582615 - Overlay and P2P Networks, Spring 2016

Exercise 1 model answers

Considering the following points in the answer granted points in the assignment.

Assignment 1 - Overlays

What does a network overlay do?

A network overlay builds a new logical topology on top of an existing network topology (the underlay). It can be used to overcome limitations of the underlay, or to provide functionality that is otherwise not present in the underlay, and they can simplify designing network applications. In overlay networks, the state is usually at the edges, i.e. clients are running software that produces the overlay, meaning that there is no need to invest into networking hardware that would provide the features.

What kind of systems can use an overlay? What advantage do they get?

- P2P applications, such as certain online games, certain messaging software, and virtual private LAN solutions can use overlays to provide connectivity between hosts that are behind different NAT devices and to provide a simple view of the virtual network despite the actual network being quite complex.
- Privacy applications such as Tor use overlays to provide anonymity and a certain level of confidentiality by automatically disguising the real origin of a network connection, with no support needed from the application making the connection.
- Content delivery networks direct clients to fetch documents from nearby hosting servers instead of the real origin server, to seamlessly provide better performance and availability to web content with no changes needed to the client applications.

Does using an overlay sacrifice something? What are the trade-offs?

- Using an overlay does increase overall complexity of the system. They're a reusable component, but it might be hard to tell if a bug is in the application or in the overlay.
- Application design becomes simpler, but communication will involve more layers, hence more communication, hence more overhead. Added simplicity may also mean giving away some control.
- The provided logical topology may not be optimal for the physical topology. The overlay can make it simple to find peers on the network, but it might also make their communication traverse through a relay point somewhere on the Internet even if the hosts are located on the same physical network.
- Overlays that provide NAT traversal to all applications reduce connectivity problems, but also increase attack surface. The intended applications get exposed to the Internet (or at least other users of the overlay) and are able to communicate freely, but so do unintended applications. Correctly configuring host-level firewalls becomes more important.

Why would one use an overlay instead of simply changing the deployed Internet protocols to do what they need?

- Replacing the established protocols almost certainly breaks compatibility with just about everything on the networks today.
- The software for the established protocols is deployed everywhere, and may not be updated easily due to technical and political reasons.
- A large part of the Internet core is handling the protocols at hardware level, implying need to replace hardware if the protocol is changed. It may be difficult to justify the cost for just about any application.
- Good for one may not be good for other. Changes to the established protocols may benefit some applications, but may also impede other applications.

Assignment 2 - Overlays and the TCP/IP-stack (5 points)

What kind of layers can be found in an overlay network stack and what do they do?

There are more than one view for what constitutes an overlay, typically at least these:

- Application layer / Service layer: Protocols or messages of any software that uses the overlay.
- Routing / Topology layer: Logical topology of the overlay, routing and forwarding messages among nodes, possibly joining and parting the overlay network. Depending on the overlay may provide services such as NAT traversal.
- Network: The underlay, underlying network, this usually is the TCP/IP stack,

Some models treat security, services management etc. as separate layers; describing such a model is fine too.

How would an overlay stack relate to the regular TCP/IP stack? How would you combine them?

Overlays by definition assume the presence of an underlay, typically the TCP/IP stack. In relation to TCP/IP, the overlay stack usually plays the role of an application. From the point of view of the overlay stack, the TCP/IP stack usually acts as the network layer. From the point of view of the application, the overlay typically acts as either the transport or network layer.

A typical stack may look something like this:

Example 1	Example 2
	Application
Application	TCP / UDP / Other
Overlay / Routing	**Overlay / Routing**
TCP / UDP	TCP / UDP
IP	IP
Ethernet / Other	Ethernet / Other

Do the layers of an overlay stack and those of the TCP/IP stack perform similar functions? If yes, what are they, are they redundant, and why?

Many overlays do perform similar functions as the underlay, for example routing and forwarding. However, they are not redundant. The difference is the context. The underlay performs its tasks in the context of the underlying topology. The overlay performs the tasks in the context of the produced logical topology. i.e. The underlay forwards packets between networked hosts, the overlay forwards messages between peers in the overlay. The events in the overlay map into more complex sets of events in the underlay.

A good example is routing and forwarding in the Tor network. The path is an ordered list of nodes. One hop on the Tor path involves many hops through the Internet. On both layers each hop requires a routing decision and sending of the data to the next hop.

Assignment 3 - Peer-to-Peer (P2P) (5 points)

What is the P2P networking model? How does it compare to the traditional Client-Server model?

