6.033 systems critique 1: Example Solution
(In the interest of space, we removed specific citations from the text)

Paper	DNS
Summary of system: What is it? What does it do?	DNS maps domain names to IP addresses, and provides support for looking up those mappings.
Does DNS respond to a previous system? What problem is solved by DNS?	Yes, to the hosts.txt "system". DNS solved the problem of scalability with that system.
What are the modules of the system? How do they interact?	- Clients and nameservers are the primary modules - Nameservers contain the mappings for certain domains - Clients look up a domain by communicating with relevant nameservers (one nameserver for each dotted component of the domain name) - Basic case: client contacts root server, which delegates to a TLD server, which will continue the delegation down the tree. - Augmentations: initial request can go to any name server; name servers can keep caches; many name servers provide recursive queries and walk the tree for the client.
Design goals (use evidence from text)	Scalability, performance

	Relevant (y/n)	If no, why not? If yes, is goal met?
Simplicity	Either is fine, if well supported	If yes: lots of examples - Met by having zones delegate their own names - Client interface is simple - Single nameserver is simple (doesn't *have* to do caching, recursion) If no: the whole thing is more complex than hosts.txt
Scalability	Yes	Performance-wise: - Caching means that many queries take only a single round-trip to resolve; recursion magnifies this effect - The look-up process is simple, so name servers can handle many queries at once

6.033 systems critique 1: Example Solution
(In the interest of space, we removed specific citations from the text)

		- Thanks to the hierarchical structure, there are many nameservers to handle lookups (contrast to a centralized system). Management-wise: - Each zone makes its own policy decisions about its name bindings, so the root server is not constantly getting slammed with requests for updates
Fault-tolerance/ handling failures	Yes	If a name server fails, zones have backup name servers that will be used. Also, at least for a brief period of time, some of the name server's mappings may be cached at other servers, so some clients might not even notice the change.
Security	Either is fine, if well supported	If Yes: Is relevant now, and is not met by DNS; there is no mechanism to ensure that a name server's response is correct If no: Was not relevant at the time DNS was designed; Internet was a safe place.

Additional design goals, if any (use extra paper if >2)	Is this goal met? Explain
Performance (Sec 4.4.1 #3) (This is the primary one we looked for, since it was explicitly called out in the text)	Much of DNS's performance improvements come from being more scalable, as well as for the three enhancements to DNS (caching, recursion, queries to any nameserver).
Ease-of-management (This is an example of an additional design goal; we did not deduct points if you didn't include it. There are also other additional goals that made sense.)	Yes. Zones only deal with their names (vs., e.g., the root having to handle all updates)

Other analysis (limitations, context, notable success)	Notable success: DNS works across the entire Internet (big success)
Criteria most important to *this system* and why	Did it scale to the size of the Internet? After all, this is the reason DNS replaced hosts.txt.

6.033 systems critique 1: Example Solution
(In the interest of space, we removed specific citations from the text)
