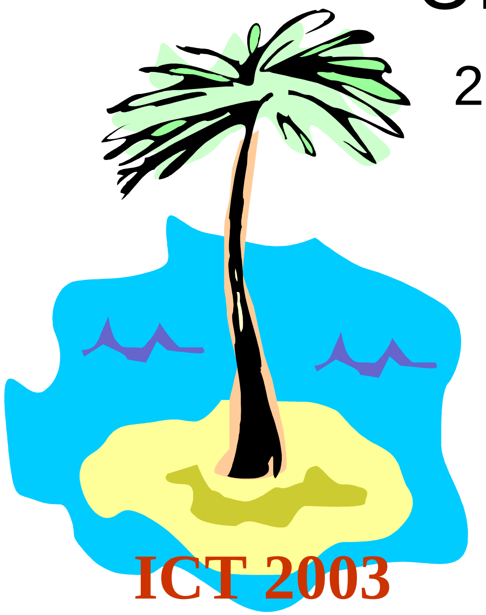
State-of-the-art in group communications: from protocols to applications

C. Pham[1] & V. Roca[2]

[1]Univ. Lyon 1, RESO/LIP, France

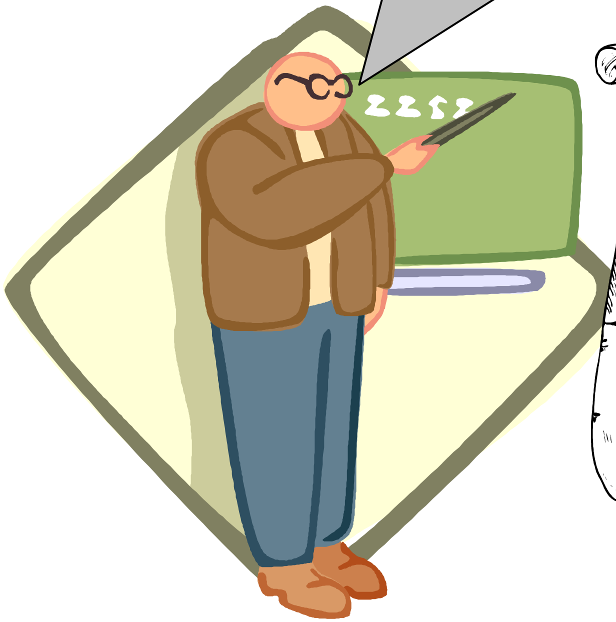
[2]INRIA R-A, Planète, France



Sunday, 23rd february, Papeete

Getting started!

Multicast has been around for more than a decade, and we've proposed many protocols!



SRM, DVMRP
CBT, RMTP,
LMS, MOSPF,
MBGP, PIM-DM,
MSDP, IGMP,
RPM, HBH,
LBRM,
DyRAM…

Yes, but very few real applications have been deployed on the Internet!

Q&A

- Q1: How many people in the audience have heard about multicast?
- Q2: How many people in the audience know basically what multicast is?
- Q3: How many people in the audience have ever tried multicast technologies?
- Q4: How many people think they need multicast?

My guess on the answers

- Q1: How many people in the audience have previously heard about multicast?
  - about 80%
- Q2: How many people in the audience know very basically what multicast is?
  - about 80%
- Q3: How many people in the audience have ever tried multicast technologies?
  - 0% !
- Q4: How many people think they need multicast?
  - almost nobody!

good guess | wrong guess

Never be pessimistic! Things are better than I thought!

- You are curious, innovative and open-minded!
- You want to be up-to-date in an ever-evolving world of high technology.
- You are not afraid of changing how things are and are always optimistic!
- This tutorial will help you go further and will comfort you in your ideas!
- Let me continue anyway!

next →

Well, I'm afraid I was right…

- Multicast has too little penetration in the Internet user community
- The research community failed in promoting the multicast technologies
- But there is now an opportunity to change all this…

Purpose of this tutorial

- Provide a comprehensive overview of current multicast technologies and deployment status
- Show what are the problems and how they can be solved
- Achieve 100%, 100%, 30% and 50% to the previous answers next time!

This tutorial will...

- explain how multicast can change the way people use the Internet
- present the main technologies behind multicast with a focus on reliable and streaming multicast solutions
- state on the current deployment of multicast technologies and the problems encountered for large scale deployment

How multicast can change the way people use the Internet?

MULTICAST!

multicast!

multicast!

Everybody's talking about multicast! Really annoying ! Why would I need multicast for by the way?

multicast!

multicas

multicast!

multicast!

mult

multicast!

multicast!

MULTICAST!

multicast!

multicast!

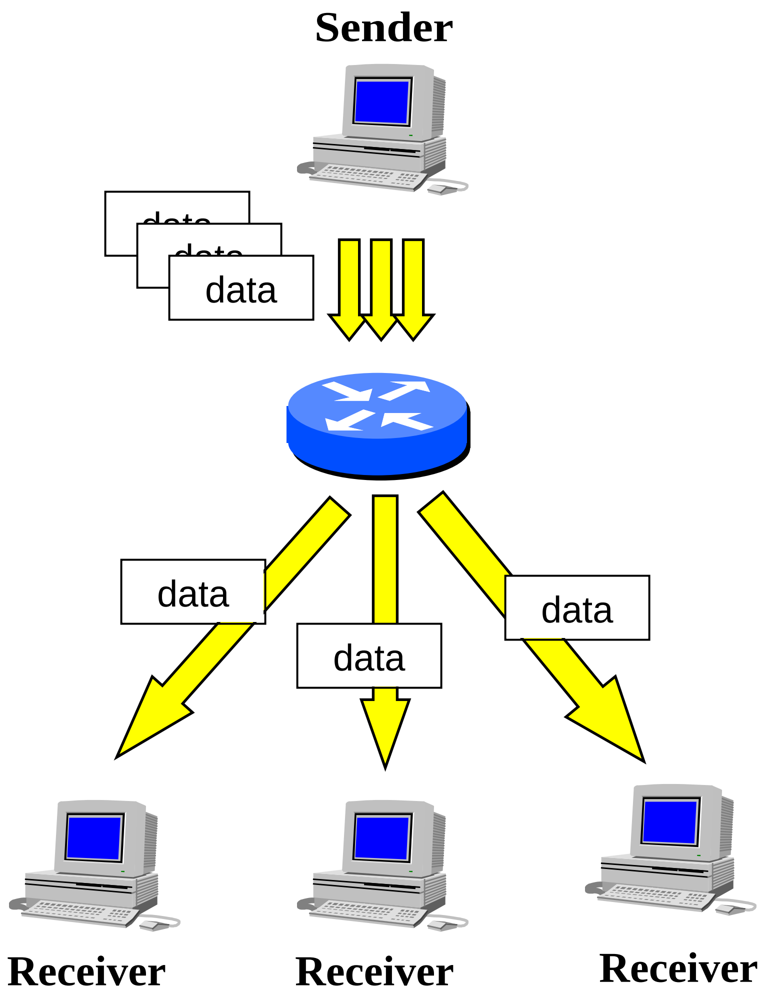
multicast!

multicast!

multicast!

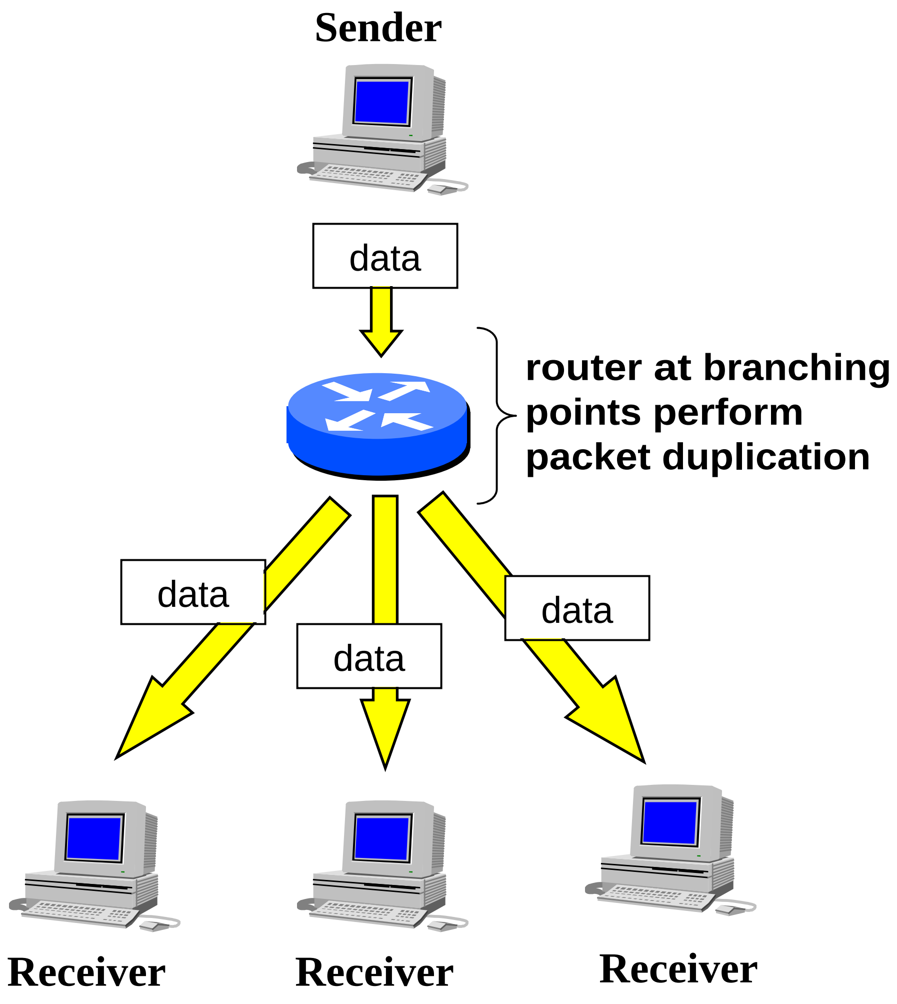
From unicast …

- Sending same data to many receivers via unicast is inefficient
- Popular WWW sites become serious bottlenecks



..to multicast on the Internet.

- Not n-unicast from the sender perspective
- Efficient one to many data distribution
- Towards low latence, high bandwidth



New applications for the Internet

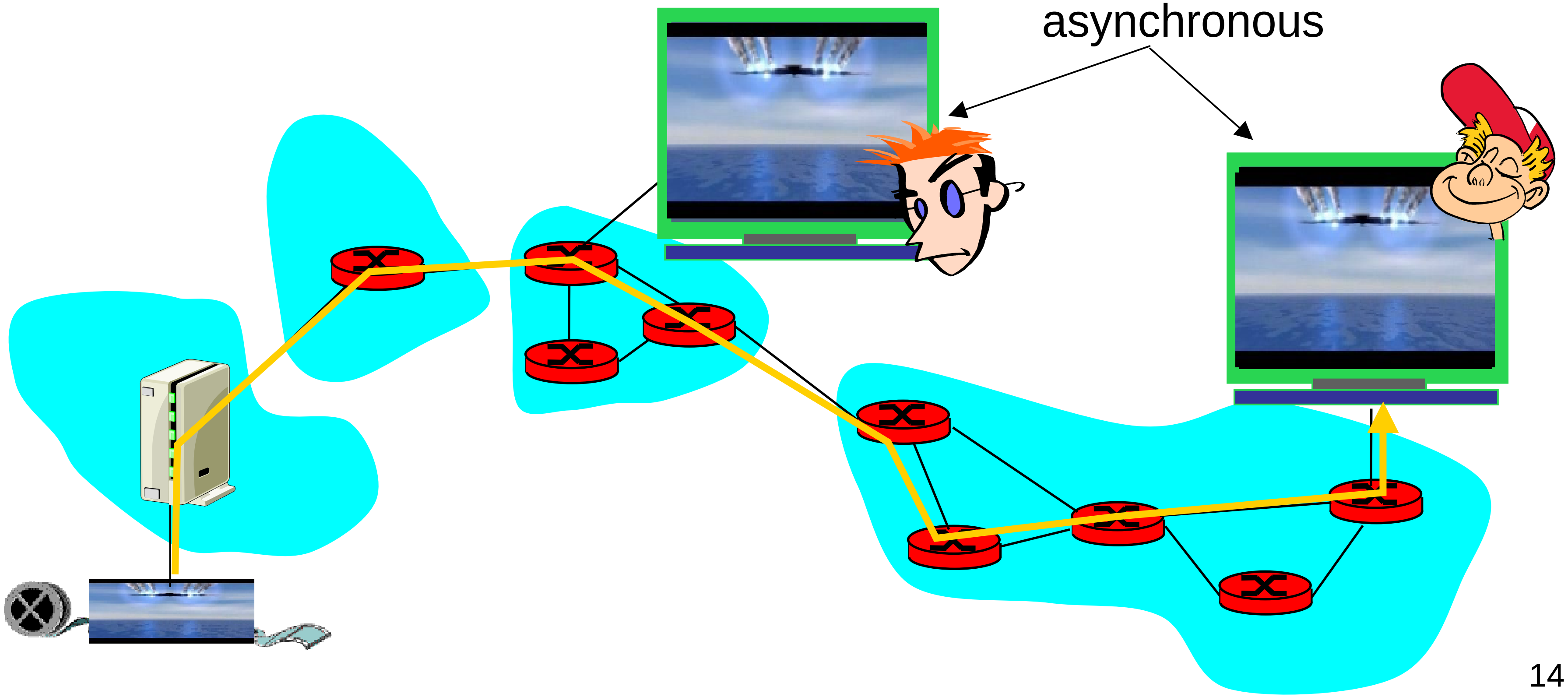
Think about…

- high-speed www
- video-conferencing
- video-on-demand
- interactive TV programs
- remote archival systems
- tele-medecine, white board
- high-performance computing, gri
- virtual reality, immersion systems
- distributed interactive simulations/ gaming…

A whole new world for multicast…

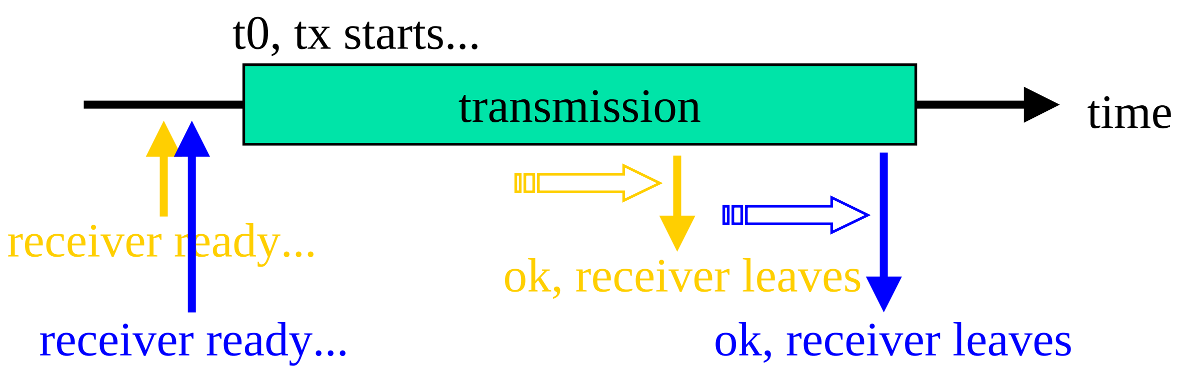
The delivery models (1)

- model 1: streaming (e.g. for audio/video)
  - multimedia data requires efficiency due to its size
  - requires real-time, semi-reliable delivery



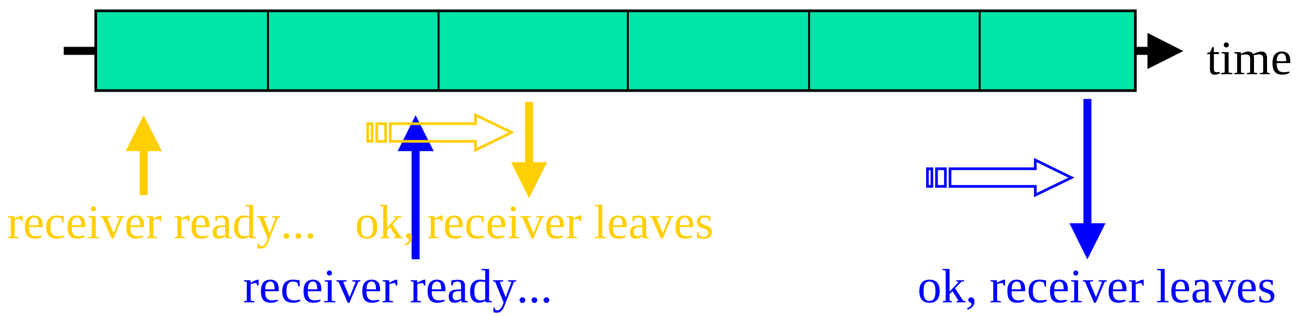
The delivery models (2)

- model 2: push delivery
  - synchronous model where delivery is started at t0
  - usually requires a fully reliable delivery, limited number of receivers
  - Ex: synchronous updates of software



The delivery models (3)

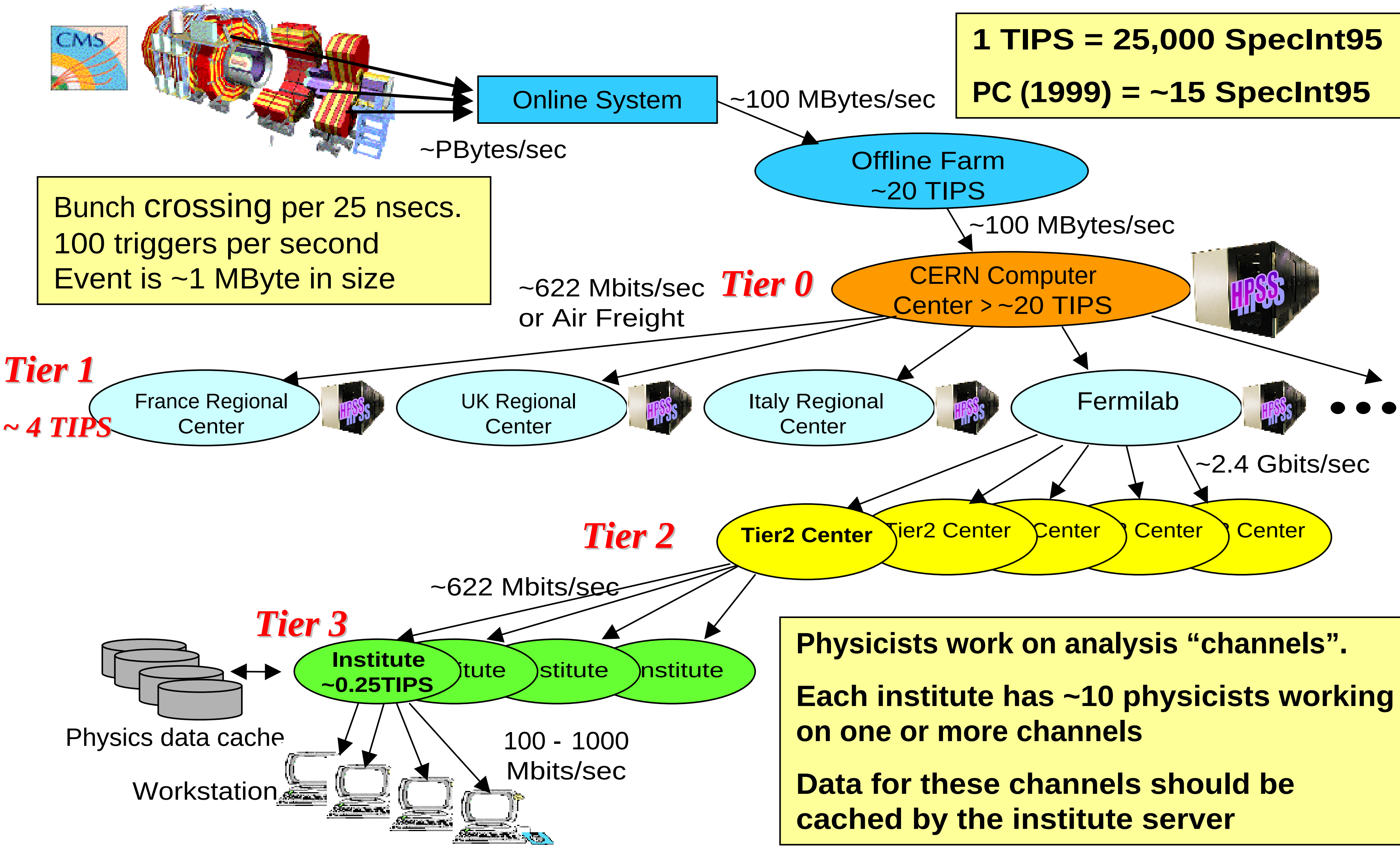
- model 3: on-demand delivery
  - popular content (video clip, software,update, etc.) is continuously distributed in multicast
  - users arrive at any time, download, and leave
  - possibility of millions of users, no real-time constraint



A very simple example in figures

- File replication (PUSH) with ftp
  - 10MBytes file
  - 1 source, n receivers (replication sites)
  - 512KBits/s upstream access
  - n=100
    - T_x= 4.55 hours
  - n=1000
    - T_x= 1 day 21 hours 30 mins!

A real example: LHC (DataGrid)



1 TIPS = 25,000 SpecInt95

PC (1999) = ~15 SpecInt95

Bunch crossing per 25 nsecs.
100 triggers per second
Event is ~1 MByte in size

Physicists work on analysis "channels".

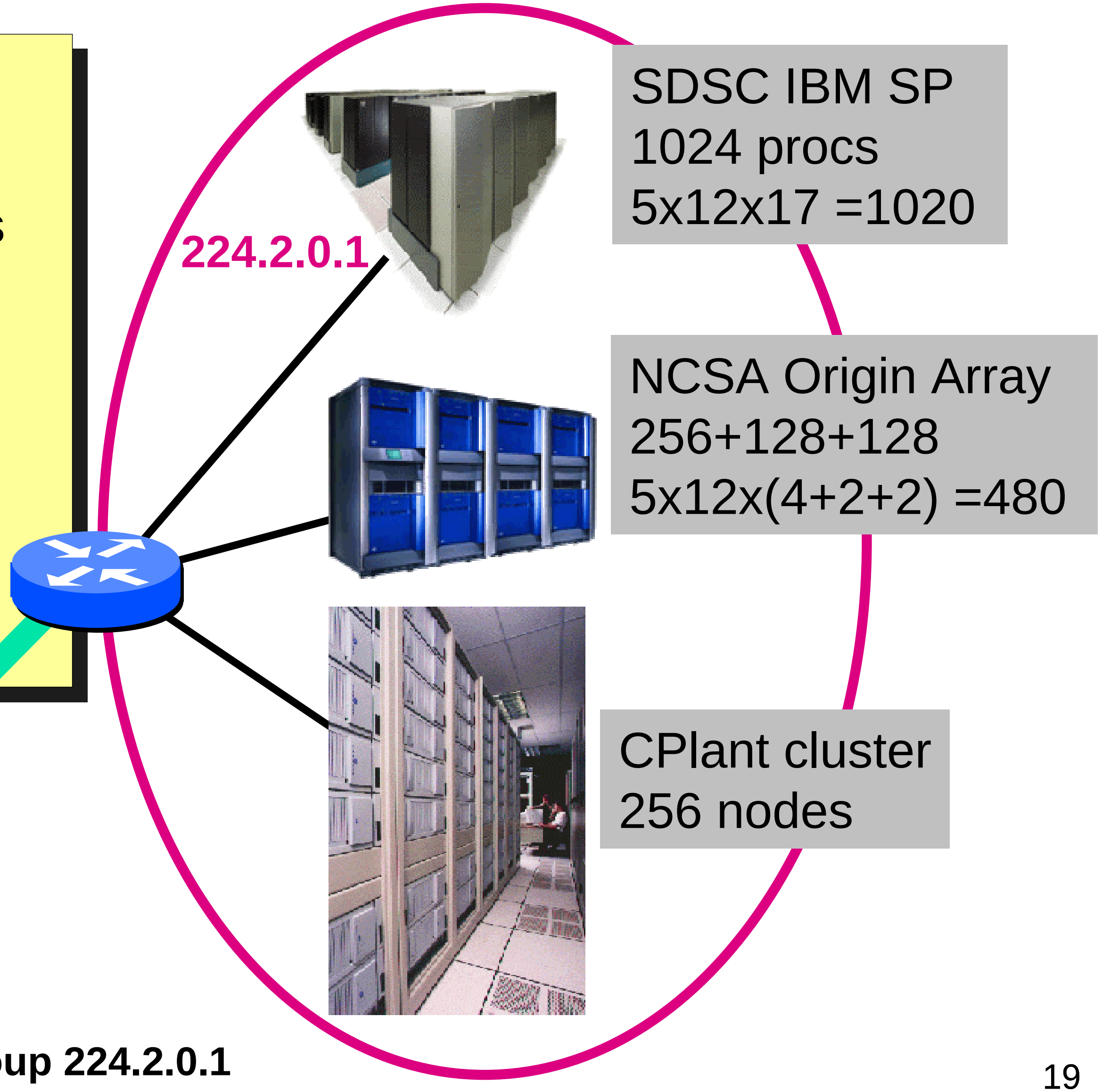
Each institute has ~10 physicists working on one or more channels

Data for these channels should be cached by the institute server

source DataGrid

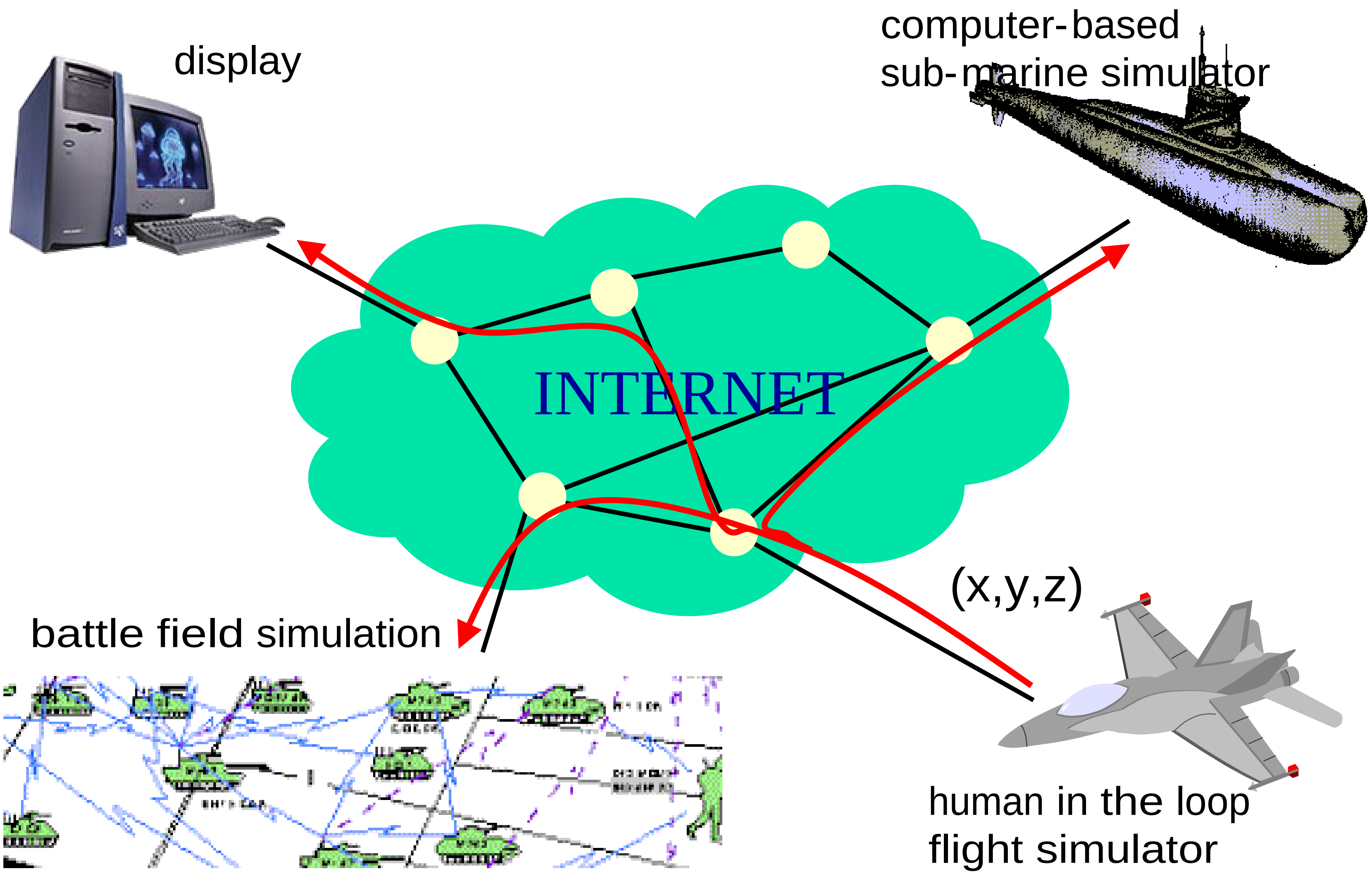
Reliable multicast: a big win for grids

- Data replications
- Code & data transfers, interactive job submissions
- Data communications for distributed applications (collective & gather operations, sync. barrier)
- Databases, directories services



Multicast address group 224.2.0.1

Wide-area interactive simulations



The challenges of multicast

SCALABILITY - SECURITY - TCP Friendliness - MANAGEMENT

SCALABILITY

SCALABILITY

SCALABILI

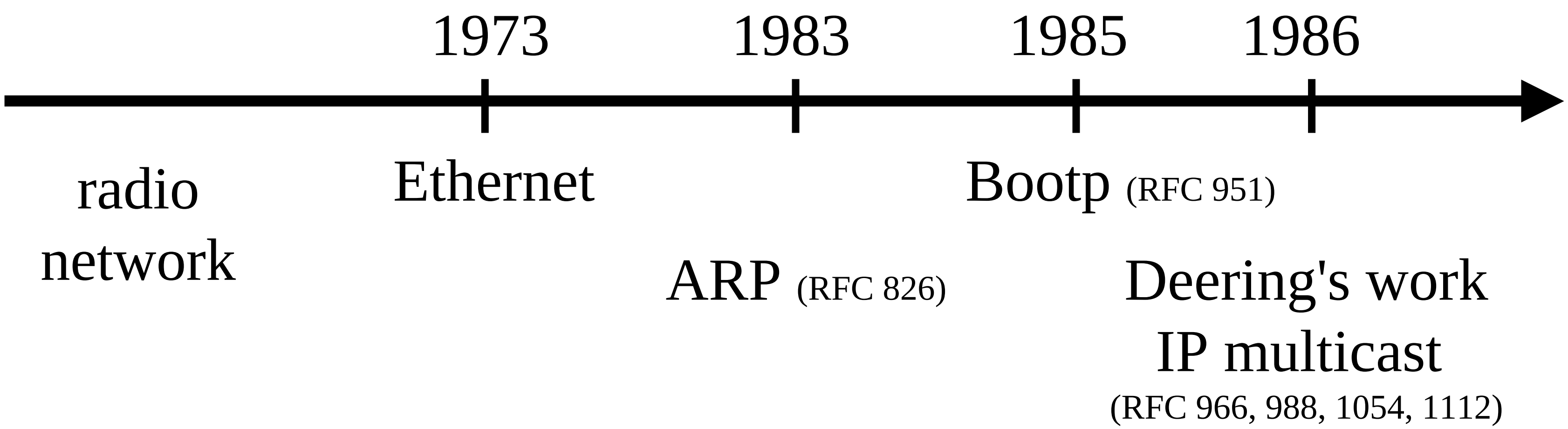
Part I

Basic of IP multicast model

IP multicast routing

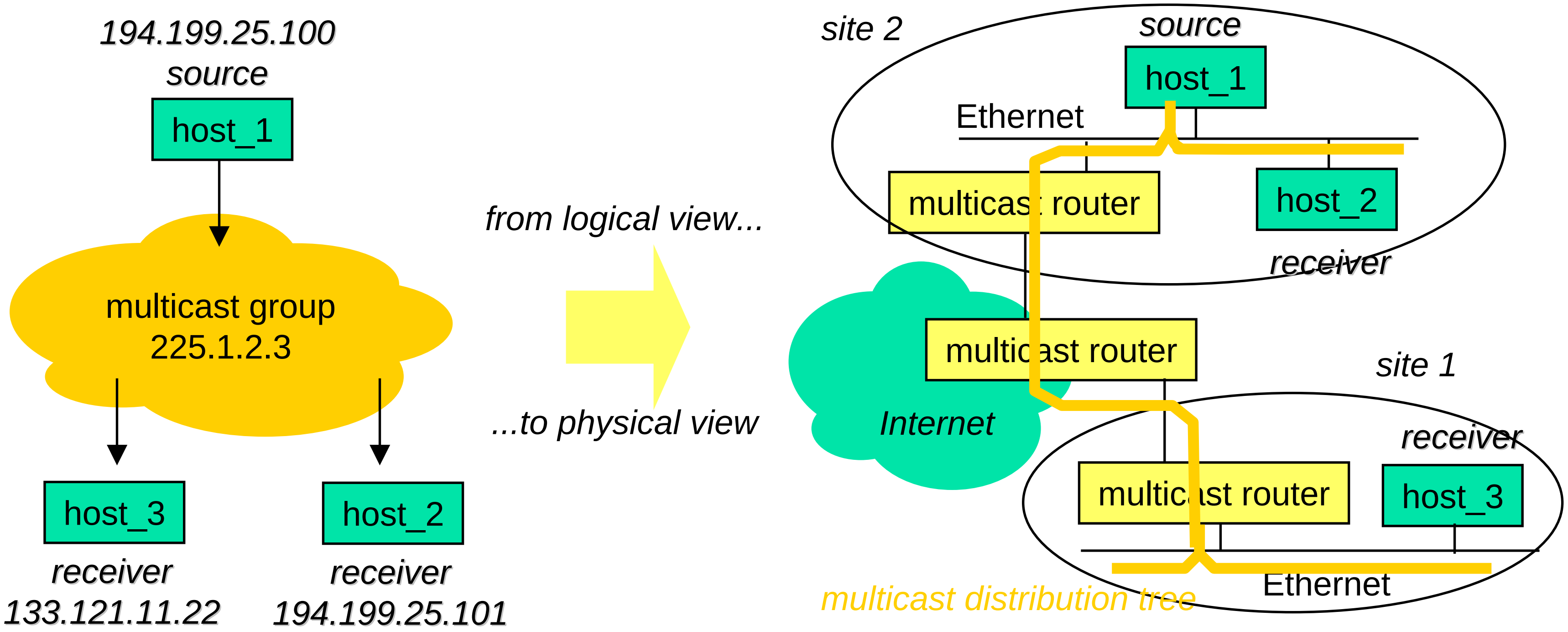
A look back in history of multicast

- History
  - Long history of usage on shared medium networks
  - Resource discovery: ARP, Bootp.



The Internet group model

- multicast/group communications means...
  - 1 → n   as well as   n → m
- a group is identified by a class D IP address (224.0.0.0 to 239.255.255.255)
  - abstract notion that does not identify any host!