In the P2P networking model, instead of clients and servers, there are *peers* that form direct network connections between each other. Peers exhibit features of both clients and servers in the sense that they both initiate and receive connections. A P2P network usually also enjoys a certain level of autonomy from a central entity.

In a P2P system, there can be a central entity that manages the P2P system and may have the power to shut it down, such as in Skype, where Microsoft controls the system. The system may also be decentralized, and depend on the ability of peers to discover each other and collaboratively maintain the system. Freenet is an example of such a system: peers maintain neighbor lists and make, receive and forward requests and responses through their neighbors. The initial peers are typically discovered from a static list of *seed nodes* and further peers are found through them.

What typical uses for P2P technologies are there?

Traditional uses for P2P technologies include:

- **Content delivery / file sharing**: Clients who are interested in some content can distribute the content to each other.
- **Online teleconferencing**: Calls can sometimes be routed directly from the caller to the destination.
- **Online gaming**: Some or all of a game's messages may be communicated directly between players. A more typical case is still under content delivery: distributing patches to games.
- **Privacy protection** and **freedom of speech**: Tor, for example, uses a network of peers to hide the true origin of connections. Freedom of speech is closely related to privacy protection. Freenet, for example, allows one to publish content to the network of peers in such a way that it's stored in the caches of those peers. Identifying the author may become very difficult, since there is no "real" hosting infrastructure owned by anybody and the content may be published anonymously.

What challenges are there for P2P systems on today's Internet? Are there solutions?

- **A bad name**: P2P technologies have gained notoriety for being used in file sharing and privacy networks to distribute files in ways that violate law or someone's intellectual property rights, and for being used in things like *bot nets*, where an orchestra of infected computers are used for executing attacks on Internet services. Increased legitimate use of P2P technologies may slowly be clearing this name.
- **Security**: As peers are not under direct control of a central entity, individual peers have more power than clients in traditional client-server setups. Some P2P network designs are more robust than others to tolerate peers that misbehave. Specific deviations from the protocols of the systems may be used to severely impede the functioning of the system, not only compromising performance, but possibly confidentiality as well. Some systems are designed to tolerate certain kinds of misbehavior, but it's difficult to give a generic answer to this.
- **Network Address Translation** (NAT): NAT breaks end-to-end connectivity on the Internet. Peers behind a NAT device may initiate connections and receive responses, but may not be able to receive new incoming connections. They may also not be aware of their real public IP address. There are techniques for NAT traversal, such as *hole punching* and *relaying*, and ways to configure the NAT to map incoming connection rules to the client, for example UPnP / IGD Protocol.
- **Operator throttling**: P2P file sharing especially causes a lot of traffic. An Internet Service Provider (ISP) often needs to pay for traffic that they send toward other ISPs. The amount of traffic, and the fact that most peers are probably not within their network, has encouraged some ISPs to throttle or filter traffic related to certain P2P systems. One solution would be if P2P systems tried to prefer peering with hosts within the same ISPs network over peering with hosts at other ISPs. This solution would be good for file distribution, but it would be a problem for privacy protection systems, where it is desired that no single entity can deduce the path that a message (even encrypted) traveled on the network.

Name three systems that use P2P technologies and briefly explain what they do.

Three examples:

- **Skype**: Skype is an online telephony / teleconferencing application, currently owned by Microsoft. If technically (and administratively) possible, calls on skype go directly from client to client. In those networking scenarios where direct connections are not possible, Microsoft hosts a collection of *Super nodes* that relay the calls.
- **BitTorrent**: BitTorrent is a file sharing system where a torrent file describes some content, and a collection of trackers keeps records on peers interested in such content and tells the peers about each other. The actual dissipation of the content takes place directly between the peers.
- **Coral**: Coral is a content delivery network for the world wide web that internally uses P2P technology for content discovery and distribution. By applying a suffix to the host part of a website URL, the website can be accessed through the Coral CDN. If the website is already in the cache of some Coral node, it is fetched from that node, otherwise it is fetched to the Coral network from the original server at the unsuffixed address.

Assignment 4 - Delivering content (5 points)

What are Content Delivery Networks (CDN)?

Content delivery networks, as the name suggests, are networks that deliver content. Typically a CDN distributes copies of a document to multiple servers around the world or some smaller geographical target and uses indirection to make clients fetch the document from a nearby CDN host instead of the original server.

What are the main benefits of using a CDN?

