0:00:00 And if I keel over and fall down, somebody call the 0:00:03.497 ambulance. OK, what we're going to do 0:00:06.015 today is, this is actually the last lecture on a topic that 0:00:10.071 some might consider part of networking, but some might 0:00:13.778 consider part of a more general topic of distributed systems. 0:00:17.975 So it actually forms a bridge between the stuff we learned 0:00:21.962 about in networking, and what we're going to see 0:00:25.249 from Wednesday over the next six or seven lectures of the class, 0:00:29.655 and of the associated recitations having to do with 0:00:33.153 fault tolerance, reliable computing. 0:00:37 And most of the interesting aspects of what we're going to 0:00:40.931 talk about involve techniques in fault tolerance, 0:00:44.172 and more techniques have to do with redundancy and replication. 0:00:48.448 And DNS, the domain name system, which is the system 0:00:51.965 we're going to look at today in the context of distributed 0:00:55.896 naming is a bridge because on the one hand it covers some of 0:00:59.965 the aspects of networking that we talked about. 0:01:03.137 On the other hand, it shows an example of how you 0:01:06.448 can achieve replication to achieve fault tolerance. 0:01:11 And were going to study these techniques systematically over 0:01:15.739 the next few lectures. So getting an example in mind 0:01:19.836 is usually a good idea. So you've already seen the 0:01:23.772 network layer, and you've seen that. 0:01:26.584 At the network layer, attachment points on the 0:01:30.199 network are identified by IP addresses in the internet are 0:01:34.778 more generally identified by network layer addresses. 0:01:40 So as an example on the Internet are IP addresses. 0:01:43.769 And, this is the name that's used by the network layer to 0:01:48.076 identify attachment points anywhere on the network. 0:01:51.923 And in fact, there's a name that's used by 0:01:55.076 the end to end layer to name one endpoint of connection. 0:02:00 So in fact if you think about it and go back to our very early 0:02:04.066 lecture on naming, an address is really just a 0:02:07.066 name, but a name that's been overloaded with information that 0:02:11.066 allows the user of that name to locate this object. 0:02:14.4 An address, really, is a name that has information 0:02:17.666 in it, overloaded information in it that actually allows for it 0:02:21.8 to be located. And in fact, 0:02:23.533 an IP address is nothing other than a name that tells you where 0:02:27.666 in the Internet topology, the entity being named by this 0:02:31.333 IP address is located. So my computer is something 0:02:35.66 like 18.31.0.35. It doesn't mean anything other 0:02:38.49 than the fact that somewhere on this Internet topology there is 0:02:42.303 this big, complicated graph. And that address allows you to 0:02:45.871 do routing in the geography of that topology. 0:02:48.577 It has nothing to do with real-world geography. 0:02:51.406 It just allows you to write it down in that topology. 0:02:54.605 Now, in principle, you could build every Internet 0:02:57.557 application, and have users interact with Internet 0:03:00.571 applications purely with these network layer addresses. 0:03:05 But that would be quite inconvenient. 0:03:07.205 I mean, you would then have to be sending e-mail to your 0:03:10.575 friends, or not, with Joe@MIT.edu. 0:03:12.781 But, you'd have to do Joe at some IP address. 0:03:15.477 And, it's pretty hard and complicated to remember. 0:03:18.479 So the first problem with just using pure network layer 0:03:21.787 addresses that we want to solve today, and we will solve to some 0:03:25.647 degree, although not completely is to come up with a better way 0:03:29.446 of naming things that are more convenient. 0:03:33 And you already know the answer. 0:03:35.324 The answer is you send e-mail to Joe@MIT.edu. 0:03:38.849 You go to the 6.033 website at MIT.edu/6.033 or some other 0:03:43.125 equivalent thing that leads to the same page. 0:03:46.425 You don't actually think about names in terms of IP addresses. 0:03:51 So in fact, to a large extent, the fact that these are human 0:03:55.425 understandable and names that are mnemonics that you can 0:03:59.55 easily remember is a good thing. And so we do actually want to 0:04:05.362 come up with a way of naming things that's independent of IP 0:04:10.168 addresses. The second goal here is to come 0:04:13.508 up with a naming scheme with a solution that allows some degree 0:04:18.558 of modularity. As you know, 0:04:20.676 names provide a level of indirection between the thing 0:04:24.994 that you want to get to, and the handle that you want to 0:04:29.474 associate with it. And that level of indirection, 0:04:33.8 if we come up with a good way of doing this, 0:04:36.379 it will allow us to do a few things like, for example, 0:04:39.56 I can tell you that the website for MIT.edu, for the institute's 0:04:43.339 homepage is MIT.edu. And, behind that, 0:04:45.56 I could change the actual computers on which the website 0:04:48.86 is located. And I could do that 0:04:50.66 independently of telling other people of any change in it, 0:04:54.079 whereas if I told them that the website was at a particular IP 0:04:57.74 address, then every time I moved a page, moved the pages from one 0:05:01.579 computer to another, I have to tell everybody in the 0:05:04.639 world that the website has changed. 0:05:08 And we'd like to minimize doing that kind of thing. 0:05:10.987 And so, the domain name system provides a solution to this 0:05:14.393 problem. DNS provides a solution to this 0:05:16.723 problem. Most of you have already heard 0:05:18.993 of this. It maps between what are 0:05:20.905 formerly called domain names, but what we are going to just 0:05:24.371 call host names for convenience. It maps between host names and 0:05:29.11 records. And it turns out there are many 0:05:31.855 different kinds of records. And we're going to look at a 0:05:35.724 few of them in this lecture today. 0:05:38.046 But for your mental model right now, just assume that it maps 0:05:42.268 between host names and IP addresses. 0:05:44.731 So it turns out that an IP address is just one example of a 0:05:48.812 record called an address record. But the general goal of DNS is 0:05:53.174 to map between host names and records. 0:05:55.778 And, there is a variety of them, as I said. 0:06:00 For now, just assume they are IP addresses. 0:06:02.733 So for example, MIT.edu might be 18 dot 0:06:05.206 whatever it is as an IP address. So what are the goals in 0:06:08.85 designing the system? So, primarily we're going to be 0:06:12.234 talking about how the domain name system is designed, 0:06:15.618 and how it works. And as we go along, 0:06:17.96 we'll see some things that it does that are different from 0:06:21.67 other systems or other ways you can solve this problem. 0:06:25.184 There are basically two goals in the design of the system. 0:06:30 And the first goal is that it should scale. 0:06:33.149 In fact, the original motivation for the domain name 0:06:36.975 system when it came about in the early 80s was that it was 0:06:41.25 becoming extremely inconvenient to manage the mappings between