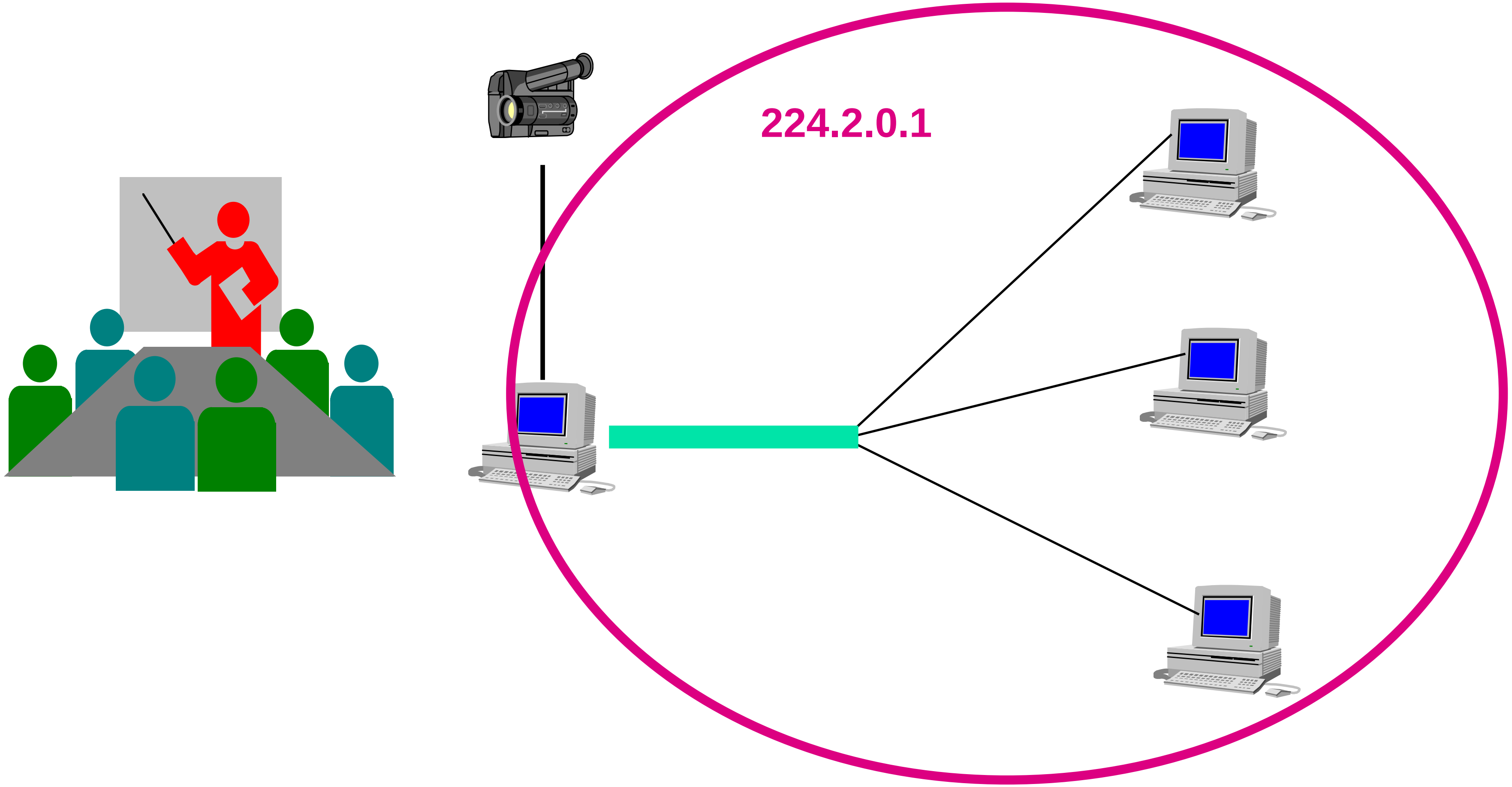


The group model is an open model

- anybody can belong to a multicast group
  - no authorization is required
- a host can belong to many different groups
  - no restriction
- a source can send to a group, no matter whether it belongs to the group or not
  - membership not required
- the group is dynamic, a host can subscribe to or leave at any time
- a host (source/receiver) does not know the number/identity of members of the group

Example: video-conferencing

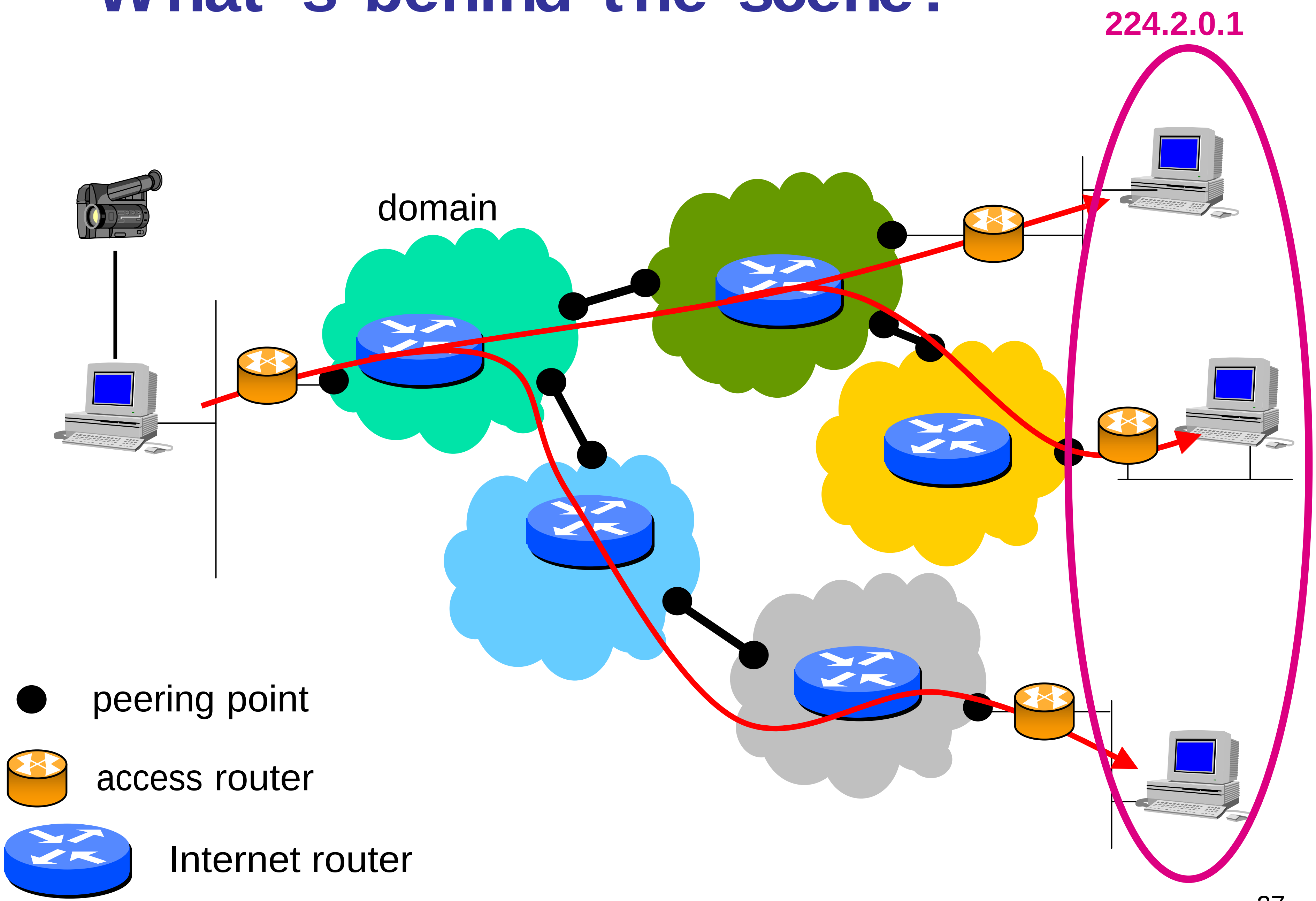
The user's perspective



Multicast address group 224.2.0.1

from UREC, http://www.urec.fr

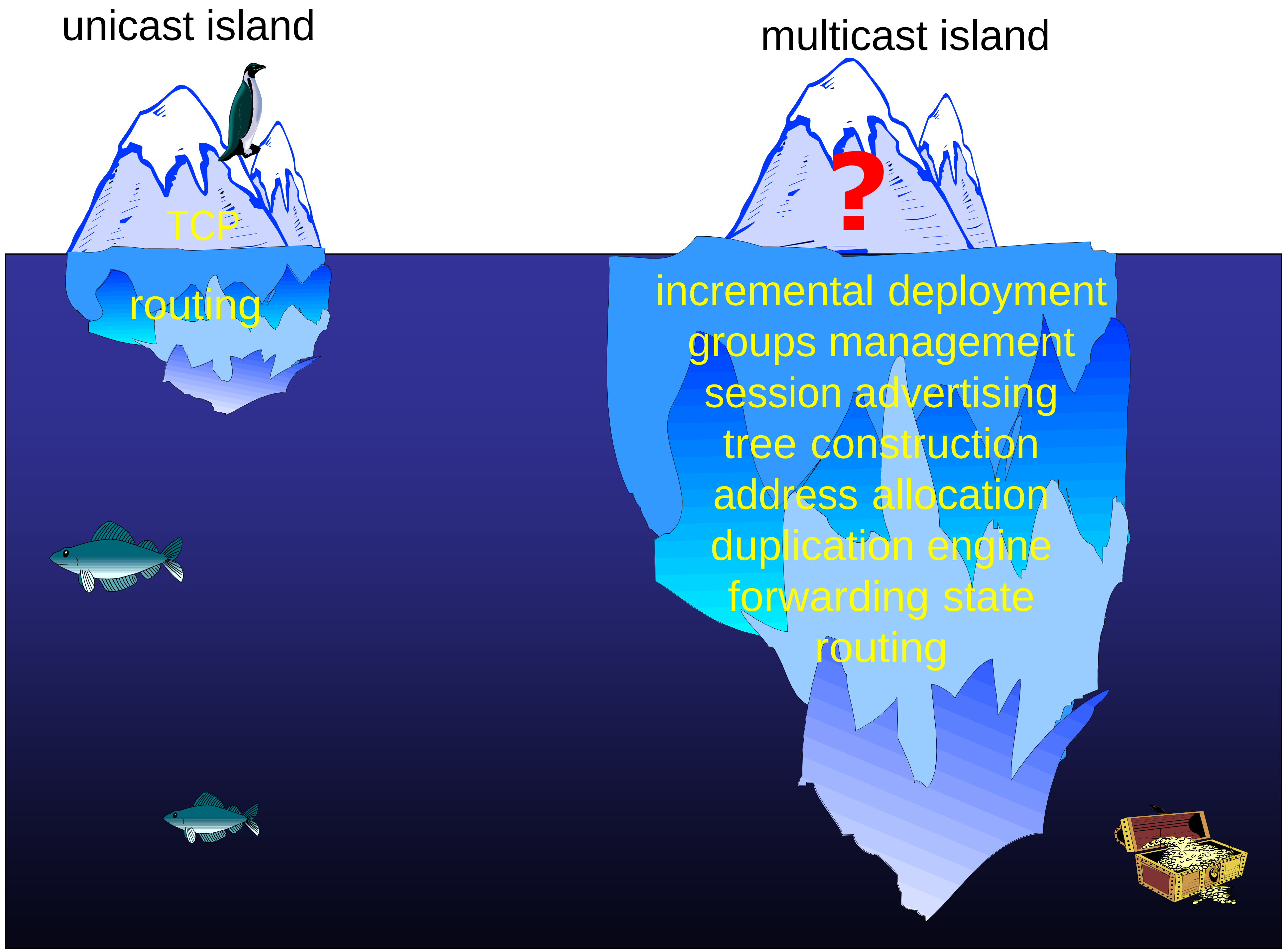
What's behind the scene?



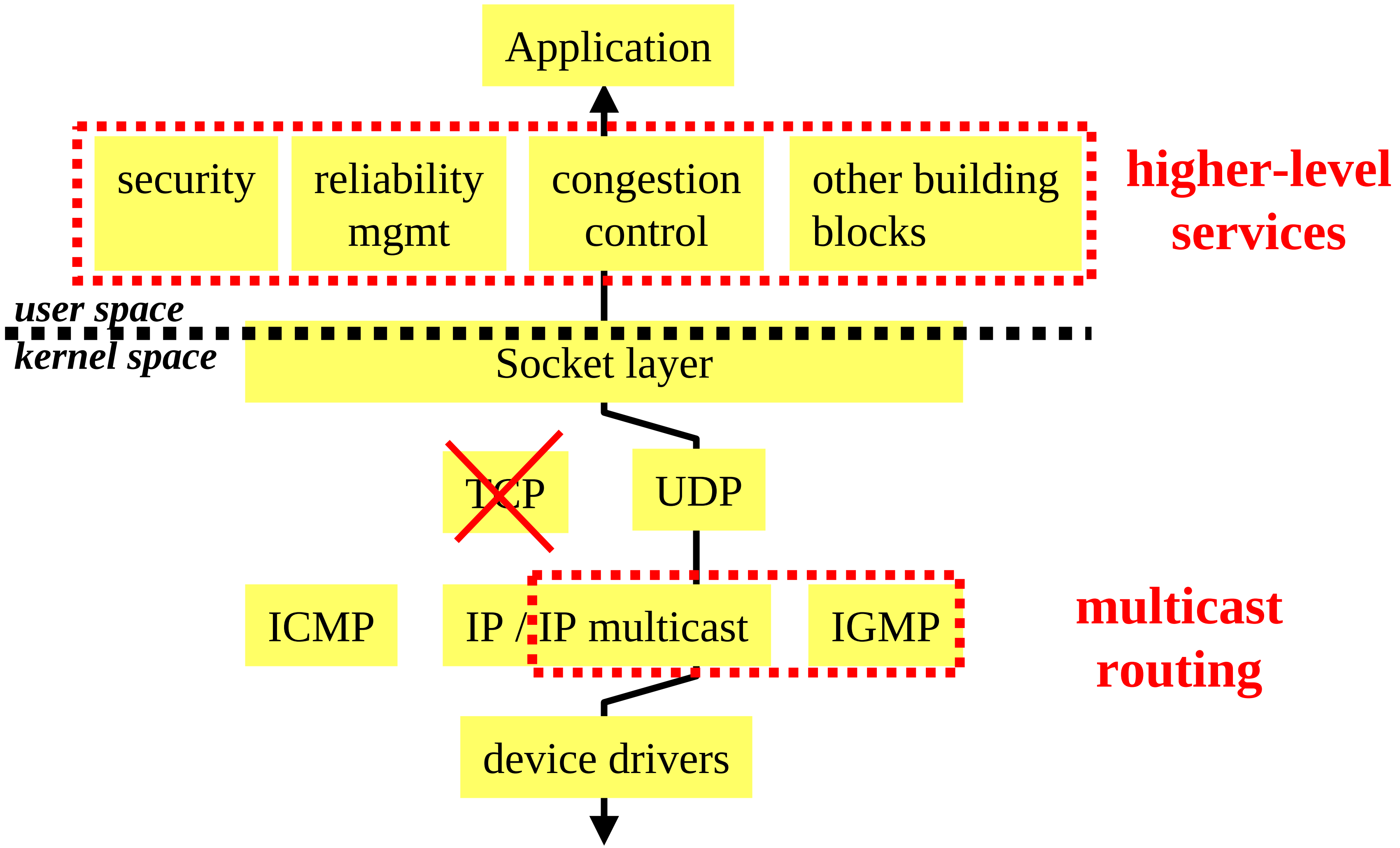
IP multicast TODO list

- ✓ Receivers must be able to subscribe to groups, need group management facilities
- ✓ A communication tree must be built from the source to the receivers
- ✓ Branching points in the tree must keep multicast state information
- ✓ Inter-domain routing must be reconsidered for multicast traffic
- ✓ Need to consider non-multicast clouds

good luck…


unicast island
TCP
routing
multicast island
?
incremental deployment
groups management
session advertising
tree construction
address allocation
duplication engine
forwarding state
routing

Multicast and the TCP/IP layered model

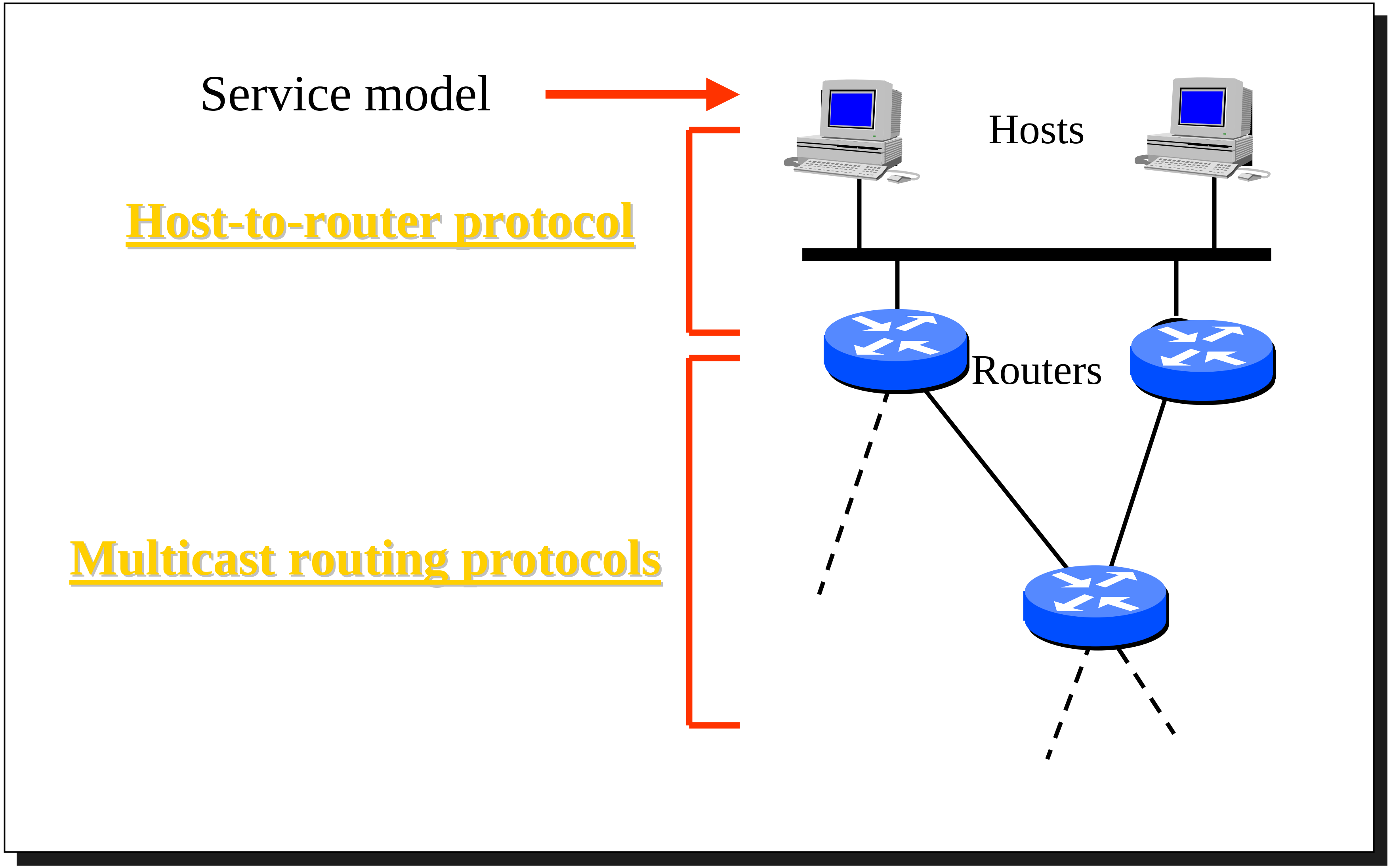


The two sides of IP multicast

- local-area multicast
  - use the potential diffusion capabilities of the physical layer (e.g. Ethernet)
  - efficient and straightforward

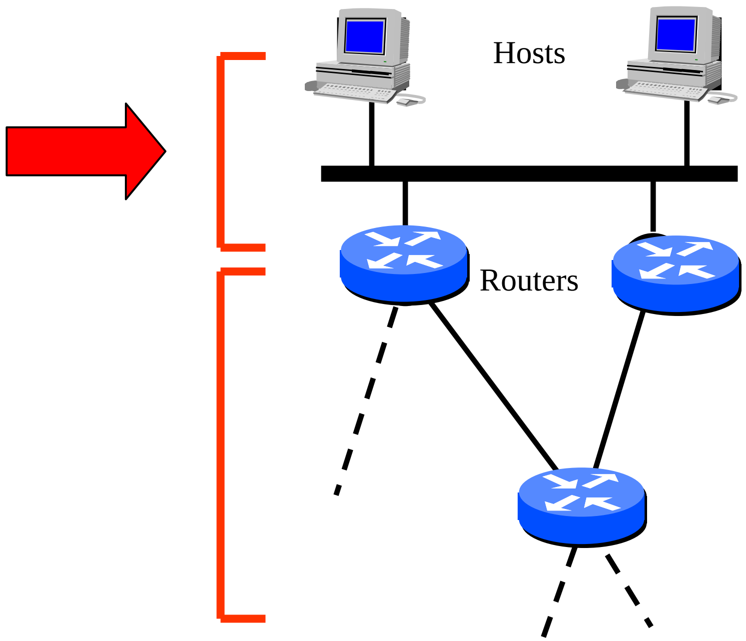
- wide-area multicast
  - requires to go through multicast routers, use IGMP/multicast routing/...(e.g. DVMRP, PIM-DM, PIM-SM, PIM-SSM, MSDP, MBGP, BGMP, MOSPF, etc.)
  - routing in the same administrative domain is simple and efficient
  - inter-domain routing is <u>complex, not fully operational</u>

IP Multicast Architecture

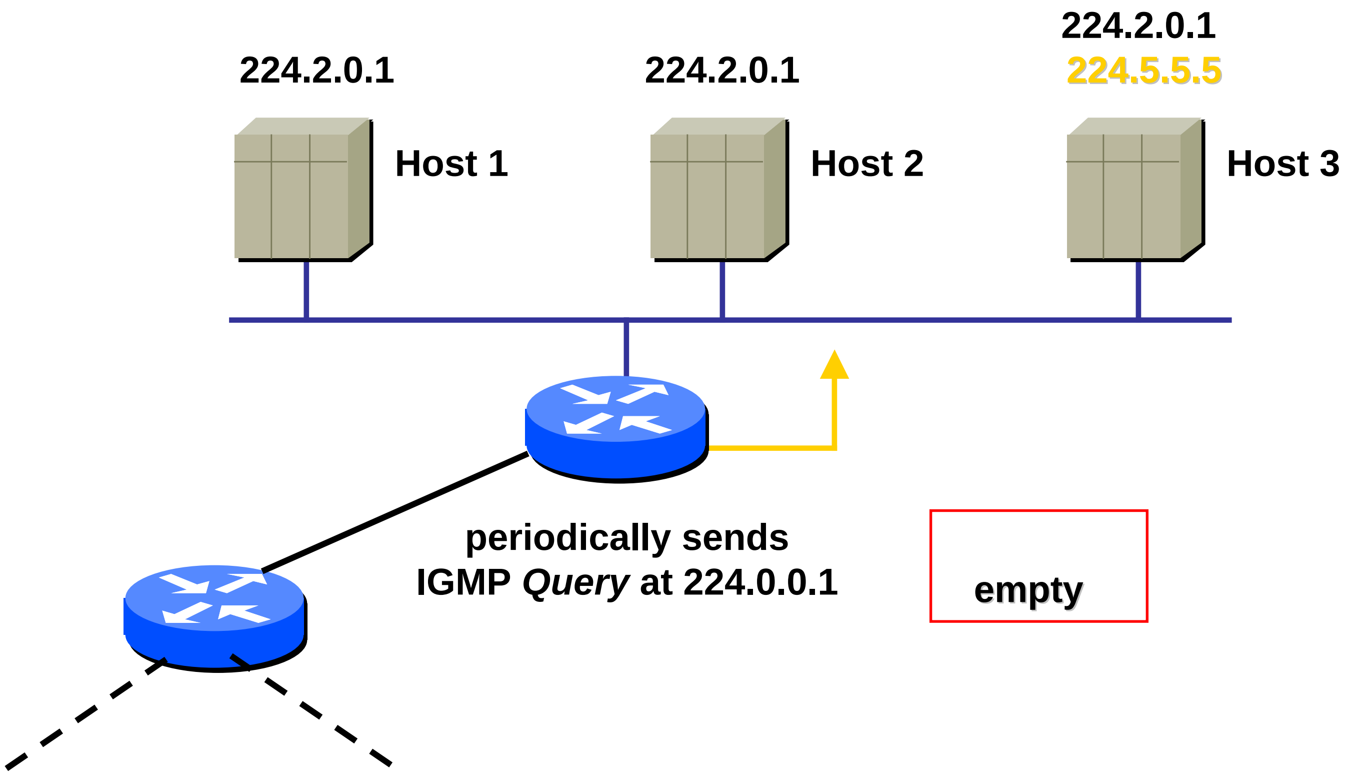


Internet Group Management Protocol (RFC 1112)

- IGMP: “signaling” protocol to establish, maintain, remove groups on a subnet.
- Objective: keep router up-to-date with group membership of entire LAN
  - Routers need not know who all the members are, only that members exist
- Each host keeps track of which mcast groups are subscribed to
  - Socket API informs IGMP process of all joins



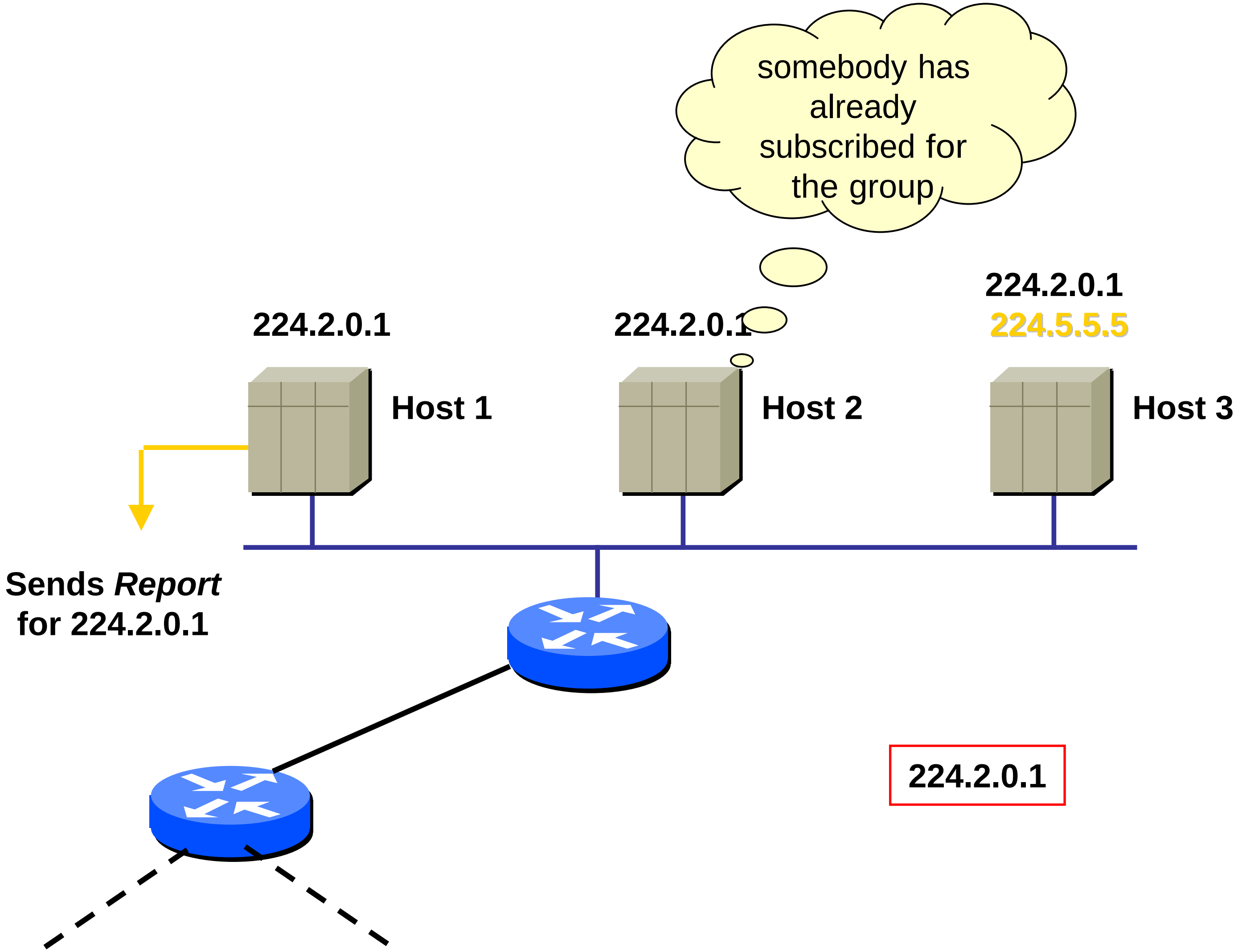
IGMP: subscribe to a group (1)



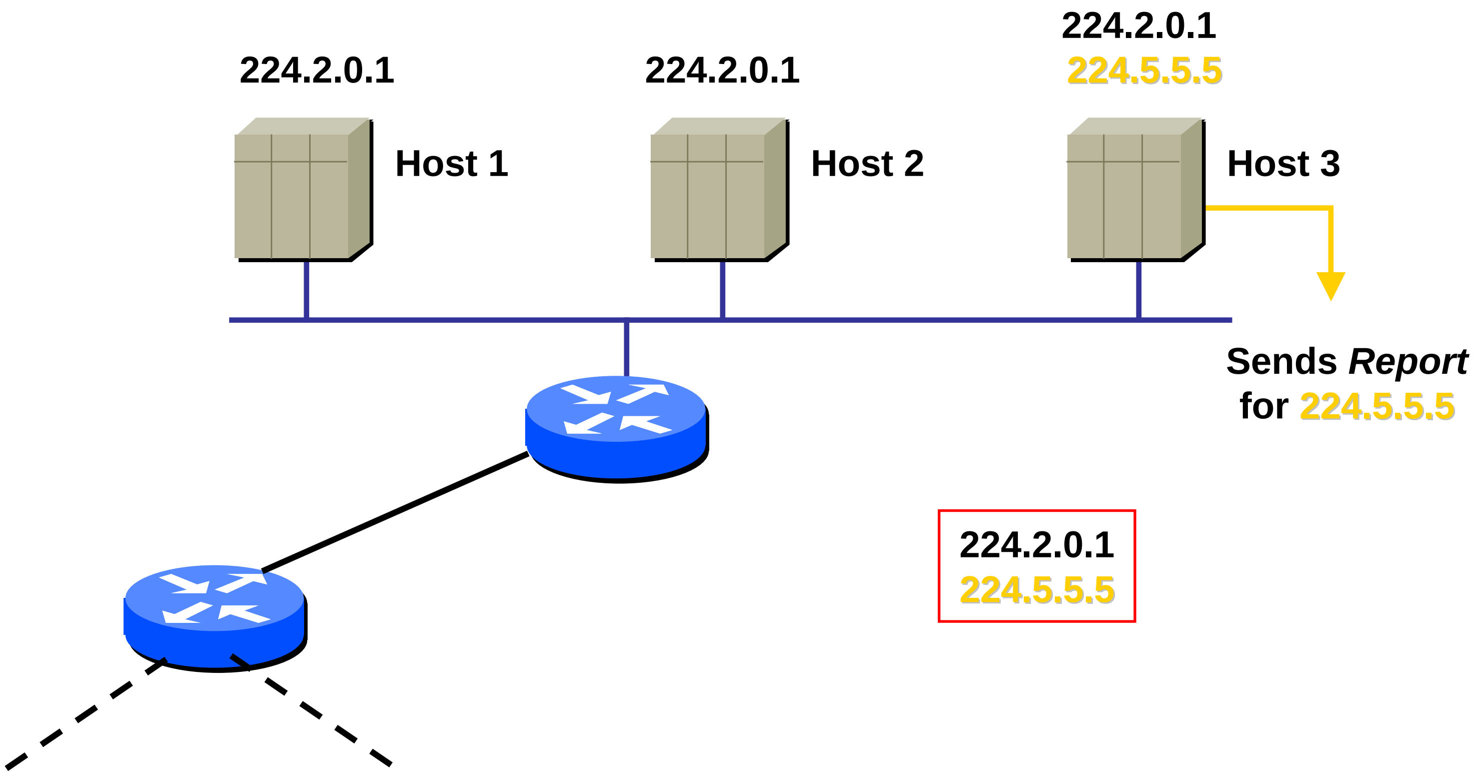
224.0.0.1 reach all multicast host on the subnet

from UREC

IGMP: subscribe to a group (2)

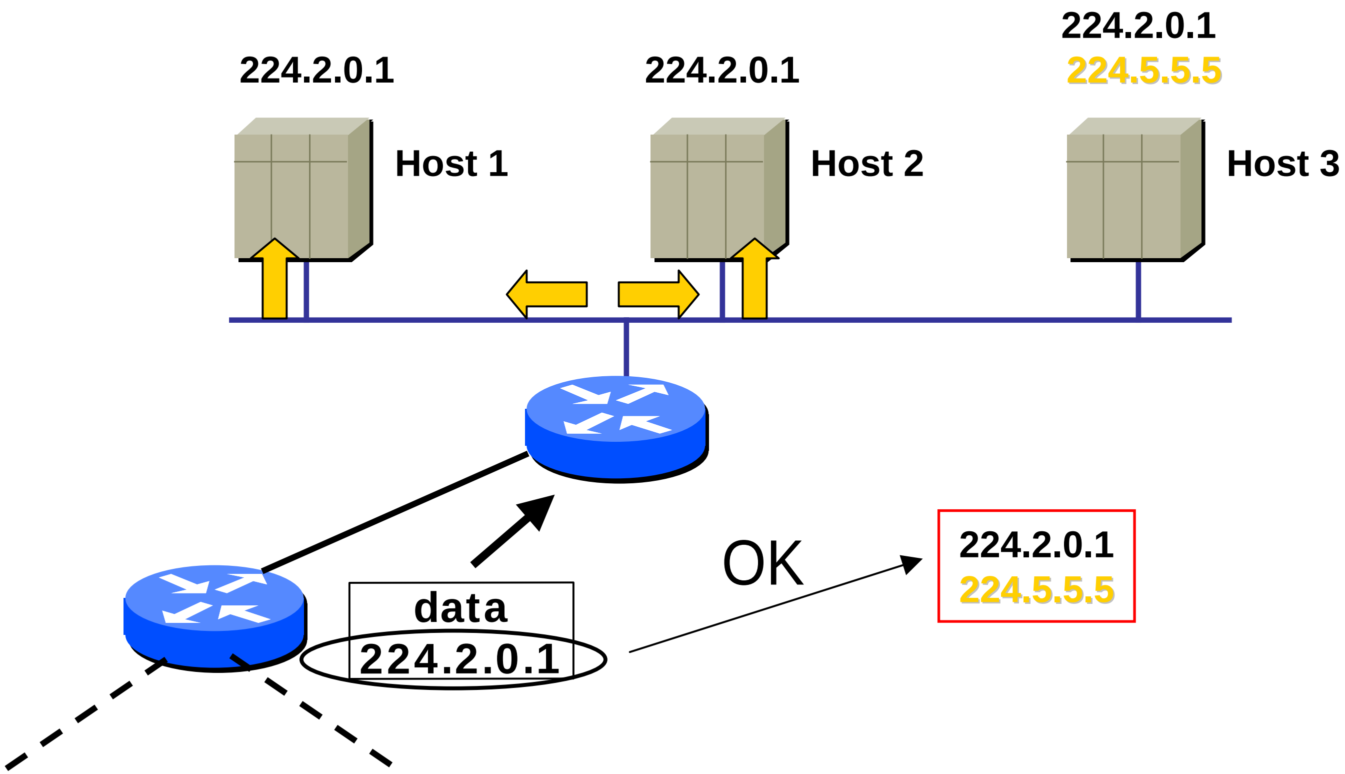


IGMP: subscribe to a group (3)