A CDN brings documents closer to those who wish to access them, meaning shorter paths on the network, and likely better bandwidth, and hence, better performance. It also provides load balancing: since copies of the documents are placed on a number of servers, the servers can divide the burden of serving the document to users. The original server does not need all the bandwidth necessary to serve all the clients. Another reason to use a CDN is availability; the original server can crash without taking the documents down with it, because the CDN can continue serving the document during the original server's downtime. The document served by a CDN can, in some systems, also be a stream, for example, a live video stream. A CDN can internally split the stream to multiple servers on the edge of the CDN, and each of these servers is able to stream it further to multiple clients, vastly increasing the potential number of simultaneous viewers for the stream.

From business perspective, using a CDN generally is profitable. Distributing servers all around the world is expensive and complicated, and there already are established businesses that do this, and provide access to their infrastructure for reasonable financial compensation.

How are typical CDN:s different from typical P2P systems that perform a similar task?

- Usually P2P content distribution / file sharing systems require special software from the hosts that use them; e.g. a BitTorrent client. CDN:n typically work with regular software for fetching documents, such as a regular web browser, and use some method of indirection to point them to the nearest copy of the document.
- Typically in P2P systems, the content is transferred between users of the system. Those who wish to access the content also help delivering it to others. In CDN:n, the documents are usually transferred once from the original server to the CDN, and then many times from the CDN:n servers to the clients, i.e. the clients are not distributing the content to each other.
- It is more common for companies to use CDNs than P2P technology for distributing content. Exceptions do exist, such as Blizzard, who uses a BitTorrent -like system to deliver updates to the World of Warcraft game client. Spotify desktop clients previously used P2P technology to deliver cached songs to other nearby listeners, but they gave it up in early 2014 to simplify their code base [link].

Can a CDN use P2P technology and still serve content to non-P2P clients?

Yes. A CDN can internally distribute the content to its servers through any means it finds suitable, and still offer the content to clients using the traditional CDN methods that do not require anything special from the client.

Turbo challenge - Teredo (5 points)

What does Teredo allow a network connected host to do?

Teredo is a tunneling solution that allows hosts that support IPv6, but only have IPv4 connectivity from the network, to communicate with hosts on the Internet using IPv6.

What are Teredo clients, servers, and relays, and what purpose do they have?

Teredo clients are hosts that use Teredo to gain connectivity to IPv6-enabled hosts on the Internet, and to accept IPv6 connections from the Internet. Teredo client software is included in Microsoft Windows since Vista (limited support in XP SP2), and is available for Linux and BSD (Miredo).

Teredo servers provide NAT traversal support and address assignations to Teredo clients, and relay bubble messages between Teredo clients and Teredo relays. Since the Teredo servers only relay control traffic, their bandwidth requirements are not very demanding per client. Since the Teredo addresses are derived from the addresses of the client and the server, a Teredo server can be designed to be stateless.

Teredo relays are servers which relay packets between the Teredo overlay and the native IPv6 Internet. The Teredo relays advertise routes to the Teredo address block to hosts on the Internet, and the other way around. Teredo relays typically require vast amounts of bandwidth.

What are Teredo bubble messages?

Teredo bubble messages are minimalistic control messages that are used to initiate hole punching for NATs. Teredo relays and clients send bubble messages through Teredo servers to clients, which then perform hole punching toward the address in the bubble message's header.

Explain the steps when a client successfully contacts a host through Teredo and receives a response.

Important information to know beforehand are that a client's teredo address contains the Teredo server's IPv4 address, and the client's IPv4 address and used port. The relays advertise all or part of the IPv6 address space and the Teredo address space.

When a client connects to an IPv6 host through Teredo, it constructs an IPv6 packet with its Teredo IPv6 address as the source address, and the target IPv6 address as the destination address and sends it over the Teredo interface. The Teredo software then encapsulates it into a UDP datagram, determines which Teredo relay to send it to (based on which IPv6 blocks it advertises), and sends it to the determined relay. The relay then decapsulates the IPv6 packet from the UDP datagram and forwards it into the native IPv6 network.

When the packet reaches its destination on the IPv6 network and the destination host responds, it sends an IPv6 packet back to the client's Teredo address. Eventually it is forwarded to a host that advertises routes to the Teredo address block, i.e. a Teredo relay. The Teredo relay encapsulates it into a UDP datagram. If this relay has seen the client recently, it forwards the UDP datagram to it. If not, it sends a bubble message to the Teredo server (determined from the client's Teredo address), which forwards the bubble message to the client. The client then, if necessary, performs NAT hole punching, i.e. connects to the relay mentioned in the header of the bubble packet, and the relay sends the UDP encapsulated IPv6 packet to the client. Teredo software on the client then decapsulates the UDP datagram and injects the IPv6 packet to the received packet queue of the Teredo network interface.