6.033 in the news

Dertouzos Distinguished Lecture: Connectivity, Dr. Robert M. Metcalfe

FEBRUARY 23, 2022
4:00 PM – 5:00 PM

LOCATION
virtual event via Zoom

SPEAKER
Robert Metcalfe

HOST
Daniela Rus
MIT SCC & CSAIL

this lecture could not be better timed with respect to the 6.033 schedule
how do modules of a system communicate if they’re on separate machines?

`Class Browser` (on machine 1)

```python
def main():
    html = browser_load_url(URL)
    ...

def browser_load_url(url):
    msg = url # could reformat
    send request
    wait for reply
    html = reply # could reformat
    return html
```

`Class Server` (on machine 2)

```python
def server_load_url():
    ...
    return html

def handle_server_load_url(url):
    wait for request
    url = request
    html = server_load_url(URL)
    reply = html
    send reply
```

client

server

network
how do modules of a system communicate if they’re on separate machines?

client

server
how do modules of a system communicate if they’re on separate machines?

client → server
how do modules of a system communicate if they’re on separate machines?

point-to-point links: get a source to talk to a directly-connected destination
how do modules of a system communicate if they’re on separate machines?
how do modules of a system communicate if they’re on separate machines?

**switches**: help forward data to destinations that are far away

switches do other things, too
how do modules of a system communicate if they’re on separate machines?
how do modules of a system communicate if they’re on separate machines?

as this system grows, we need to think about how to turn this set of links into a network
how do modules of a system communicate if they’re on separate machines?

as this system grows, we need to think about how to turn this set of links into a network

communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)
how do modules of a system communicate if they’re on separate machines?

As this system grows, we need to think about how to turn this set of links into a network.

Link

Communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)
how do modules of a system communicate if they’re on separate machines?

As this system grows, we need to think about how to turn this set of links into a network.

Network naming, addressing

Link communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)
how do modules of a system communicate if they’re on separate machines?

As this system grows, we need to think about how to turn this set of links into a network.

Network naming, addressing, routing

Link communication between two directly-connected nodes

Examples: ethernet, bluetooth, 802.11 (wifi)
how do modules of a system communicate if they’re on separate machines?

As this system grows, we need to think about how to turn this set of links into a network.

Link
- Communication between two directly-connected nodes
  - Examples: Ethernet, Bluetooth, 802.11 (WiFi)

Network
- Naming, addressing, routing

Transport
- Sharing the network, reliability (or not)
how do modules of a system communicate if they’re on separate machines?

as this system grows, we need to think about how to turn this set of links into a network

application
the things that actually generate traffic

transport
sharing the network, reliability (or not)

network
naming, addressing, routing

link
communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)
application  the things that actually generate traffic

transport  sharing the network, reliability (or not)

network  naming, addressing, routing

link  communication between two directly-connected nodes

examples: ethernet, bluetooth, 802.11 (wifi)
1970s:
ARPAnet

- **application**: the things that actually generate traffic
- **transport**: sharing the network, reliability (or not)
- **network**: naming, addressing, routing
- **link**: communication between two directly-connected nodes

*examples: ethernet, bluetooth, 802.11 (wifi)*
1970s: ARPAnet

**Figure 6.1** Drawing of September 1969 (Courtesy of Alex McKenzie)

- **Application**: the things that actually generate traffic
- **Transport**: sharing the network, reliability (or not)
- **Network**: naming, addressing, routing
- **Link**: communication between two directly-connected nodes
  
e.g., ethernet, bluetooth, 802.11 (wifi)

[Link to historical context](https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html)
1970s: ARPAnet

**Application**
The things that actually generate traffic

**Transport**
Sharing the network, reliability (or not)

**Network**
Naming, addressing, routing

**Link**
Communication between two directly-connected nodes

Examples: Ethernet, Bluetooth, 802.11 (WiFi)

---

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html

Katrina LaCurts | lacurts@mit.edu | 6.033 2022
1970s:
ARPAnet

hosts.txt

 application the things that actually generate traffic

 transport sharing the network, reliability (or not)

 network naming, addressing, routing

 link communication between two directly-connected nodes
 examples: ethernet, bluetooth, 802.11 (wifi)

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
1970s:
ARPAnet

hosts.txt

ARPANET LOGICAL MAP, MARCH 1977

application
the things that actually generate traffic

transport
sharing the network, reliability (or not)

network
naming, addressing, routing

link
communication between two directly-connected nodes

elements: ethernet, bluetooth, 802.11 (wifi)

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
1970s: ARPAnet

- hosts.txt
- distance-vector
- routing

**ARPANET LOGICAL MAP, MARCH 1977**

**application**
the things that actually generate traffic

**transport**
sharing the network, reliability (or not)

**network**
naming, addressing, routing

**link**
communication between two directly-connected nodes

examples: ethernet, bluetooth, 802.11 (wifi)

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
1970s: ARPAnet

hosts.txt distance-vector routing

**application** the things that actually generate traffic

**transport** sharing the network, reliability (or not)

**network** naming, addressing, routing

**link** communication between two directly-connected nodes

*examples: ethernet, bluetooth, 802.11 (wifi)*

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
1970s: ARPAnet

ARPANET GEOGRAPHIC MAP, JUNE 1977

hosts.txt  distance-vector routing

1978: flexibility and layering

application  the things that actually generate traffic

transport  sharing the network, reliability (or not)

network  naming, addressing, routing

link  communication between two directly-connected nodes

examples: ethernet, bluetooth, 802.11 (wifi)

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
1970s: ARPAnet
1978: flexibility and layering

hosts.txt distance-vector routing

with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html

![ARPANET Geographic Map, June 1977](https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html)