0:06:45.824 names of hosts and their IP addresses in a way that as the 0:06:50.1 network grew, and it was growing pretty 0:06:52.949 rapidly even then, as the network grew it turned 0:06:56.475 out to be a management nightmare. 0:07:00 And to understand this, basically there's three ways in 0:07:03.006 which you could imagine; there are more than three ways. 0:07:05.958 But there are three more obvious ways in which you could 0:07:09.02 imagine mapping between these names and these records, 0:07:11.972 so the names and IP addresses. And the first one is, 0:07:14.812 in fact, the way the Internet names were being managed until 0:07:18.097 DNS came along. And that's to use a model that 0:07:20.603 you might think of as the telephone book model. 0:07:24 0:07:28 It's actually astonishing the telephone companies use the 0:07:31.294 telephone book model where every six months at your doorstep 0:07:34.764 there are these three thick books that show up. 0:07:37.47 And you're just looking at it and going, what do I do with 0:07:40.823 this? And you lug it to your home, 0:07:42.764 and by the time you might use a couple of times, 0:07:45.529 and then the next big three books come along. 0:07:48.117 You actually wonder why they don't just put their phone books 0:07:51.647 on the Web and make it easy to get at. 0:07:53.823 But the telephone book model really is, there's this central 0:07:57.294 repository of information. And everybody gets this 0:08:01.43 information from the center repository. 0:08:03.794 But the repository actually pushes it to everybody. 0:08:06.904 So it's not really a queryable repository. 0:08:09.827 It's not something you go in contact on an as needed basis. 0:08:13.435 So, in fact, every few months, 0:08:15.239 or in the old days of the Internet, everyday in the 0:08:18.349 morning, every computer would go and pull the current mapping 0:08:22.081 between names and addresses from a central site. 0:08:26 And this model kind of works for a little bit, 0:08:28.528 but stopped working. It stops scaling on multiple 0:08:31.224 levels. First of all, 0:08:32.348 the resources required for everybody to have to go and 0:08:35.325 collect this information every day turns out to be significant. 0:08:38.808 But it also turns out to be a scaling bottleneck more 0:08:41.73 fundamentally from the standpoint of any time you add a 0:08:44.764 machine to your local organization, 0:08:46.674 you have to go and tell the central person that you've added 0:08:49.988 this machine. And so, at a human level it 0:08:52.067 doesn't scale very well because you can't allow for this 0:08:55.157 automatically happen because then people would sort of 0:08:58.134 willy-nilly do all sorts of, you know, just claim that they 0:09:01.393 own various machines or various names at various places. 0:09:06 So the telephone book model is something that the Internet used 0:09:09.894 to have. It used to be a file card 0:09:11.967 host.txt. And, every computer had a copy 0:09:14.417 of this file that usually was current as of a 24 hour period. 0:09:18.185 So, the second approach that you could adapt knowing this 0:09:21.703 approach of just sort of pushing a telephone book periodically 0:09:25.535 doesn't work. You might actually adapt a 0:09:27.984 model, a centralized server model. 0:09:31 0:09:35 And the central server model is, the main difference from the 0:09:38.339 telephone book model is that nothing is pushed to you. 0:09:41.289 Instead, imagine a search engine like, say, 0:09:43.627 Google or something like that where if you want to know the 0:09:46.856 mapping between a name and in it the address, you go and contact 0:09:50.362 this name service, which turns out to be a central 0:09:53.09 server. And it does essentially the 0:09:54.982 same thing that Google might do. And this actually isn't that 0:09:58.322 hard to implement from a technical standpoint because 0:10:01.216 Google does much more than this. And clearly it works. 0:10:05.342 But the real problem with the central server model is what I 0:10:09.114 alluded to the last time. It doesn't handle, 0:10:11.863 when you assign names to machines, you want to make sure 0:10:15.378 that people don't conflict on these names because these are 0:10:19.086 well-defined names for machines on the Internet. 0:10:22.091 And you need a model by which you can decide who is allowed to 0:10:25.99 name a machine as fool.MIT.edu and who's 0:10:28.675 allowed to take X.CNN.com, and so on. 0:10:32 And so, for every computer in the Internet, 0:10:34.932 having a central person deal with deciding whether that's OK 0:10:39.051 or not isn't a scalable solution. 0:10:41.285 And that's why we don't really adopt that model for DNS. 0:10:45.125 The model that's adapted for DNS, and has a number of other 0:10:49.174 attractive properties, which we'll talk about, 0:10:52.316 is the distributed database model, or more generally, 0:10:55.946 a distributed, federated model where every 0:10:58.809 organization, and organizations could be 0:11:01.531 recursively defined to have sub-organizations. 0:11:06 Every organization sort of manages a portion of this 0:11:09.03 overall global namespace. And it manages everything about 0:11:12.358 it. It manages all of the mappings 0:11:14.319 between names and records for that part of the namespace. 0:11:17.647 And the more names they have, the more work they have to do. 0:11:21.153 But nobody else really has to do that much more work. 0:11:24.243 And it's a pretty nice model because everybody does some 0:11:27.512 amount of work in terms of technical resources, 0:11:30.245 and more importantly in terms of human administrative 0:11:33.336 resources. And overall, 0:11:35.779 that's one component of the scalability of the system. 0:11:39.222 And that's one of the main reasons why the system turns out 0:11:42.989 to scale and work very well. And the second goal is 0:11:46.237 reliability. Once you have a system like 0:11:48.77 this, and people start getting used to it, and applications 0:11:52.538 start using names, it had better be the case that 0:11:55.656 the system is in fact generally available. 0:11:58.319 And it better not be the case that the DNS be the Achilles 0:12:02.022 heel of the Internet in terms of, you know, 0:12:04.75 the network infrastructure has a very, very high reliability 0:12:08.582 because all this running stuff works out great. 0:12:13 And we find all these alternate paths. 0:12:15.312 And things don't work because I'm not able to get to my DNS. 0:12:19 That had better not be the case. 0:12:20.937 So we'll see what techniques DNS uses to get pretty good 0:12:24.375 reliability. But the jury is till out as to exactly how 0:12:27.687 reliable it is. But overall most users will 0:12:30.312 admit that generally it seems to work about as well as the rest 0:12:34.187 of the network. So, what does DNS do? 0:12:37.466 Fundamentally, it provides an abstraction 0:12:40.259 called a lookup abstraction. You give it a name. 0:12:43.541 It returns a record to you. And you can ask for particular 0:12:47.73 types of records that you want. We'll get into that in a 0:12:51.571 second. So you ask a name, 0:12:53.316 and you get back a record. And, there's many ways, 0:12:56.738 and different operating systems have different ways in which 0:13:00.857 this exact function is called. For example, 0:13:04.888 get host by name might be a commonly used way of going from 0:13:09.185 a name to an IP

