Rebuild from Scratch: Internet 2.0

Things I like to do during my sabbatical

Bernd Paysan

EuroForth 2009, Exeter

Outline

Get Rid of Accumulated Cruft

- The Intenet is finally starting to show its age — a lot of things don't work as well as they should
- At the same time, dependency increases
- IPv6 solves some of the problems, but creates others, and ignores many
- Too complex, not following its own specifications (RFCs) in parts, too many protocols
- Postel problem: If you are generous in what you accept, produced rubbish will increase over time
- Solution: Throw it away and rebuild from scratch

Get Rid of Accumulated Cruft

- The Intenet is finally starting to show its age — a lot of things don't work as well as they should
- At the same time, dependency increases
- IPv6 solves some of the problems, but creates others, and ignores many
- Too complex, not following its own specifications (RFCs) in parts, too many protocols
- Postel problem: If you are generous in what you accept, produced rubbish will increase over time
- Solution: Throw it away and rebuild from scratch

Get Rid of Accumulated Cruft

- The Intenet is finally starting to show its age — a lot of things don't work as well as they should
- At the same time, dependency increases
- IPv6 solves some of the problems, but creates others, and ignores many
- Too complex, not following its own specifications (RFCs) in parts, too many protocols
- Postel problem: If you are generous in what you accept, produced rubbish will increase over time
- Solution: Throw it away and rebuild from scratch

Get Rid of Accumulated Cruft

- The Intenet is finally starting to show its age — a lot of things don't work as well as they should
- At the same time, dependency increases
- IPv6 solves some of the problems, but creates others, and ignores many
- Too complex, not following its own specifications (RFCs) in parts, too many protocols
- Postel problem: If you are generous in what you accept, produced rubbish will increase over time
- Solution: Throw it away and rebuild from scratch

Get Rid of Accumulated Cruft

- The Intenet is finally starting to show its age — a lot of things don't work as well as they should
- At the same time, dependency increases
- IPv6 solves some of the problems, but creates others, and ignores many
- Too complex, not following its own specifications (RFCs) in parts, too many protocols
- Postel problem: If you are generous in what you accept, produced rubbish will increase over time
- Solution: Throw it away and rebuild from scratch

Get Rid of Accumulated Cruft

- The Intenet is finally starting to show its age — a lot of things don't work as well as they should
- At the same time, dependency increases
- IPv6 solves some of the problems, but creates others, and ignores many
- Too complex, not following its own specifications (RFCs) in parts, too many protocols
- Postel problem: If you are generous in what you accept, produced rubbish will increase over time
- Solution: Throw it away and rebuild from scratch

Requirements

Scalability Must work well with low and high bandwidths, loose and tightly coupled systems, few and many hosts connected together over short to far distances.

Easy to implement Must work with a minimum of effort, must allow small and cheap devices to connect. One idea is to replace "busses" like USB and firewire with cheap LAN links.

Security Users want authentication and authorization, but also anonymity and privacy. Firewalls and similar gatekeepers (load balancers, etc.) are common.

Media capable This requires real-time capabilities, pre-allocated bandwidth and other QoS features, end-to-end.

Transparency Must be able to work together with other networks (especially Internet 1.0).

Requirements

Scalability Must work well with low and high bandwidths, loose and tightly coupled systems, few and many hosts connected together over short to far distances.

Easy to implement Must work with a minimum of effort, must allow small and cheap devices to connect. One idea is to replace "busses" like USB and firewire with cheap LAN links.

Security Users want authentication and authorization, but also anonymity and privacy. Firewalls and similar gatekeepers (load balancers, etc.) are common.

Media capable This requires real-time capabilities, pre-allocated bandwidth and other QoS features, end-to-end.

Transparency Must be able to work together with other networks (especially Internet 1.0).

Requirements

Scalability Must work well with low and high bandwidths, loose and tightly coupled systems, few and many hosts connected together over short to far distances.

Easy to implement Must work with a minimum of effort, must allow small and cheap devices to connect. One idea is to replace "busses" like USB and firewire with cheap LAN links.

