's go over how the Internet works. We'll go more into depth on some of this later on, but having some background will be useful next week and going forward in general, and will highlight some security risks that we'll address later on



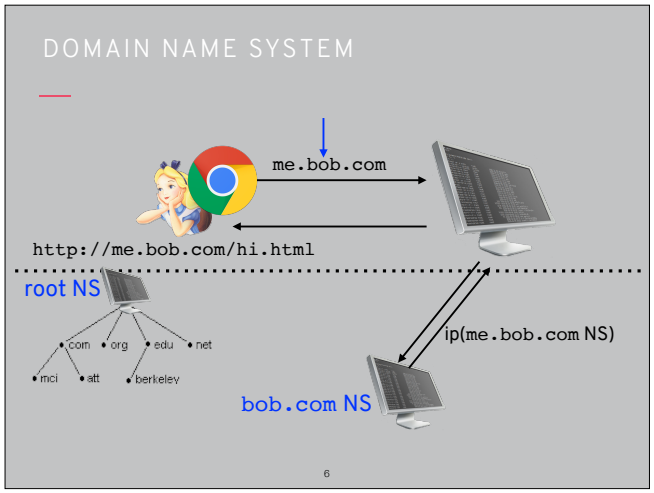
How does Alice find content for the website?



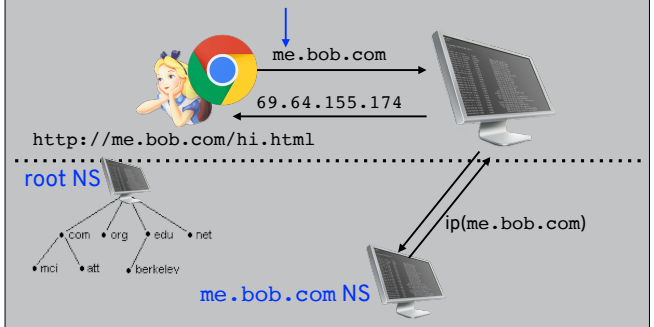
Start by finding IP address, which is a unique identifier



Need to perform domain-name resolution using domain name system (DNS). Start from a top-level domain (TLD) like .com and go down in hierarchy. These days there are a lot more TLDs like .london and there are some interesting business models here



DOMAIN NAME SYSTEM



DOMAIN NAME SYSTEM

FAQs

q: do we really do this every time we go to a website?

a: no! DNS results are cached by your browser.

q: where do you find first name server? or root server?

a: basically, they're hard-coded.

First: we absolutely don't do this every time we visit a website, answers are cached

Second: the way we find name servers is hard-coded in our computer and/or browser

DOMAIN NAME SYSTEM

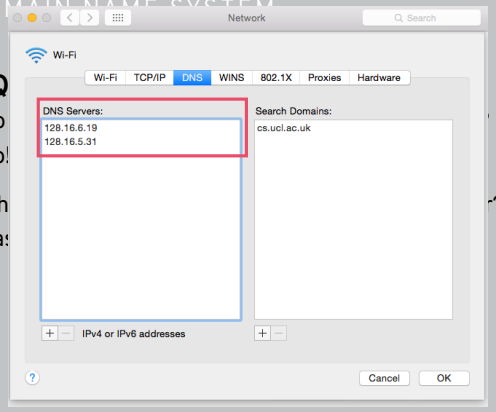
FAQ

q: do

a: no!

q: wh ?

a: ba



DOMAIN NAME SYSTEM

FAQs

q: do we really do this every time we go to a website?

a: no! DNS results are cached by your browser.

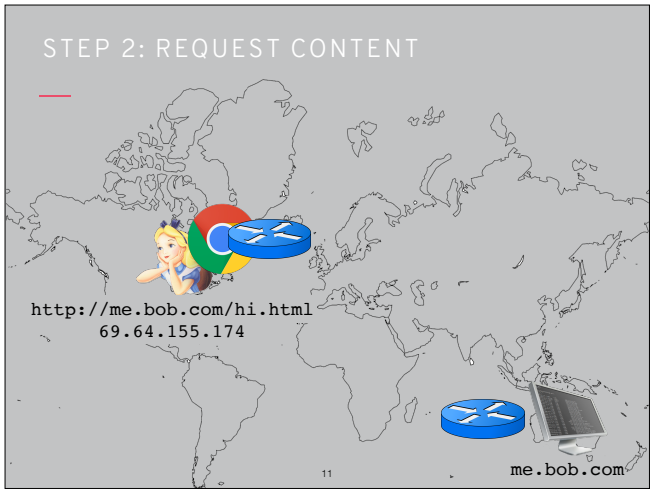
q: where do you find first name server? or root server?

a: basically, they're hard-coded.