address. But, we are going to just say 0:13:13.111 DNS resolved just more abstractly. 0:13:16 0:13:21 So an application can invode DNS resolve, DNS name, 0:13:25.076 and get an IP address or some other record associated with it. 0:13:30 Now if you think back at the way we did our idealized naming 0:13:34.561 model or genetic naming model, we actually had a resolve which 0:13:39.276 took a name and a context as an argument. 0:13:42.369 And, it returned back the value that was bound to that name. 0:13:46.93 So you might ask what the context is for a DNS resolve. 0:13:51.104 And there is actually multiple answers to this question. 0:13:55.124 The first answer is that applications specify a context 0:13:59.299 usually. 0:14:01 0:14:05 So, for example, an application might specify 0:14:08.067 that. It wishes to know the IP 0:14:10.088 address of this name. For example, 0:14:12.389 if it's a web browser that wants to know some server's IP 0:14:16.293 address, so it can connect to it using the TCP connection. 0:14:20.266 Alternatively, an application like an e-mail 0:14:23.264 program might specify that it wants not the IP address of this 0:14:27.516 machine, but the name of a machine that can handle mail on 0:14:31.49 behalf of this name. So, for example, 0:14:35.066 if I send somebody email to ABC.MIT.edu, my mail program, 0:14:38.799 somewhere along the way some server would do a lookup for a 0:14:42.666 special kind of record turns out to be called the MX record, 0:14:46.6 which is a mail record, which would then return to 0:14:49.866 caller not the IP address of MIT.edu, but the IP address of, 0:14:53.799 in fact, we learn a name of a machine that's capable of 0:14:57.399 handling mail for MIT.edu. And in general, 0:15:01.239 that would be very different from the IP address associated 0:15:06.033 with MIT.edu. So that's established as a 0:15:09.256 context. And more generally, 0:15:11.487 there is a DNS configuration file. 0:15:14.214 There is some DNS configuration on registry, or whatever it is 0:15:19.256 depending on the system that you have that tells this resolve 0:15:24.38 step what context in which the name must be resolved. 0:15:30 So just for example, you might have a machine, 0:15:33.185 cricket.MIT.edu, another machine 0:15:35.38 cricket.berklee.edu. And DNS resolve might call DNS 0:15:38.92 resolve of cricket. And when the application 0:15:41.964 calls that, the program doing the 0:15:44.442 resolution needs to know what context to do it in. 0:15:47.911 And often, that's specified in the DNS configuration. 0:15:51.592 So on Windows and UNIX, there is this thing called a 0:15:55.203 search path. And you go through a set of 0:15:57.964 search paths that provide this context. 0:16:02 So will see this in more detail later today. 0:16:04.794 So users themselves, and there's an important point 0:16:08.044 here about lookups, and how lookups are to be 0:16:10.903 distinguished from something else that are called search. 0:16:14.542 Now, these names are generally very useful for programs because 0:16:18.572 they allow modularity to occur, where it no longer are you 0:16:22.276 worried about services being associated with IP addresses. 0:16:25.981 You can move your website between IP addresses of the back 0:16:29.685 without telling everybody about it. 0:16:33 So from that standpoint this name to record mapping a DNS 0:16:36.79 with lookups is very, very useful. 0:16:39.024 In terms of convenience, the DNS is not the whole story 0:16:42.679 because although you most often send email to people with 0:16:46.469 fool.MIT.edu so you remember that or you have some other file 0:16:50.53 in which you find that information very easily, 0:16:53.644 more generally speaking people often don't, we need something 0:16:57.705 in addition to just lookups. Human users need something in 0:17:01.563 addition to lookups. And the general term given to 0:17:05.857 that is search. So for example Google provides 0:17:08.739 search on the Internet. And tomorrow's recitation talks 0:17:12.198 a little bit about one aspect of that. 0:17:14.568 And in fact, it's an interesting discussion 0:17:17.258 because you will find from the reading tomorrow that users were 0:17:21.229 using a search engine to essentially do a lookup task 0:17:24.559 where they would go to Google and type CNN.com when in fact if 0:17:28.466 they already knew to CNN.com they could just as well have 0:17:32.053 typed it on their URL window. And that's sort of the way real 0:17:36.819 users turned out to use the Web. But over peer-to-peer 0:17:39.93 applications that most of you might be more familiar with than 0:17:43.51 I am have a form of search. And those applications are 0:17:46.62 interesting because they do searches on all sorts of 0:17:49.613 attributes of the content that you want, and by and large don't 0:17:53.252 really use DNS in a particularly, 0:17:55.13 if they use it at all it's sort of incidental. 0:17:57.772 They probably don't even need to use it. 0:18:01 So it's not like all Internet applications require DNS in 0:18:04.966 order to work. In fact, there are plenty of 0:18:07.941 applications that don't need DNS at all. 0:18:10.704 So, and they benefit a lot from search. 0:18:13.395 OK, so let's get back to DNS and talk a little bit about a 0:18:17.433 few different topics. We're going to start first with 0:18:21.116 the namespace, and how it works. 0:18:23.312 And then were going to talk about how name resolution works. 0:18:27.491 And then we're going to talk about things like performance 0:18:31.529 and scalability and robustness. So the DNS namespace has two 0:18:37.015 properties to it. The first is, 0:18:39.174 in both of these are nice ideas, and hit upon a theme, 0:18:42.989 or at least one of which hits upon a theme we've covered in 0:18:47.164 6.033. DNS is a hierarchical system. 0:18:49.683 And it's a structured namespace. 0:18:51.915 I'll describe what structured means in a moment. 0:18:55.298 So the way our DS namespace works is it's a hierarchical 0:18:59.257 system which is structured as a tree. 0:19:03 And at the top of the tree there is a little circle which 0:19:06.763 really is a dot. And I'm going to call that the 0:19:09.854 root. So, everything starts at the 0:19:12.072 root. And the root is the root of 0:19:14.223 this namespace. And it's a tree. 0:19:16.306 The namespace is divided into A, a bunch of domains. 0:19:19.733 And domains are divided into subdomains. 0:19:22.354 And subdomains are recursively divided into sub-subdomains, 0:19:26.252 and so on. And in fact, 0:19:28.845 the depth is arbitrary. It could be arbitrarily long. 0:19:32.511 In practice, nobody really has need for 0:19:35.189 depth more than four or five. And, in fact, 0:19:38.149 90% of the names, or more than 90% of the names 0:19:41.392 are pretty flat. You don't go more than two 0:19:44.352 levels down. So at the top level, 0:19:46.607 this is pretty familiar to most of you. 0:19:49.286 You would have com, and edu, and net, 0:19:51.823 and org, and gov, and a few others. 0:19:54.22 And these things, there is actually about 13 of 0:19:57.462 them right now. These things are called 0:20:01.147 generic, top-level domains, so generic in the sense that 0:20:03.892 they are not really associated with any country, 0:20:06.238 for example. And then in addition to this, 0:20:08.284 you have a whole bunch of country codes like dot US, 0:20:10.829 and I don't know