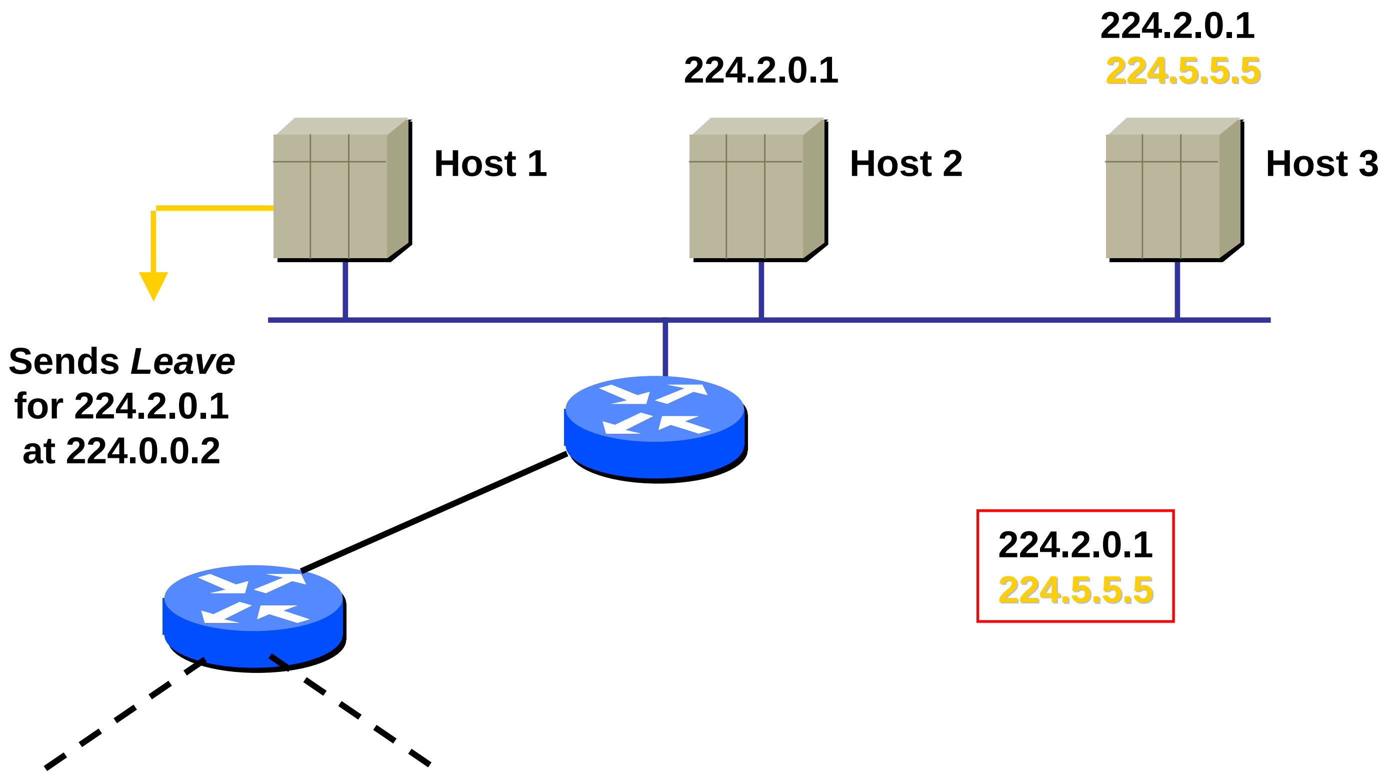
from UREC

Data distribution example



from UREC

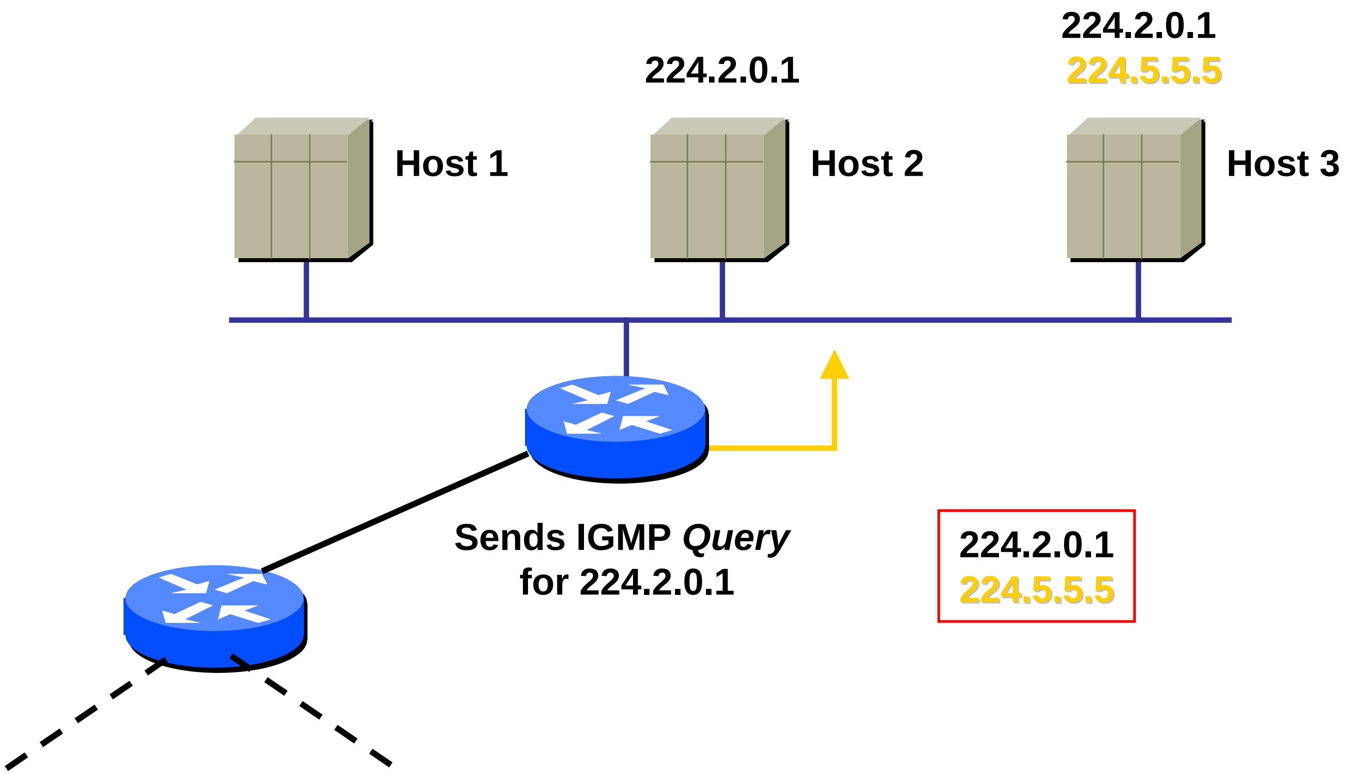
IGMP: leave a group (1)



224.0.0.2 reach the multicast enabled router in the subnet

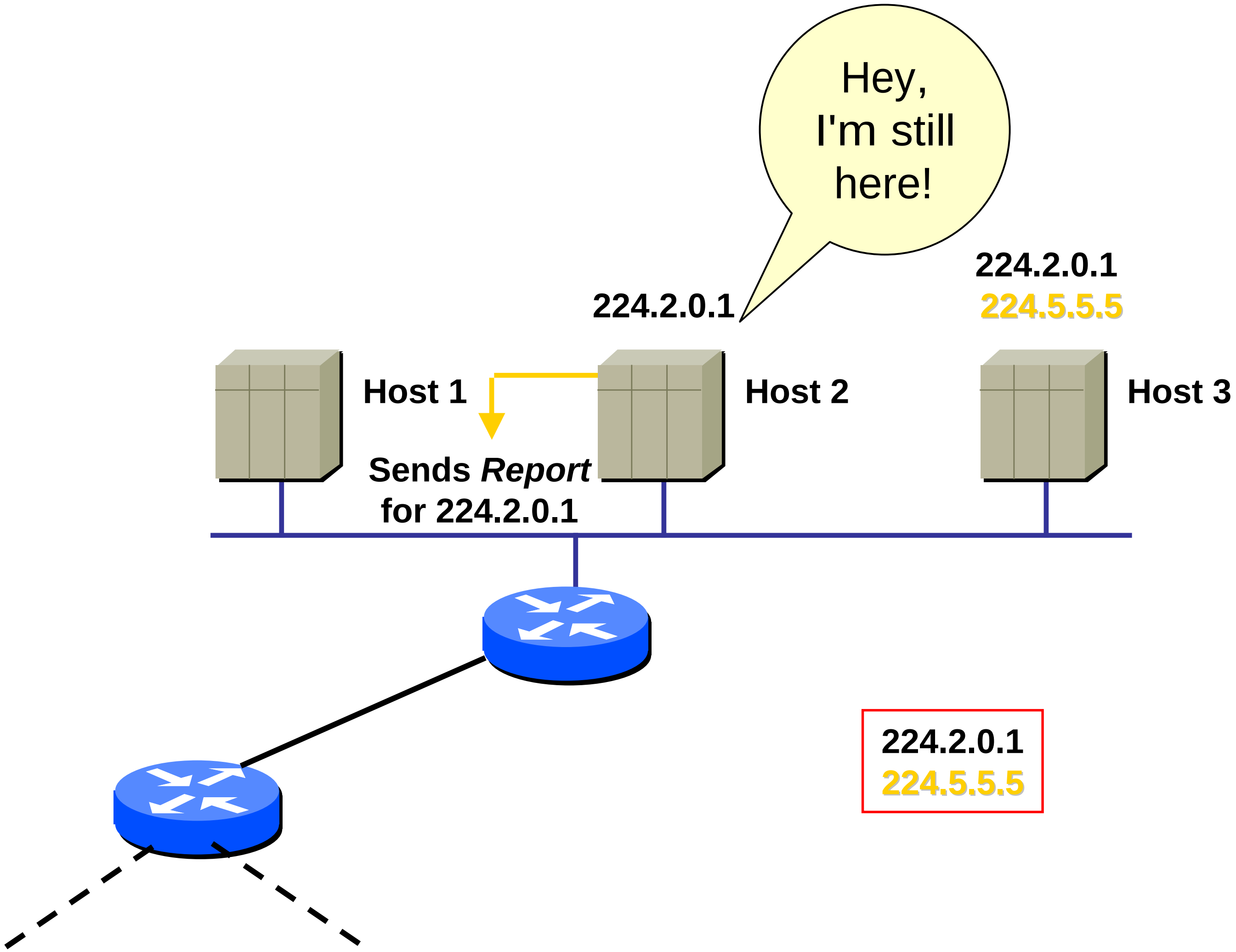
from UREC

IGMP: leave a group (2)



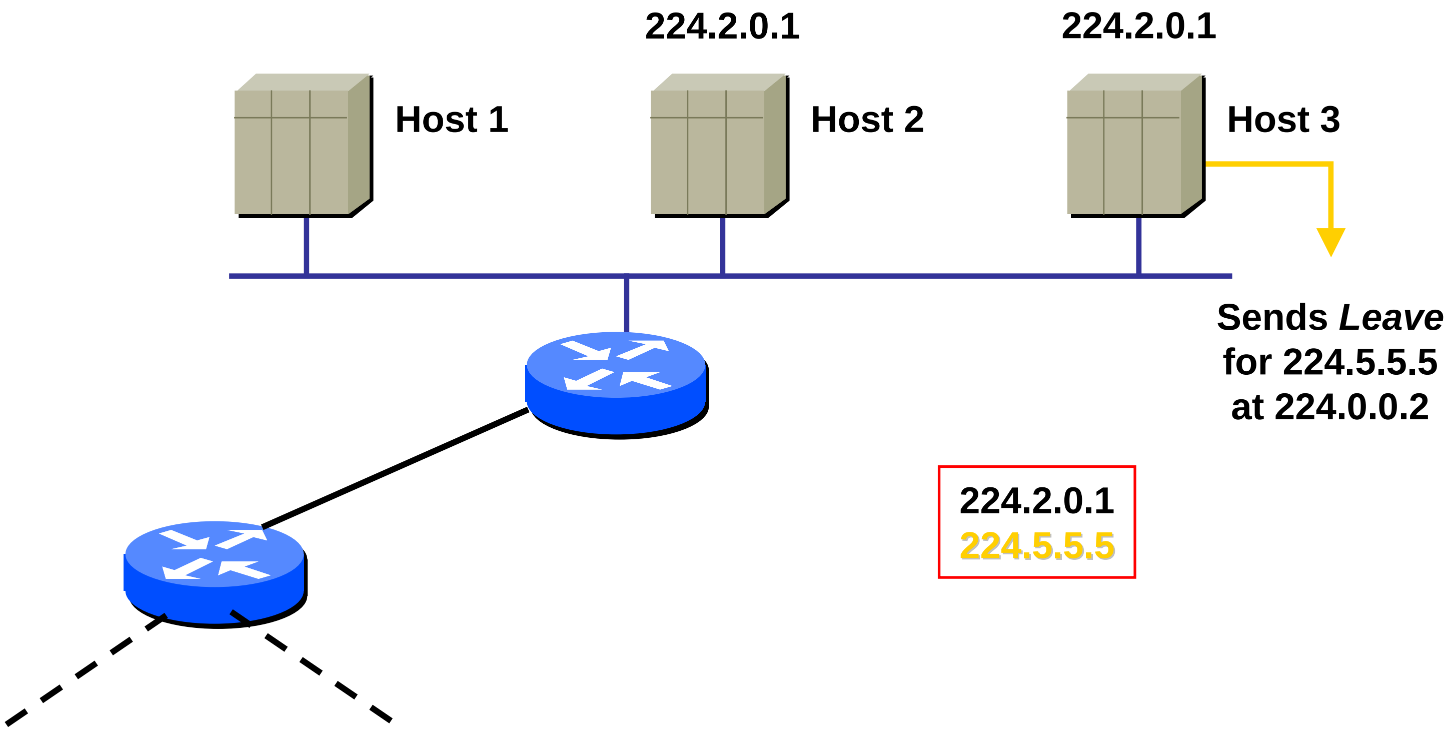
from UREC

IGMP: leave a group (3)

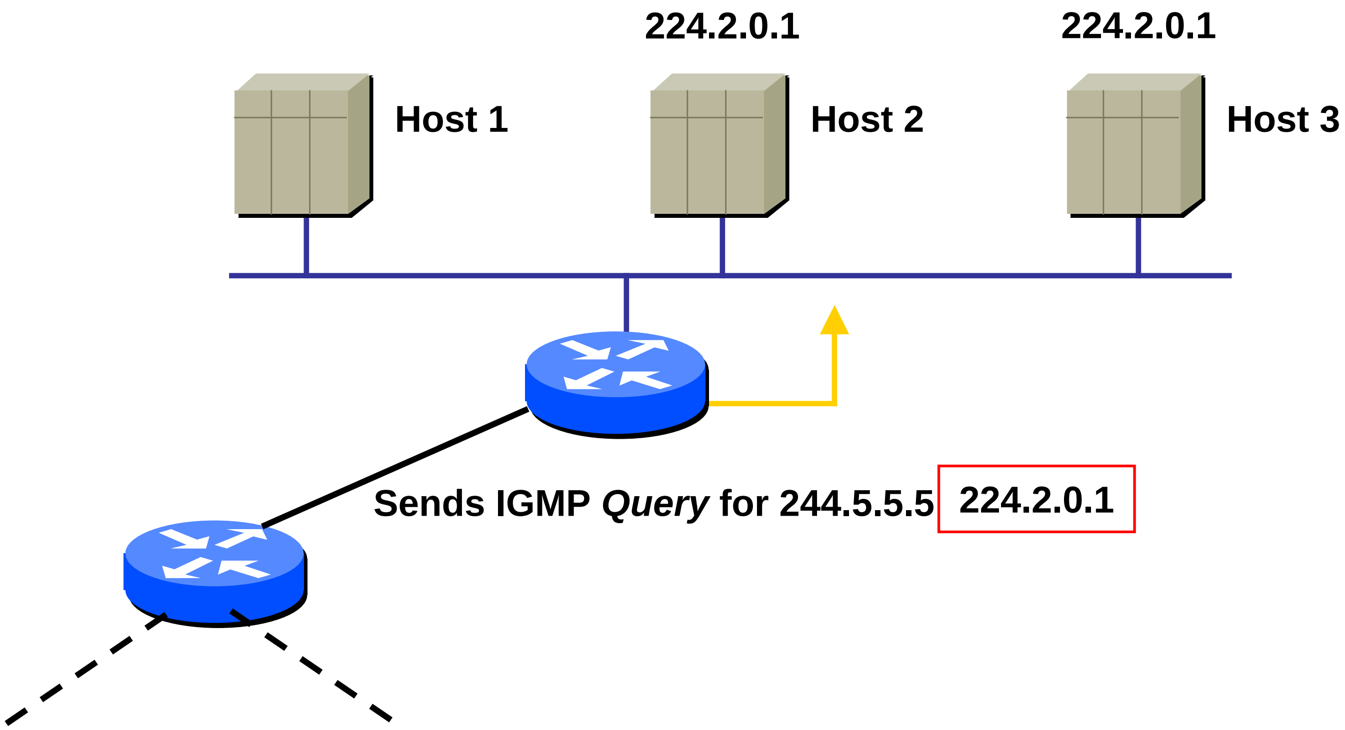


from UREC

IGMP: leave a group (4)

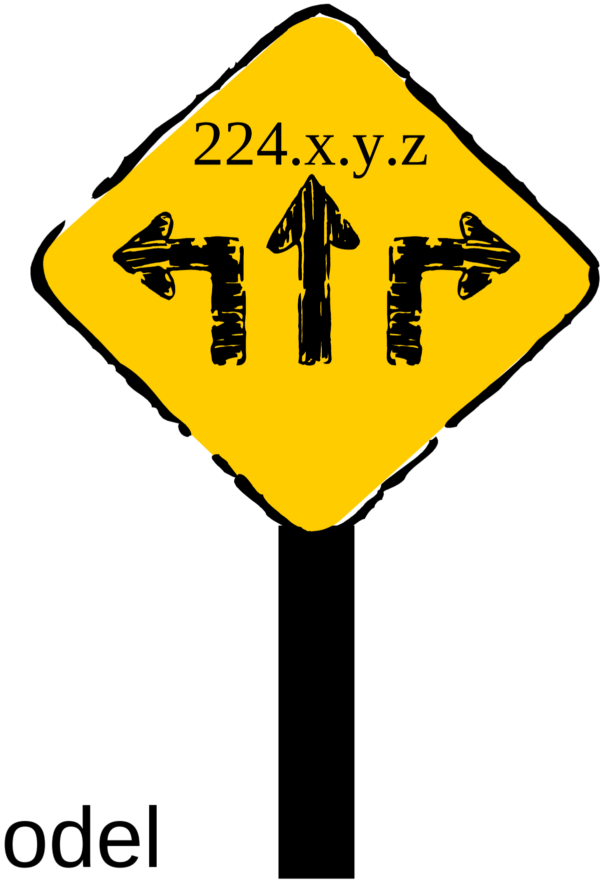


IGMP: leave a group (5)



from UREC

Part I



Basic of IP multicast model

<u>IP multicast routing</u>

2. IP multicast routing

- We'll see in this section

  2.1- traditional dense mode multicast routing

  2.2- sparse mode multicast routing

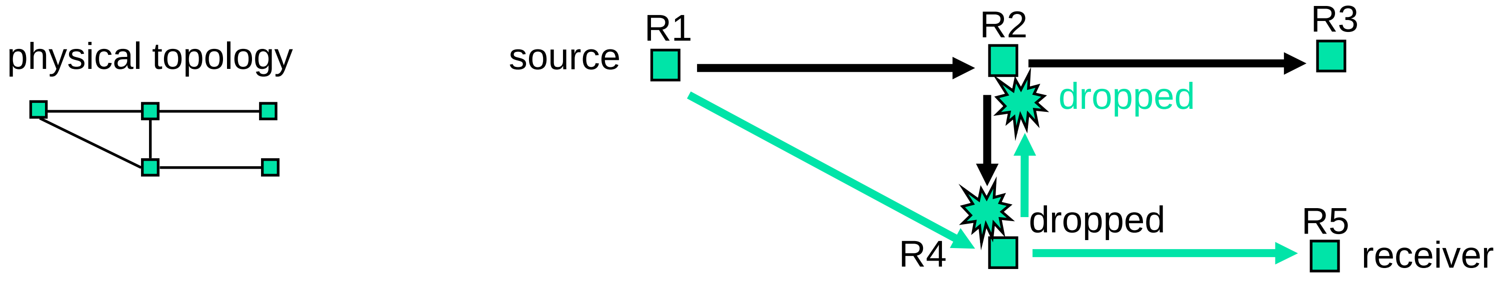
  2.3- source specific multicast routing

  we don't go into the details, we merely give the main ideas…

2.1- Dense mode protocols, DVMRP

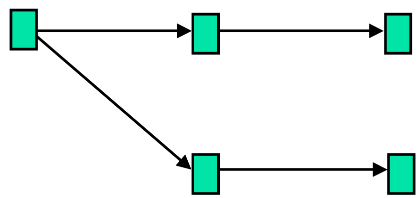
- The Ancestor: DVMRP (Distance Vector Multicast Routing)
  - based on Reverse Path Forwarding (RPF) algo.

    A multicast router forwards packets received from a line which is on the shortest path to the source, and drops other packets

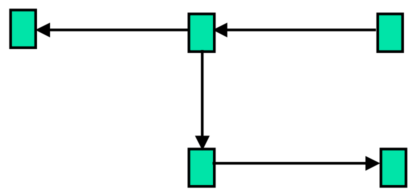


DVMRP… (cont’)

- resulting multicast distribution tree



- different sources lead to diff. trees
  - ⇒ improves load distribution on the links

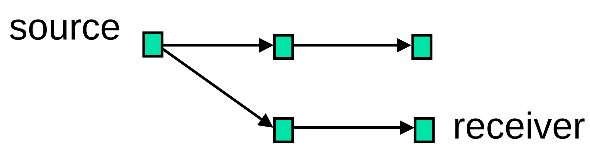


- creates a spanning tree…

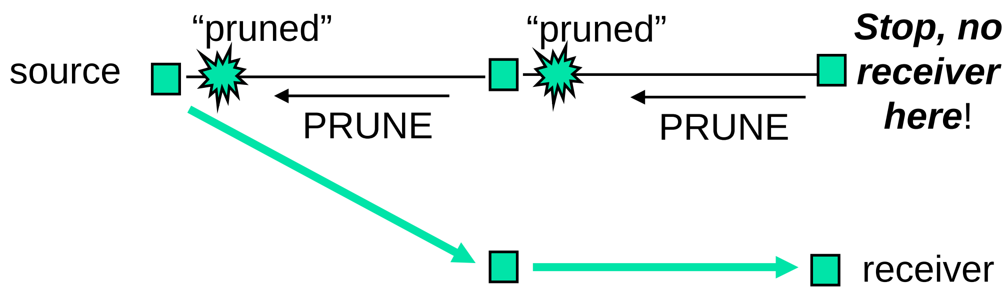
DVMRP… (cont’)

- add “flood and prune” algorithm to dynamically update the tree

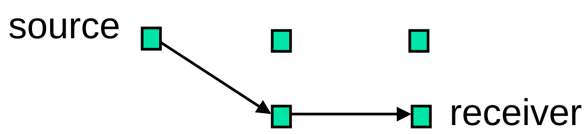
step 1: flood the Internet (only limited by the packet’s TTL)



step 2: prune useless branches



- resulting pruned multicast distribution tree



DVMRP... (cont')

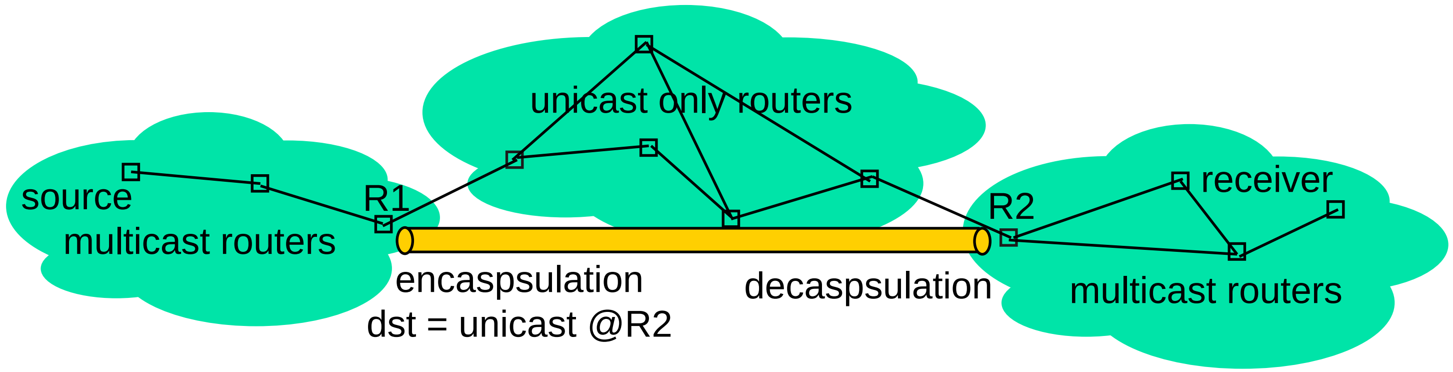
- flooding/pruning is done periodically to update the tree
  - required to discover new receivers and remove branches to receivers who left the session
- limitations:
  - creates signaling load (PRUNE message)
  - periodically creates important traffic (flooding)
  - all routers keep some state for all the multicast groups in use in the Internet

DVMRP… (cont’)

- large scale deployment of DVMRP in the MBONE (multicast backbone) since 1992
- tunnels are set up to link “multicast islands” through unicast areas

  within a multicast area: native multicast

  in a tunnel: multicast packets are encapsulated in unicast IP packets



DVMRP... (cont')

- it works but... this is far from perfect
  - periodical flooding creates a heavy load on routers/links
  - each multicast router must keep some forwarding state for each group
  - tunneling quickly became anarchic
  - this is a flat architecture (the same protocol is used everywhere)
- conclusion: “dense mode protocols” like DVMRP are not scalable enough for WAN multicast routing
  - dense mode means that we assume a dense distribution of receivers, wrong in practice!

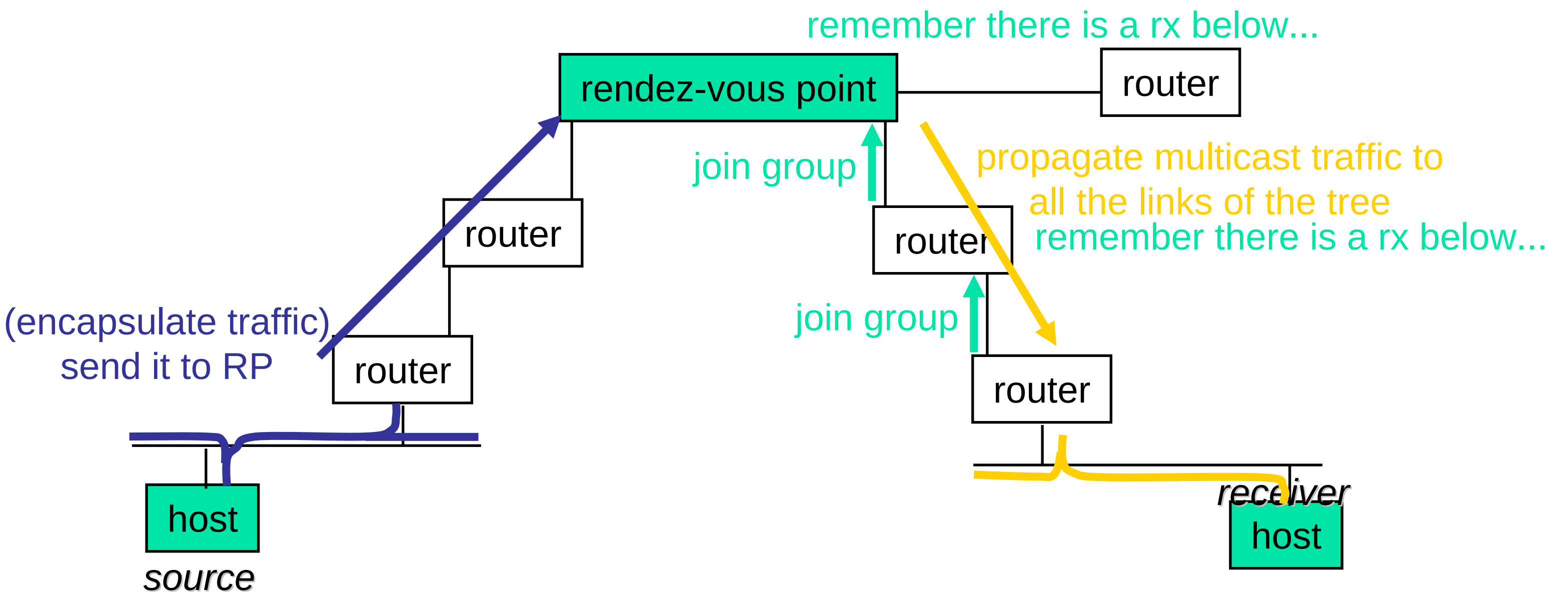
2.2- Sparse mode protocols

- The newcomers: PIM-SM/MSDP/MBGP
  - **PIM-SM** (Protocol Independent Multicast - Sparse Mode)
  - **MSDP** (Multicast Source Discovery Protocol)
  - **MBGP** (Multicast Border Gateway Protocol)
- domain ≅ site, or ISP network

  similar to “autonomous systems” of unicast routing
- intra-domain mcast routing uses PIM-SM
- inter-domain mcast routing requires MBGP
- the discovery of sources in other domains requires MSDP

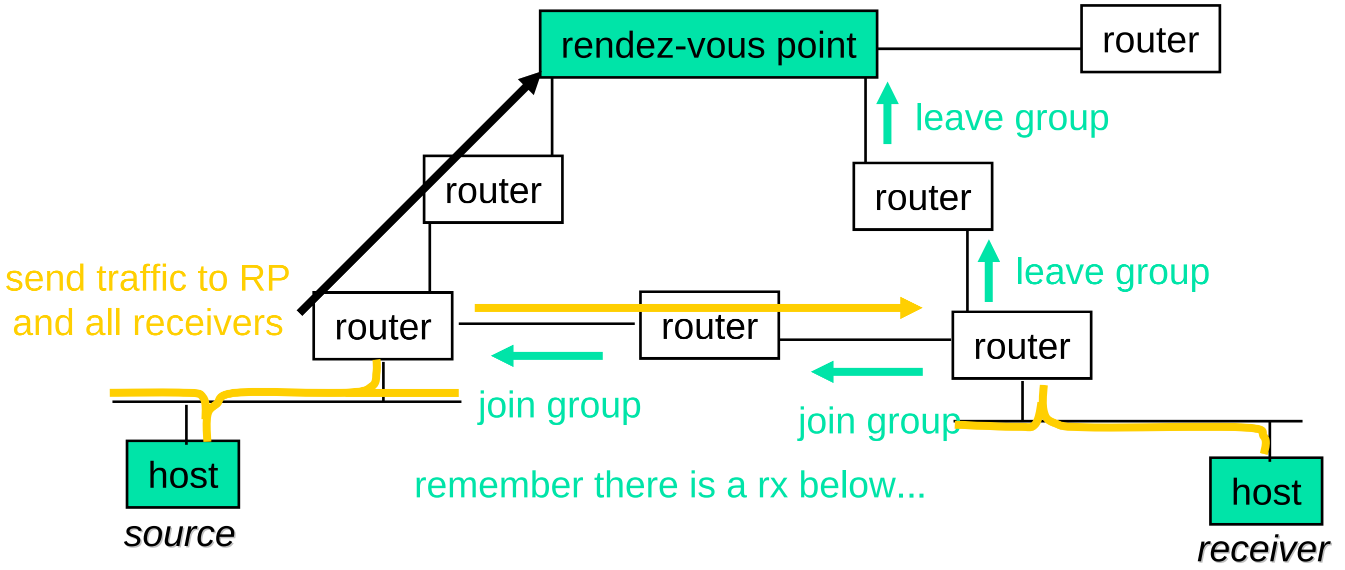
PIM-SM for intra-domain multicast

- Based on a “rendez-vous point” (RP)
  - assumes receivers are sparsely distributed
    ⇒ concentrating traffic on a RP is relevant
  - **STEP1:** a single “shared tree” is built, no matter how many sources there are



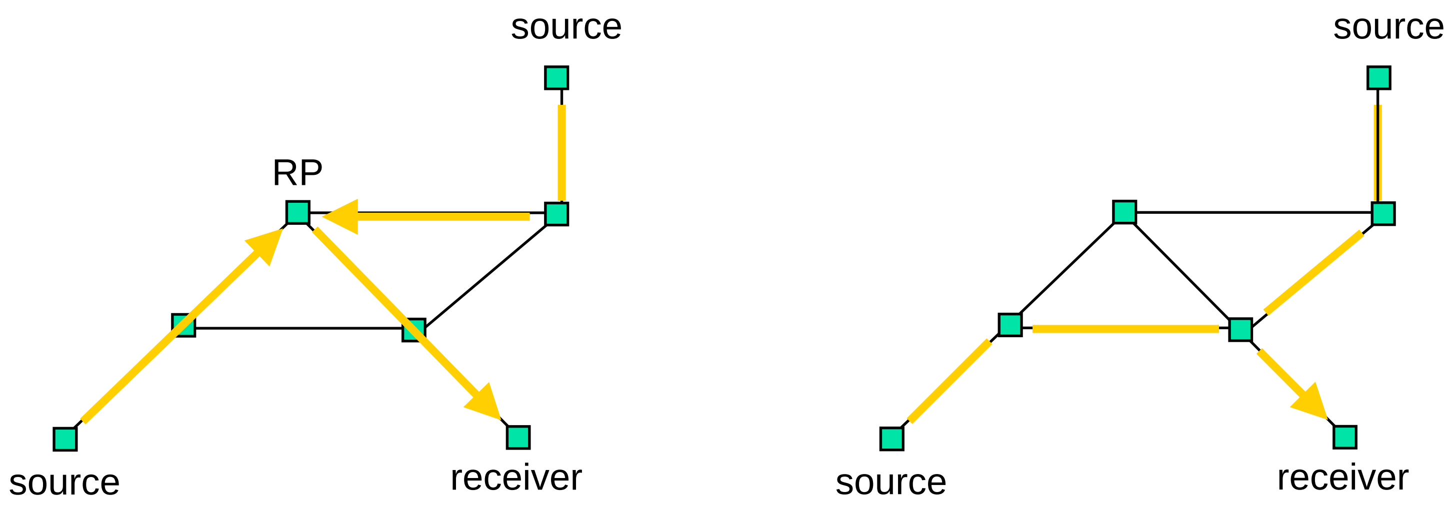
PIM-SM… (cont’)

- **Step2:** build a per-source tree now that the receiver knows who is(are) the source(s)
- in practice move from shared tree to per-source tree upon first packet reception !