Security Users want authentication and authorization, but also anonymity and privacy. Firewalls and similar gatekeepers (load balancers, etc.) are common.

Media capable This requires real-time capabilities, pre-allocated bandwidth and other QoS features, end-to-end.

Transparency Must be able to work together with other networks (especially Internet 1.0).

Requirements

Scalability Must work well with low and high bandwidths, loose and tightly coupled systems, few and many hosts connected together over short to far distances.

Easy to implement Must work with a minimum of effort, must allow small and cheap devices to connect. One idea is to replace "busses" like USB and firewire with cheap LAN links.

Security Users want authentication and authorization, but also anonymity and privacy. Firewalls and similar gatekeepers (load balancers, etc.) are common.

Media capable This requires real-time capabilities, pre-allocated bandwidth and other QoS features, end-to-end.

Transparency Must be able to work together with other networks (especially Internet 1.0).

Requirements

Scalability Must work well with low and high bandwidths, loose and tightly coupled systems, few and many hosts connected together over short to far distances.

Easy to implement Must work with a minimum of effort, must allow small and cheap devices to connect. One idea is to replace "busses" like USB and firewire with cheap LAN links.

Security Users want authentication and authorization, but also anonymity and privacy. Firewalls and similar gatekeepers (load balancers, etc.) are common.

Media capable This requires real-time capabilities, pre-allocated bandwidth and other QoS features, end-to-end.

Transparency Must be able to work together with other networks (especially Internet 1.0).

Switching, not Routing

- The Internet is connected by routers, which route every packet
- Most Internet traffic is connection oriented; even DNS usually goes to a local cache server
- IPv6 increases the routing tables even more

Physical Route

- Take first n bits of target address and select destination
- Shift target address by n
- Insert bit-reversed source into address field

Switching, not Routing

- The Internet is connected by routers, which route every packet
- Most Internet traffic is connection oriented; even DNS usually goes to a local cache server
- IPv6 increases the routing tables even more

Physical Route

- Take first n bits of target address and select destination
- Shift target address by n
- Insert bit-reversed source into address field

Switching, not Routing

- The Internet is connected by routers, which route every packet
- Most Internet traffic is connection oriented; even DNS usually goes to a local cache server
- IPv6 increases the routing tables even more

Physical Route

- Take first n bits of target address and select destination
- Shift target address by n
- Insert bit-reversed source into address field

Switching, not Routing

- The Internet is connected by routers, which route every packet
- Most Internet traffic is connection oriented; even DNS usually goes to a local cache server
- IPv6 increases the routing tables even more

Physical Route

- Take first n bits of target address and select destination
- Shift target address by n
- Insert bit-reversed source into address field

Switching, not Routing

- The Internet is connected by routers, which route every packet
- Most Internet traffic is connection oriented; even DNS usually goes to a local cache server
- IPv6 increases the routing tables even more

Physical Route

- Take first n bits of target address and select destination
- Shift target address by n
- Insert bit-reversed source into address field

Switching, not Routing

- The Internet is connected by routers, which route every packet
- Most Internet traffic is connection oriented; even DNS usually goes to a local cache server
- IPv6 increases the routing tables even more

Physical Route

- Take first n bits of target address and select destination
- Shift target address by n
- Insert bit-reversed source into address field

Routing Sever

- Routing server resolves name $\longrightarrow$ destination. Routing service is a cache, distributed data base like DNS.
- Routing happens once per host and destination, and is cached
- Net interconnects and end nodes separated — no problem of abusing end nodes
- Return path not spoofable, and shared with destination

Routing Sever

- Routing server resolves name $\longrightarrow$ destination. Routing service is a cache, distributed data base like DNS.
- Routing happens once per host and destination, and is cached
- Net interconnects and end nodes separated — no problem of abusing end nodes
- Return path not spoofable, and shared with destination