how many countries there are, 0:20:13.074 but there's a large number of them. 0:20:14.771 So those things are country code top level domains. 0:20:17.266 And they are not that interesting in that there's 0:20:19.662 nothing different about them from anything else. 0:20:22.007 So we may as well just look at the generic top-level domains. 0:20:26 OK, and edu gets divided into MIT and other places that don't 0:20:31.158 matter and so on and so forth. So you might end up down here. 0:20:36.316 You might have www, or website, or for whatever the 0:20:40.615 student's from. I don't know who actually owns 0:20:44.484 this. C sale might be here. 0:20:46.719 EECS might be here. And I might have a machine 0:20:50.588 underneath here, let's say, X just a random 0:20:54.457 machine. And, the thing about the DNS 0:20:57.552 namespace is that every label here, this is a label, 0:21:01.936 right? So the way you read this is if 0:21:06.283 you just start at any label here and go upward, you can read it out 0:21:10.926 from bottom to top. So you would say com is an 0:21:14.351 example of a label. MIT, edu, root is a label. 0:21:17.776 So that's read as MIT.edu dot. And usually people omit the 0:21:22.115 trailing dot because it's implicit. 0:21:24.703 Or you X.csale.MIT.edu dot is another fully formed, 0:21:28.509 what's it called, fully qualified domain name. 0:21:33 OK, that's sort of the correct way to read it out is from 0:21:36.329 bottom to top. Now, every node here is 0:21:38.528 associated with some information. 0:21:40.43 And that's what this record means. 0:21:42.392 OK, some nodes might have nothing in it, 0:21:44.711 but in general, every node is associated with 0:21:47.326 some information. So, for example, 0:21:49.288 X might have information here. I'm just going to call it INFO 0:21:52.855 for now. But, X might have associated 0:21:54.995 with it an A record, which is an IP address record. 0:21:57.968 I'll describe that in a moment. But it might have other things 0:22:02.551 associated with it. In fact, DNS is pretty 0:22:04.672 flexible. You could define your own 0:22:06.431 record and put that into the system, and have applications 0:22:09.379 that read from it. It's pretty opaque to it. 0:22:11.603 It doesn't really require; if you have something new that 0:22:14.5 comes up and you want to use it, you could stick it into DNS and 0:22:17.758 retrieve it out. So people put all sorts of 0:22:19.931 things now into DNS. For example, 0:22:21.586 there are weird proposals. But in GPS coordinates of a 0:22:24.327 name. So if you know MIT.edu isn't 0:22:26.034 going to move very much, then put in the GPS 0:22:28.258 coordinates. You might find applications 0:22:30.275 that find it useful. For example, 0:22:31.931 if you are some mobile computing application, 0:22:34.206 you might find it useful. So there's all sorts of things 0:22:39 you could stick into the DNS. And people have come up with 0:22:42.799 all sorts of very wacky things, including telephone numbers in 0:22:46.866 DNS. It's very flexible. 0:22:48.4 So not only are the leaves associated with things with 0:22:51.933 information, but you can have information at any level, 0:22:55.533 any node in the tree has information associated with it. 0:23:00 And the scale of this namespace today is extremely vague. 0:23:03.853 I mean, I don't know the exact number of things there are, 0:23:07.776 but I've read that about 500 million, I don't know if it's 0:23:11.698 250 million or 500 million, somewhere in between, 0:23:15.002 there are that many registered names in the system in 0:23:18.58 aggregate. That's a pretty big number. 0:23:21.126 Now, it's a very big number. So you need a way in which you 0:23:25.118 can make the system scale. And that's done using a really 0:23:28.972 nice idea called delegation. And more than any technical 0:23:33.824 decision that was made in DNS, I mean, we're going to talk 0:23:37.675 about some of them like caching and all this other stuff. 0:23:41.459 But more than any technical decision, delegation is really 0:23:45.31 the reason DNS scales. And, it's ultimately really the 0:23:48.891 reason why DNS is pretty successful. 0:23:51.256 So, what is delegation? The best way to understand it 0:23:54.77 is recursively. So the root at the top is 0:23:57.472 centrally owned, and it's owned by a trusted 0:24:00.378 entity. So what that means is that any 0:24:03.912 name that ends in root, ultimately the authority for 0:24:07.163 that name rests with whoever owns and runs root. 0:24:10.159 And if you paid attention to the press, and the newspapers, 0:24:13.856 or magazines, you'll see that there's this 0:24:16.47 fight for the root that's ongoing right now over the past 0:24:20.039 few years. And, the current owner of the 0:24:22.525 root and things associated with it, and who essentially controls 0:24:26.541 the namespace, is an entity called ICANN, 0:24:29.091 I-C-A-N-N, and there's a lot of politics associated with it. 0:24:34 Now, continuing down on the delegation idea, 0:24:37.868 the next layer down from the root, the technical term for it 0:24:43.176 is top-level domain because it's at the top level: 0:24:47.584 TLD, OK? And, these top-level domains 0:24:50.823 that are delegated from the root, and it's really hard to 0:24:55.861 come up with a new top-level domain. 0:25:00 In relation to these, you have a few more. 0:25:02.315 But you don't really come up with them willy-nilly. 0:25:05.138 You kind of have to go through a long procedure before the root 0:25:08.639 decides to delegate the top-level domain to somebody 0:25:11.519 else. And now it's recursive from 0:25:13.326 here on. Every label here can be 0:25:15.076 sub-delegated arbitrarily by only contacting the owner of 0:25:18.238 that label. So once you get to com, 0:25:20.158 you don't have to go further down. 0:25:22.021 You just have to go to whoever happens to own the com label and 0:25:25.522 tell them that you want to register a new portion of the 0:25:28.628 namespace with that. So, for example, 0:25:32.181 MIT must have gone at some point to the owner of edu and 0:25:36.243 said, I want MIT. And, there's some 0:25:38.754 out-of-band human procedure that occurs before the other 0:25:43.037 party is convinced that this MIT is a legitimate entity, 0:25:47.099 and allocates this name to it, and likewise. 0:25:50.275 Whoever wanted CSALE went to the person who runs MIT's name. 0:25:54.632 This is called a zone, this DNS namespace and told it 0:25:58.473 that it wanted CSALE, and established by human 0:26:01.796 efforts rather than anything technical that it wanted that 0:26:06.005 portion. Now, the reason it scales is that 0:26:09.521 you can kind of add machines, you know, names at the bottom, 0:26:12.728 not machines but names anywhere here without really having to go 0:26:16.152 all the way up. You just need to go up to 0:26:18.326 whoever owns your parent label and convince them that you want 0:26:21.641 a name. So if I want to add a machine, 0:26:23.652 Y, I don't really have to go and talk to even anybody at MIT. 0:26:26.913 I just have to talk to the person who runs the CSALE 0:26:29.684 namespace and tell them that I want a name, Y. 0:26:33 And I can, then, sub-delegate that name. 0:26:35.146 I could, today, connect a computer 0:26:36.962 X.CSALE.MIT.edu have in this IP address with it, 0:26:39.548 and tomorrow decide I don't really want X to be a computer. 0:26:42.74 I want it to be the name of my research group or whatever. 0:26:45.876 and then have machines underneath it which are 0:26:48.352