PIM-SM... (cont')

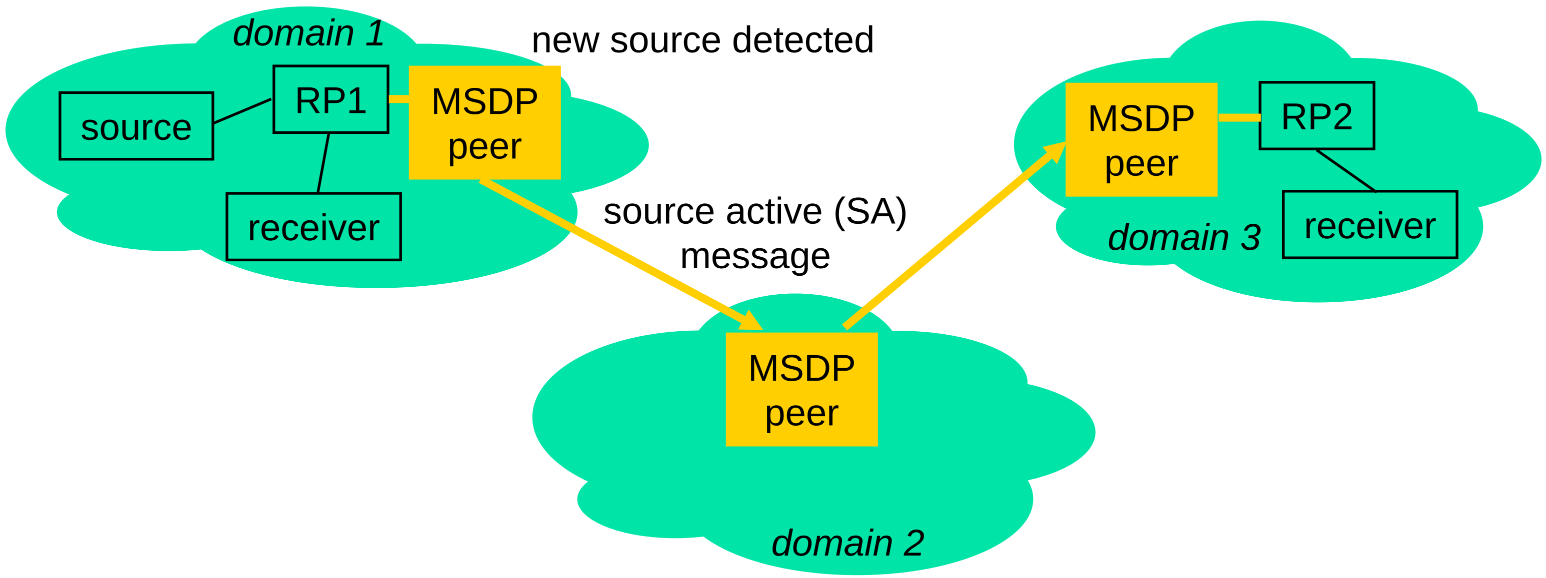
- moving to a per-source tree is efficient for bulk data transfer, but has a higher cost in case of multiple sources
  - one tree per source versus a single shared tree



from shared tree... *...to per-source tree*

MSDP for inter-domain src discov.

- each domain runs PIM-SM with its own local RP to avoid third-party dependency
- problem: how can a receiver in a domain be informed of a source located in another domain... with MSDP!



MSDP… (cont’)

- problem with some applications
  - reducing the join latency requires using a cache in each peer of active sources
  - follows a soft-state model, where entries must be periodically refreshed
  - does not work with low frequency bursty applications
    - soft-state is lost each time a packet sent… receivers never get any packet

- limited scalability in terms of nb groups
  - each peer informs every other peer of local sources, and everybody knows everything !

Conclusions PIM-SM/MBGP/MSDP

- works, currently operational
  - deployed in the American Internet2 network
  - deployed in the GEANT European network

    http://www.dante.net/nep/GEANT-MULTICAST/
- but this is not the long term solution...
  - high signaling load for dynamic groups
  - problems with low frequency bursty applications
  - limited scalability with the number of groups

  - long term solution may be quite different...

2.3- Source-specific mcast routing

- new trend: **source specific multicast**
  - a group, called channel, is identified by:

    {source@, multicast@}
- single-source 1 $\rightarrow$ n model
    - {S, M} and {S', M} are disjoint
    - only S can send some traffic to {S, M}
  - n $\rightarrow$ m still possible with many 1 $\rightarrow$ n channels...
  - follows the express multicast proposal

    H. Holbrook, D. Cheriton, "IP multicast channels: EXPRESS support for large-scale single-source applications", ACM SIGCOMM'99, September 1999.

Source specific multicast... (cont')

- many benefits:
  - disjoint mcast addressing space per source

    ... instead of a single global addressing space

    ⇒ no address conflict

  - no need for a bootstrap protocol (like MSDP) for discovering the sources

    ⇒ it is carried in the {S, M} channel identifier

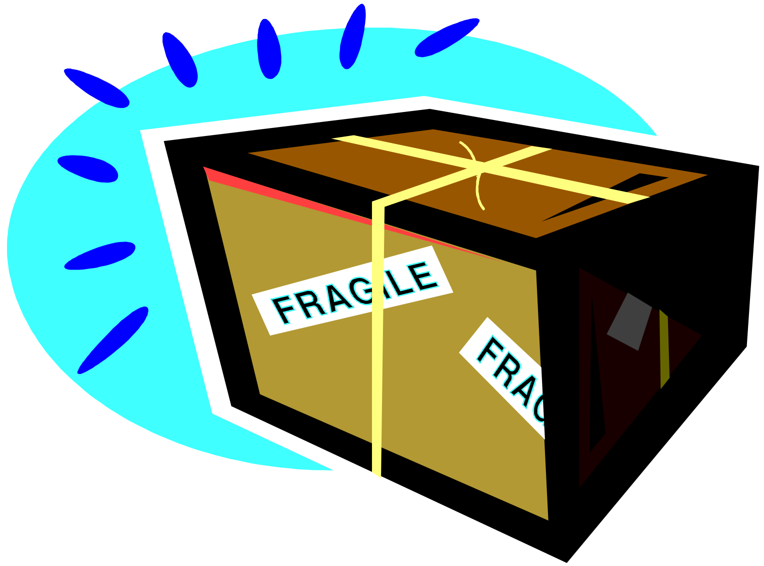
  - more security

    ⇒ only the source can send to {S, M}

Source specific multicast… (cont’)

- works with limited modifications of current protocols
  - use IGMPv3 in hosts and 1st hop routers
  - use a modified version of PIM-SM (no RP, use directly to the per-source tree)
- probably the future of IP Multicast routing…
  - unless the importance of many-to-many applications overwhelms SSM

Part II



Introducing reliability

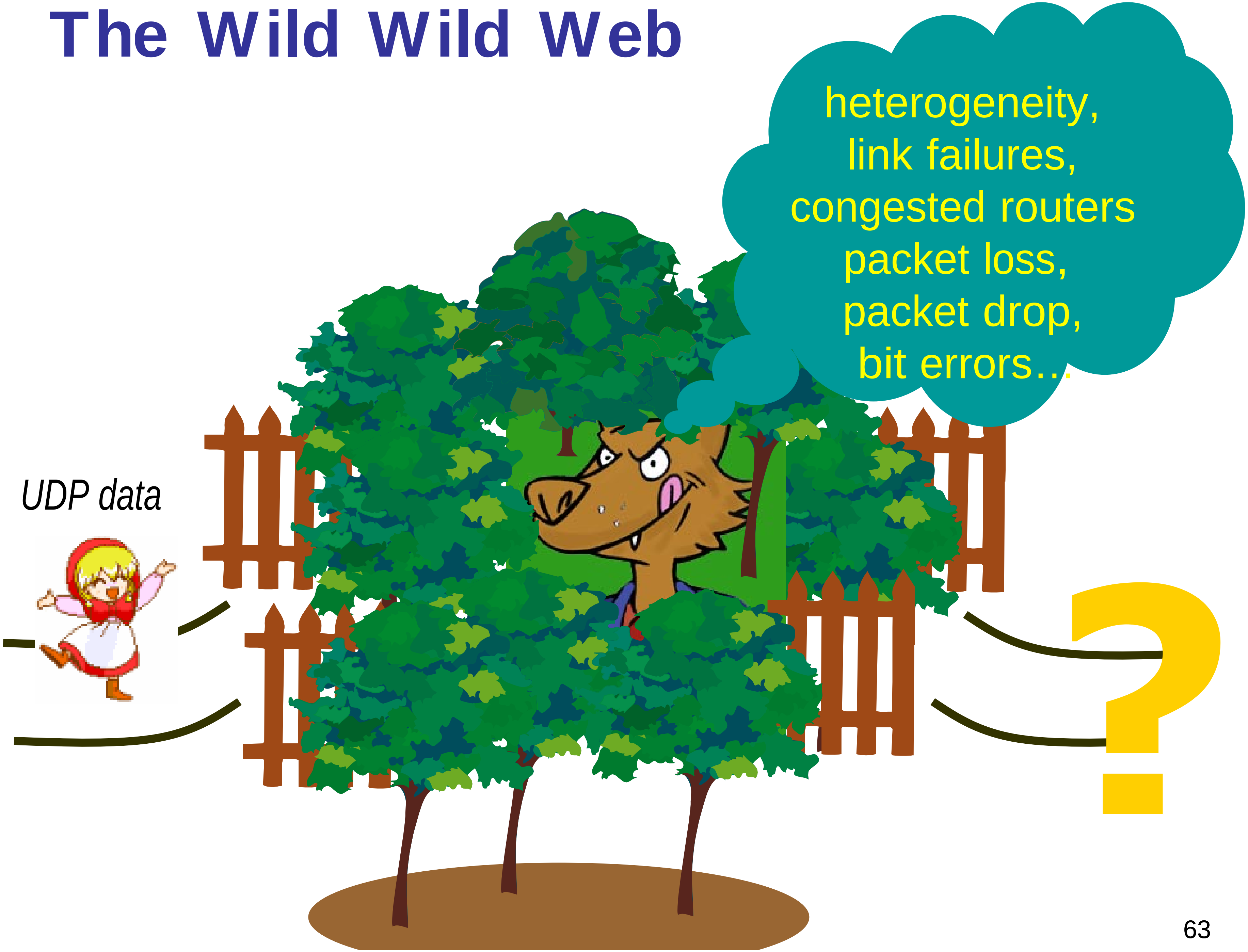
End-to-end solutions

FEC-based solutions

Layered solutions

Router-assisted solutions

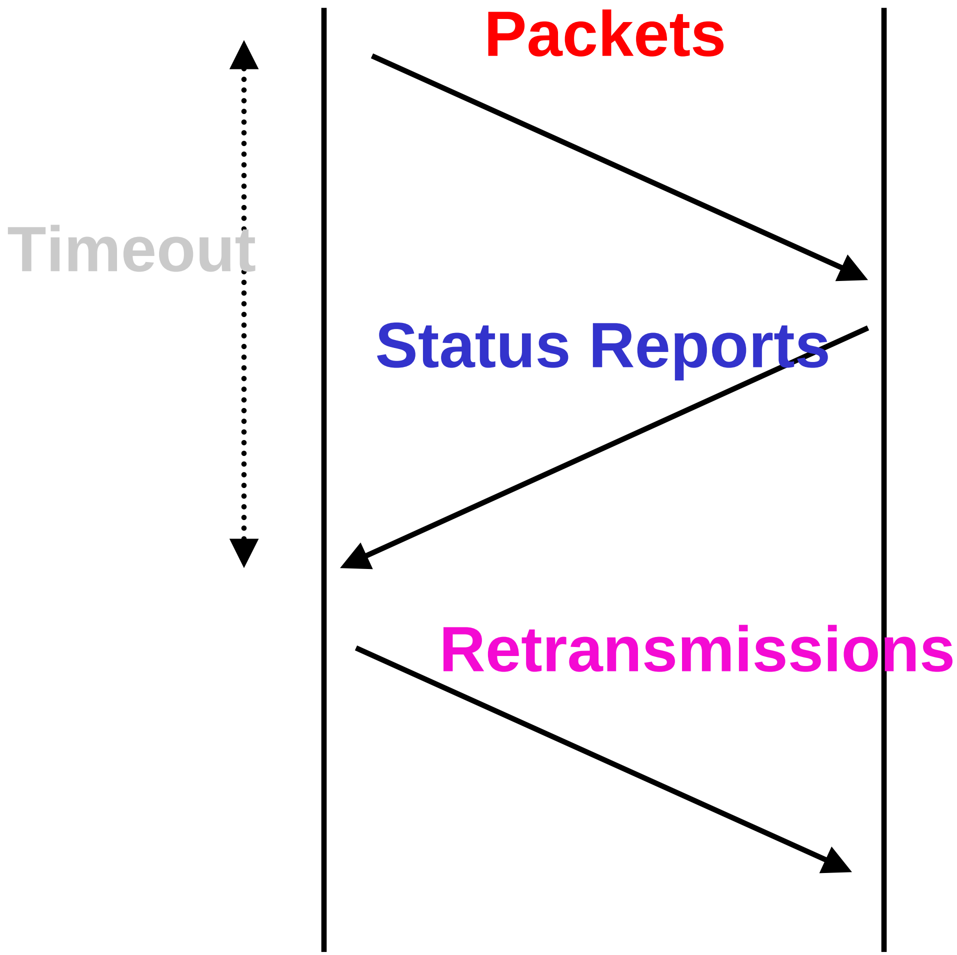
The Wild Wild Web



Reliability Models

- Reliability => requires redundancy to recover from uncertain loss or other failure modes.
- Two types of redundancy:
  - Spatial redundancy: independent backup copies
    - Forward error correction (FEC) codes
    - Problem: requires huge overhead, since the FEC is also part of the packet(s) it cannot recover from erasure of all packets
  - Temporal redundancy: retransmit if packets lost/error
    - Lazy: trades off response time for reliability
    - Design of status reports and retransmission optimization important

Temporal Redundancy Model



- Sequence Numbers
- CRC or Checksum

- ACKs
- NAKs,
- SACKs
- Bitmaps

- Packets
- FEC information

Part II

Introducing reliability

ACK/NACK end-to-end solutions

FEC-based solutions

Layered solutions

Router-assisted solutions

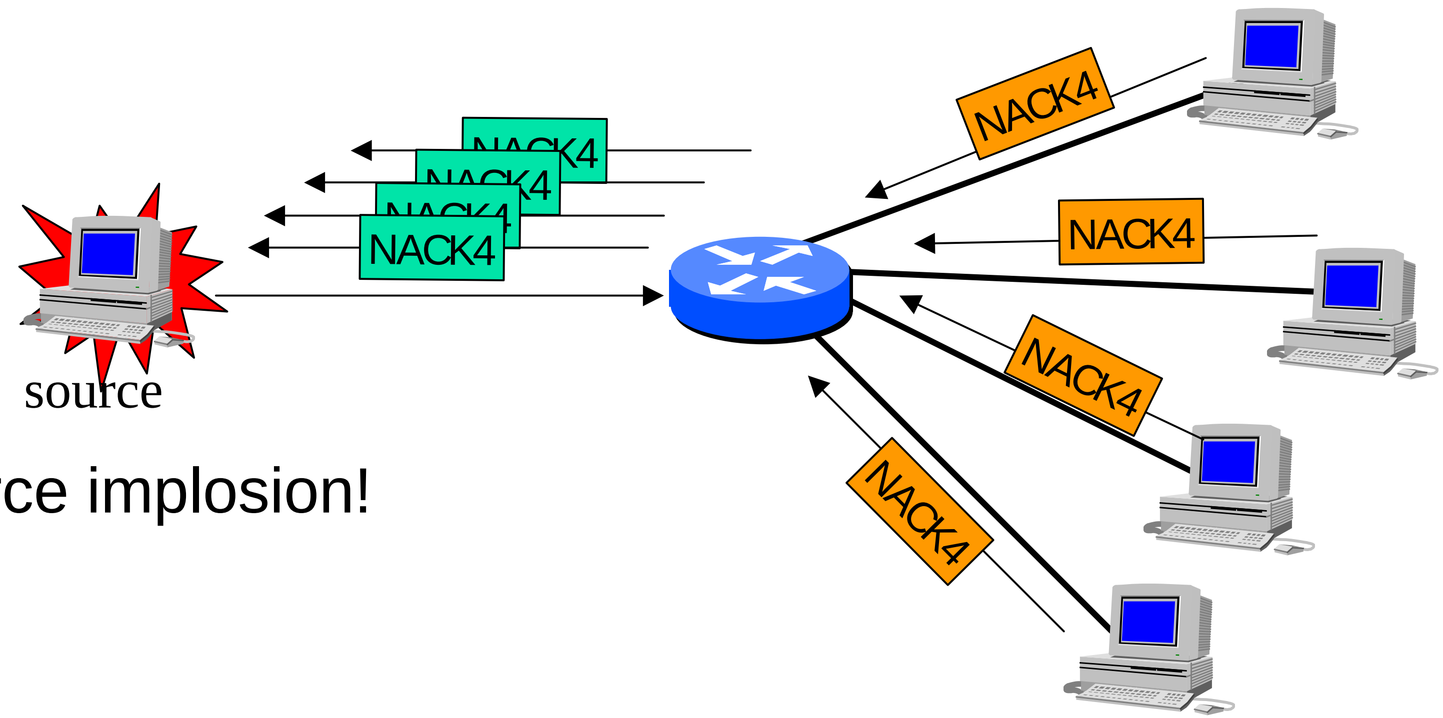
End-to-end reliability models

- Sender-reliable
  - Sender detects packet losses by gap in ACK sequence
  - Easy resource management

- Receiver-reliable
  - Receiver detect the packet losses and send NACK towards the source

Challenge: scalability (1)

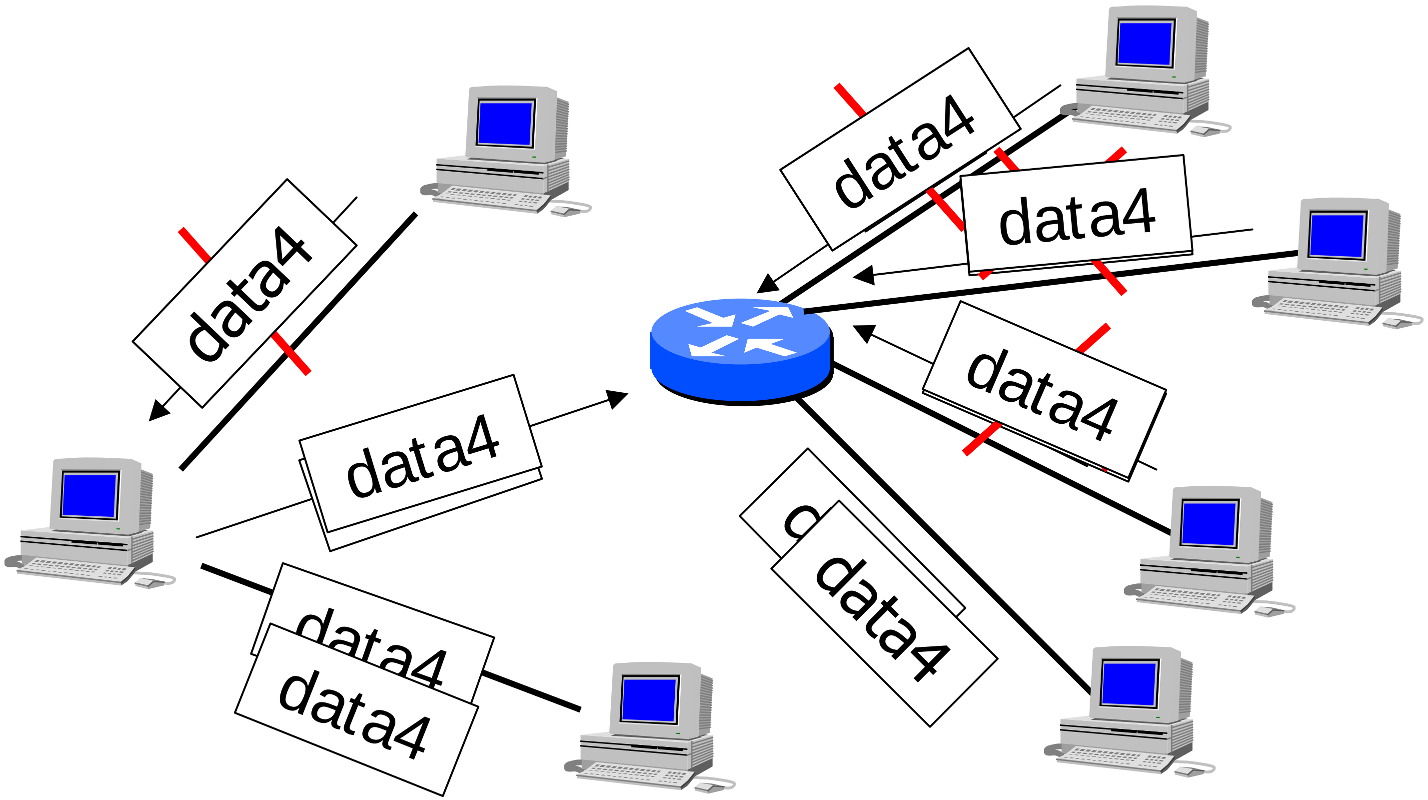
- many problems arise with 10,000 receivers...
- Problem 1: scalable control traffic
  - ACK every 2 packets (à la TCP)...oops, 10000ACKs / 2 pkt!
  - NAK (negative ack) only if failure... oops, if pkt is lost close to the source, 10000 NAKs!



source implosion!

Challenge: scalability (2)

- problem 2: scalable repairs/ exposure
  - receivers may receive several time the same packet



A piece of the solutions (1)

- solutions to problem 1: scalable control traffic
  - solution 1: feedback suppression at the receivers
    - each node picks a random backoff timer
    - send the NAK at timeout if loss not corrected
  - solution 2: proactive FEC (forward error correction)
    - send data plus additional FEC packets
    - any FEC packet can replace any lost data packet
  - solution 3: use a tree of intelligent routers/servers
    - use a tree for ACK aggregation and/or NAK suppression
    - PGM, ARM, DyRAM

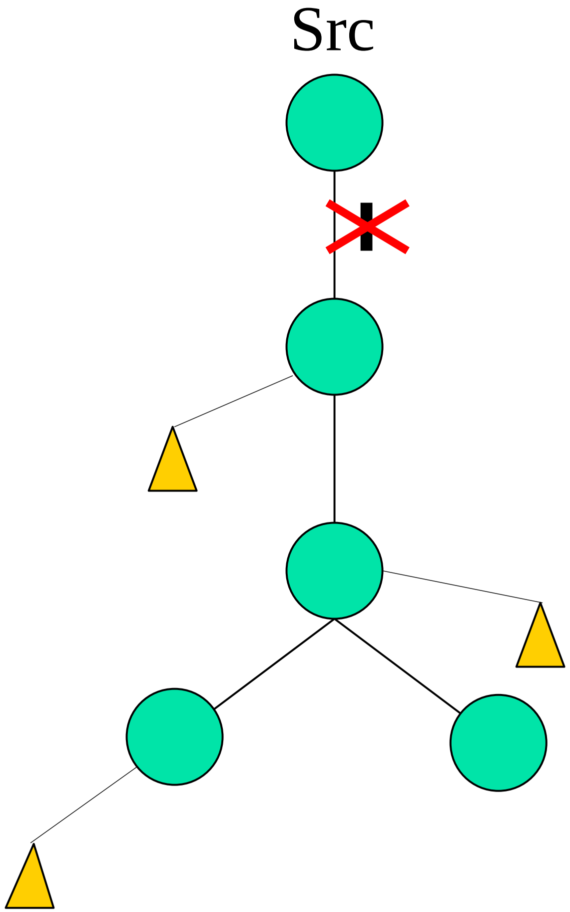
A piece of the solutions (2)

- solutions to problem 2: scalable repairs
  - solution 1: use TTL-scoped retransmissions
    - repair packets have limited scope
  - solution 2: use proactive/reactive FEC
    - proactive: always send data + FEC
    - reactive: in case of retransmission, send FEC
  - solution 3: use a tree of retransmission servers
    - a receiver can be a retransmission server if he has the requested data

Scalable Reliable Multicast
Floyd et al., 1995

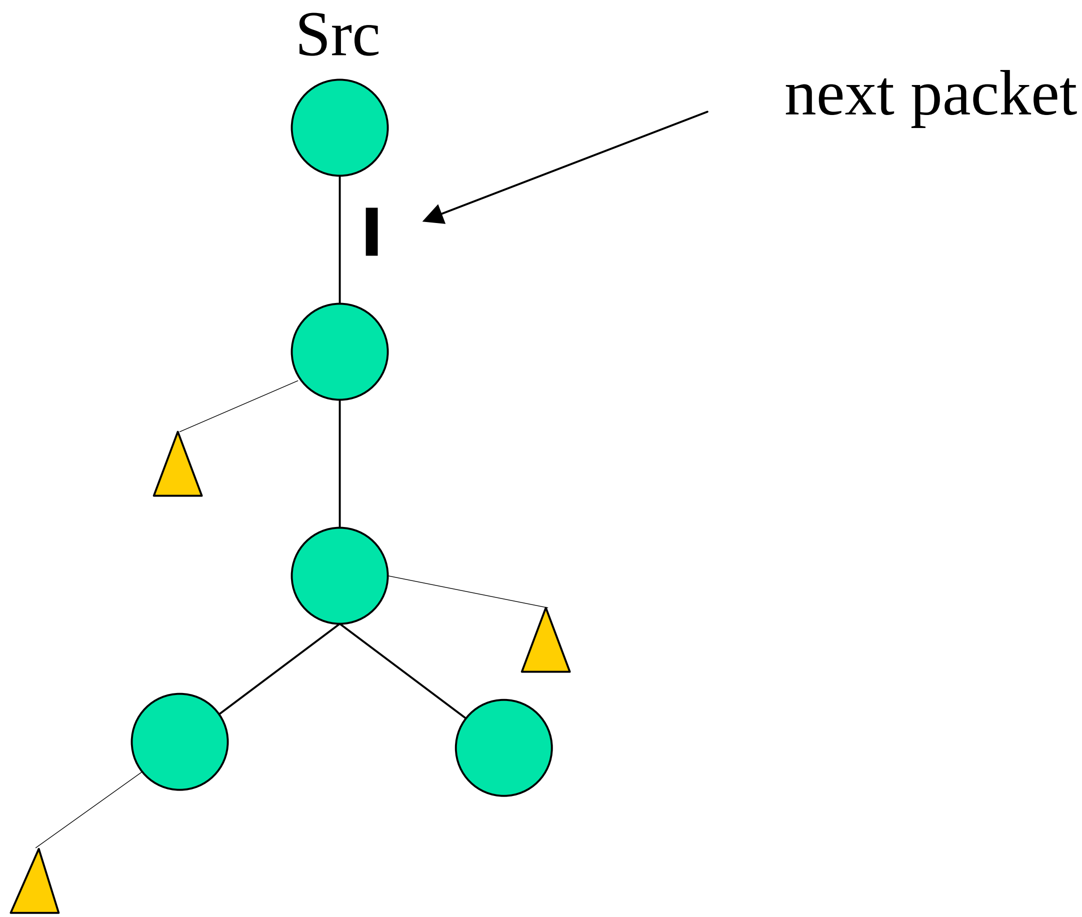
- Receiver-reliable, NACK-based
- NACK local suppression
  - Delay before sending
  - Based on RTT estimation
  - Deterministic + Stochastic
- Every member may multicast NACK or retransmission
- Periodic session messages
  - Sequence number: detection of loss
  - Estimation of distance matrix among members