Routing Sever

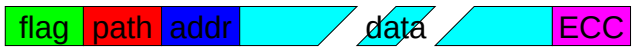
- Routing server resolves name $\rightarrow$ destination. Routing service is a cache, distributed data base like DNS.
- Routing happens once per host and destination, and is cached
- Net interconnects and end nodes separated — no problem of abusing end nodes
- Return path not spoofable, and shared with destination

Routing Sever

- Routing server resolves name $\rightarrow$ destination. Routing service is a cache, distributed data base like DNS.
- Routing happens once per host and destination, and is cached
- Net interconnects and end nodes separated — no problem of abusing end nodes
- Return path not spoofable, and shared with destination

Packet Header

	Size
Flags	*2*
Path	*2/8*
Address	*2/8*
Junk	*0/8*
Data	*32/128/512/2k*
ECC	*L1 dependent*



Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

Flow Control

TCP/IP

- End node flow control
- Slow start (scalabiliy problem)
- Randomly dropped packets in case of overload
- Retransmits and fake or arbitrary delayed acknowledges cause problems
- Remember: End node is possibly hostile!

Network flow control

- Keep statistics
- Send "jam" messages back to worst offenders
- Notify early about available resources
- Topological fairness, not per–connection fairness

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

1:n Connections

Multicasting

- Route is a tree, not a path
- Use table to select multiple destinations
- Hosts joining a multicast search upstream switch and add address to table
- Multicasting is a scarce resource — use when apropriate
- No regulation needed

Broadcasting

- Use bitmap to select destinations
- Small end user routers may use CAM[a]
- Broadcasts use global/region numbers: Regulation needed

[a] Content Addressed Memory

Legacy

Internet 2.0 over Ethernet Put packets into ordinary Ethernet frames — needs jumbo frames for 2k packets

Internet 2.0 over IP Put packets into UDP datagrams, use VoIP/P2P techniques to route through NAT

IP over Internet 2.0 Similar to MPLS:[1] Build tunnels per destination group

Access to Content Protocol translation (all kind of different implementations possible)

[1] Multi Layer Protocol Switching

Legacy

Internet 2.0 over Ethernet Put packets into ordinary Ethernet frames — needs jumbo frames for 2k packets

Internet 2.0 over IP Put packets into UDP datagrams, use VoIP/P2P techniques to route through NAT

IP over Internet 2.0 Similar to MPLS:[1] Build tunnels per destination group

Access to Content Protocol translation (all kind of different implementations possible)

[1] Multi Layer Protocol Switching

Legacy

Internet 2.0 over Ethernet Put packets into ordinary Ethernet frames — needs jumbo frames for 2k packets

Internet 2.0 over IP Put packets into UDP datagrams, use VoIP/P2P techniques to route through NAT

IP over Internet 2.0 Similar to MPLS:[1] Build tunnels per destination group

Access to Content Protocol translation (all kind of different implementations possible)

[1] Multi Layer Protocol Switching

Legacy

Internet 2.0 over Ethernet Put packets into ordinary Ethernet frames — needs jumbo frames for 2k packets

Internet 2.0 over IP Put packets into UDP datagrams, use VoIP/P2P techniques to route through NAT

IP over Internet 2.0 Similar to MPLS:[1] Build tunnels per destination group

Access to Content Protocol translation (all kind of different implementations possible)

[1] Multi Layer Protocol Switching

Abstraction

Avoid unnecessary abstraction

Distributed Shared Memory Packet transfer data blocks from one node to another, on this level, it's just addresses and data

Active Messages Separate data and "metadata." Metadata is really code — coded in a (sandboxed) stack–VM that suits the abstraction model, and allows expanding it. Protocol hierarchy build on a common foundation of commands

Files with Attributes Most Internet protocols deal with files of sorts. Files often have attributes (sender/receiver and subjects on E-Mails, data type, date, name, possibly references in Hypertext files). Add properties to files, and allow querying for those.

Abstraction

Avoid unnecessary abstraction

Distributed Shared Memory Packet transfer data blocks from one node to another, on this level, it's just addresses and data

Active Messages Separate data and "metadata." Metadata is really code — coded in a (sandboxed) stack–VM that suits the abstraction model, and allows expanding it. Protocol hierarchy build on a common foundation of commands

Files with Attributes Most Internet protocols deal with files of sorts. Files often have attributes (sender/receiver and subjects on E-Mails, data type, date, name, possibly references in Hypertext files). Add properties to files, and allow querying for those.