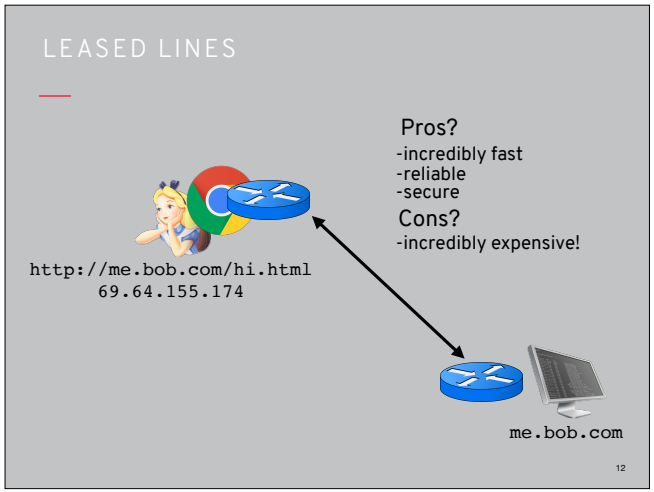
q: how do you actually find these name servers?

a: the magic of routing!

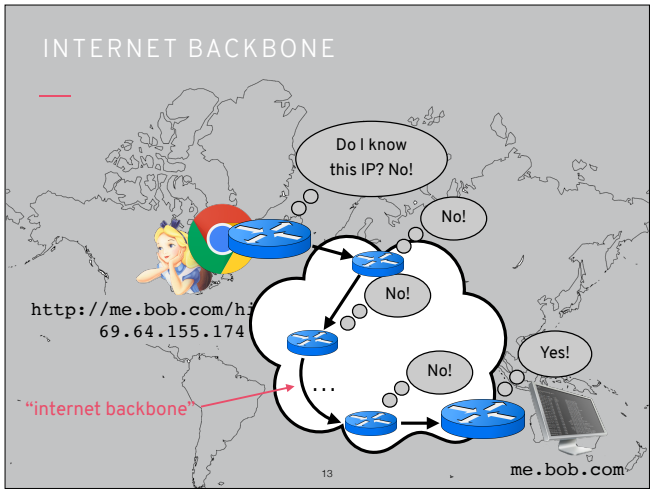
Third: we find servers using routing. This is also how we proceed once we have the IP address for the website we want to visit



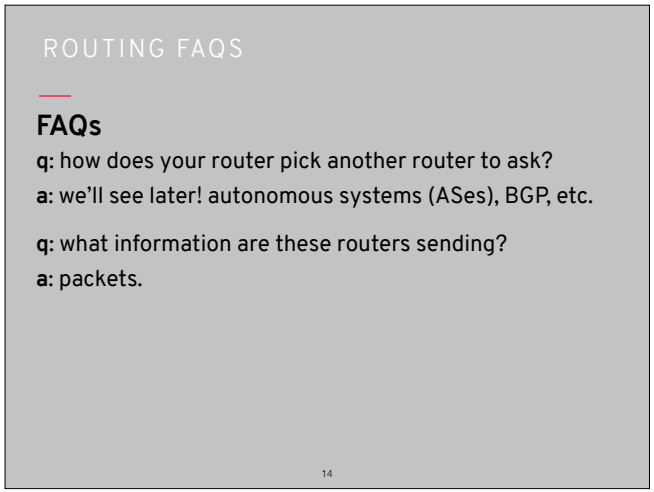
Reminder: Alice and Bob are both physical entities, based somewhere in the world



Simplest thing to imagine is direct line (Internet cable) between Alice and Bob. This would be great but it's incredibly expensive and can't exist between every pair of users on the Internet

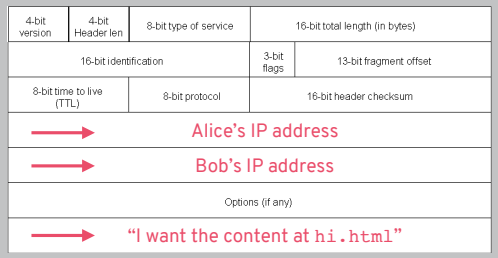


Most routing is done instead via the Internet backbone, which we'll learn more about in later weeks



Internet backbone is run by ISPs, huge business and again an interesting governance structure

PACKET



ROUTING FAQS

FAQs

q: how does your router pick another router to ask?

a: fascinating topic! autonomous systems (ASes), BGP, etc.