Y.X.CSALE.MIT.edu. What I do with the label is my 0:26:50.994 business. And there's no rule that these 0:26:53.14 are IP addresses. In fact, these are nothing. 0:26:55.561 These are just labels. And I can associate arbitrary 0:26:58.368 amounts of information I want with that label. 0:27:02 0:27:06 So, domains can be formed anywhere in tree. 0:27:11 0:27:17 And that's really nice because you don't have to go back to a 0:27:20.646 central entity in order to add these names. 0:27:23.198 And that's the main reason why the central server model doesn't 0:27:26.966 really fly. 0:27:28 0:27:34 OK, so examples of records, we've already seen a few of 0:27:37.298 these. So let's look in more detail at 0:27:39.558 what this info could contain. Like I said, 0:27:42.063 it's very flexible. You can have all sorts of 0:27:44.751 things you add in here. But there's a few very common 0:27:47.927 ones. The first one is called an A 0:27:49.943 record, which stands for an address record. 0:27:52.509 And, it's what you might expect. 0:27:54.402 It's an IP version for a address for a name. 0:27:56.968 So, X.CSALE.MIT.edu, whatever its IP address is. 0:28:01 So, that's stuck in this database. 0:28:03.062 It's really maintained in a file on the name server that 0:28:06.5 handles that name. You might have MX, 0:28:08.75 which stands, the X is for mail exchanger. 0:28:11.312 So, it's mail exchanger. So, that's for email. 0:28:14.125 So, when I send email to you@MIT.edu, somewhere along the 0:28:17.625 way there's a lookup done for not the IP address of MIT.edu, 0:28:21.312 but the MX record for MIT.edu. And in general, 0:28:24.125 the MX record could be anywhere. 0:28:26.062 If MIT decided to outsource its email functionality to some 0:28:29.687 other company, the MX record would just bind 0:28:32.375 to some name of a mail server in that other company. 0:28:37 So it doesn't even have to be local to us. 0:28:40.037 There's another one that's interesting in the context of 0:28:44.111 stuff we've seen before called a C name, which stands for a 0:28:48.407 canonical name. But a C name is really a 0:28:51.296 synonym. This is where you can say there 0:28:54.185 are many names that really mean the same thing. 0:28:57.592 So, for example, to take a very real example, 0:29:00.851 there used to be AI Lab and LCS, and now there's the same 0:29:05 lab, CSALE. Now, there are a lot of 0:29:08.428 subdomains from LCS.MIT.edu, and AI.MIT.edu. 0:29:11.099 And sort of it's a nightmare to have to go and change all of the 0:29:15.012 DNS entries for all of the machines. 0:29:17.186 So the standard way in which you manage this kind of thing is 0:29:20.913 to set up these C names, which says [UNINTELLIGIBLE] 0:29:24.08 NMS.LCS.MIT.edu. That's what it was. 0:29:27 You just set up a C name that says NMS.LCS.MIT.edu, 0:29:30.005 there's a C name for NMS.CSALE.MIT.edu. 0:29:32.289 So, you don't have to do anything more other than set up 0:29:35.595 these synonyms. And everything else just sort 0:29:38.24 of continues to work out. There are other useful things 0:29:41.486 you could do with C names. For example, 0:29:43.77 if you decide you want, you're running your Web server 0:29:46.956 on one machine, and then you want to change it 0:29:49.661 over to another machine but not have to tell the whole world 0:29:53.207 about it, what you do is you tell everybody that your Web 0:29:56.573 server's running, let's say, MIT.EDU. 0:30:00 And then you set up a C name for that MIT.edu to 0:30:02.774 machineone.MIT.edu. It's another name. 0:30:04.959 And then someday you decide to change from machineone.MIT.edu 0:30:08.501 to machinetwo.MIT.edu. All you have to do is just 0:30:11.335 change this one mapping in the backhand in the DNS that maps 0:30:14.819 from MIT.edu, set up a C name mapping to a 0:30:17.239 different machine, and that's all you have to do. 0:30:20.073 You don't have to tell anybody else about the change that 0:30:23.38 you've made. So, it's very useful. 0:30:25.328 And this is just an example of a more general concept that 0:30:28.693 we've seen already called a synonym. 0:30:32 And the fourth thing, which is actually what we're 0:30:36.376 going to spend most of our time on today, for the rest of today, 0:30:42.12 is called an NS record, which is a name server record. 0:30:46.952 And, this is the thing that's really going to help us figure 0:30:52.332 out how to implement DNS resolve in a scalable way. 0:30:56.891 I will describe what a name server record is in a moment. 0:31:03 So if you look at this tree, associated with many of the 0:31:07.294 labels in the tree are not just A records, which are generally 0:31:12.057 associated, but also things called NS records. 0:31:15.571 And, what an NS record says is that, let's say there's an NS 0:31:20.178 record associated with MIT.edu. What it says is that that NS 0:31:24.785 record gives the name of the machine that's responsible for 0:31:29.314 managing all of the names that end with MIT.edu. 0:31:34 So, for example, edu wouldn't know anything 0:31:36.537 about, in general, edu may not know anything about 0:31:39.497 how CSALE.MIT.edu is really mapped. 0:31:41.552 But all edu needs to know is what the NS record is for 0:31:44.754 MIT.edu so that as you'll see in this procedure, 0:31:47.593 you'll see that occasionally things at the top of the tree 0:31:51.037 will get requests for a given name, and they need to know what 0:31:54.722 to do with the full name. And, the way they'll do it is 0:31:57.985 they'll find out that they don't know anything about the full 0:32:01.61 name. But they know about other 0:32:04.674 people who know more about that name. 0:32:07.086 And that's going to be implemented using this thing 0:32:10.435 called an NS record. So I'll show this with an 0:32:13.449 example. I think it'll become pretty 0:32:15.794 clear. So the way in which 0:32:17.468 applications use DNS is you have an application that wants to 0:32:21.488 call DNS resolve on a DNS name. 0:32:23.631 And, there's a piece of software that usually runs on 0:32:27.114 every computer called a stub resolver. 0:32:31 OK, and a stub resolver is just something that allows 0:32:34.698 applications to not have to worry about this whole RPC 0:32:38.467 mechanism that DNS involves, that DNS entails. 0:32:41.668 So, that's just abstracted way into the 0:32:44.797 stub resolver. So the stub resolver really 0:32:47.713 does all of the work. So the way the stub resolver 0:32:51.198 works is, and the way DNS resolution works is that the 0:32:54.967 stub resolver really can send a DNS request that it has from the 0:32:59.448 application to any name server that it wants to. 0:33:04 And later on we'll talk a bit about how you pick this name server. 0:33:07.617 And there are lots of these name servers around. 0:33:10.5 On the internet, it's a massive distributed 0:33:13.075 infrastructure. And, the infrastructure 0:33:15.405 consists of people, of nodes that have 0:33:17.674 responsibility for different portions of this namespace. 0:33:21.047 So, you send a request to any name server, and they all 0:33:24.358 participate in this system together, these name servers to 0:33:27.853 help resolve names. So let's take an example here. 0:33:32.52 Let's say it's X.CSALE.MIT.edu. So, at this point in general 0:33:37.476 the name server knows nothing about X.CSALE.MIT.edu. 0:33:41.761 And it's plan's going to be that it's going to send this 0:33:46.382 request out to