Abstraction

Avoid unnecessary abstraction

Distributed Shared Memory Packet transfer data blocks from one node to another, on this level, it's just addresses and data

Active Messages Separate data and "metadata." Metadata is really code — coded in a (sandboxed) stack–VM that suits the abstraction model, and allows expanding it. Protocol hierarchy build on a common foundation of commands

Files with Attributes Most Internet protocols deal with files of sorts. Files often have attributes (sender/receiver and subjects on E-Mails, data type, date, name, possibly references in Hypertext files). Add properties to files, and allow querying for those.

Abstraction II

Caching, P2P, Clouds Distributed file system: Use a cryptographic hash ("URI") to identify documents and look them up where you can get them cheapest — neighbour computer, cloud cohosting, P2P network.

Text Formatting Wikis show: HTML too complicated. Use wiki-style formatting plus separated style sheets for layout.

Single Frontend With consolidated protocols, the browser should do everything neatly. Add AJAX–like capabilities directly to the data model the browser understands.

Abstraction II

Caching, P2P, Clouds Distributed file system: Use a cryptographic hash ("URI") to identify documents and look them up where you can get them cheapest — neighbour computer, cloud cohosting, P2P network.

Text Formatting Wikis show: HTML too complicated. Use wiki-style formatting plus separated style sheets for layout.

Single Frontend With consolidated protocols, the browser should do everything neatly. Add AJAX–like capabilities directly to the data model the browser understands.

Abstraction II

Caching, P2P, Clouds Distributed file system: Use a cryptographic hash ("URI") to identify documents and look them up where you can get them cheapest — neighbour computer, cloud cohosting, P2P network.

Text Formatting Wikis show: HTML too complicated. Use wiki-style formatting plus separated style sheets for layout.

Single Frontend With consolidated protocols, the browser should do everything neatly. Add AJAX–like capabilities directly to the data model the browser understands.

Examples

Open connection

```
s'' http'' protocol s'' foo.com'' host
<addr> <len> data-window
<addr'> <len'> command-window
open-port
```

Get hypertext document

```
:? expand-preload  BEGIN  dup list-len >r
  dup s'' preload'' get-attribute-list append
  uniquify-list  dup list-len r> =  UNTIL ;
[[ s'' bar/url.wiki'' ]] expand-preload
' send-mime-file map-list
```

Examples

Open connection

```
s'' http'' protocol s'' foo.com'' host
<addr> <len> data-window
<addr'> <len'> command-window
open-port
```

Get hypertext document

```
:? expand-preload  BEGIN  dup list-len >r
  dup s'' preload'' get-attribute-list append
  uniquify-list  dup list-len r> =  UNTIL ;
[[ s'' bar/url.wiki'' ]] expand-preload
' send-mime-file map-list
```

Security Requirements

- Encryption to avoid eavesdropping
- Authentication and Authorization for access control
- Key exchange and trust network (PKI)
- Anonymity if necessary (onion routing)

Security Requirements

- Encryption to avoid eavesdropping
- Authentication and Authorization for access control
- Key exchange and trust network (PKI)
- Anonymity if necessary (onion routing)

Security Requirements

- Encryption to avoid eavesdropping
- Authentication and Authorization for access control
- Key exchange and trust network (PKI)
- Anonymity if necessary (onion routing)

Security Requirements

- Encryption to avoid eavesdropping
- Authentication and Authorization for access control
- Key exchange and trust network (PKI)
- Anonymity if necessary (onion routing)

Summary

- This is pure vaporware now
- Path to reality: Reference implementation, RFC, IETF discussion

Summary

- This is pure vaporware now
- Path to reality: Reference implementation, RFC, IETF discussion

For Further Reading I

Bernd Paysan

Internet 2.0

http://www.jwdt.com/~paysan/internet-2.0.html