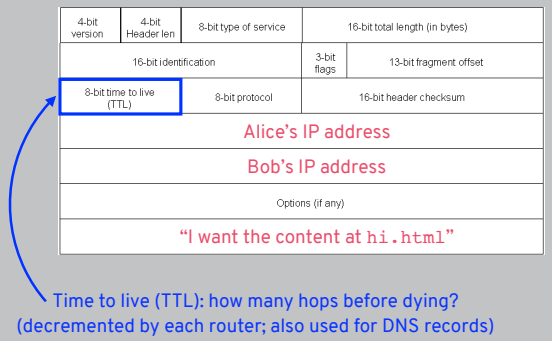
q: what information is Alice's router forwarding?

a: packets.

q: could requests just go around in a circle?

a: no! packets contain information on when to give up.

TIME TO LIVE



Time to live (TTL): how many hops before dying?
(decremented by each router; also used for DNS records)

TTL helps to eliminate noise (in the form of unhelpful or unnecessary packets)

ROUTING FAQS

FAQs

q: how does your router pick another router to ask?

a: fascinating topic! autonomous systems (ASes), BGP, etc.

q: what information is Alice's router forwarding?

a: packets.

q: could requests just go around in a circle?

a: no! packets contain information on when to give up.

q: what happens once Bob's server gets this request?

a: everyone lives happily ever after!

ALICE'S REQUEST

4-bit version	4-bit Header len	8-bit type of service	16-bit total length (in bytes)	
16-bit identification			3-bit flags	13-bit fragment offset
8-bit time to live (TTL)		8-bit protocol	16-bit header checksum	
Alice's IP address				
Bob's IP address				
Options (if any)				
"I want the content at `hi.html`"				

BOB'S RESPONSE

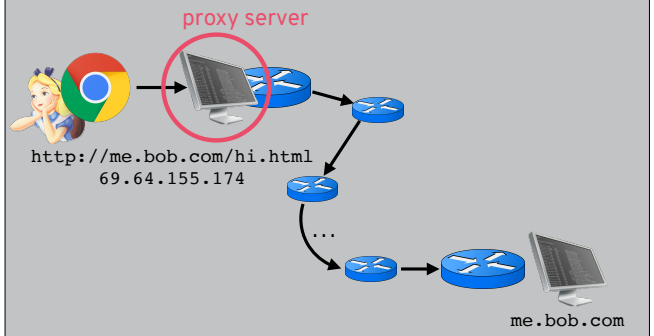
4-bit version	4-bit Header len	8-bit type of service	16-bit total length (in bytes)	
16-bit identification			3-bit flags	13-bit fragment offset
8-bit time to live (TTL)		8-bit protocol	16-bit header checksum	
Bob's IP address				
Alice's IP address				
Options (if any)				
<Content at `hi.html` (part 1 of N)>				

as is, anyone can see which sites you're visiting

source and destination IP addresses never get modified!

Think of a packet as an envelope and a letter combined (so, a postcard)

ASIDE: PROXY SERVERS AND VPNS

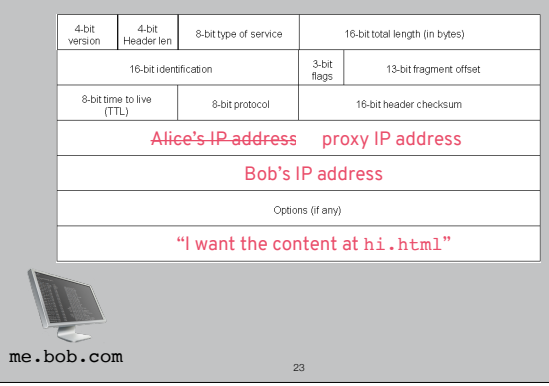


ASIDE: PROXY SERVERS AND VPNS

4-bit version	4-bit Header len	8-bit type of service	16-bit total length (in bytes)
16-bit identification		3-bit flags	13-bit fragment offset
8-bit time to live (TTL)	8-bit protocol	16-bit header checksum	
Alice's IP address			
~~Bob's IP address~~ proxy IP address			
Options (if any)			
"I want the content at `hi.html`"			