the root name server, OK? 0:33:49.658 And for now, assume that the root name 0:33:52.766 server is well-known, that is, the IP address of the 0:33:57.051 name server in charge of the root. 0:34:01 So everybody has a name server that got associated with 0:34:04.262 themselves. Assume that the IP address of 0:34:06.679 the name server of the root is just well known to everybody, 0:34:10.243 OK? So it sends X.CSALE.MIT.edu all 0:34:12.297 the way to the root name server. And the root name server is 0:34:15.862 actually going to look at this thing and say, 0:34:18.52 well, you want to know the A record. 0:34:20.635 You might want to know the A record for X.CSALE.MIT.edu. 0:34:23.958 It says, but I don't actually know what the IP address 0:34:27.16 associated with this name is. But I do know somebody who can 0:34:31.875 tell you more because I do know who runs the name service for 0:34:35.625 edu. So, it comes back with the 0:34:37.5 response that's also called a referral saying, 0:34:40.312 well, I don't know the answer, but here's where you need to go 0:34:44.125 in order to find out more. And that's done by sending back 0:34:47.687 the name server, or in fact more generally a set 0:34:50.625 of name servers, but sending back the NS records 0:34:53.562 for nodes that handle the edu domain. 0:34:55.812 So that just comes back at you. And now, you now know one or 0:34:59.5 more name servers for the edu domain. 0:35:03 So, let me write that here. And this name server then goes 0:35:06.327 back to this edu domain and says, OK, tell me what the IP 0:35:09.596 address for X.CSALE.MIT.edu is. And, notice that at all stages, 0:35:13.216 it's sending the full name because there's always some 0:35:16.31 chance that these nodes have the answer, and we'll get to why 0:35:19.813 that might be in a few minutes. But you always send the full 0:35:23.258 name. And the edu name server record 0:35:25.301 in this case is, in general, going to say, 0:35:27.694 well, I don't know about X.CSALE.MIT.edu. 0:35:30.03 But I do know that MIT came and delegated from me. 0:35:34 So I do know the name server record associated with MIT 0:35:37.383 because it delegated from me. And in general, 0:35:40.139 everybody has to know the name server records for the people 0:35:43.836 one level down from them. So it sends back this 0:35:46.718 information saying, well, I don't know what it is. 0:35:49.787 But here's a referral to the name server for MIT.edu. 0:35:53.045 And this procedure basically continues. 0:35:55.426 So this is MIT. Actually it's just the MIT.edu 0:35:58.245 name server. And you just go back and forth 0:36:02.119 until eventually you get to the main server for CSALE.MIT.edu, 0:36:06.428 which by definition maintains the mapping for everything of 0:36:10.525 the form NAME.CSALE.MIT.edu. And so, you get the answer. 0:36:14.41 Or you get something that says X is not actually registered in 0:36:18.719 which case you get a no-such-domain error message. 0:36:22.181 Actually it's a no-such-domain error code, which you then 0:36:26.136 interpret as saying, OK, there is no such name 0:36:29.315 that's been registered. Now there's a couple of things 0:36:34.329 of note here. The name server records 0:36:37.124 associated with the domain really have to have very little 0:36:41.55 to do with that domain. In fact, the name server record 0:36:45.743 for, forget the root names for a moment. 0:36:48.771 The name server records for the edu domain don't actually have 0:36:53.508 to end in edu, or don't have to end in 0:36:56.381 anything that relates to edu. In practice, 0:36:59.564 in fact, today they're of the form something.NSTLD.com. 0:37:05 I mean, they have nothing to do with the edu domain. 0:37:08.531 And, this is an important point. 0:37:10.678 You could associate here any label, any name of a name server 0:37:14.832 that's willing to manage the delegation, manage the entries 0:37:18.849 in your database. They don't actually have to 0:37:21.896 match the same domain name. And, this is a very powerful 0:37:25.704 feature of the namespace of the way DNS works because it's not 0:37:29.928 like this thing has to be something.edu, 0:37:32.629 and here the name servers for MIT.edu have to be actually 0:37:36.507 something.MIT.edu. In practice, 0:37:39.782 edu is not managed by anything.edu. 0:37:42 It's something, NSTLD.com. 0:37:43.63 In practice, it so happens that MIT.edu 0:37:46.108 happens to run its own name servers; that's something like 0:37:49.826 bitsy.MIT.edu. But that's by no means a 0:37:52.5 requirement. 0:37:54 0:38:09 So there's a couple of problems that we need to solve about this 0:38:12.871 basic mechanism it does what we saw was that once you know the 0:38:16.62 root name server record, then you could go back and 0:38:19.692 forth because everybody knows how the names one level down 0:38:23.195 from them have been delegated and knows the name server 0:38:26.513 entries for those delegates. And then you get the final 0:38:29.832 answer. So there's a few things we need 0:38:33.246 to solve. The first one is bootstrap. 0:38:35.855 There's actually a couple of aspects of this bootstrap. 0:38:39.768 The first one is, how does this any name server 0:38:43.101 here know the name servers of the root? 0:38:45.855 And, the answer here is that there's no magic. 0:38:49.115 I mean, you just have to know. And, in some sense, 0:38:52.666 in all naming systems, ultimately there is, 0:38:55.71 at some level of this name discovery procedure, 0:38:59.043 at some level there's out of band machinery that has to get 0:39:03.246 in and be involved. And the way this works out is 0:39:07.702 that people publish the name server records for the root. 0:39:10.991 They post it on websites. They publish it on mailing 0:39:13.985 lists. And you can figure DNS software 0:39:16.158 is configured to manage those entries. 0:39:18.33 And there are proposals and protocols for automatically 0:39:21.501 updating it and so on. But you've got to be very 0:39:24.261 careful with technical solutions like that because you want to 0:39:27.843 make sure that malicious bodies don't pretend that they're 0:39:31.19 telling you the correct name server entry because once they 0:39:34.596 usurp DNS functioning, they could do a lot of damage. 