**application** the things that actually generate traffic

**transport** sharing the network, reliability (or not)

**network** naming, addressing, routing

**link** communication between two directly-connected nodes

examples: ethernet, bluetooth, 802.11 (wifi)
1970s: ARPAnet hosts.txt distance-vector routing TCP, UDP

1978: flexibility and layering

with a layered model, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers

application the things that actually generate traffic

transport sharing the network, reliability (or not)
examples: TCP, UDP

network naming, addressing, routing
examples: IP

link communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1970s:</td>
<td>ARPAnet</td>
</tr>
<tr>
<td>1978:</td>
<td>flexibility and layering</td>
</tr>
<tr>
<td>early 80s:</td>
<td>growth → change</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>the things that actually generate traffic</td>
</tr>
<tr>
<td>transport</td>
<td>sharing the network, reliability (or not)</td>
</tr>
<tr>
<td>network</td>
<td>naming, addressing, routing</td>
</tr>
<tr>
<td>link</td>
<td>communication between two directly-connected nodes</td>
</tr>
</tbody>
</table>

with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers.

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html
with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers

---

**application**

the things that actually generate traffic

**transport**

sharing the network, reliability (or not)

examples: TCP, UDP

**network**

naming, addressing, routing

examples: IP

**link**

communication between two directly-connected nodes

examples: ethernet, bluetooth, 802.11 (wifi)
with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers
1970s: ARPAnet  
1978: flexibility and layering  
early 80s: growth → change  
late 80s: growth → problems

| hosts.txt | distance-vector routing | TCP, UDP | OSPF, EGP, DNS |

application

the things that actually generate traffic

transport

sharing the network, reliability (or not)
examples: TCP, UDP

network

naming, addressing, routing
examples: IP

link

communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)

https://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography/atlas/historical.html

with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers
1970s: ARPAnet  
1978: flexibility and layering  
early 80s: growth → change  
late 80s: growth → problems

hosts.txt  
distance-vector routing  
TCP, UDP  
OSPF, EGP, DNS

application  
the things that actually generate traffic

transport  
sharing the network, reliability (or not)  
examples: TCP, UDP

network  
naming, addressing, routing  
examples: IP

link  
communication between two directly-connected nodes  
examples: ethernet, bluetooth, 802.11 (wifi)

with a layered model, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers

https://www.vox.com/a/internet-maps
1970s: ARPAnet

1978: flexibility and layering

early 80s: growth \(\rightarrow\) change

late 80s: growth \(\rightarrow\) problems

| hosts.txt | distance-vector routing | TCP, UDP | OSPF, EGP, DNS | congestion collapse |

1970s: ARPAnet

early 80s: growth \(\rightarrow\) change

late 80s: growth \(\rightarrow\) problems

application

the things that actually generate traffic

transport

sharing the network, reliability (or not)

examples: TCP, UDP

network

naming, addressing, routing

examples: IP

link

communication between two directly-connected nodes

examples: ethernet, bluetooth, 802.11 (wifi)

https://www.vox.com/a/internet-maps

with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers
1970s: ARPAnet
1978: flexibility and layering
early 80s: growth → change
late 80s: growth → problems

hosts.txt  distance-vector routing  TCP, UDP  OSPF, EGP, DNS  congestion collapse  policy routing

application  the things that actually generate traffic
transport  sharing the network, reliability (or not)
examples: TCP, UDP
network  naming, addressing, routing
examples: IP
link  communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)

with a layered model, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers

https://www.vox.com/a/internet-maps
1970s: ARPAnet | 1978: flexibility and layering | early 80s: growth → change | late 80s: growth → problems

- hosts.txt
- distance-vector routing
- TCP, UDP
- OSPF, EGP, DNS
- congestion collapse
- policy routing
- CIDR

**application**
the things that actually generate traffic

**transport**
sharing the network, reliability (or not)
examples: TCP, UDP

**network**
naming, addressing, routing
examples: IP

**link**
communication between two directly-connected nodes
examples: ethernet, bluetooth, 802.11 (wifi)

with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers
with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers.
<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1970s: ARPAnet</td>
<td>1978: flexibility and layering</td>
<td>TCP, UDP OSPF, EGP, DNS congestion collapse policy routing CIDR</td>
</tr>
</tbody>
</table>
| 1970s: ARPAnet | 1993: commercialization |the things that actually generate traffic  
transport sharing the network, reliability (or not) examples: TCP, UDP  
network naming, addressing, routing examples: IP  
link communication between two directly-connected nodes examples: ethernet, bluetooth, 802.11 (wifi)  

with a **layered model**, we can swap out protocols at one layer without much (or perhaps any) change to protocols at other layers

1970s: ARPAnet  
1978: flexibility and layering  
early 80s: growth → change  
late 80s: growth → problems  
1993: commercialization

<table>
<thead>
<tr>
<th>Hosts.txt</th>
<th>Distance-vector routing</th>
<th>TCP, UDP</th>
<th>OSPF, EGP, DNS</th>
<th>Congestion collapse</th>
<th>Policy routing</th>
<th>CIDR</th>
</tr>
</thead>
</table>

- **application**: the things that actually generate traffic
- **transport**: sharing the network, reliability (or not)
  - examples: TCP, UDP
- **network**: naming, addressing, routing
  - examples: IP
- **link**: communication between two directly-connected nodes
  - examples: ethernet, bluetooth, 802.11 (wifi)

---

On the Internet, we have to solve all of the “normal” networking problems (addressing, routing, transport) at massive scale, while supporting a diverse group of applications and competing economic interests.