SRM Request Suppression

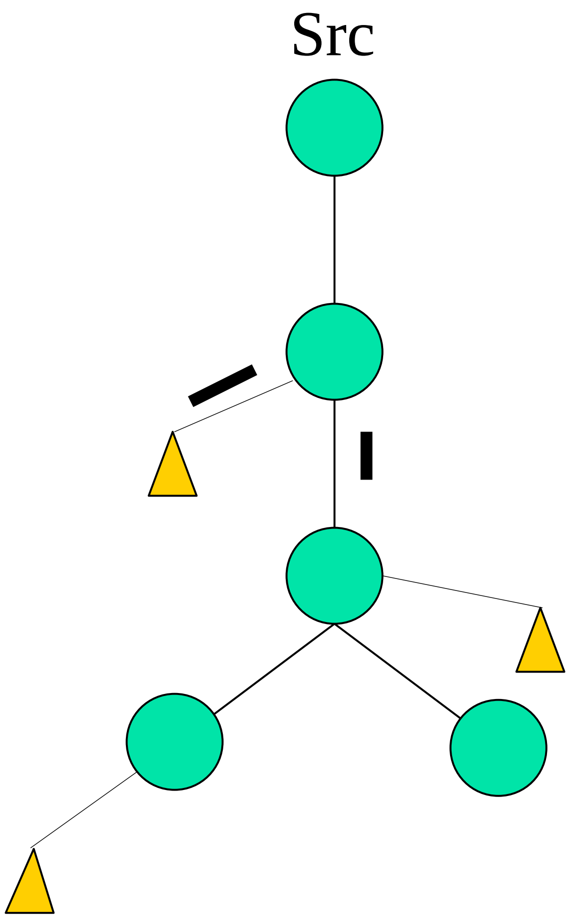


from Haobo Yu , Christos Papadopoulos

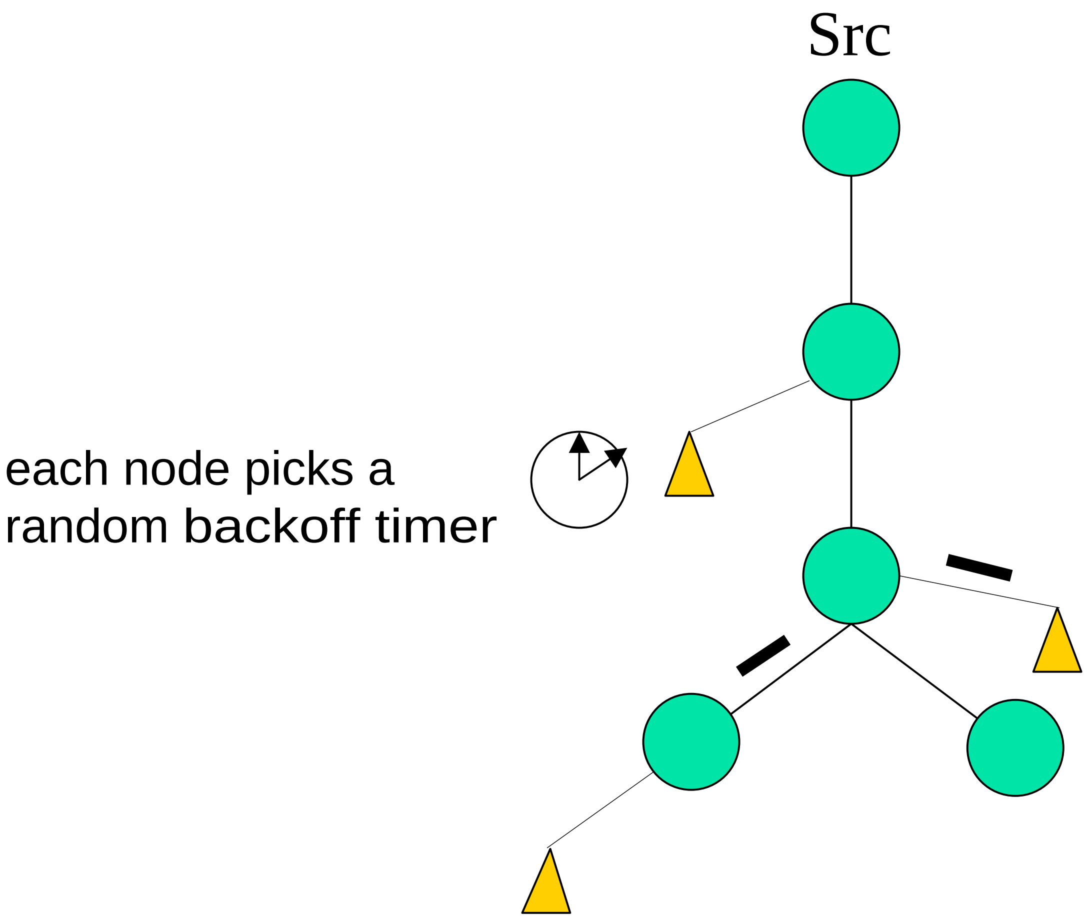
SRM Request Suppression



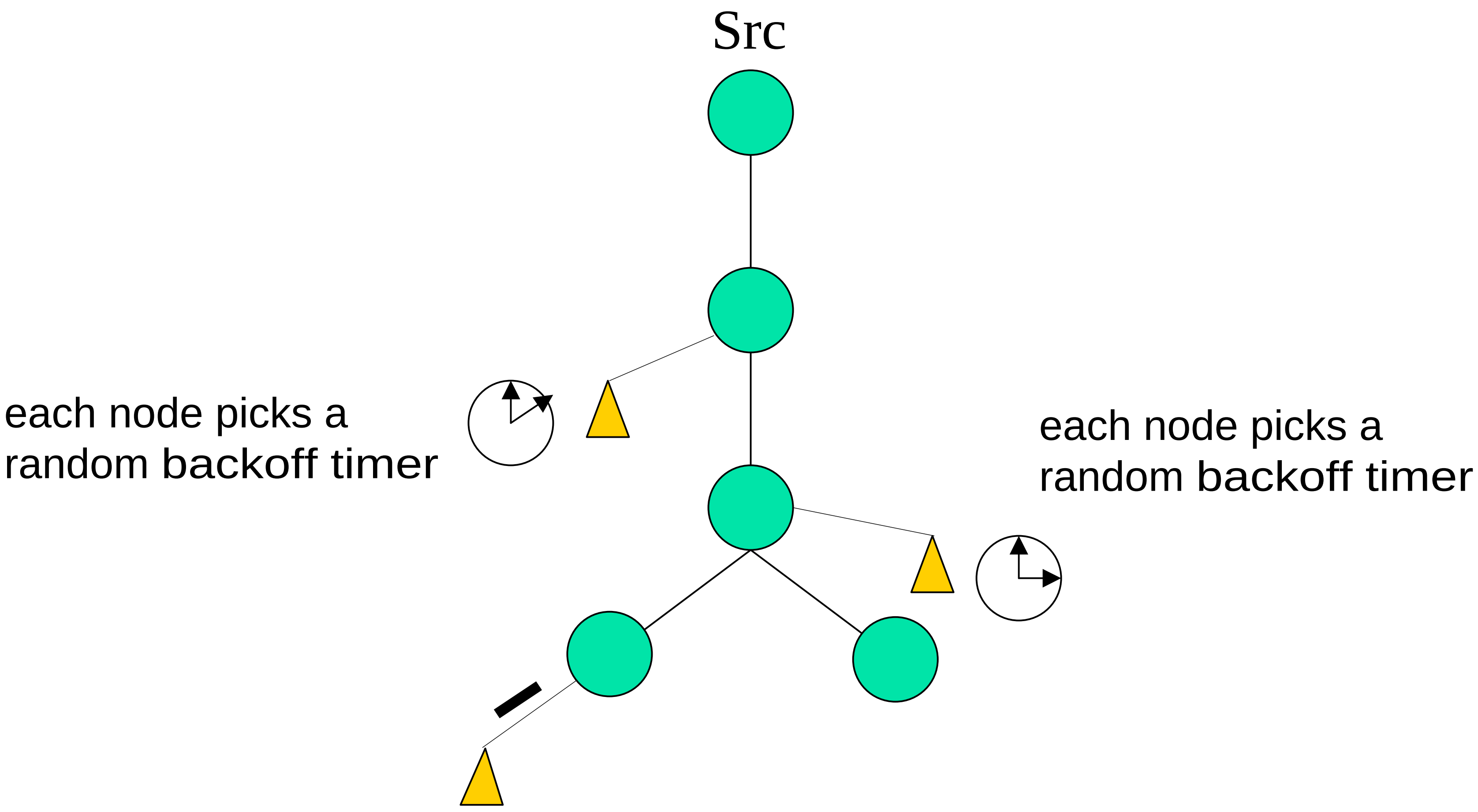
SRM Request Suppression



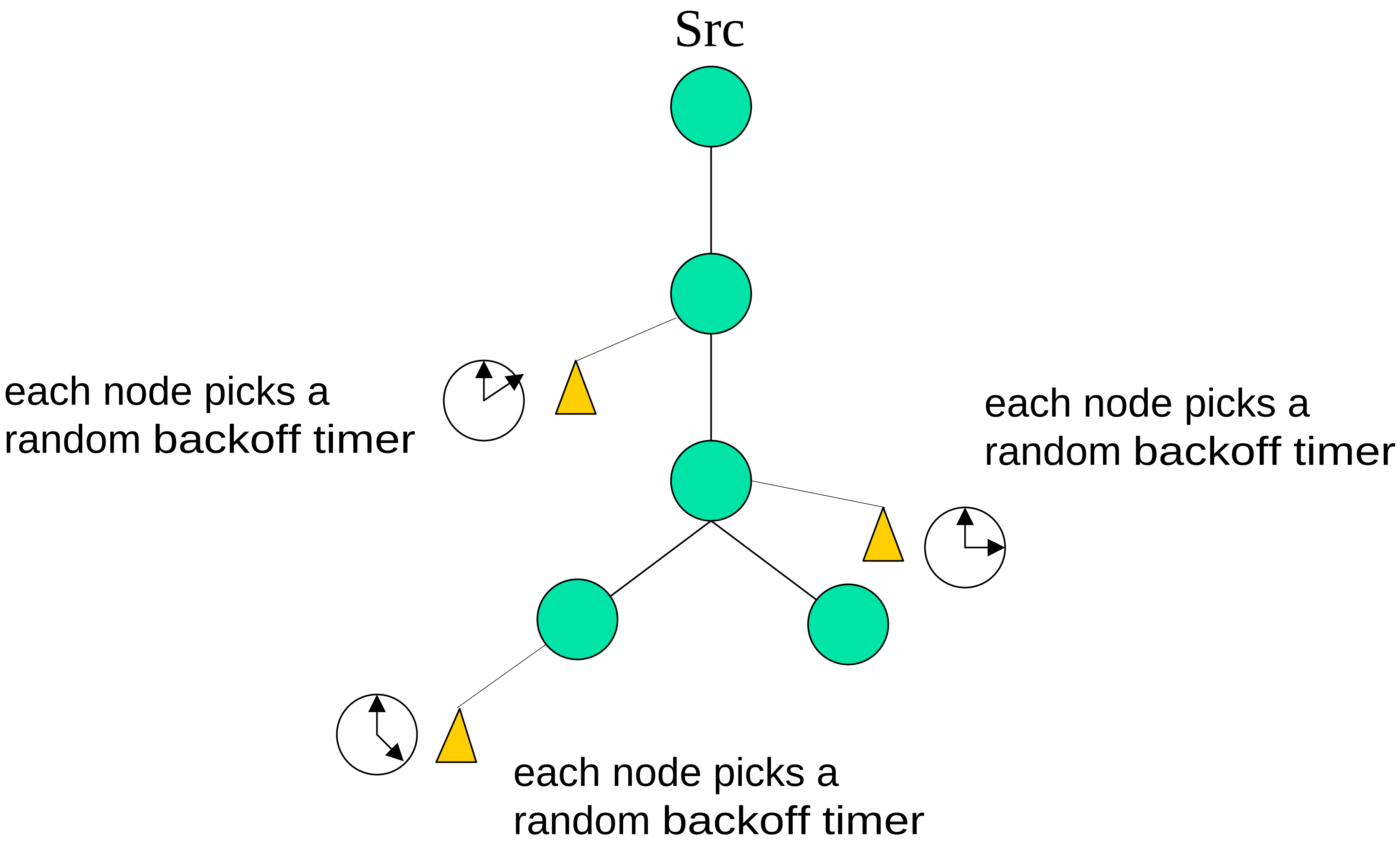
SRM Request Suppression



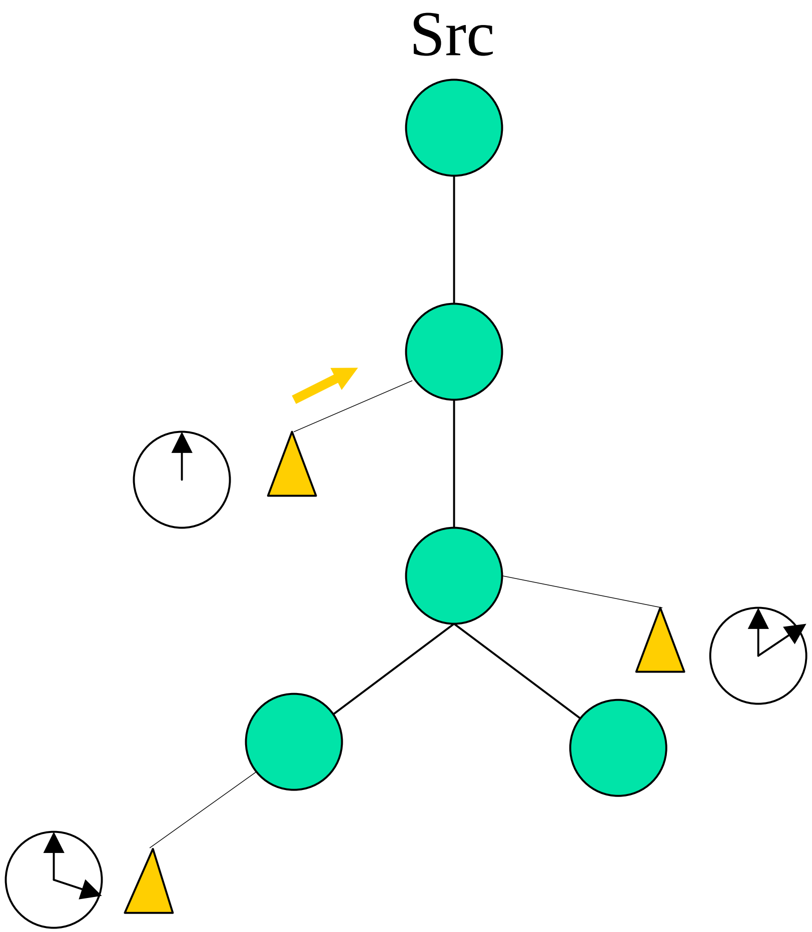
SRM Request Suppression



SRM Request Suppression

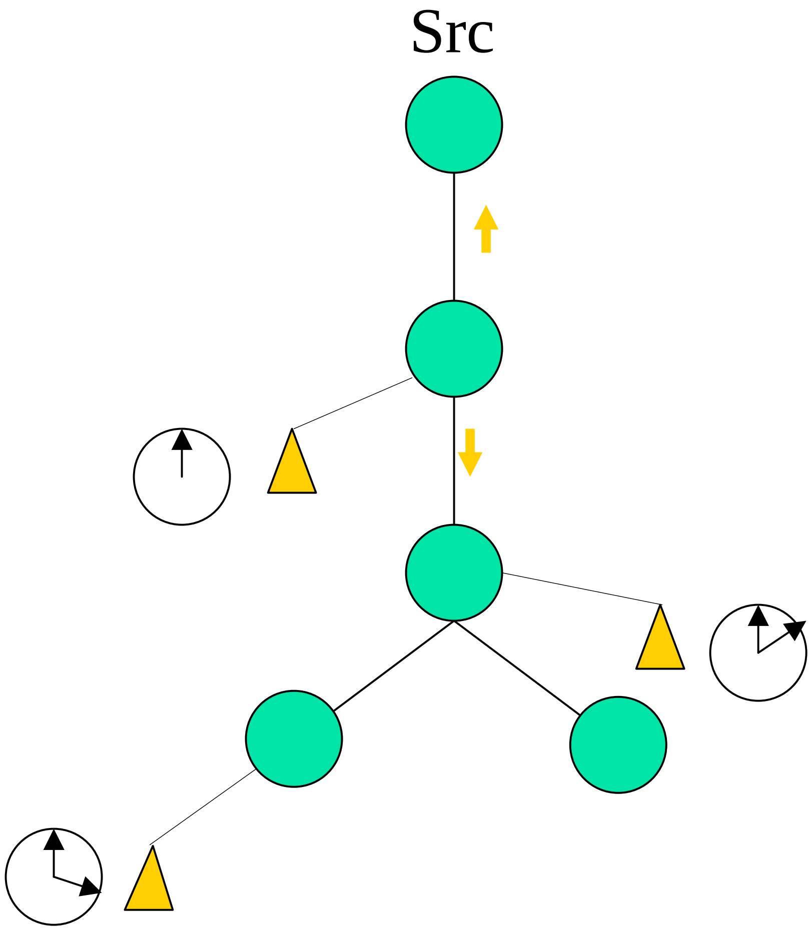


SRM Request Suppression



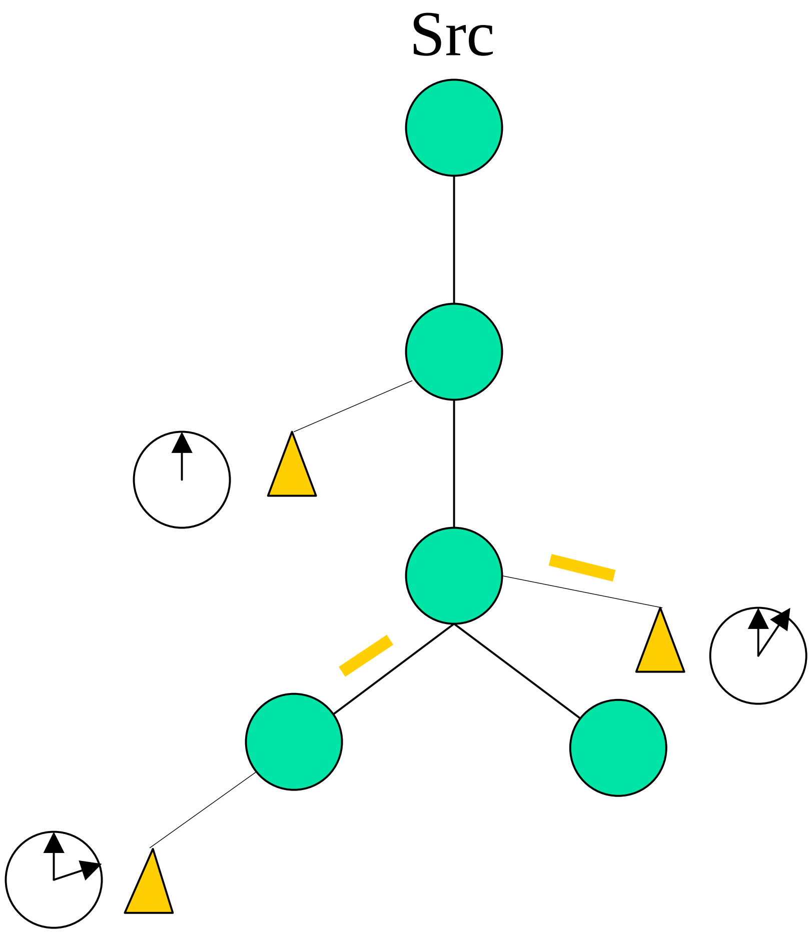
from Haobo Yu , Christos Papadopoulos

SRM Request Suppression



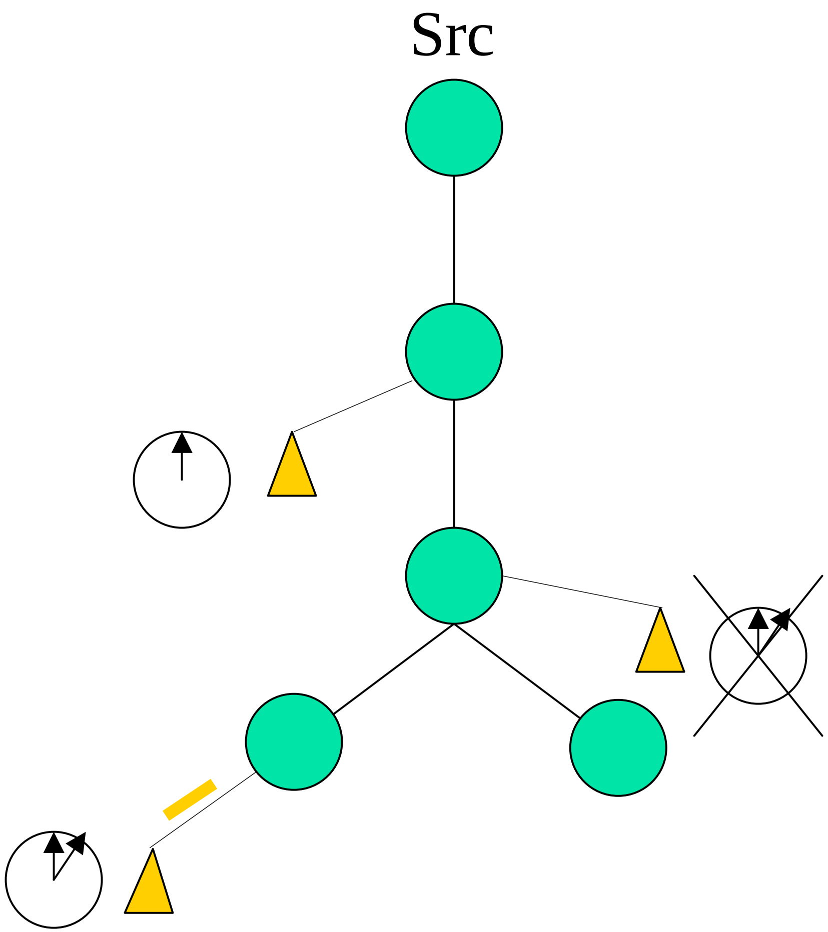
from Haobo Yu , Christos Papadopoulos

SRM Request Suppression



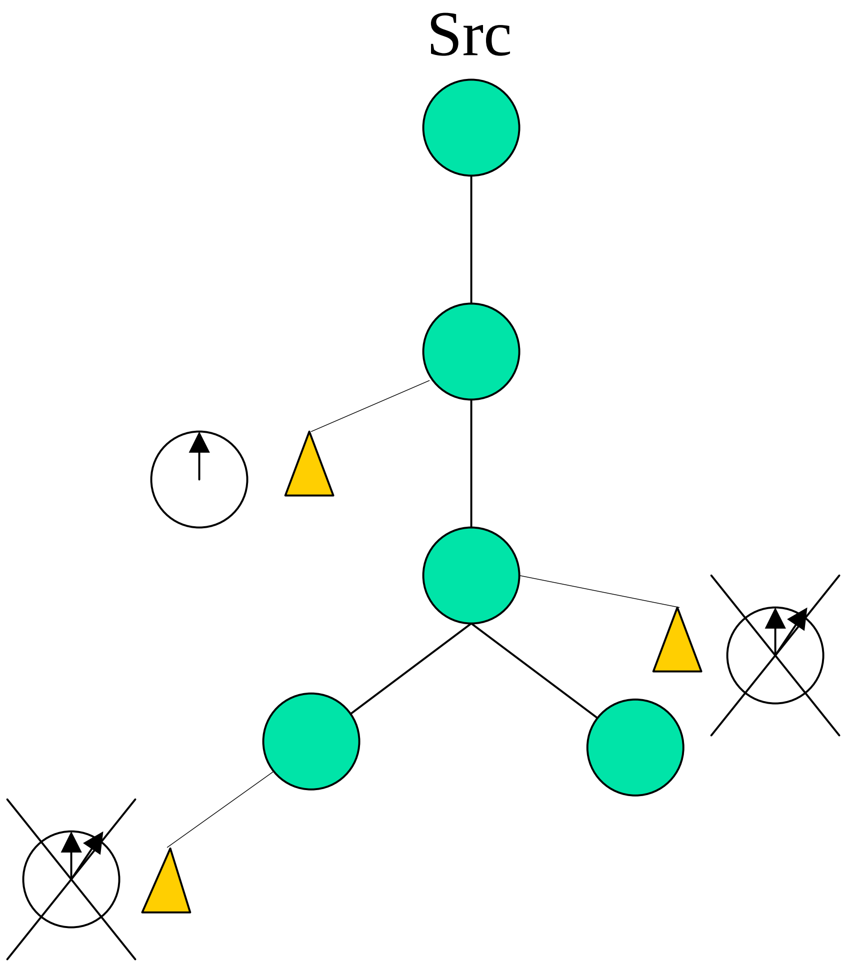
from Haobo Yu , Christos Papadopoulos

SRM Request Suppression



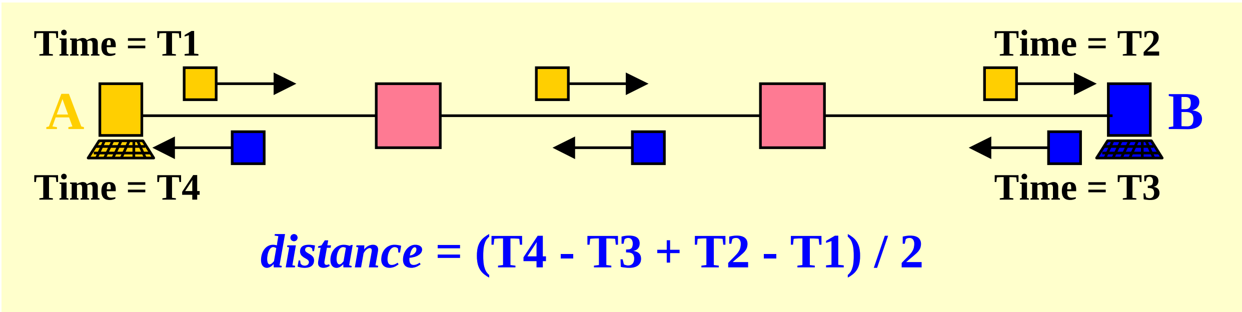
from Haobo Yu , Christos Papadopoulos

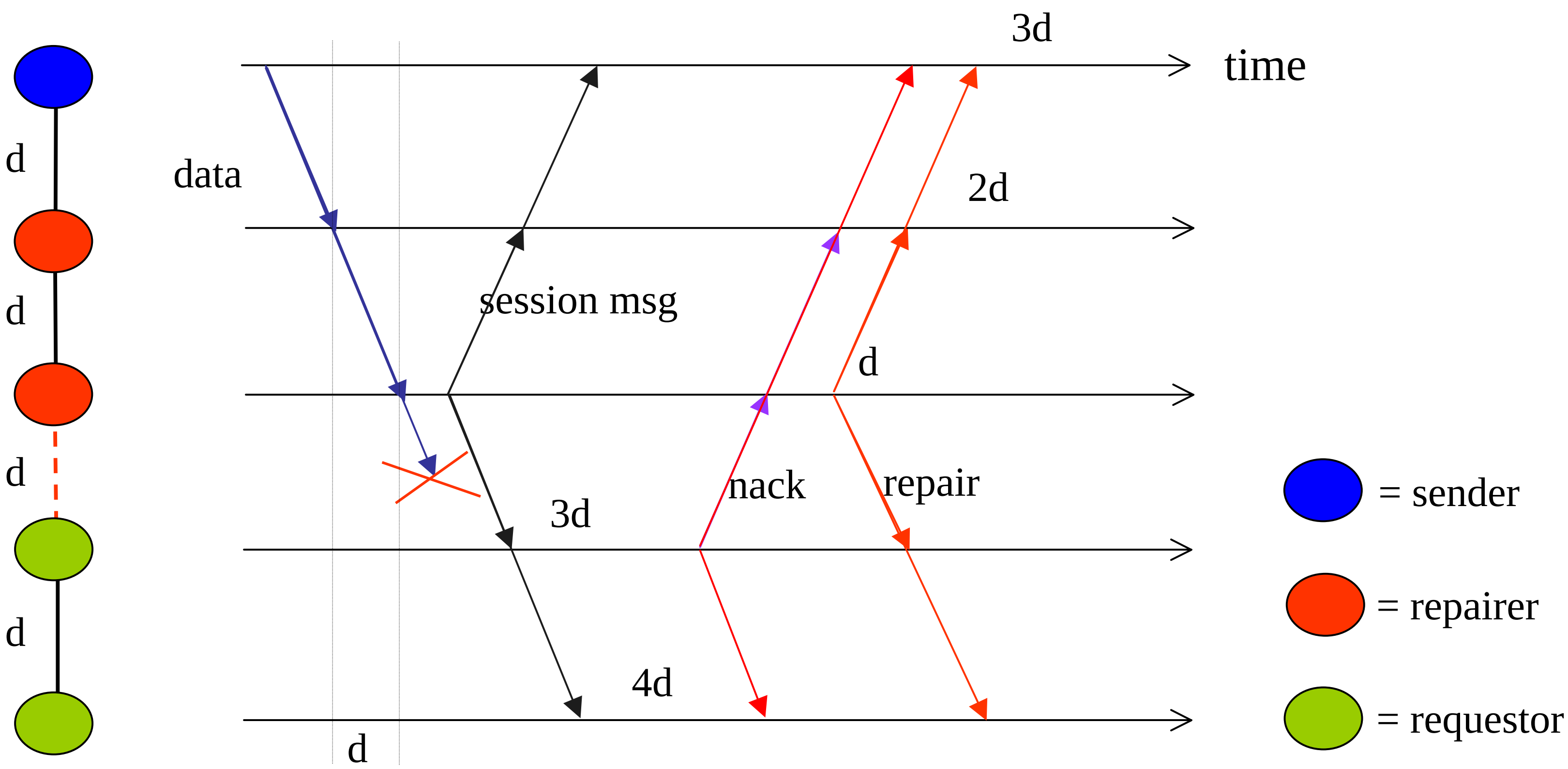
SRM Request Suppression



from Haobo Yu , Christos Papadopoulos

Deterministic Suppression



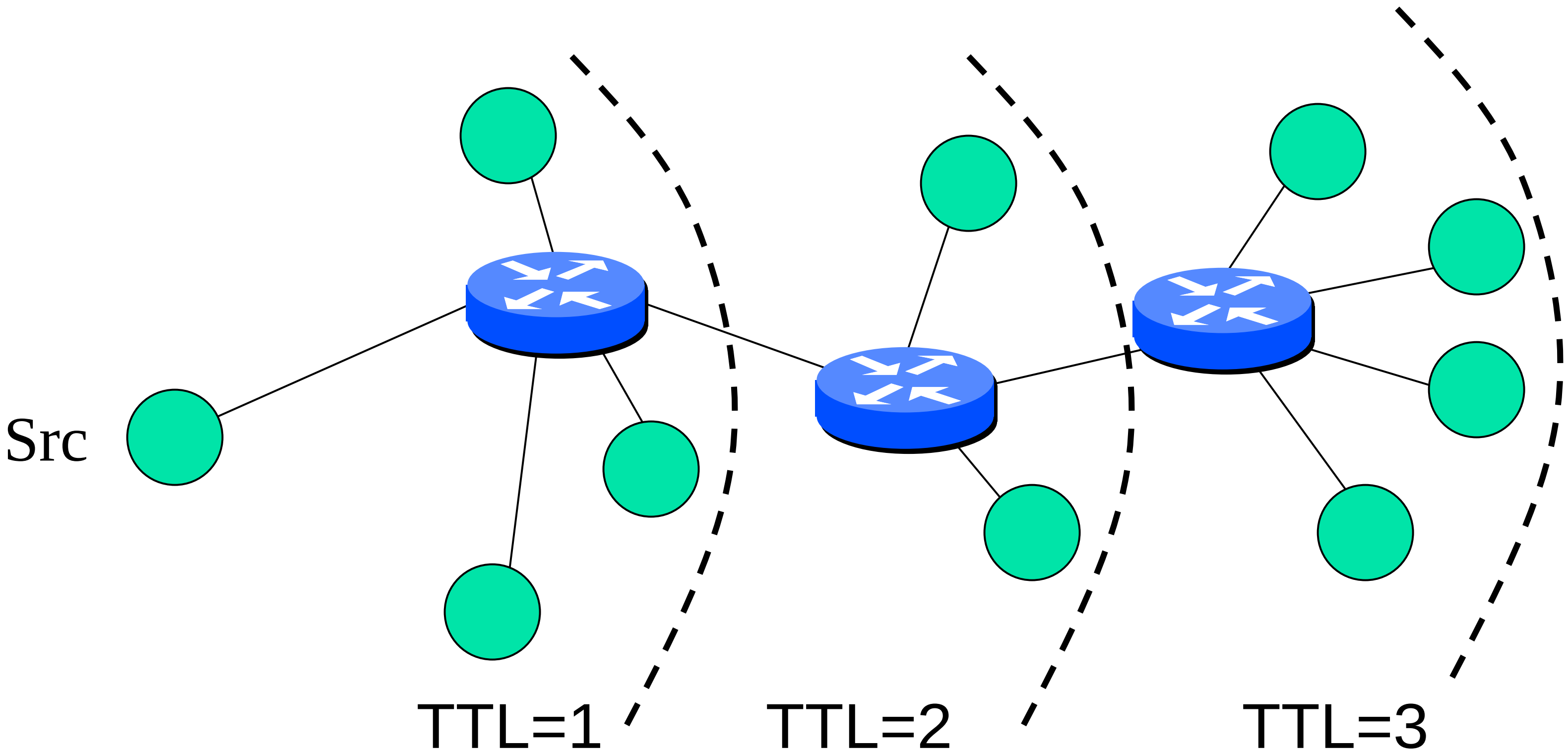


$$\text{Delay} = C_1 \times d_{S,R}$$

from Haobo Yu , Christos Papadopoulos

Simple TTL-scoped of repairs

- use the TTL field of IP packets to limit the scope of the repair packet



Summary: reliability problems

- What is the problem of loss recovery?
  - feedback (ACK or NACK) implosion
    - ACK/NACK aggregation based on timers are approximative!
  - replies/repairs duplications
    - TTL-scoped retransmissions are approximative!
  - difficult adaptability to dynamic membership changes

- Design goals
  - reduces the feedback traffic
  - reduces recovery latencies
  - improves recovery isolation

Current IETF standardization work

- "One size does not fit all"
  - "requirements" x "conditions/problems" matrix is too large for a single solution!!!
  - define Building Blocks (BB)
    - logical, reusable component
    - used by the PI
    - example: Forward Error Correction (FEC) BB
  - define several classes of protocols for reliable multicast: Protocol Instantiation (PI)
    - non reusable
    - glue between the various BBs
    - provides an operational solution

IETF standardization work...(cont')

- Flat NORM
  - for small to medium sized groups
  - simplicity, uses NAK

- Hierarchical TRACK
  - for medium sized to large groups
  - requires tree building (manual/automatic

Part II

Introducing reliability

ACK/NACK end-to-end solutions

FEC-based solutions

Layered solutions

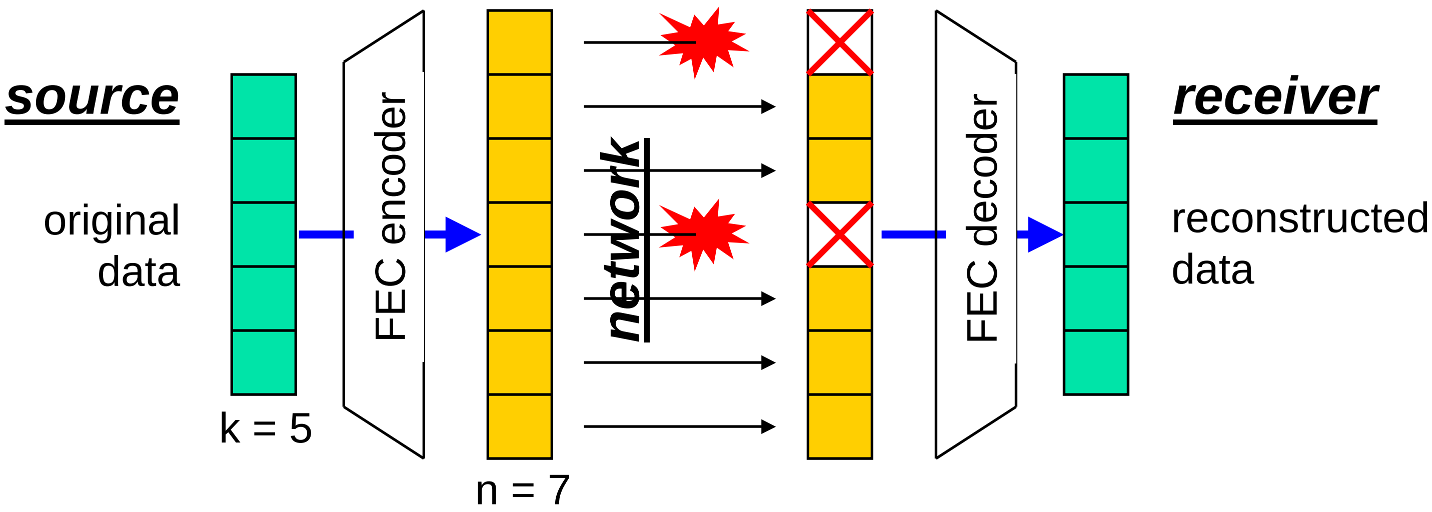
Router-assisted solutions

FEC (Forward Error Correction)

- add some redundancy to the data flow
- reliable multicast is almost impossible without FEC !
  - a single FEC packet can recover different losses at different receivers $\Rightarrow$ improves scalability
- we only consider packet-based erasure channels (like the Internet)
  - packets are either perfectly received or lost
  - mimics the effects of congested routers
  - FEC operates on a packet basis

FEC... (cont')

- more precisely (MDS FEC code)...
  - sender: FEC (k, n)
    - for k original data symbols, add n-k FEC symbols
      $\Rightarrow$ total of n symbols (or packets) sent
  - receiver:
    - as soon as it receives any k symbols out of n, a receiver can reconstruct the original k symbols
    - a FEC code with this property is called "MDS"



FEC classification

- [FECinfo02] provides a classification based on the (k, n) parameters
  - small block FEC codes (small k)

    Reed-Solomon (based on Vandermonde matrices, or Cauchy matrices), Reed-Muller…
  - large block FEC codes (large k)

    LDPC, Tornado

    belong to the "codes on graph" category
  - expandable FEC codes (large k and n)

    LT

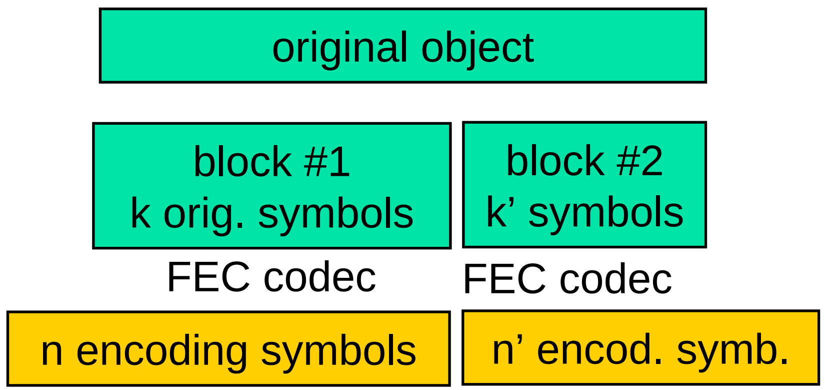
FEC classification… (cont’)

- other codes exist but are
  - either lossy codes (ok for video/audio transmission)
  - or dedicated to bit stream transmissions over noisy channels

  - not for us!