0:39:39 In practice, in fact, DNS is not 0:39:41.233 particularly secure. It's a different discussion as 0:39:44.835 to whether that's important or not. 0:39:47.285 But the way this root name server mapping works is that 0:39:51.175 everybody just knows. It's widely published. 0:39:54.273 You go to Google and you'll just find the answer. 0:39:57.731 It's also on all sorts of mailing lists. 0:40:00.541 So the first one is root identity. 0:40:04 The second issue is, how does the stub connect to 0:40:07.138 any name server? Would it just pick at random? 0:40:10.08 How does it find the name server that it can connect to? 0:40:13.676 And the answer to this is that this is actually running on your 0:40:17.73 computer. The stub result was running on 0:40:20.28 your machine. It's a library in your machine. 0:40:23.157 So, one approach, and the most common approach 0:40:26.099 today is that you might obtain it when you obtain an IP address 0:40:30.153 using a protocol like DHCP where you turn on your computer and 0:40:34.076 you get an IP address using some protocol. 0:40:38 That protocol, some gateway upstream might 0:40:41.257 tell you which

address using some protocol or else. What protocol, some gateway upstream might service tell you which name server to use. 0:40:43.879 So if you have a computer at home, your Internet service 0:40:48.249 provider would have something set up. 0:40:51.109 So they would tell you what name server to use. 0:40:54.764 So this can be done using a mechanism like DHCP. 0:40:58.419 Or like in the old days, and these days if you want to 0:41:02.63 do this kind of work, it's done manually. 0:41:07 For example, you could go into some registry 0:41:09.852 somewhere or file like Kazaa.com. 0:41:11.975 Then you just start stuff in. The second issue that we need 0:41:15.823 to spend some time about, which we are going to spend the 0:41:19.538 next five or so minutes on is performance. 0:41:22.258 And the main point about performance is that if you look 0:41:25.907 at that picture for how names are being resolved, 0:41:29.092 if somebody at Berkeley wants to resolve 0:41:32.276 X.CSALE.MIT.edu, there's a huge number of 0:41:34.93 roundtrips that they have to do to all sorts of name servers all 0:41:39.11 over the world, or at least all over the 0:41:41.697 country before they can figure out what the IP address is from 0:41:45.744 a machine. And this is a little bit silly 0:41:49.854 because often, they might want to connect to 0:41:52.512 your webpage and get a 4 kB thing, which takes a couple of 0:41:56.036 roundtrips to get. And, in order to do that, 0:41:58.694 they're spending a huge number of roundtrips latency, 0:42:01.909 use latency in getting this right answer. 0:42:05 So the approach that we're going to take to solve the 0:42:08.026 too-much-too-many roundtrips problem is an approach you've 0:42:11.343 already seen. We're going to use caching. 0:42:13.671 And, in order for this caching to work, there is actually 0:42:16.931 something in this picture that you have to understand in a 0:42:20.248 little bit more detail about two different kinds of name 0:42:23.449 resolutions that are really going on. 0:42:25.544 The first kind of name resolution that's going on in 0:42:28.513 DNS is the kind of resolution that the edu name server or the 0:42:32.005 MIT.edu name server is doing in this picture, 0:42:34.566 and that's called iterative resolution. 0:42:38 Iterative resolution says the following. 0:42:40.135 If I ask you to do some work for me to resolve a name, 0:42:43.036 if you don't know the answer you just tell me, 0:42:45.5 I don't know the answer, but you go here and find out 0:42:48.346 the answer. So you're getting these 0:42:50.208 pointers referring you to the right place. 0:42:52.452 The second kind of resolution that's going on is the kind of 0:42:55.682 resolution that this box is doing, this any name server box. 0:42:58.912 And it's doing something called recursive resolution because 0:43:02.142 what's going on here is the stub resolver's telling it, 0:43:05.098 here's a name; resolve it for me. 0:43:08 And what he's doing is basically saying, 0:43:10.256 OK, I'll resolve it for you and get you the final answer. 0:43:13.495 I'm not going to give you a referral back to other places, 0:43:16.793 which means eventually once it figures out the answer, 0:43:19.859 if there is one, it's going to know the answer 0:43:22.462 even though it did not originate the query. 0:43:24.892 And the moment you have recursive resolution, 0:43:27.438 it means that you can cache the answer because if somebody else 0:43:31.024 comes and asks you the same query, you already have the 0:43:34.148 answer cached. And notice that this benefit 0:43:37.847 doesn't accrue if you're doing purely iterative resolution. 0:43:41.675 If everybody was just sending referrals back to the stub 0:43:45.306 resolver, the only caching you'd really primarily be gaining is 0:43:49.398 for all of the requests that are common to this computer because 0:43:53.556 nobody is actually cashing in getting any answers along the 0:43:57.384 way. Nobody's even getting any 0:43:59.297 referrals along the way. Only the node that's starting 0:44:02.795 the name resolution is going to be getting any answers or any 0:44:06.755 referrals at all. So the secret to getting good 0:44:10.977 DNS performance is for certain nodes, for certain name servers, 0:44:15.204 to agree to do recursive resolution -- 0:44:18 0:44:23 And an example of that is any name server in this picture. 0:44:26.293 So now if you have, a lot of the computers here 0:44:28.951 with a lot of applications running on it, 0:44:31.262 all of which use the same name server, and that name server is 0:44:34.786 configured to recursively resolve names, 0:44:37.097 then that name server benefits from being able to cache the 0:44:40.448 answers to previous DNS lookups. But notice that they can cache 0:44:44.031 two kinds of answers, and one of which is much more 0:44:46.92 important than the other. The first kind of answer they 0:44:50.039 can cache is the final answer that you get back that says 0:44:53.275 X.CSALE.MIT.edu is at a particular IP address. 0:44:55.875 So the next time somebody goes to X.CSALE.MIT.edu, 0:44:58.706 they have the answer right there. 0:45:02 But if you look at the statistics of DNS requests, 0:45:04.731 there is some commonality in that everybody wants to go to 0:45:07.908 www.CNN.com or Google.com or Yahoo.com, and so on. 0:45:10.639 But, there's a huge number of requests going to machines like 0:45:13.983 yours and mine which aren't running anything interesting. 0:45:17.104 And you are really the only people interested in those 0:45:20.059 machines. So really what's going on, 0:45:22.009 and why caching helps is that not only are the final answers 0:45:25.298 being cached, these referrals are being 0:45:27.416 cached. So for example, 0:45:28.642 any name already knows the root, but now after the first 0:45:31.875 request it knows the mapping for edu's name service. 0:45:36 And after the first one to MIT.edu, it can cache the 0:45:39.313 mapping for MIT.edu's name server as well. 0:45:41.976 So, the next time somebody asks for anything.MIT.edu, 0:45:45.354 this name server doesn't have to go all the way back to the 0:45:49.122 root. In fact, it doesn't even have 0:45:51.331 to go all the way back to edu. It just has to start with 0:45:54.904 MIT.edu whose name service entry it already has in its cache. 