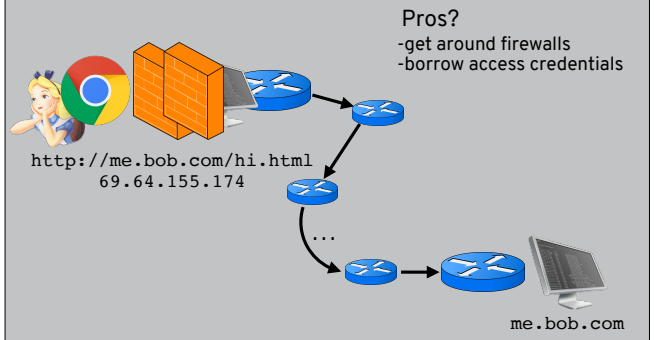
To Alice's computer it looks like Alice is talking to the proxy server

ASIDE: PROXY SERVERS AND VPNS



And to Bob's computer it looks like Bob is talking to the proxy server

ASIDE: PROXY SERVERS AND VPNS



So Alice and Bob are talking to each other, but everyone thinks they're talking to the proxy. This is useful for dealing with firewalls (like Great Firewall) or using credentials (like VPNing into UCL to read research articles)

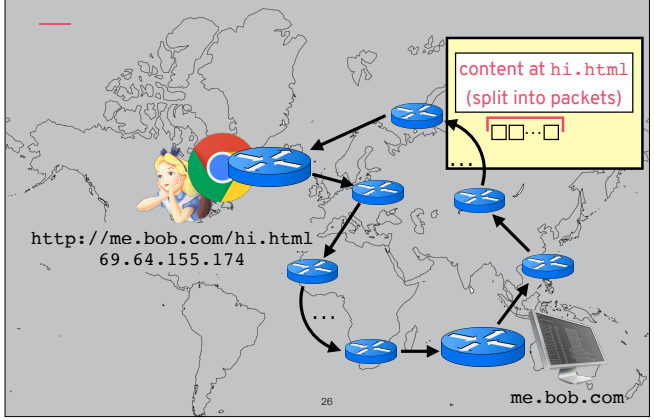
BOB'S RESPONSE

4-bit version	4-bit Header len	8-bit type of service	16-bit total length (in bytes)	
16-bit identification			3-bit flags	13-bit fragment offset
8-bit time to live (TTL)		8-bit protocol	16-bit header checksum	
Bob's IP address				
Alice's IP address				
Options (if any)				
<Content at `hi.html` (part 1 of N)>				

25

Back to where we were: this is the content of the website going back to Alice

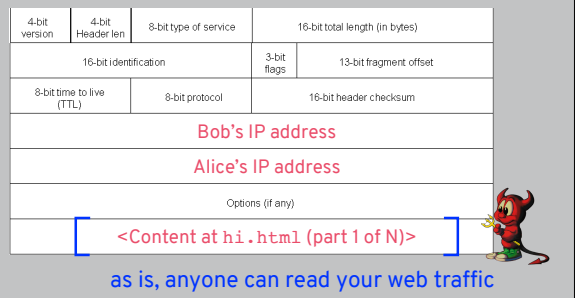
STEP 3: RECEIVE CONTENT



26

Packets go back, in sequence, using potentially different route

BOB'S RESPONSE

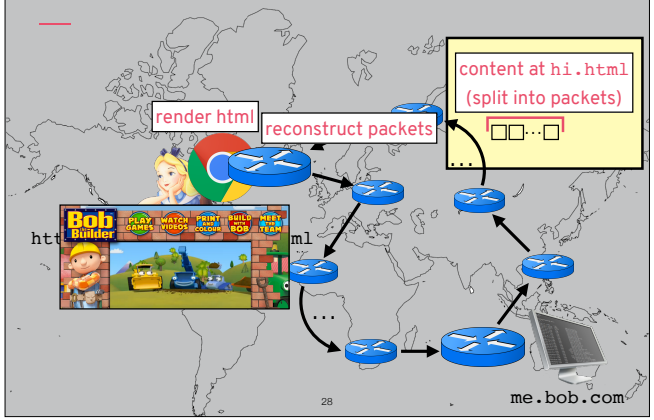


as is, anyone can read your web traffic

27

Again, packets are not just an envelope but also a letter. We'll get back to the security issues around this next week

STEP 3: RECEIVE CONTENT



28

Once Alice has the packets, can reconstruct them (in order), render the HTML code in the browser, and see the website

HOW DOES THE INTERNET WORK?

goal: get Alice to that website!

find content host

get IP address for `me.bob.com` ("domain name resolution")

request content

send `GET` request to IP address (routing via the internet backbone)

`i.html`

receive content

wait for response from IP address, then render `hi.html` and enjoy

QUIZ!

Please go to

`https://moodle.ucl.ac.uk/mod/quiz/view.php?id=2723780`

to take this week's quiz!