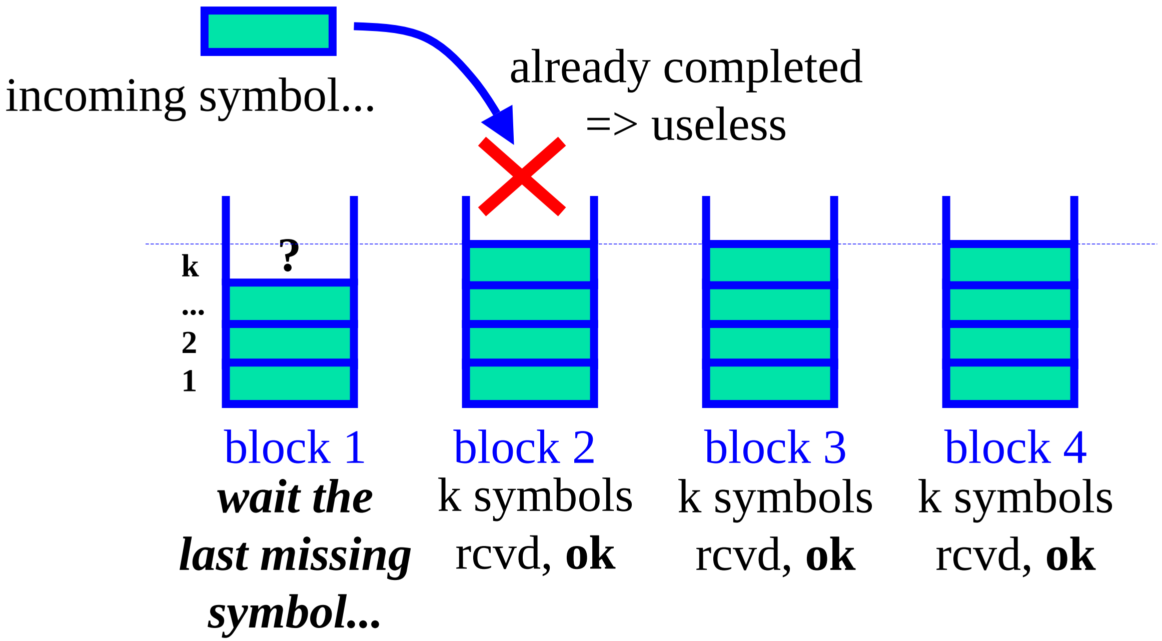
Small block FEC codes

- e.g. Reed-Solomon codes [Rizzo97]
- this is an “MDS code”
  - any k out of n is sufficient to build original pkts
- the k parameter is < a few tens for computational reasons
  - split large data objects into several blocks
  - limits correction capability of a FEC symbol
  - limits the global efficiency



Small block FEC codes… (cont’)

- an example of problem generated by a small k



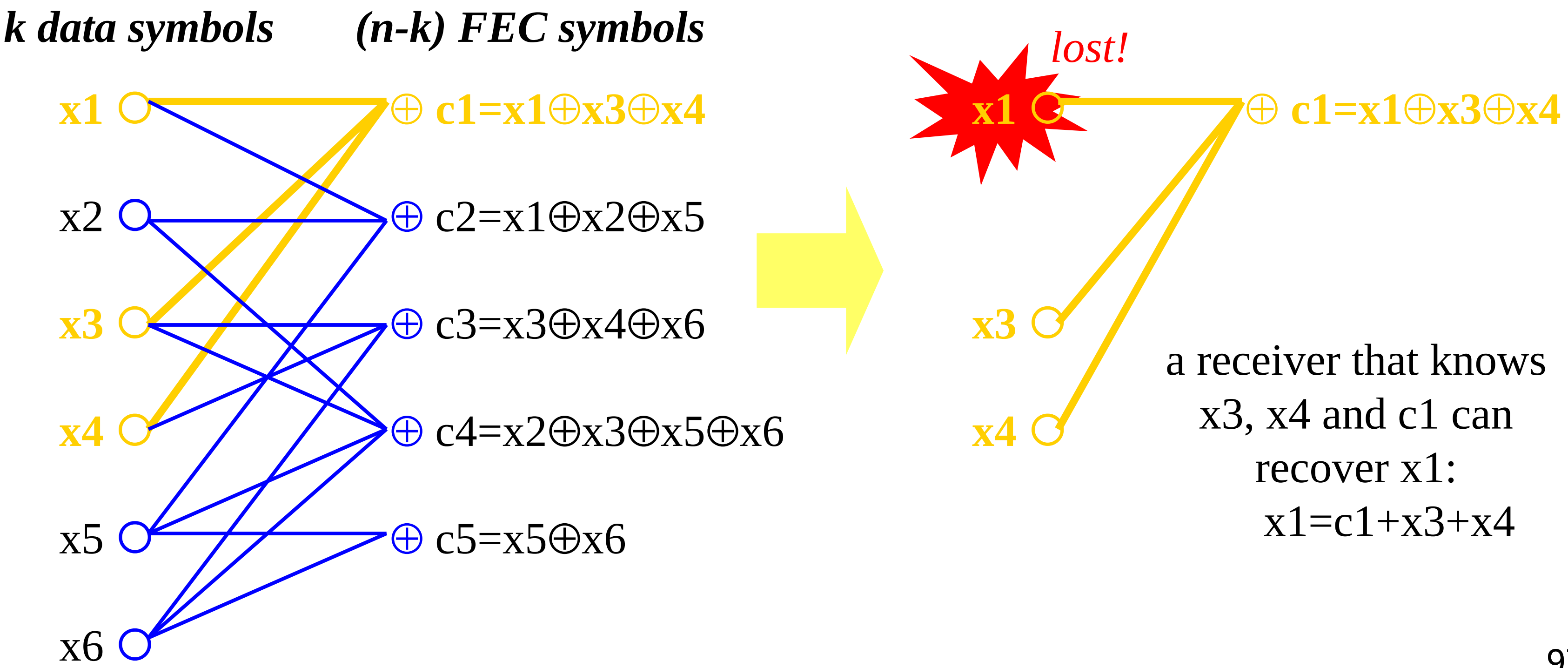
- limited number of n-k FEC symbols created
  - ⇒ can lead to packet duplications
- high quality open-source implementation available

Large block FEC codes

- e.g. LDPC and Tornado codes
- (k,n) with a **very large k**
- but n is limited in practice (e.g. n = 2k)
- decoding requires $(1+\varepsilon)k$, i.e. a bit more than k symbols
  - ε is around %10 (for the best codes) to 40%
  - this is not an MDS code
- high-speed encoding/decoding

Large block FEC codes… (cont’)

- an example: LDPC code
  - based on XOR operations (⊕)
  - uses bipartite graphs between source and FEC symbols
  - iterative decoding



Expandable FEC codes

- expandable FEC codes
  - no predefined limit to the n parameter

    consequence: FEC symbols can be produced on-demand, no symbol duplication
  - no technical information ever published (as far as I know)
  - patents owned by Digital Fountain

Use of FEC in RM protocols

- what FEC for what reliable multicast protocol…

	NORM	TRACK	ALC
small block code	YES	YES	far from the best solution
large block code	not the best solution	not the best solution	YES
expandable block code	not the best solution	not the best solution	YES YES

Part II

Introducing reliability

ACK/NACK end-to-end solutions

FEC-based solutions

Layered solutions

Router-assisted solutions

ALC: Asynchronous Layered Coding

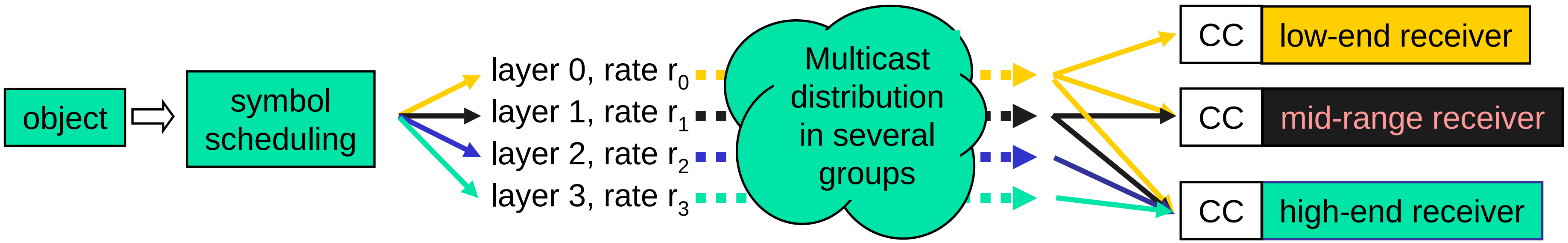
- ALC/LCT standard
  - one the three reliable multicast protocols being standardized at the RMT IETF working group
  - **RFC 3450 up to RFC 3453**
  - offers unlimited scalability (no feedback)
  - supports receiver heterogeneity
  - supports ``push'', ``on-demand'' and ``streaming'' delivery modes
  - suited to the distribution of popular content

ALC… (cont')

- Building blocks required by ALC
  - LCT (glue + header definition)
  - FEC (any FEC code)
  - layered congestion control (e.g. WEBRC)
  - security (e.g. TESLA authentication)

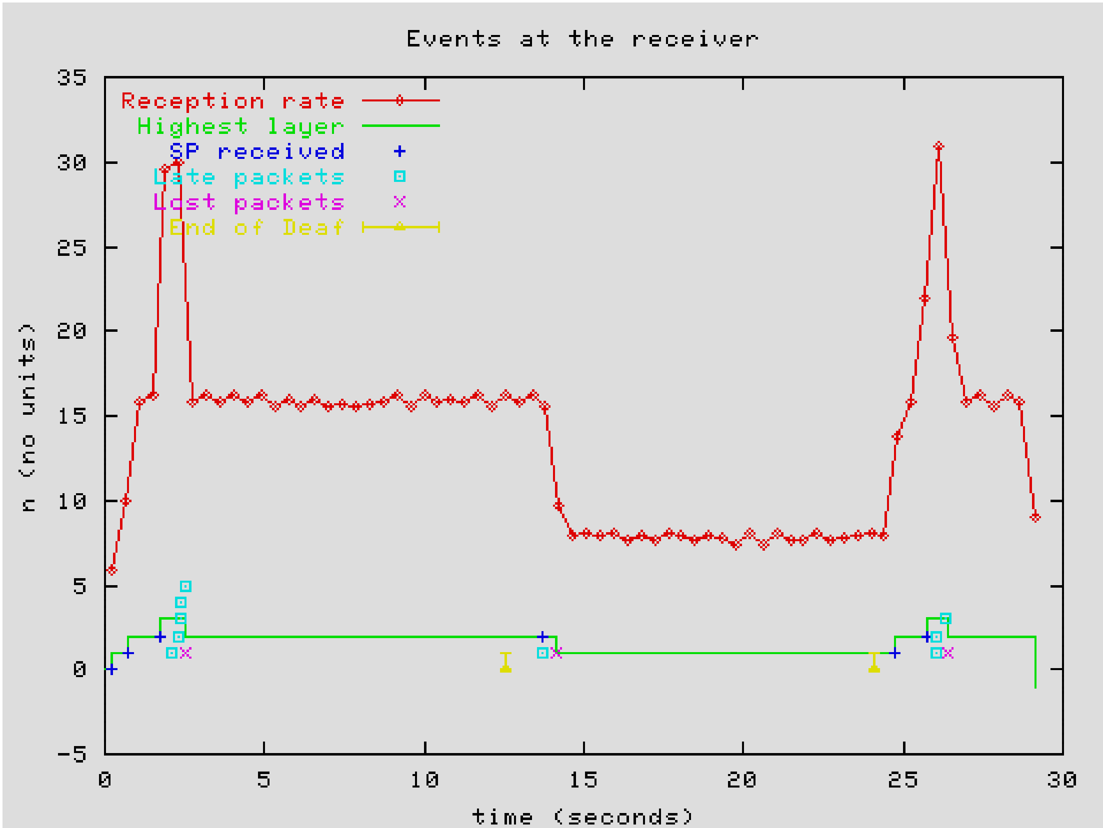
ALC… (cont’)

- How does it work?
  - multi-rate transmissions, over several multicast groups, one per layer
  - the congestion control BB (e.g. RLC) tells a receiver when to add or drop a layer



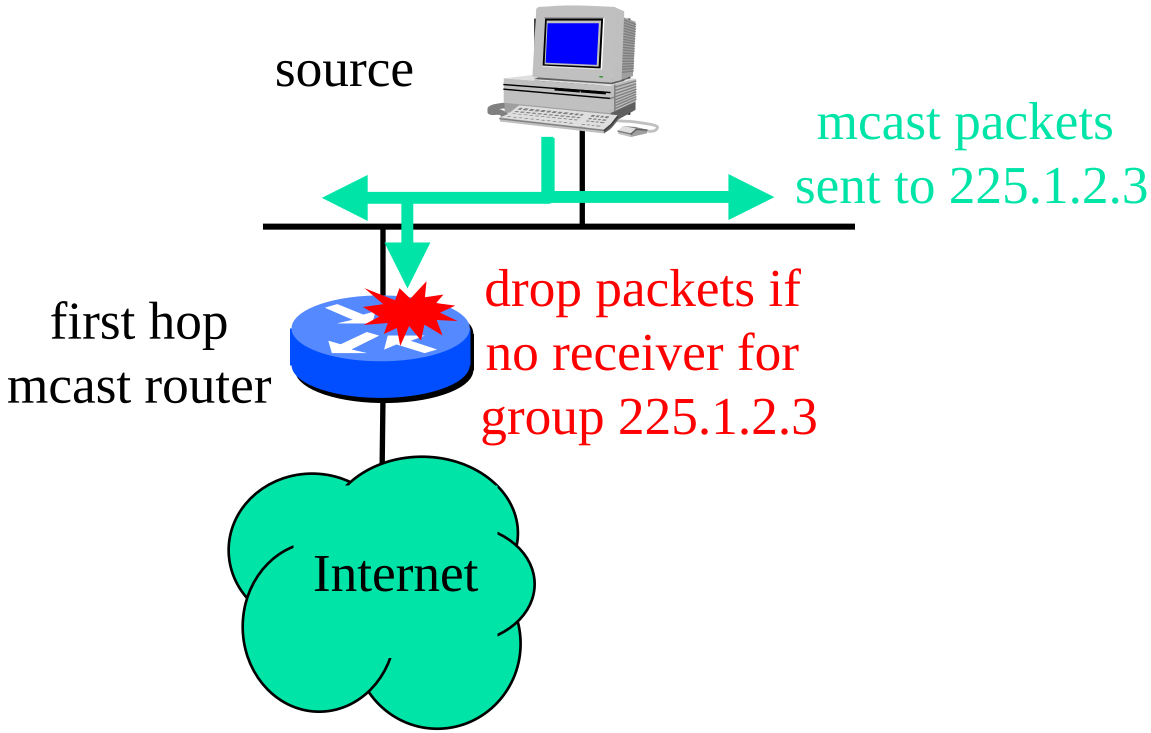
ALC… (cont’)

- number of layers received is dynamic
  - depends on losses experienced
  - symbol scheduling must take it into account!
- example



ALC… (cont’)

- How does it work… (cont’)
  - sending to a multicast group with no receiver attached is not a problem…
  - packets are dropped by the first hop router !



The ALC PI … (cont')

- How does it work... (cont')
  - mix **randomly** all the data+FEC packets and send them on the various layers
  - required to counter the random losses and random layer addition/removal
  - other more intelligent organizations are possible (and can avoid duplications) but only work in an ideal world... (e.g. a LAN)
    - in practice losses, layer dynamic, layer de-synchronization lead to catastrophic performances…

The ALC PI . . . (cont')

- a transmission approach completely different from NORM/TRACK
  - file transmission with NORM/TRACK

source recvs:								NAK(2)						NAK(4)			
source sends:	0	1	2	3	4	5	6	FEC1	7	8	9	10	11	FEC2	12	13	14	END

  - file transmission with ALC (just an example!)

	Layer 3	F3	F12	F0	F1	F4	F11	F6	F5	F14	F7	F8	F2	F9	F10	F13	END
	Layer 2	2	4	10	8	5	9	11	14	7	3	0	12	1	6	13	END
	Layer 1	F12	F9	F2	F1	F10	F7	F6	F4	F13	F3	F5	F11	F14	F0	F8	END
source sends:	Layer 0	11	2	4	9	0	13	10	7	8	1	3	14	5	12	6	END

time →

What is ALC really good at?

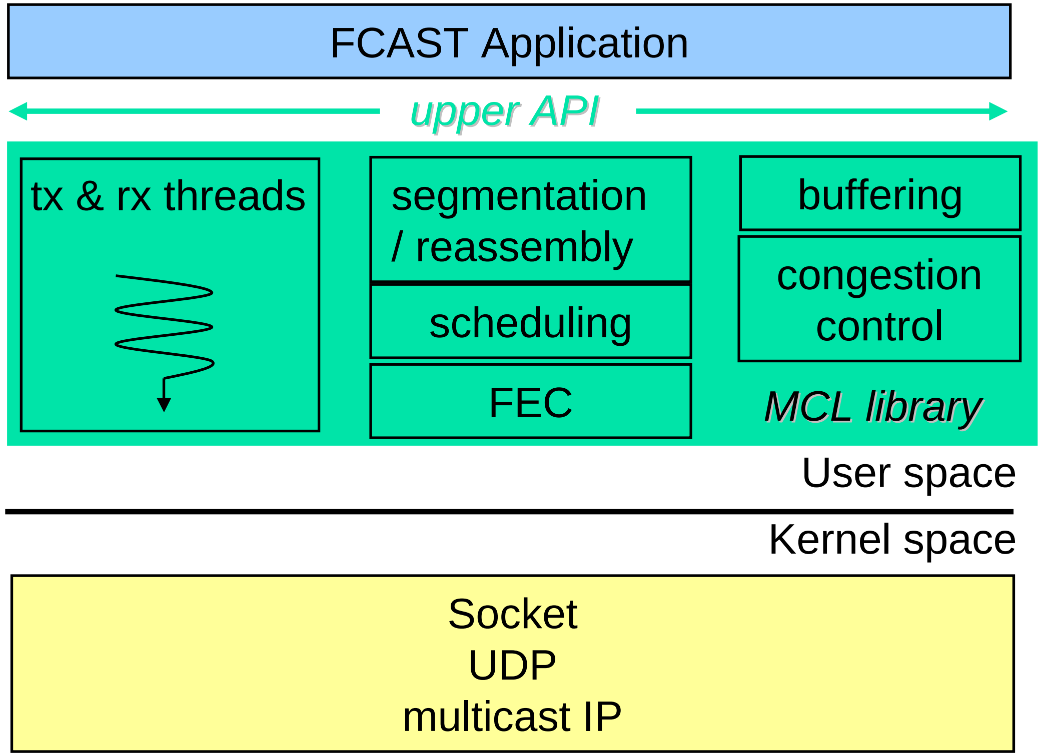
- On-demand delivery mode
  - **yes**, this is the only RM solution supporting it!
- Streaming delivery mode
  - **yes**, partial reliability is possible too
- Push delivery mode
  - **no** for the general case, **yes** when there is no (or a very small) feedback channel (e.g. satellite)
- Scalability
  - **yes**, this is the only RM solution supporting it
- Heterogeneity
  - **yes**, this is the only RM solution supporting it

What is ALC really good at…(cont')

- Robustness
  - **yes**, reception can be stopped and restarted several times without any problem
  - a source is never impacted by the receiver behavior, neither are other receivers (in general)

ALC demo with MCL/FCAST

- MCL: a library implementing ALC/LCT/layered congestion control
  - OpenSource/GPL; for linux/solaris/windows
  - http://www.inrialpes.fr/planete/people/roca/mcl



ALC demo... (cont')

- FCAST, a file transfer application built on top of MCL
  - add a trailer with several meta-data
    - Content_base: path to the file
    - Content-location: file name
    - Content-length: length of file

file data	trailer	trailer_length	checksum

  - multi-slices mode (useful with large files)

file - slice 0	trailer	trailer_length	checksum
file - slice 1	trailer	trailer_length	checksum
file - slice 2	trailer	trailer_length	checksum
file - slice 3	trailer	trailer_length	checksum

Part II

Introducing reliability

ACK/NACK end-to-end solutions

FEC-based solutions

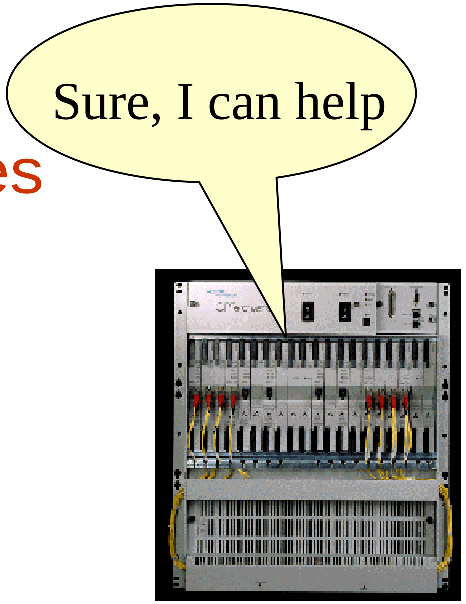
Layered solutions

Router-assisted solutions

Additional functions in routers

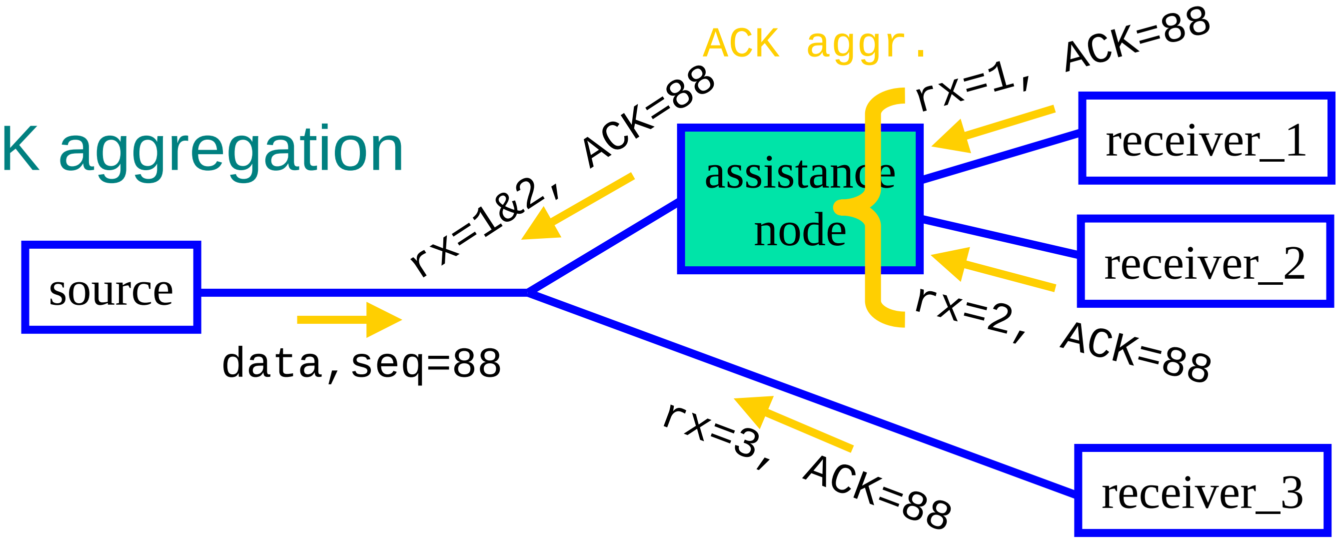
- Traditional
  - end-to-end retransmission schemes
  - scoped retransmission with the TTL fields
  - receiver-based local NACK suppression

- Router-assisted contributions
  - feedback aggregation
  - cache of data to allow local recoveries
  - subcast
  - early lost packet detection
  - ...

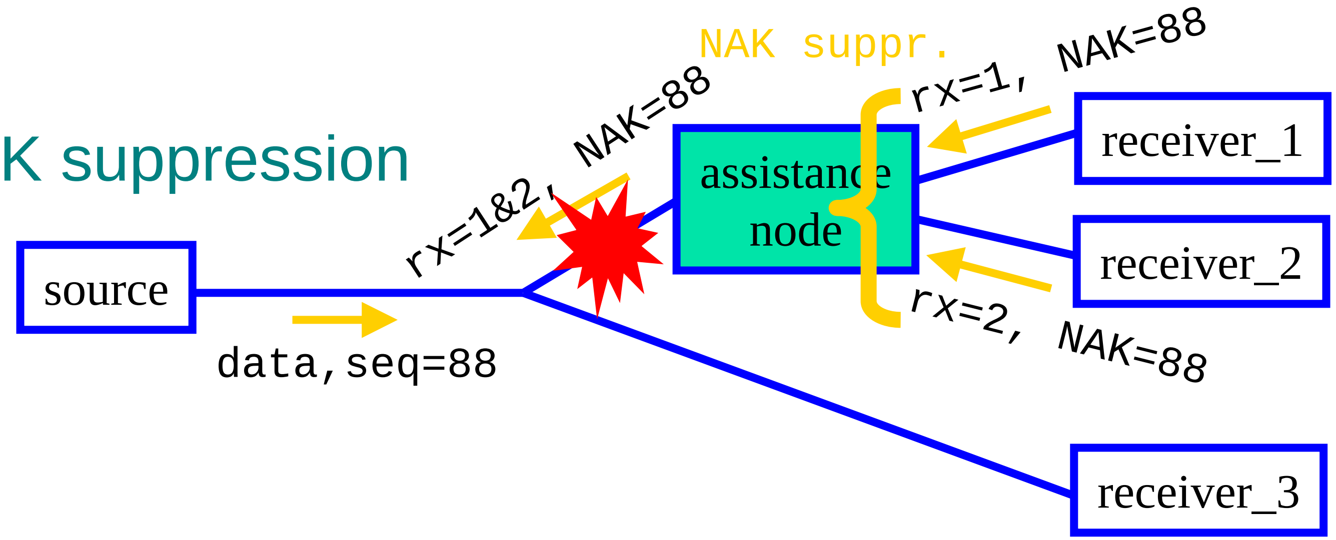


Feedback aggregation with router assistance

- ACK aggregation

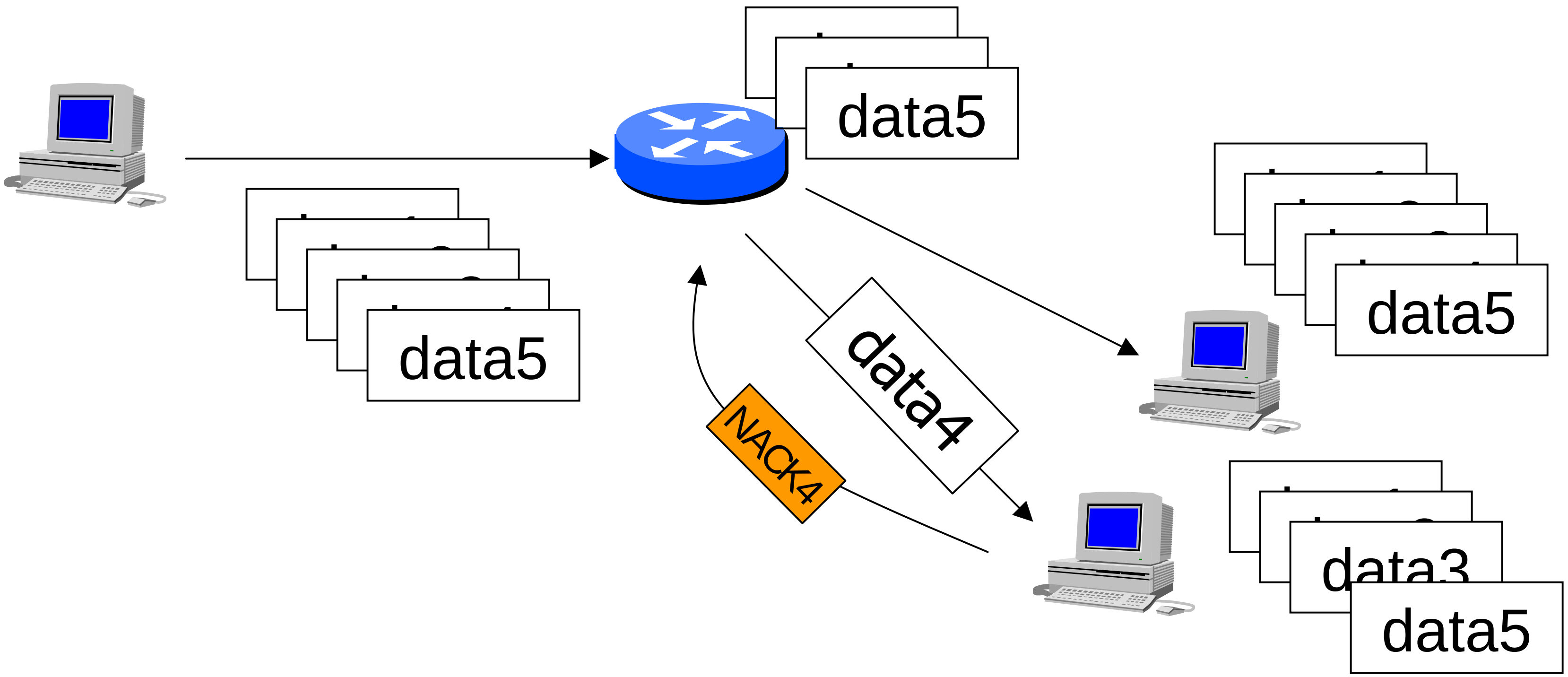


- NAK suppression



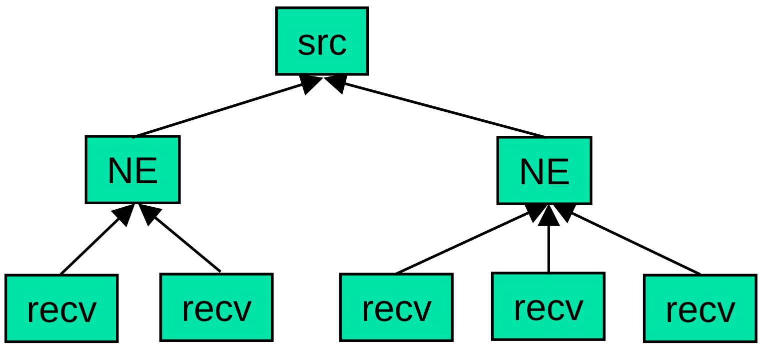
Local recovery with router assistance

- routers perform cache of data packets
- repair packets are sent by routers, when available



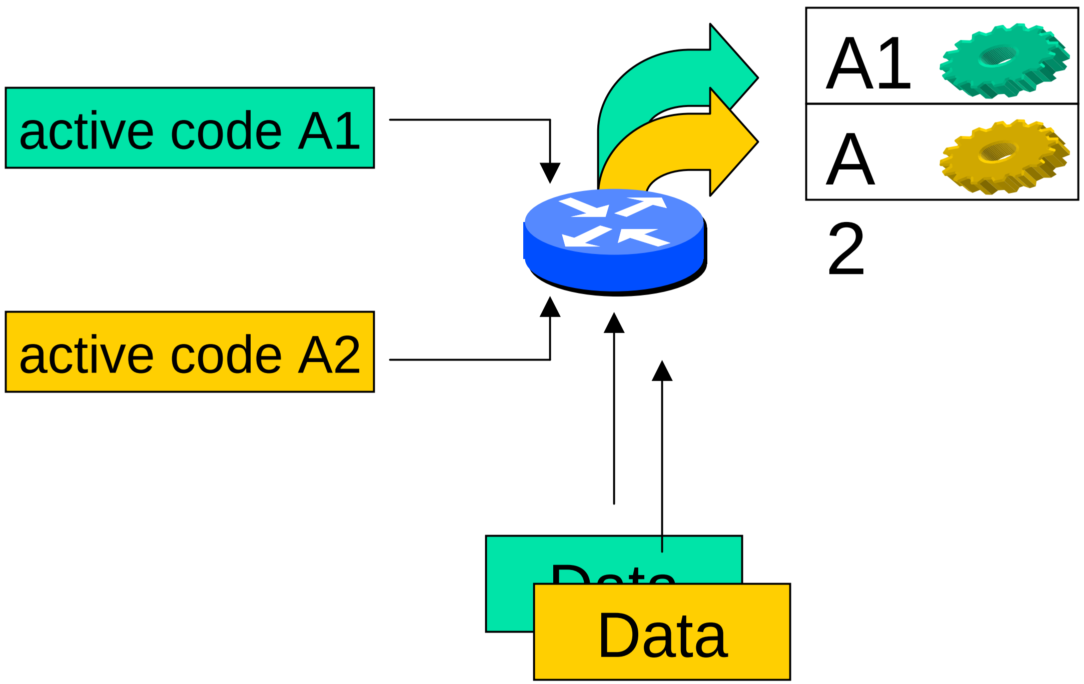
PGM Speakman et al, 1999

- CISCO & TIBCO (pragmatic multicast):
  - build a tree of NE (Network Elements) (server or router) that perform:
    - ACK aggregation along the tree
    - NACK suppression along the tree
    - localized retransmission in a subset of the tree
    - retransmission (if data is cached)
  - FEC possible for increased scalability/lower latency

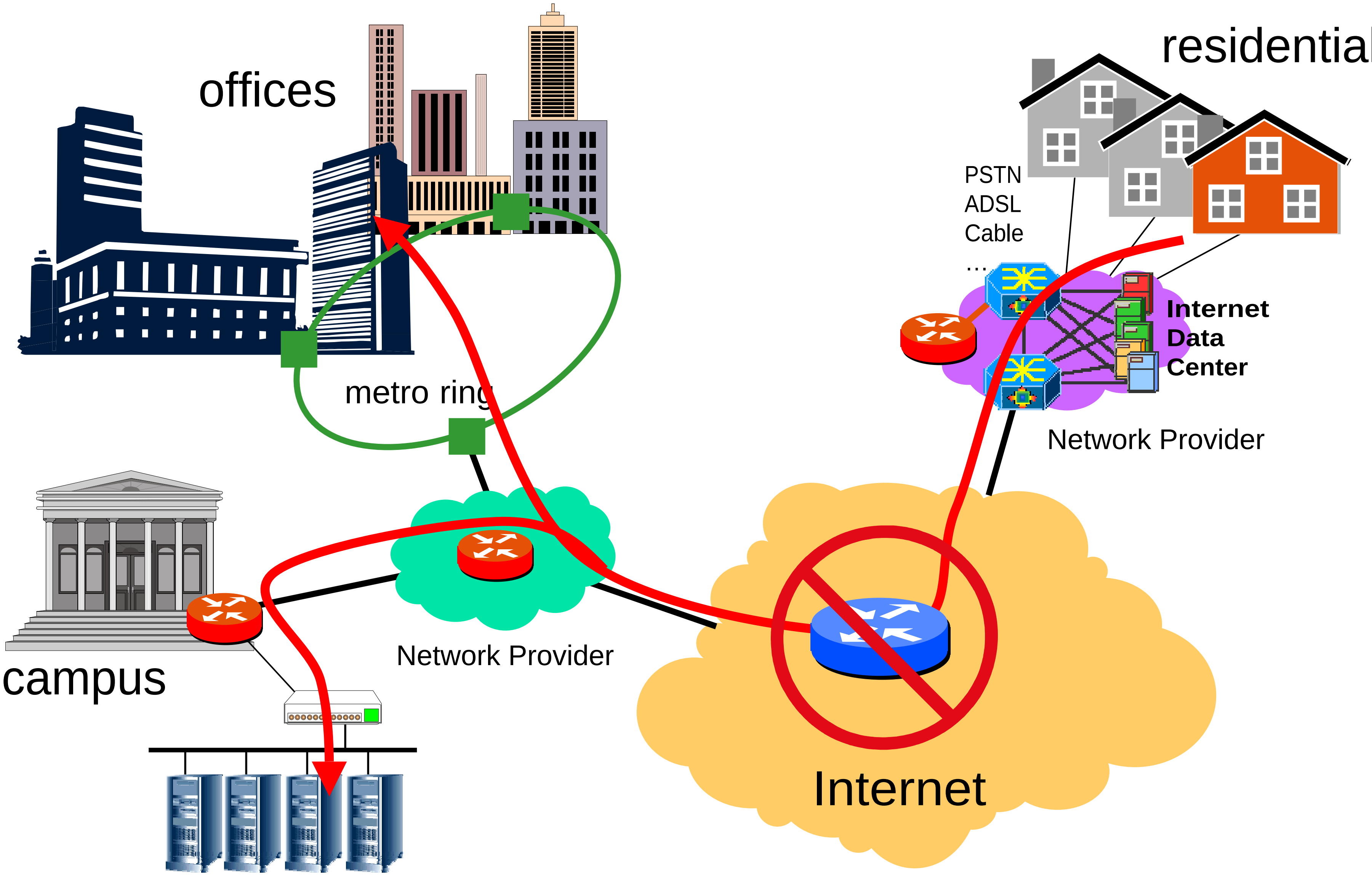


Router assistance with active networking

- Programmable nodes/ routers
- Customized computations on packets
- Standardized execution environment and programming interface

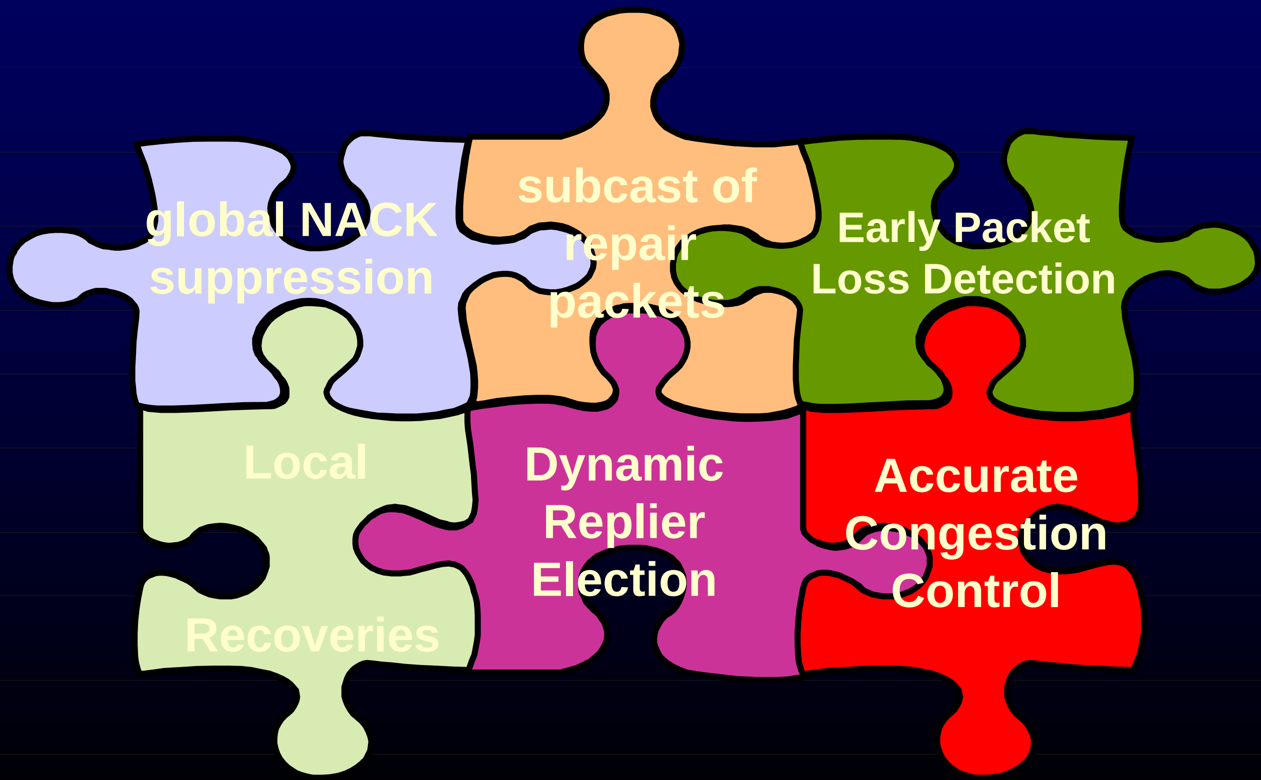


Intelligence at the edge

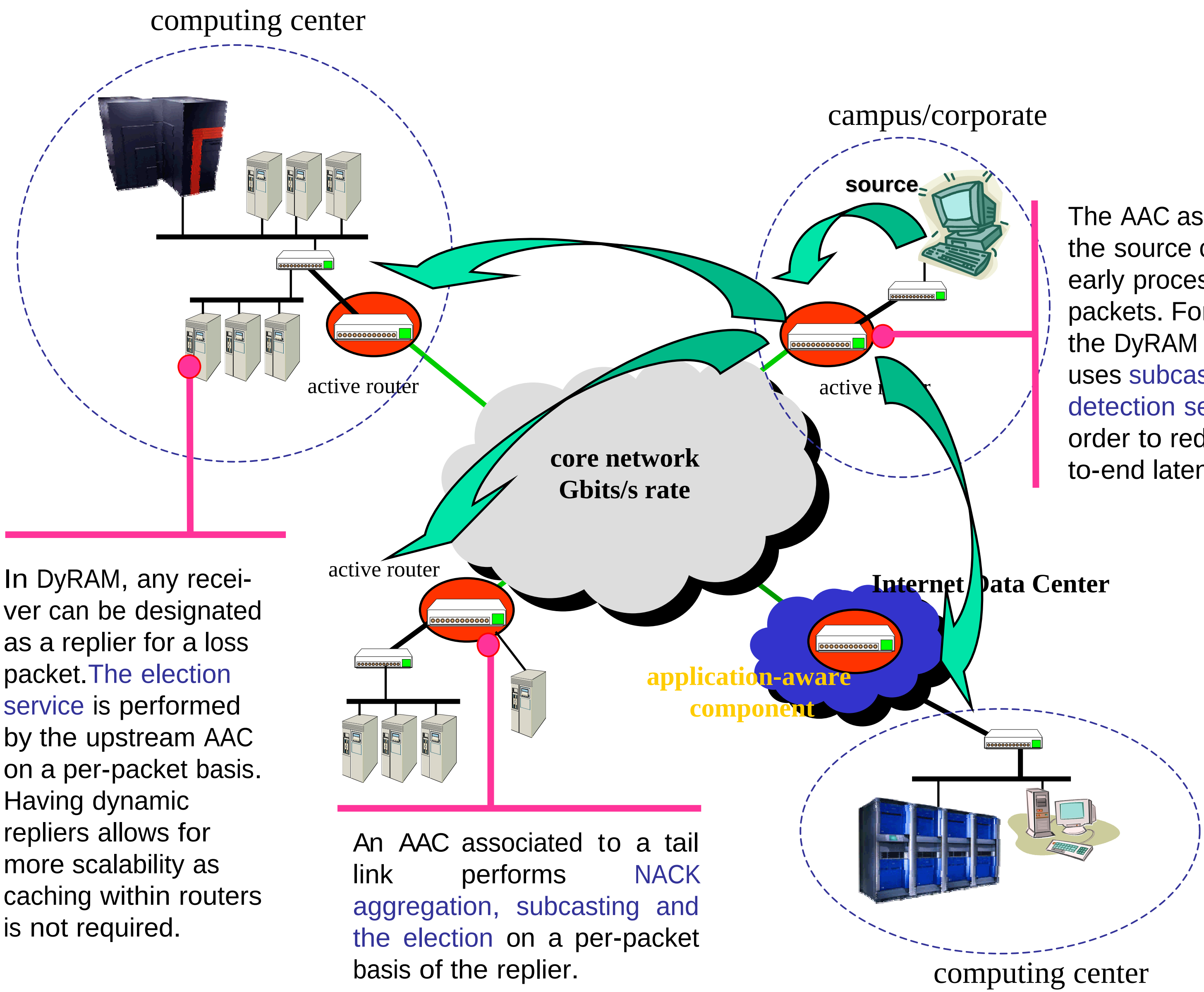


DyRAM Maimour & Pham, 2001

Protocol with modular services for achieving reliability, scalability and low latencies