0:45:58.802 And, that really is the reason why the DNS scales very, 0:46:02.31 very well. It's because it does caching. 0:46:05.639 But the real key is that it's caching referrals. 0:46:08.295 It's caching these name server entries associated with these 0:46:11.63 labels. It's getting some benefit from 0:46:13.721 caching the final answers, but it's really getting a lot 0:46:16.83 of benefit from caching referrals. 0:46:18.695 Now, of course, when you cache things, 0:46:20.786 you have to worry about being still because you certainly 0:46:23.952 don't want to cache it forever. If you cached it forever, 0:46:27.117 then nobody could change anything. 0:46:30 So let's say you decide to change the mapping of 0:46:32.972 www.MIT.edu's A record from one IP address to another. 0:46:36.324 How do you tell the whole world that you've changed it? 0:46:39.74 Well, there's two high-level strategies for this. 0:46:42.776 One is to somehow keep track of all the people who have cached 0:46:46.634 it and invalidate entries. And a few lectures from now, 0:46:50.049 we'll look at ways in which that kind of approach might be 0:46:53.654 made to work for different systems. 0:46:55.805 DNS deals with it in sort of a different way. 0:46:58.588 It doesn't worry about invalidation. 0:47:00.801 Instead, it sets expiration

time on entries also called TTL 0:47:04.47 is a time to live. That's an abused and overloaded 0:47:08.642 tone in networks. But it's really an expiration 0:47:11.079 time. It says that here's the answer 0:47:12.933 to any of these questions, to a referral or to the final 0:47:15.847 answer. Here's the mapping. 0:47:17.225 And, it's valid for such and such a time, OK, 0:47:19.556 like it could be anywhere from 15 seconds or 30 seconds to 0:47:22.576 three hours or a day. Usually it's a day or two. 0:47:25.066 It's usually never more than a couple of days because you reach 0:47:28.35 the point of diminishing returns. 0:47:31 One request every two days is not a big deal. 0:47:33.708 So usually it's on the order of several seconds if you want the 0:47:37.526 mapping to be fine-grained, or an hour, or a day. 0:47:40.481 And, after that, any access made after that to 0:47:43.251 the same entry, whether it be a referral or 0:47:45.837 whether it be a final answer, has to go back to the server 0:47:49.347 that's authoritative that's responsible for that entry. 0:47:52.671 So, for example, you went for the first time 0:47:55.319 doing the lookup of this name, and you got back that MIT.edu's 0:47:59.074 name service was some name, and that it was valid for an 0:48:02.46 hour. Then the first request that 0:48:05.825 happens for anything.MIT.edu from here after an hour has to 0:48:09.745 go here. I mean, assuming that edu is 0:48:12.178 still a valid, hasn't yet expired. 0:48:14.409 This entry has to go here. So, if you look at a time 0:48:17.856 sequence of when you go, when you actually go back to 0:48:21.371 the server responsible for a name, you can kind of divide up 0:48:25.359 time into these chunks which are the expiration time intervals 0:48:29.483 assuming they don't change. They are set by the server. 0:48:33.133 And then, you might have accesses in between like this. 0:48:38 And the only accesses that go to the server responsible for 0:48:42.167 the name are the first ones after every expiration. 0:48:45.76 So, this is the basic story for how you get reasonably good 0:48:49.928 performance, and save a lot of roundtrips in DNS. 0:48:53.377 And the real reason for the scalability of the domain name 0:48:57.473 system is a combination of administrative delegation. 0:49:02 So you don't have some human being involved in every name 0:49:05.629 that's being added to the network. 0:49:07.768 It's distributed across different organizations. 0:49:10.814 And the fact that you have caching, in particular your 0:49:14.25 caching these name server records, which means that you 0:49:17.75 can save a lot of load on the root and on the edu or com name 0:49:21.638 servers. Originally, the designers of 0:49:23.972 DNS, one thought that they had was that DNS would scale very 0:49:27.796 well because the name space is extremely divided and 0:49:31.101 hierarchical, which means that you were 0:49:33.564 gaining a lot from the hierarchy. 0:49:37 And that's why it would scale. But that's actually not true 0:49:40.471 because more than 90% of the domain namespace is not 0:49:43.523 hierarchical in any deep sense. Everybody is something.com. 0:49:46.995 Everybody of importance is something.com, 0:49:49.389 or wants to be something.com. And so, most of the load gets 0:49:52.86 here. So, there's no real deep 0:49:54.596 hierarchy that you're benefiting from here. 0:49:57.109 That's usually some flatname.com. 0:49:59.024 But it's divided. And it's delegated. 0:50:02.019 And that's the really nice thing about it. 0:50:04.22 So you are gaining much more from the fact that it's 0:50:06.957 delegated. You'd probably gain the same 0:50:08.996 scalability is everything were just something.root, 0:50:11.679 OK? We didn't really need too much 0:50:13.45 of this in terms of scalability, although it's very convenient 0:50:16.724 to be able to do delegation. But primarily it seems to be 0:50:19.729 universities that take advantage of this kind of depth here. 0:50:22.896 Most companies tend not to pay much attention to depth. 0:50:25.794 But yet, the system scales because it is, 0:50:27.94 in fact, administratively delegated. 0:50:31 So one word on replication. The DNS name servers 0:50:33.514 responsible for these names are replicated. 0:50:35.761 You can't set up a DNS name and have a name service associated 0:50:39.025 with it. A DNS name server record has to 0:50:41.112 have at least two entries, and you have to be on two 0:50:43.841 different networks. And that's one way so that if 0:50:46.409 one of them is down you can get to the other. 0:50:48.763 And unfortunately it turns out that that simple rule is often 0:50:51.974 violated. Probably the most celebrated 0:50:53.953 incident here was Microsoft's update site, or one of the sites 0:50:57.217 was down for more than 24 hours. And in the end it turned out in 0:51:01.655 all these cases to be a complicated set of reasons for 0:51:04.484 why it failed. Nothing fails for a simple 0:51:06.62 reason. But one of the root causes was 0:51:08.596 that they had DNS name servers that were replicated but that 0:51:11.746 happened to be behind the same Ethernet switch on the same 0:51:14.789 subnet, which is not recommended. 0:51:16.498 But that's what they had. So it's the kind of thing that 0:51:19.434 you need to be careful about doing. 0:51:21.249 So I'm going to stop here. And from Wednesday, 0:51:23.652 we will talk about fault tolerance. 0:51:25.574 The recitation for tomorrow is a very short paper called Google 0:51:28.884 and 9/11. But tomorrow you'll see a 0:51:30.7 little bit about how Google works. 0:51:32.462 51:30 52:00 52:30 53:00 53:30 54:00 54:30 55:00 55:30 56:00 0:51:35.558 56:30 57:00 57:30 58:00 58:30 59:00 59:30 60:00