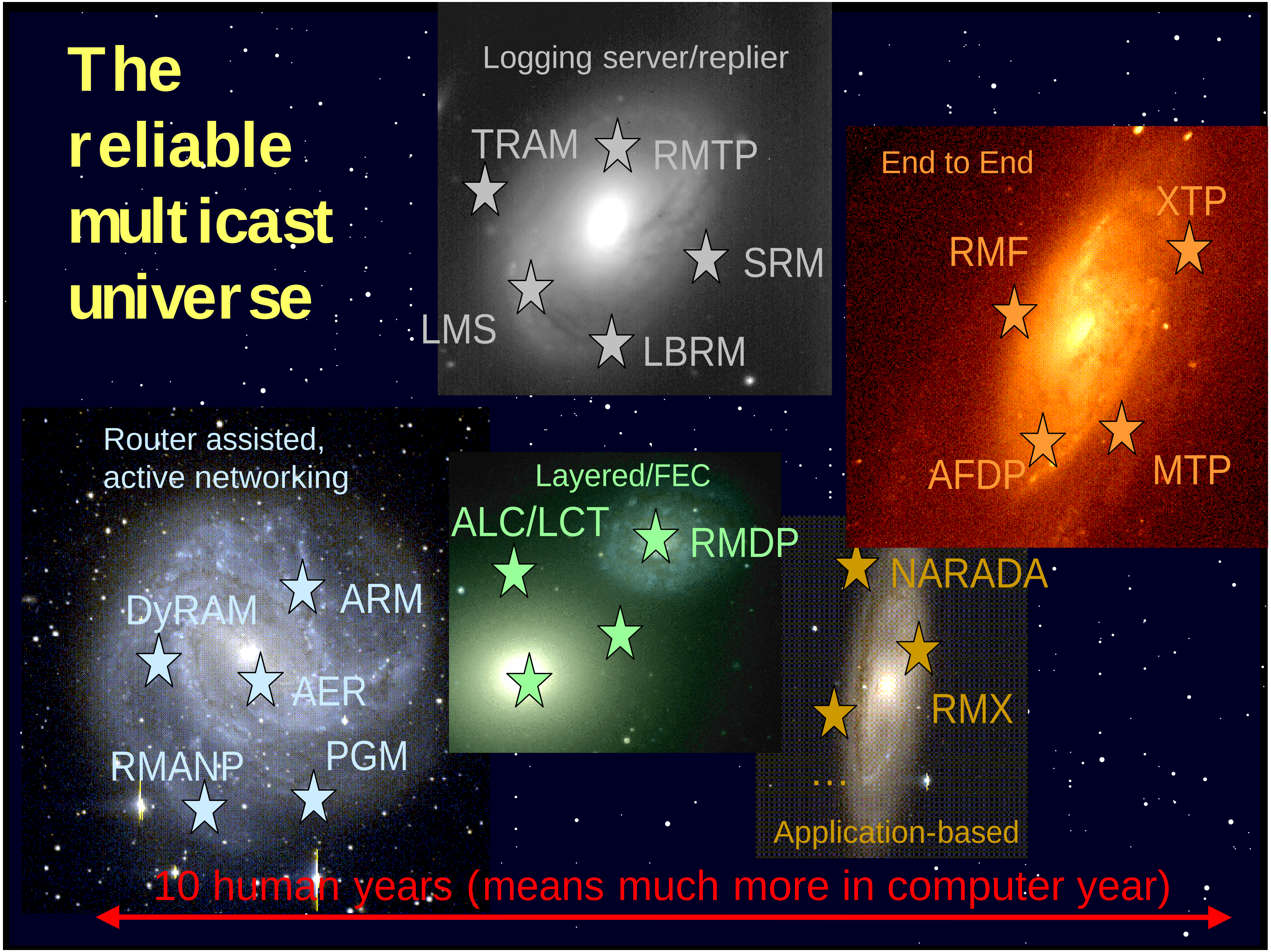
DyRAM on a grid infrastructure



The AAC associated to the source can perform early processing on packets. For instance the DyRAM protocol uses subcast and loss detection services in order to reduce the end-to-end latency.

In DyRAM, any receiver can be designated as a replier for a loss packet. The election service is performed by the upstream AAC on a per-packet basis. Having dynamic repliers allows for more scalability as caching within routers is not required.

An AAC associated to a tail link performs NACK aggregation, subcasting and the election on a per-packet basis of the replier.

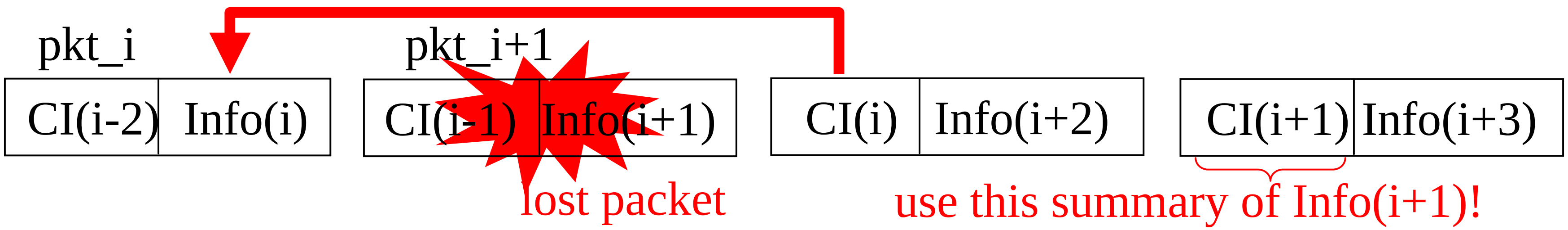

The reliable multicast universe
Logging server/replier
TRAM
RMTP
SRM
LMS
LBRM
End to End
XTP
RMF
AFDP
MTP
Router assisted,
active networking
DyRAM
ARM
AER
RMANP
PGM
Layered/FEC
ALC/LCT
RMDP
NARADA
RMX
...
Application-based
10 human years (means much more in computer year)

Part III

Semi-reliable and Streaming for Multimedia/Real Time Applications

Semi-reliable multicast

- why partial reliability ?
  - sufficient for video/audio (real-time cannot afford retransmissions)
- solution 1:
  - each packet contains compressed information of a previous packet



- solution 2:
  - add proactive FEC to the data flow
  - a FEC packet can replace any lost packet

Video streaming

- two classes of solutions
  - single layer streaming approaches
    - unicast   the natural approach
    - multicast   limitation: same flow to everybody
  - layered streaming approaches
    - exploits the video scalability features (i.e. hierarchical video encoding)
    - unicast   not suited!
    - multicast   the natural approach

  - let's only consider multicast streaming…

Single layer streaming

- approach
  - single stream, mapped on a single multicast group
  - source adapts the transmission rate (video encoding) according to feedback (e.g. RTCP)
  - limitation: everybody receive the same flow!

  - several streams at different rates can be used
  - clients joins the group that best matches their reception capabilities
  - partial solution to the above limitation (same flow for all clients of a group)

Layered streaming

- exploit video scalability
  - AKA hierarchical encoding
  - Available with MPEG-2, H263+, MPEG-4, H26L
- several scalability schemes
  - SNR

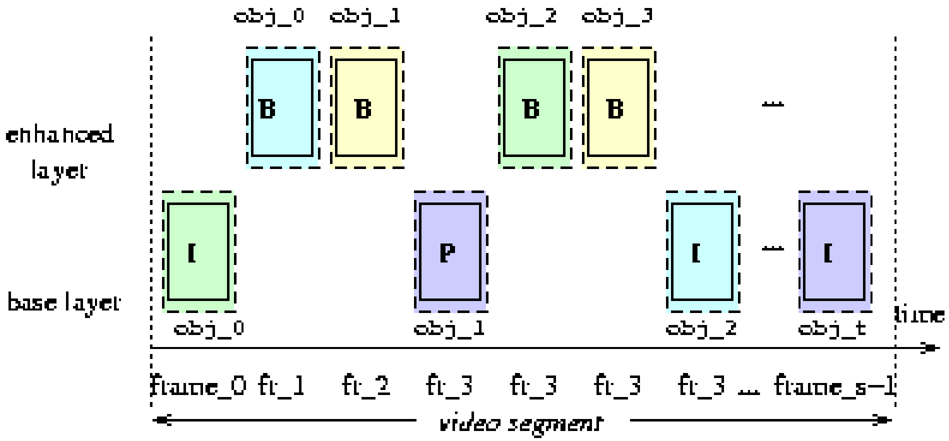
    Two video layers at same spatial/temp. scalability, with different quantization accuracy
  - Temporal scalability

    Relies on IPB frames; several ways to map P/B frames in one or more enhancement layers
  - Spatial scalability

    Two video layers at same rate but different spatial resolution

Layered streaming...(cont')

- most recent codecs (MPEG-4) add a Fine Grain Scalability (FGS) refinement
  - a receiver can benefit from a partially received enhancement layer
  - spatial (or mixed spatial/temp.) scalability
- there is often a single enhancement layer, except with temporal scalability which is more flexible!



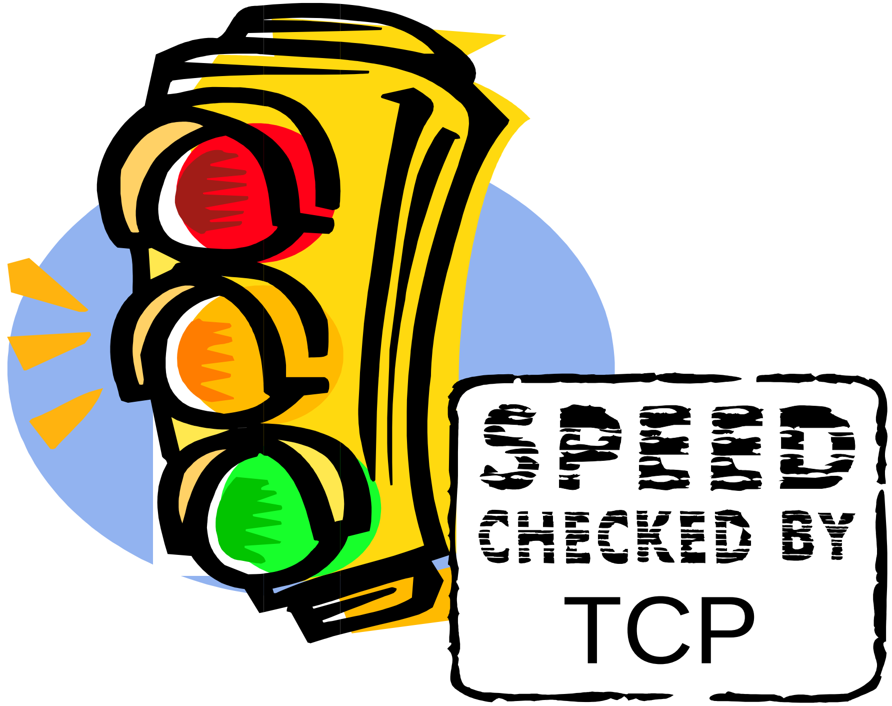
Layered streaming...(cont')

- the layered streaming approach
  - map each video layer to a different mcast group
    - requires a fine granularity (usually assume temporal scalability)
  - some proposals require the support of QoS to protect the base layer
    - without this information, no possible use of data sent on the enhancement layers
    - not very realistic !

Layered streaming... (cont')

- some proposals require feedback and/or assistance in the network (e.g. for number/rate of each layer)
  - not very scalable !
- a totally different solution is based on ALC/reliable multicast
  - solves all problems above but create a one minute latency
  - C. Neumann, V. Roca, ``Multicast Streaming of Hierarchical MPEG4 Presentations'', ACM Multimedia 2002, December 2002

Part IV



Congestion Control and
TCP-friendliness

Congestion Control

- general goals of CC
  - be **fair** with other data flows (be “TCP friendly”)
    - should a multicast transfer use as much resource as a TCP connection or n times as much ?
    - no single definition
    - be responsive to network conditions
  - be **stable**, i.e. avoid oscillations
  - utilize network resources **efficiently**
    - if only one flow, then use all the available bandwidth

Congestion Control…(cont’)

- single layer versus layered transmissions
  - two completely different schemes
  - single layer
    - sender oriented
    - based on ACK / NACK feedbacks
  - layered
    - receiver oriented
    - based on losses experienced

Single rate congestion control

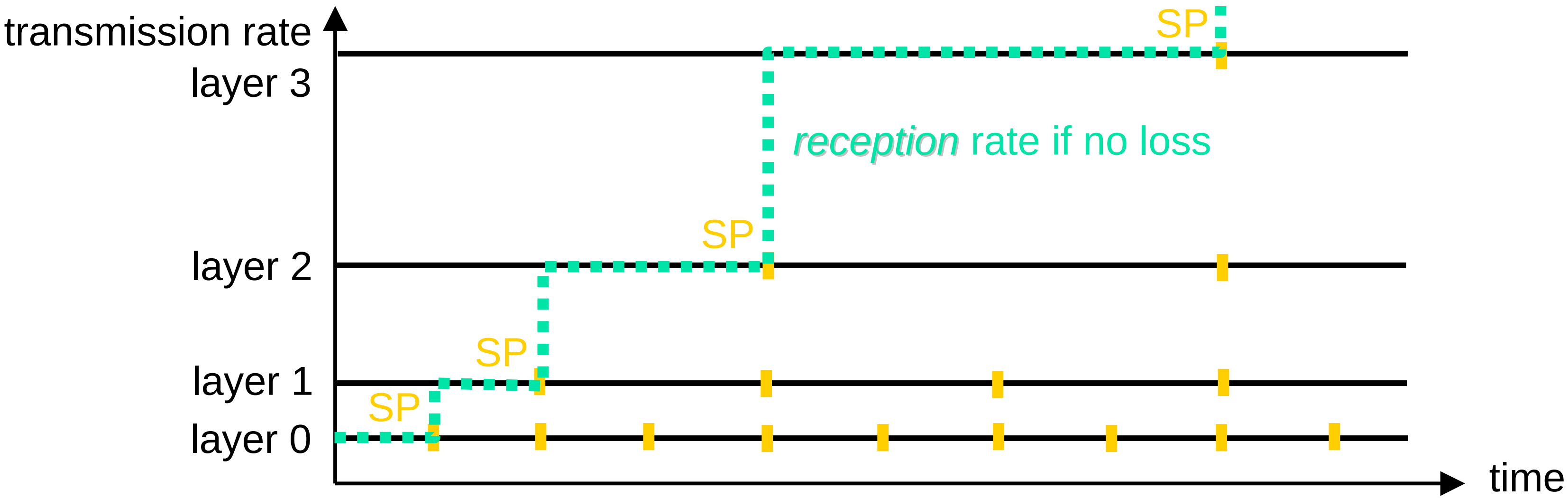
- Example: PGMCC
  - used with single-rate (i.e. layer) protocols like NORM, TRACK
  - relies on a window based transmission
    - mimics TCP
    - evolves according to the ACKs sent by the ``Acker''
  - relies on an ``Acker'' selection process
    - the ``Acker'' is the receiver with the lowest equivalent TCP throughput

      equivTCPthroughput = α / (RTT * sqrt(loss_rate))
    - the ``Acker'' changes dynamically

Layered Congestion Control

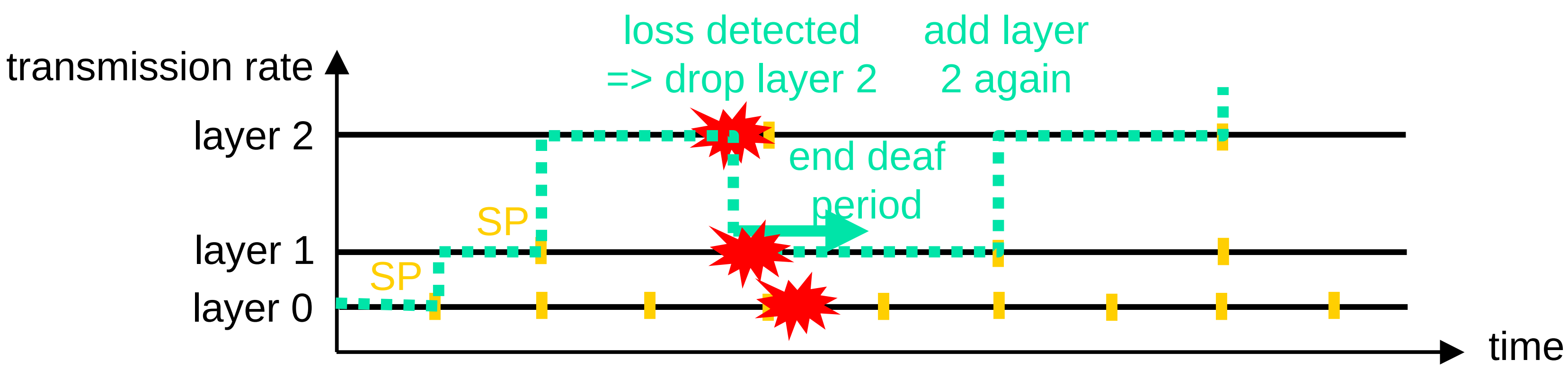
- Example: RLC
  - add synchronization points (SP) / probes
    - adding a layer is only possible at a SP if no loss has been experienced before
    - exponential spacing of SP among the layers

    ⇒ more difficult to add higher layers



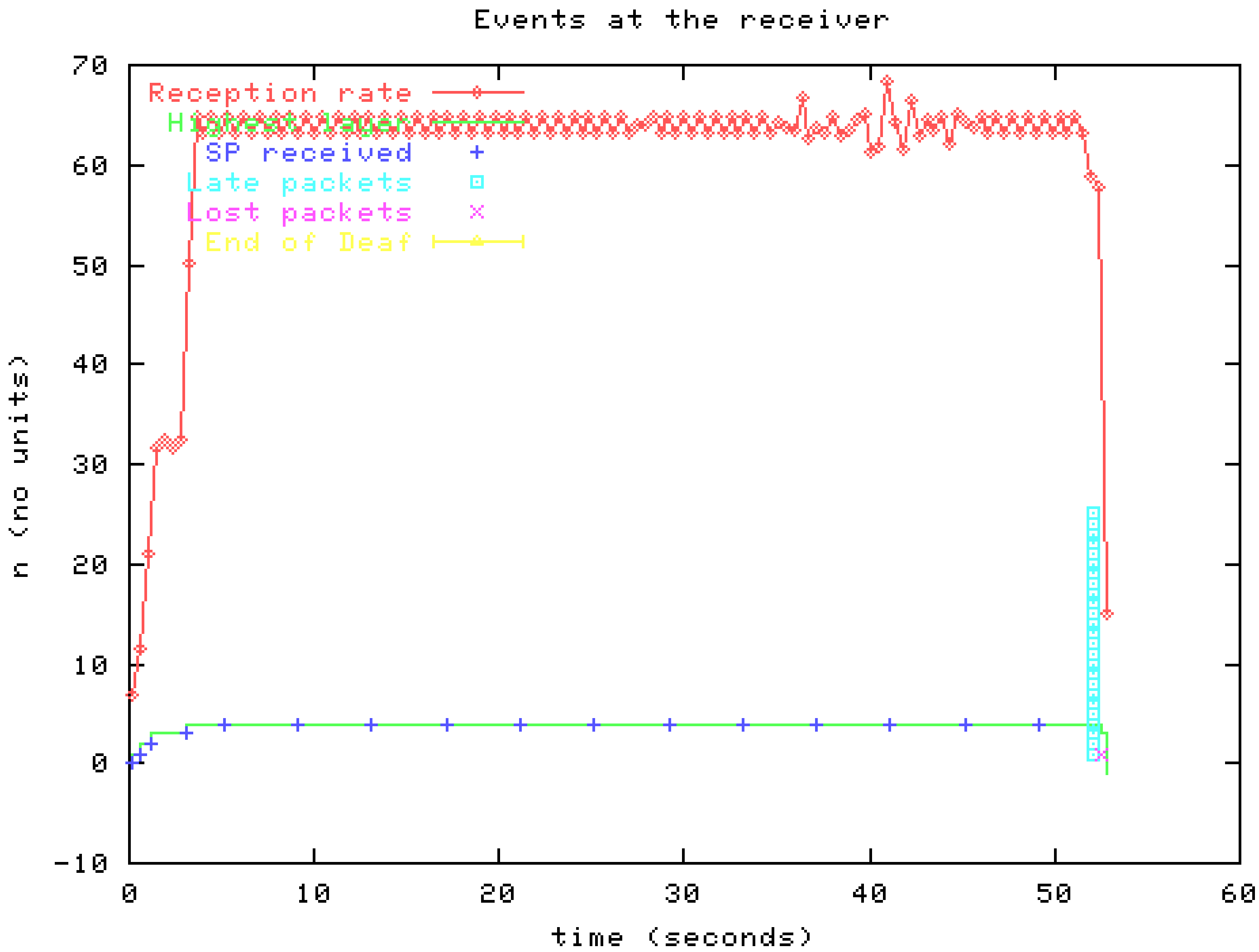
Layered congestion control…(cont’)

- requires “deaf periods”
  - because of IGMP leave latency, after dropping a layer, wait some time, until the distribution tree is updated, before restarting the normal behavior



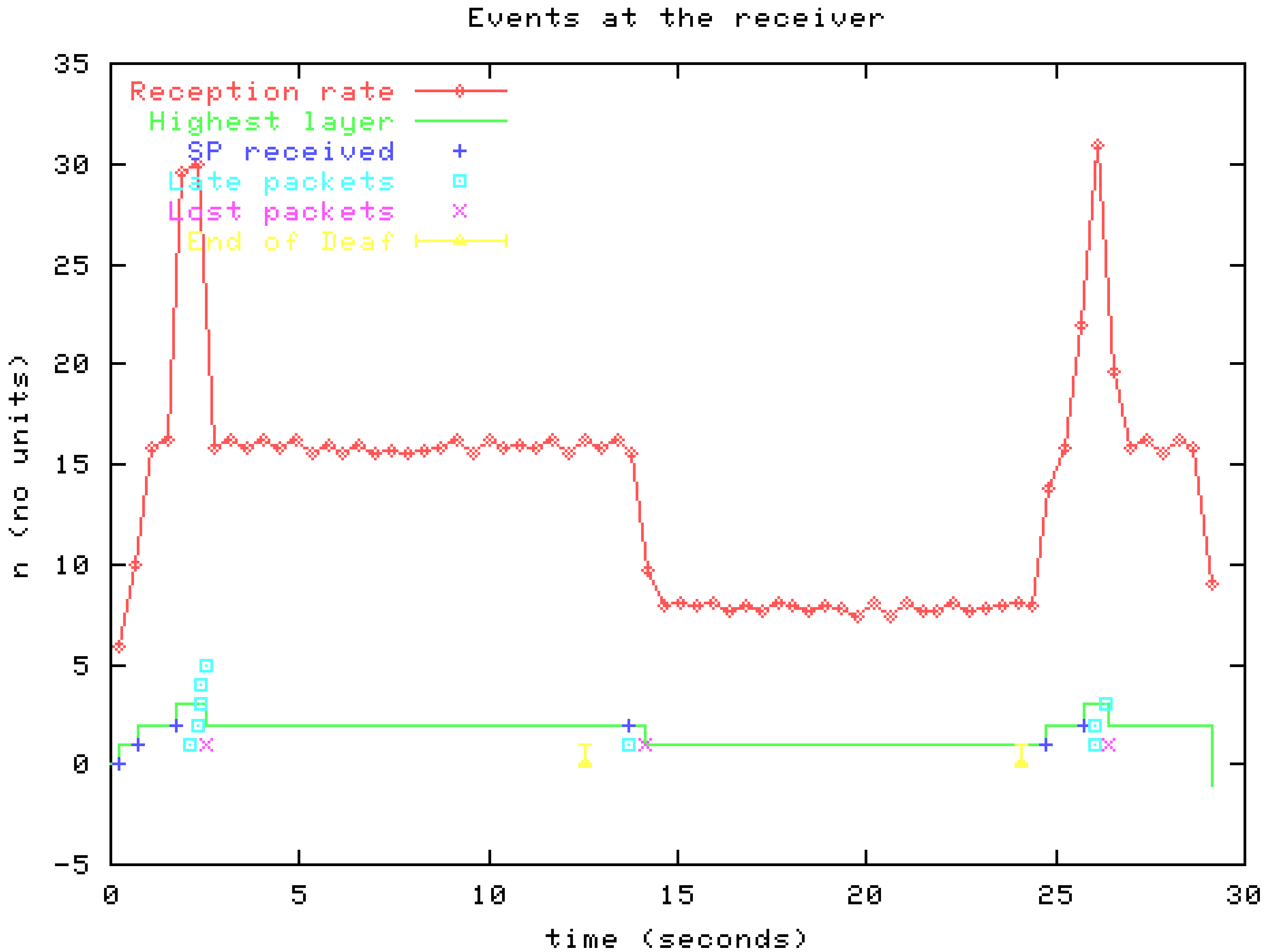
Layered congestion control…(cont’)

- ALC, RLC, receiver events, no loss



Layered congestion control…(cont’)

- ALC, RLC, receiver events, with losses



Layered congestion control…(cont')

- RLC Limitations:
  - limited by IGMP leave latency (a few seconds)
  - AIMD behavior only over long periods
    - adding a layer multiplies reception rate by 2 which is too much
  - only adapts to packet loss, not to RTT

    different from TCP where:rate ~1/(RTT*sqrt(p))

- RLC is not a good CC protocol…but it is simple!

Layered congestion control

- Other protocols exist...
  - FLID-SL (Fair Layer Increase/Decrease - Static Layering)
    - similar to RLC, without SP, with a finer rate granularity (ratio 1.3 instead of 2)
  - FLID-DL (Dynamic Layering)
    - completely different approach
    - behaves better than RLC/FLID-SL that are limited by IGMP leave latency
    - … but creates a high IGMP/Routing protocol signaling

Layered congestion control…(cont')

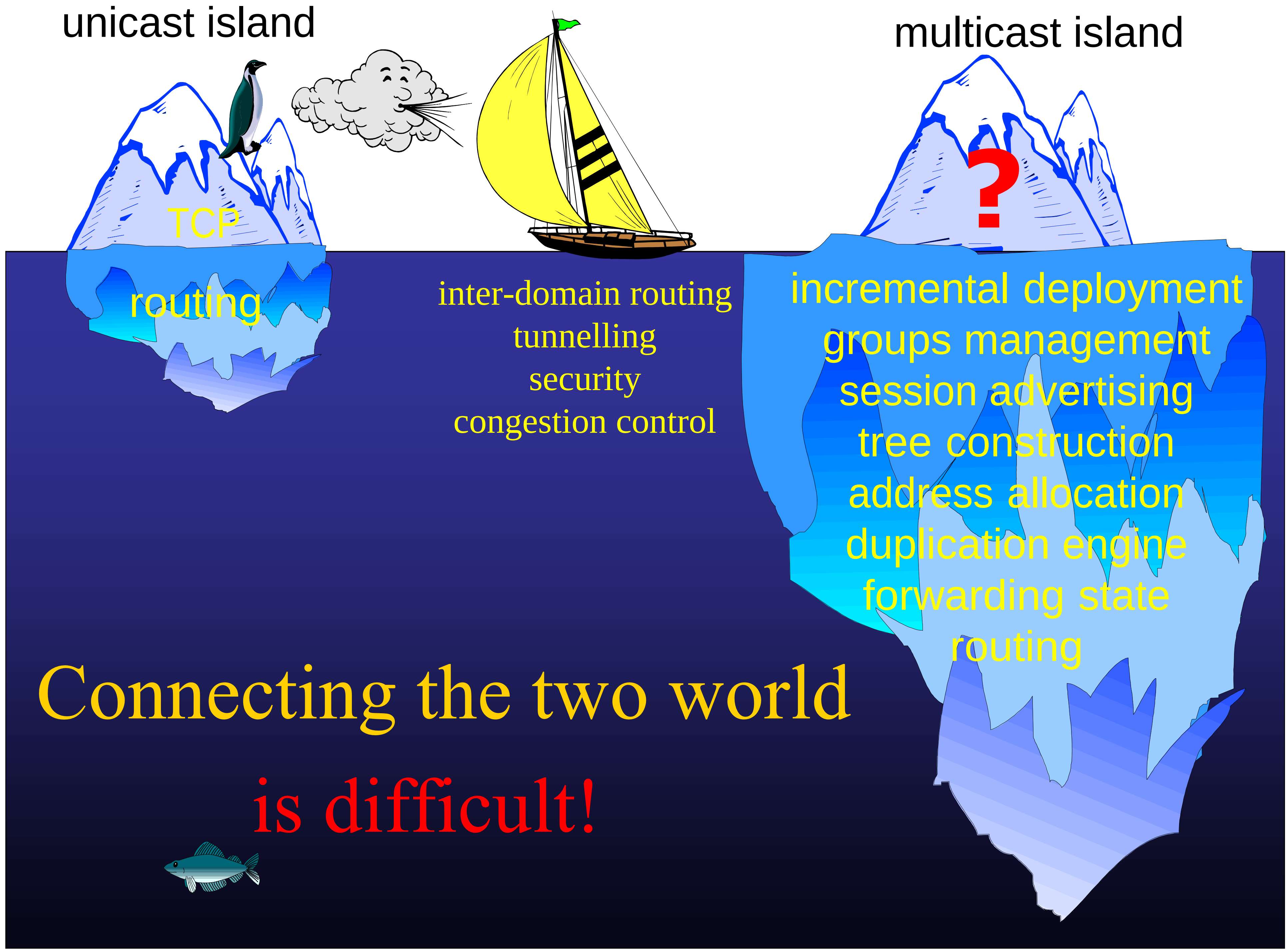
- WEBRC
  - [WEBRC02]
  - uses the dynamic layering approach of FLID-DL
  - improves throughput estimation using an equivalent TCP throughput model
  - bypasses the IGMP leave latency problem and solves the IGMP/routing load of FLID-DL

- probably the best solution today...

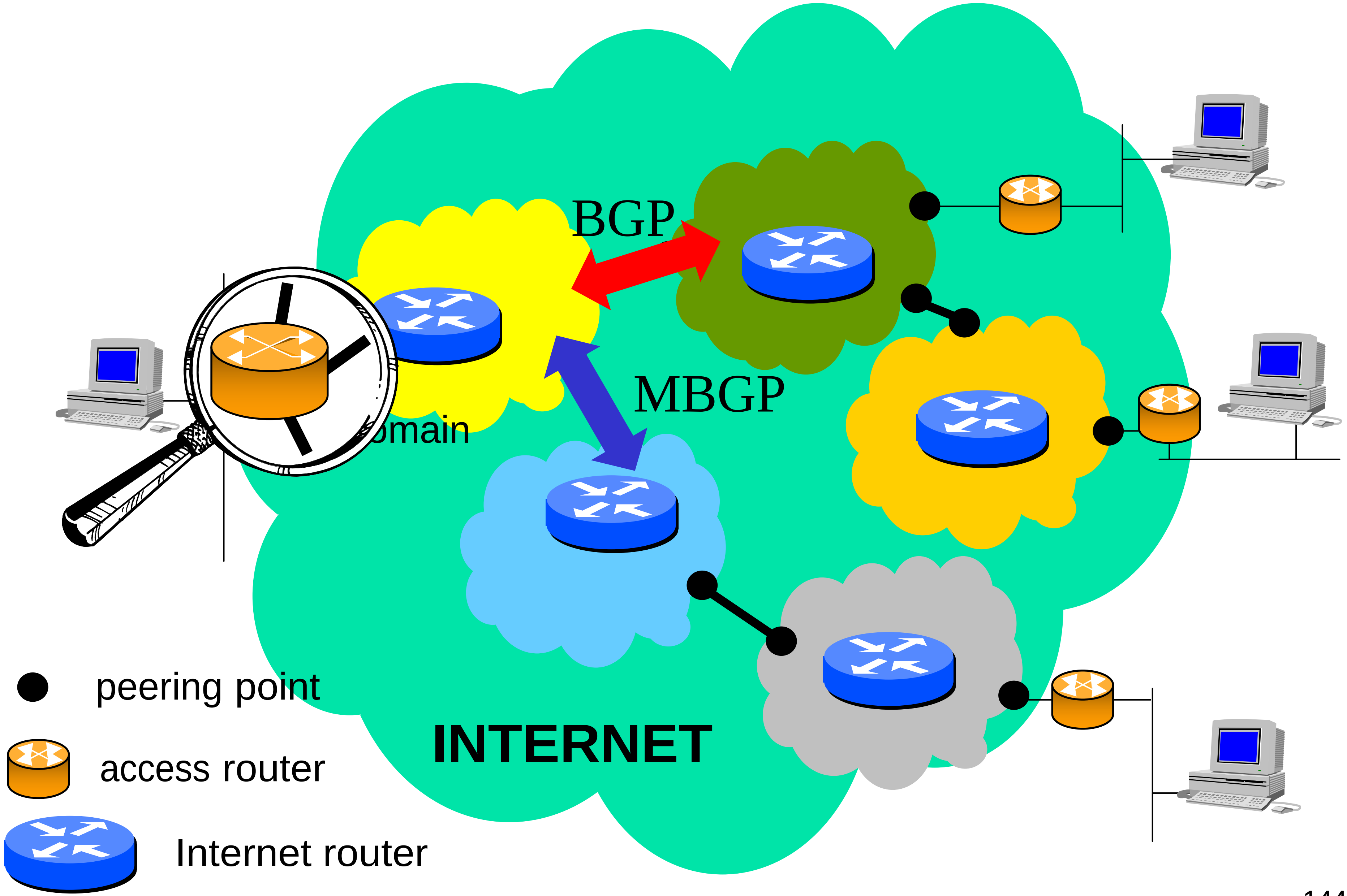
  …but also by far the most complex !

Part V

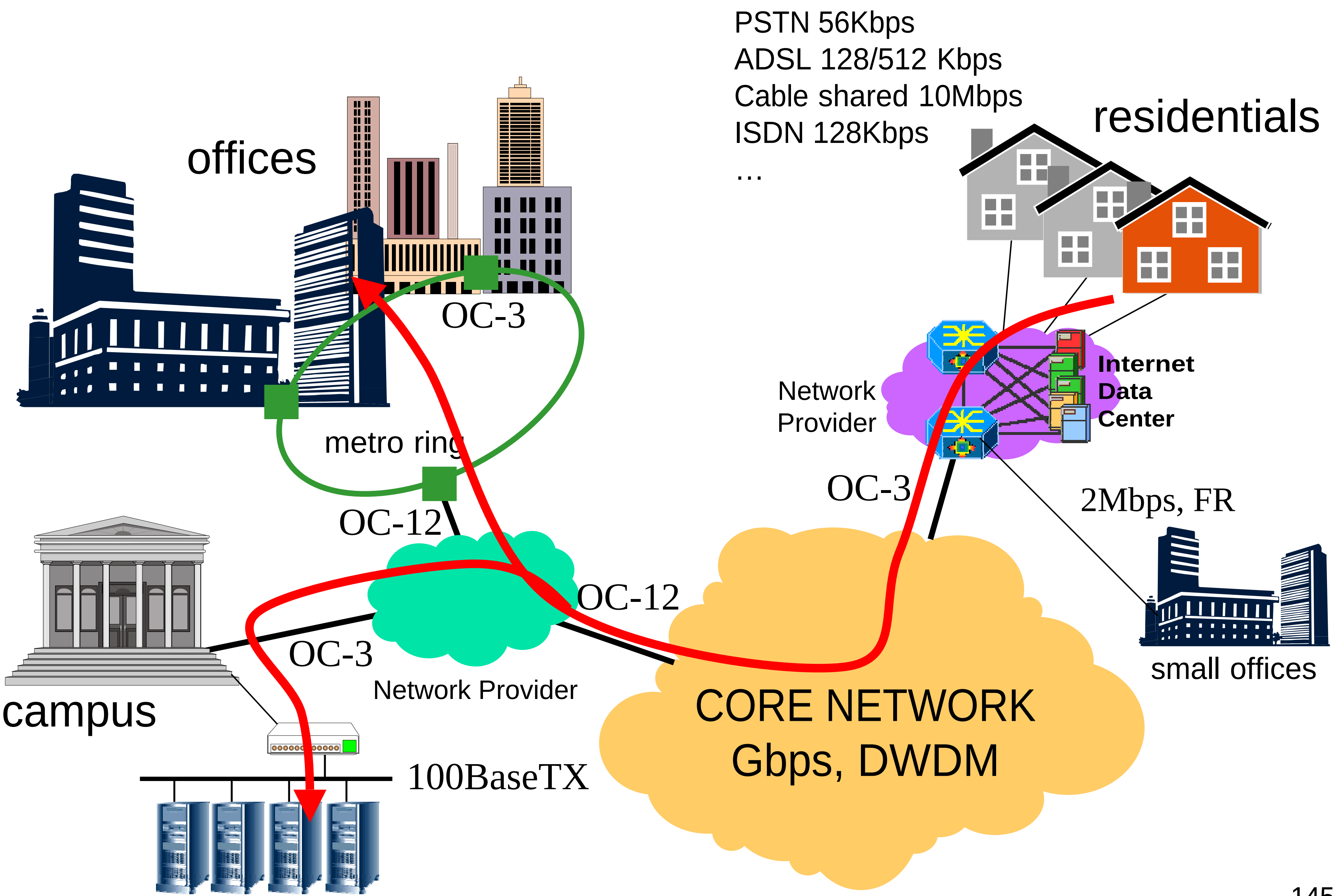
Status and Deployment of Multicast Technologies


unicast island
multicast island
TCP
routing
?
inter-domain routing
tunnelling
security
congestion control
incremental deployment
groups management
session advertising
tree construction
address allocation
duplication engine
forwarding state
routing
Connecting the two world
is difficult!

Inter-domain agreement



Users' accesses



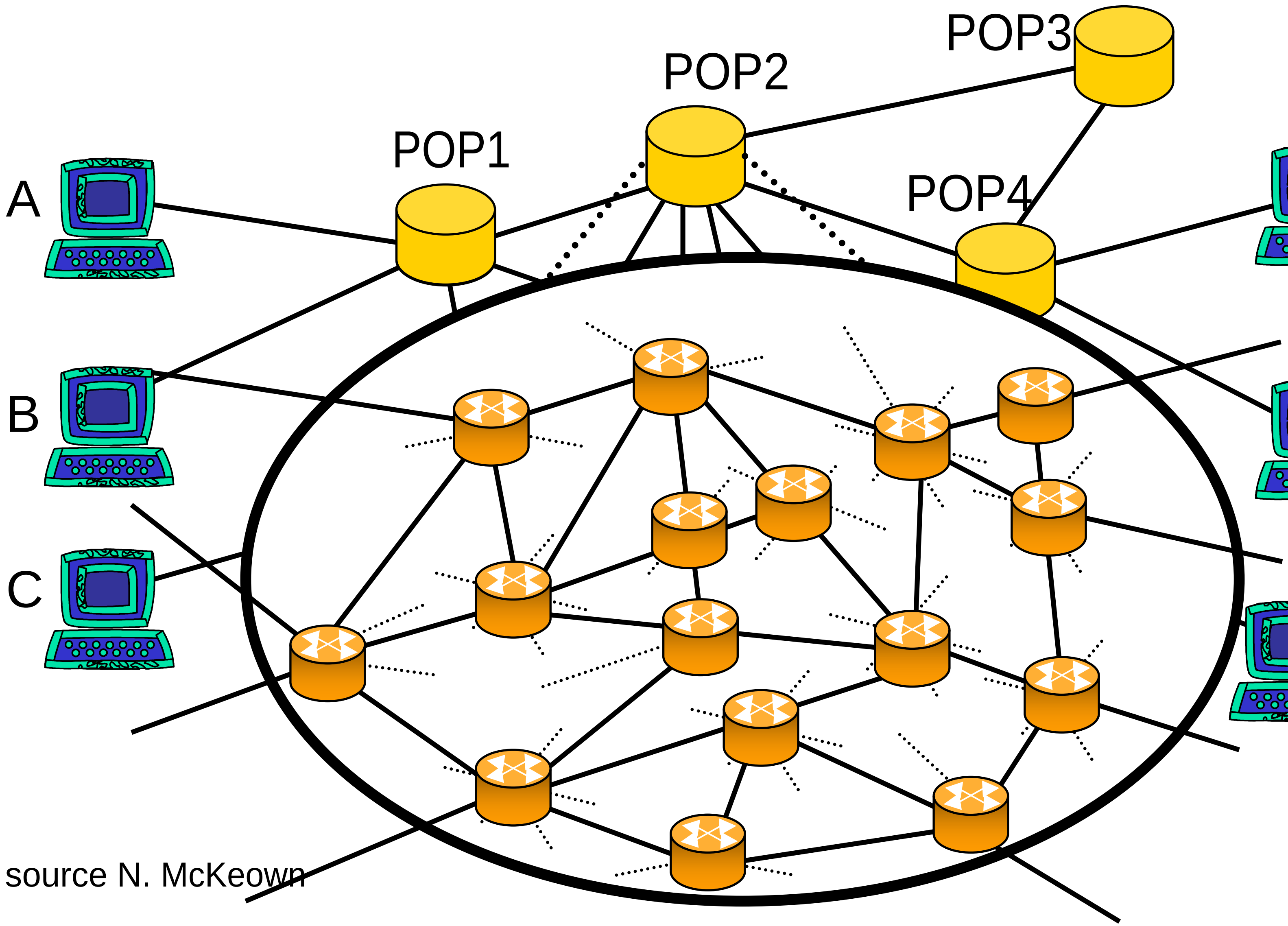
Links heterogeneity

- Backbone links
  - optical fibers
  - 2.5 to 160 Gbps with DWDM techniques
- End-user access
  - 9.6Kbps (GSM) to 2Mbps (UMTS) V.90 56Kbps modem on twisted pair
  - 64Kbps to 1930Kbps ISDN access
  - 128Kbps to 2Mbps with xDSL modem
  - 1Mbps to 10Mbps Cable-modem
  - 155Mbps to 2.5Gbps SONET/SDH

Internet routers: key elements of internetworking

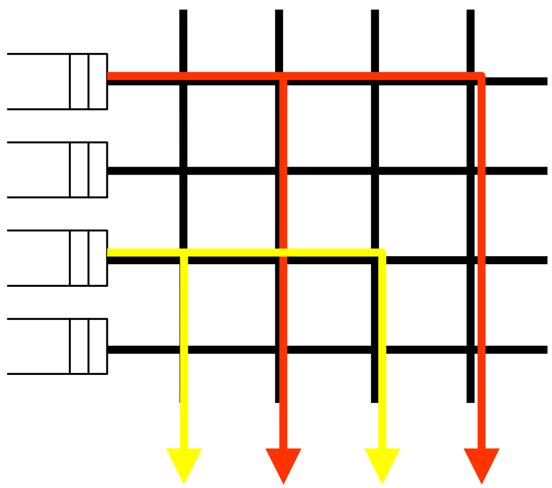
- Routers
  - run routing protocols and build routing table,
  - receive data packets and perform relaying,
  - may have to consider Quality of Service constraints for scheduling packets,
  - are highly optimized for packet forwarding functions.

Multicast in Points of Presence



source N. McKeown

Multicast, a threat for high-performance routers!



The ~~open~~ model
no-security

CONTRACT

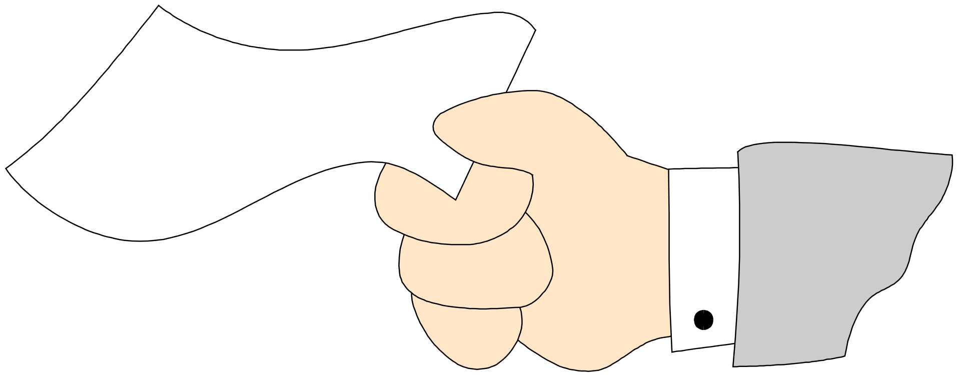
Can not control sources

Can not control receivers

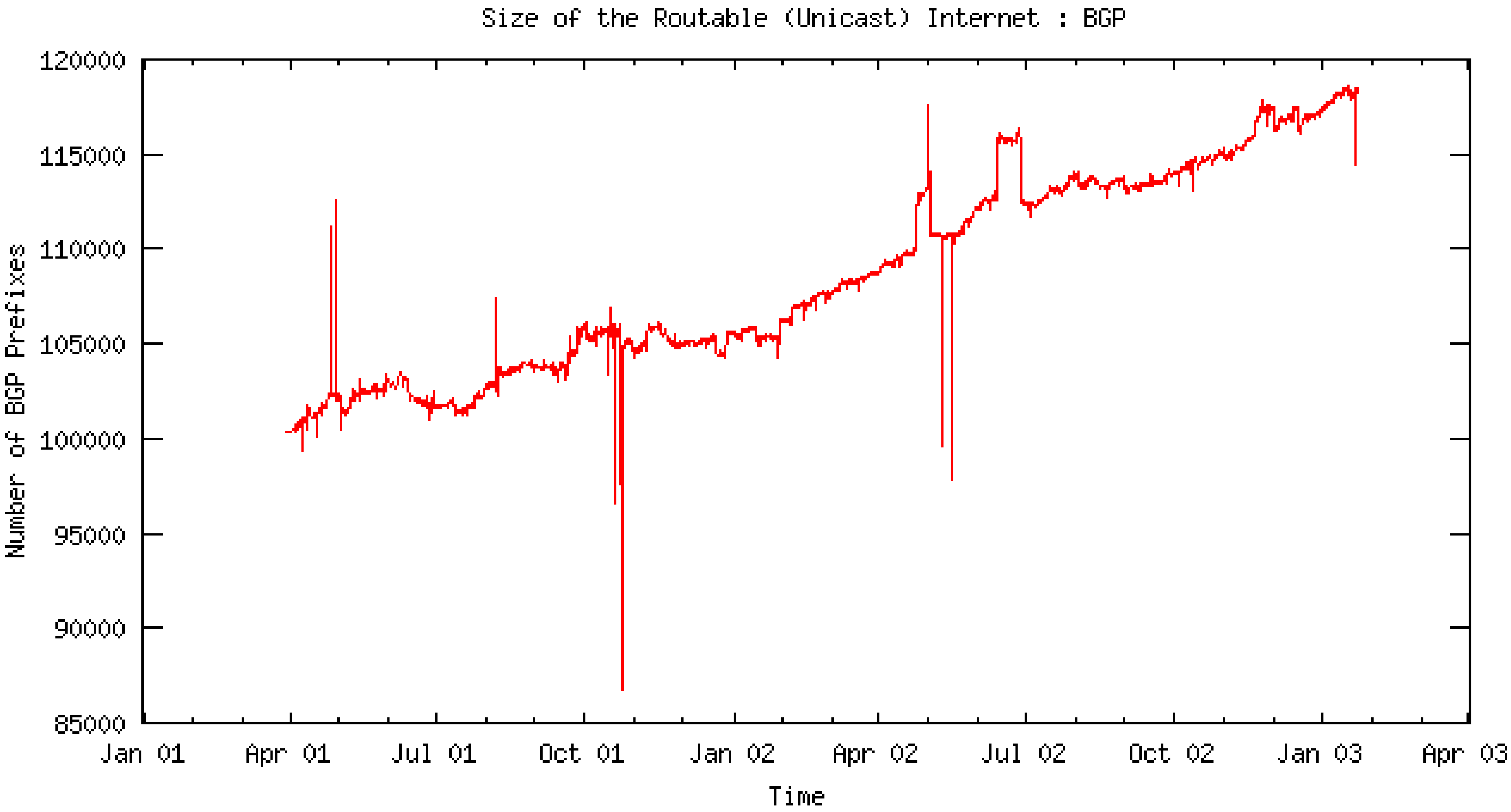
Can not control groups

Can not control traffic

Please sign

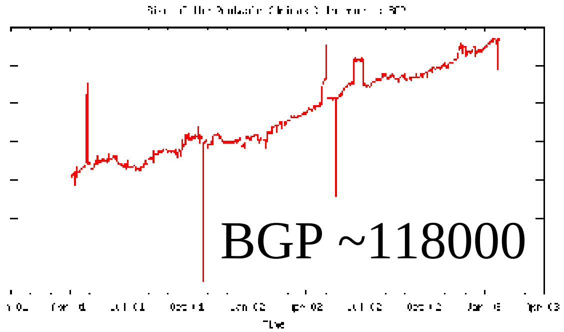


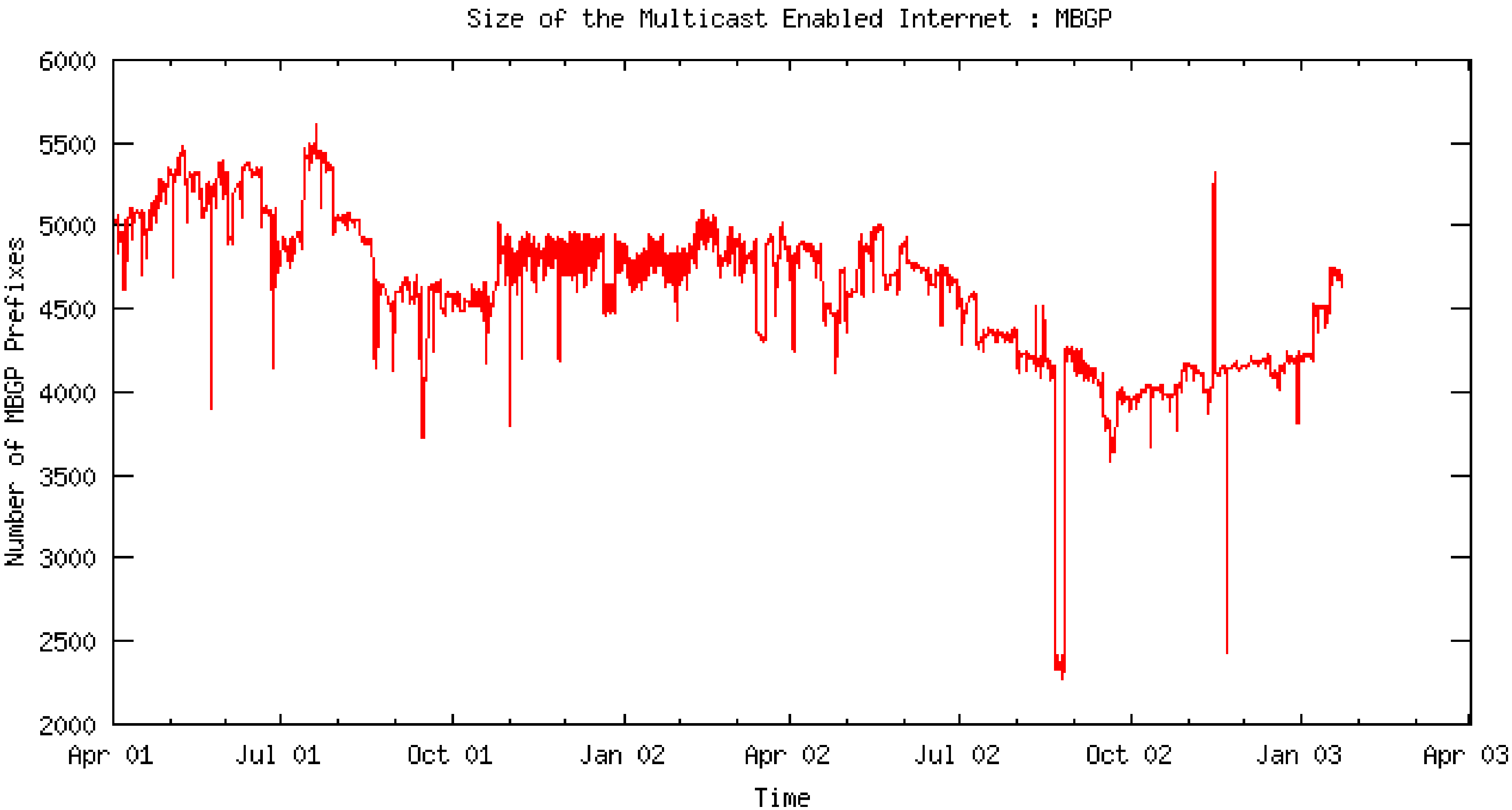
BGP table size



source www.multicasttech.com/status

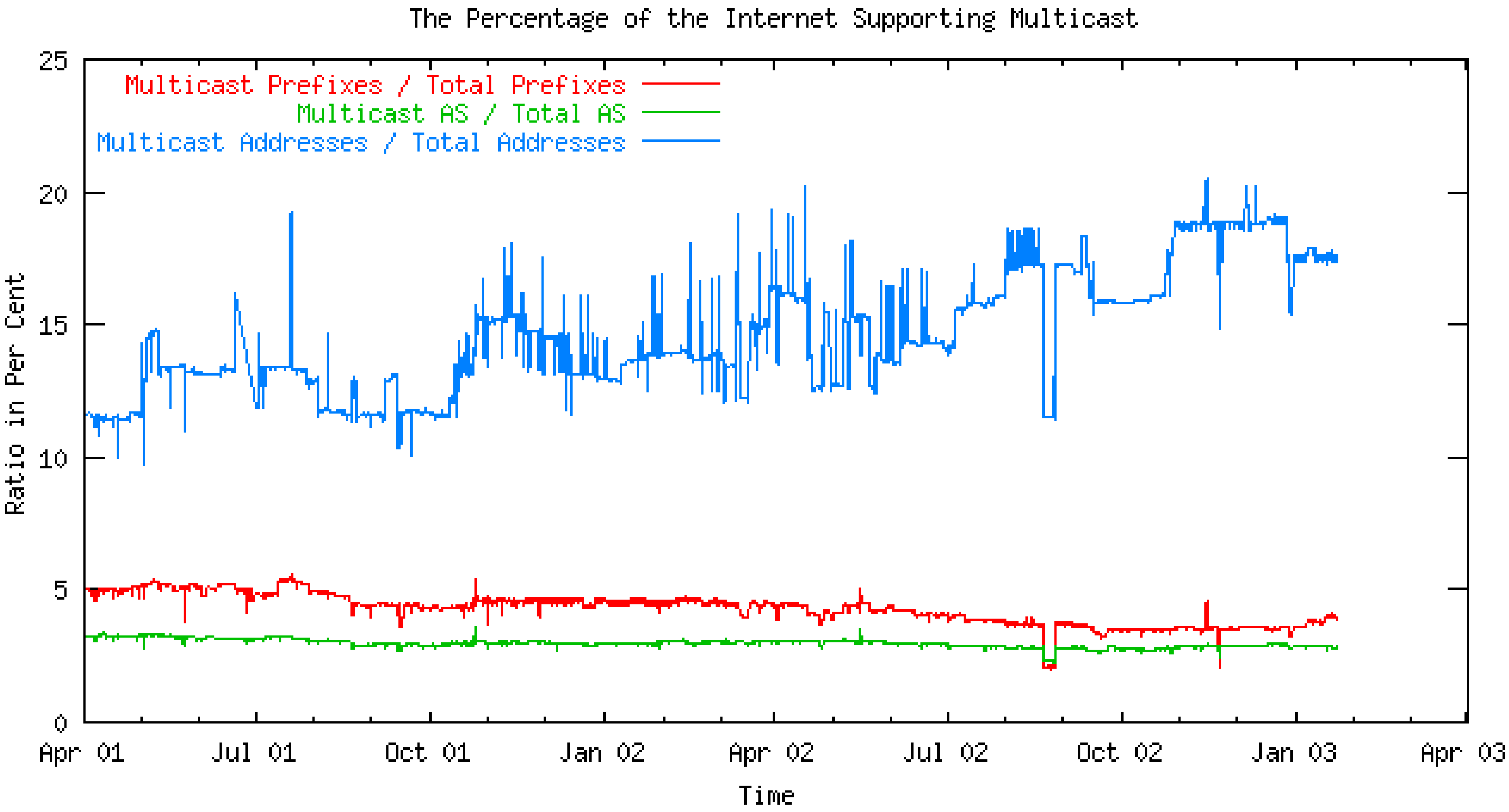
MBGP table size





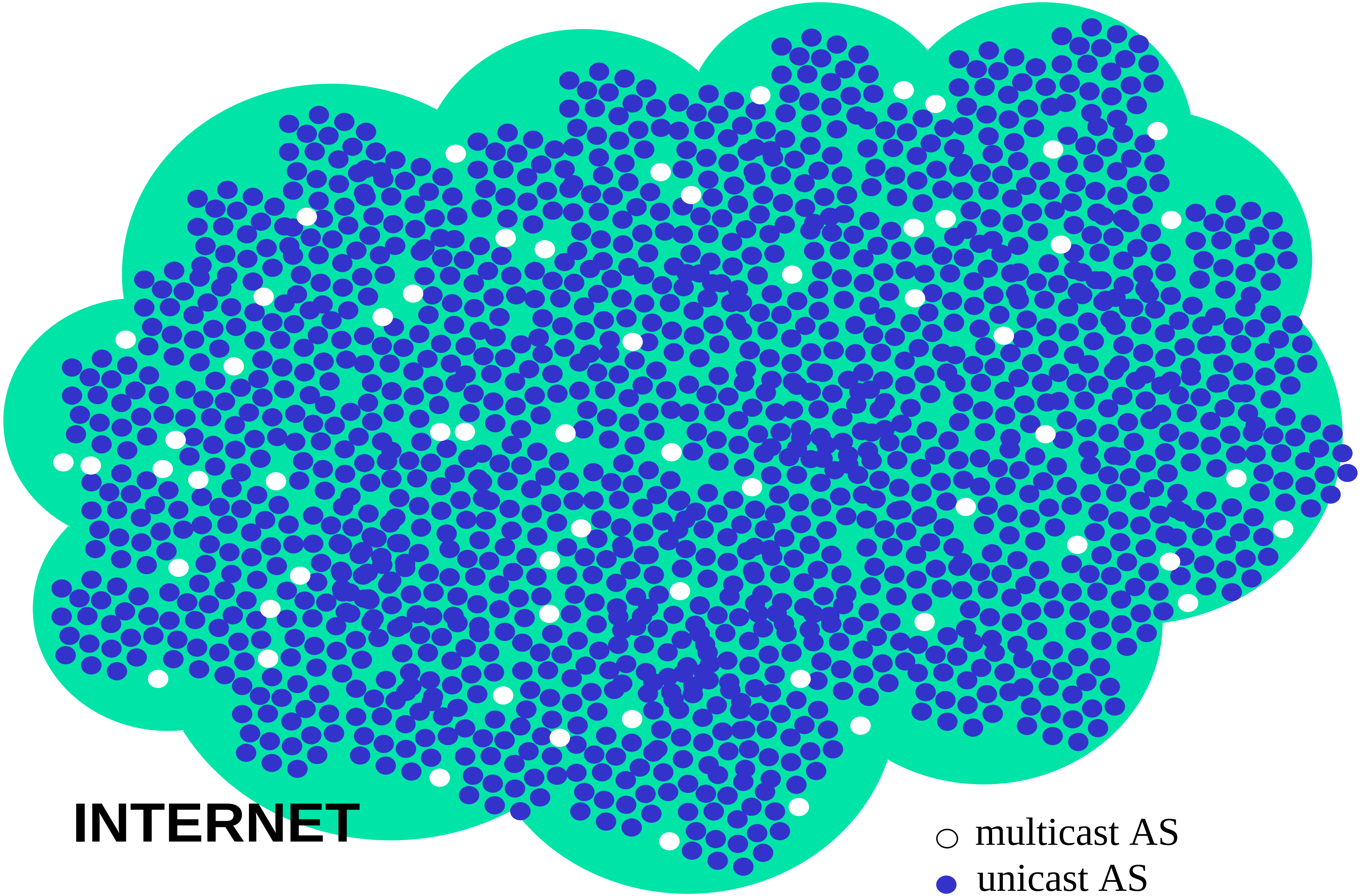
source www.multicasttech.com/status

Relative Size of the Multicast Enabled Internet

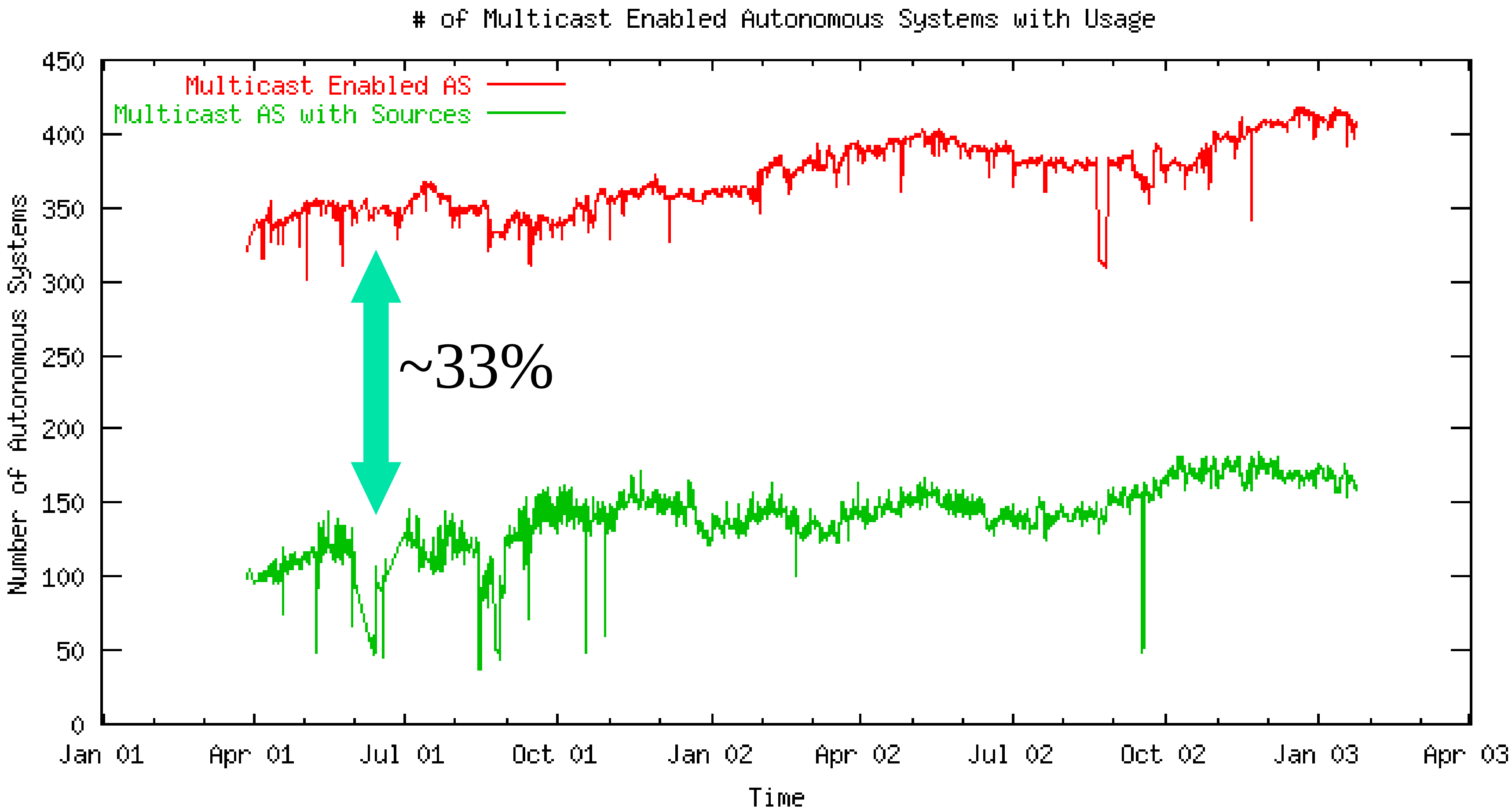


source www.multicasttech.com/status

The gap in images



Autonomous Systems in the Multicast Enabled Internet: Totals and Those With Active Sources



source www.multicasttech.com/status

The MBone (Multicast Bone)

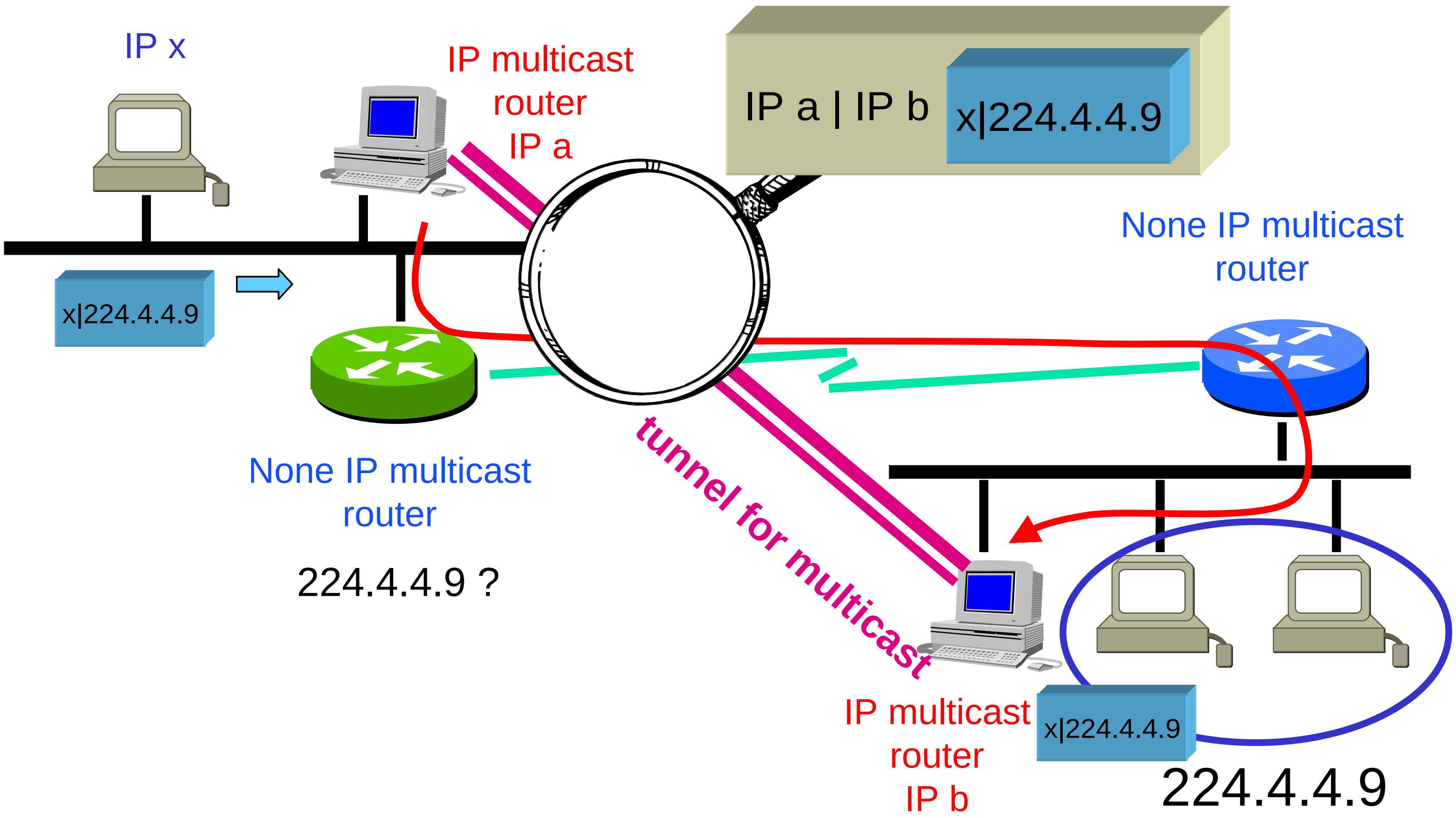
- In March 1992, a new venue quietly debuted on the Internet -- one in which people worldwide could meet in a common electronic window and not only see and talk to one another, but work on a shared "whiteboard." This conferencing network -- called the Multicast Backbone, or MBone -- has the potential to launch a new era in scientific collaboration.

http://www.lbl.gov/ICSD/MBONE/

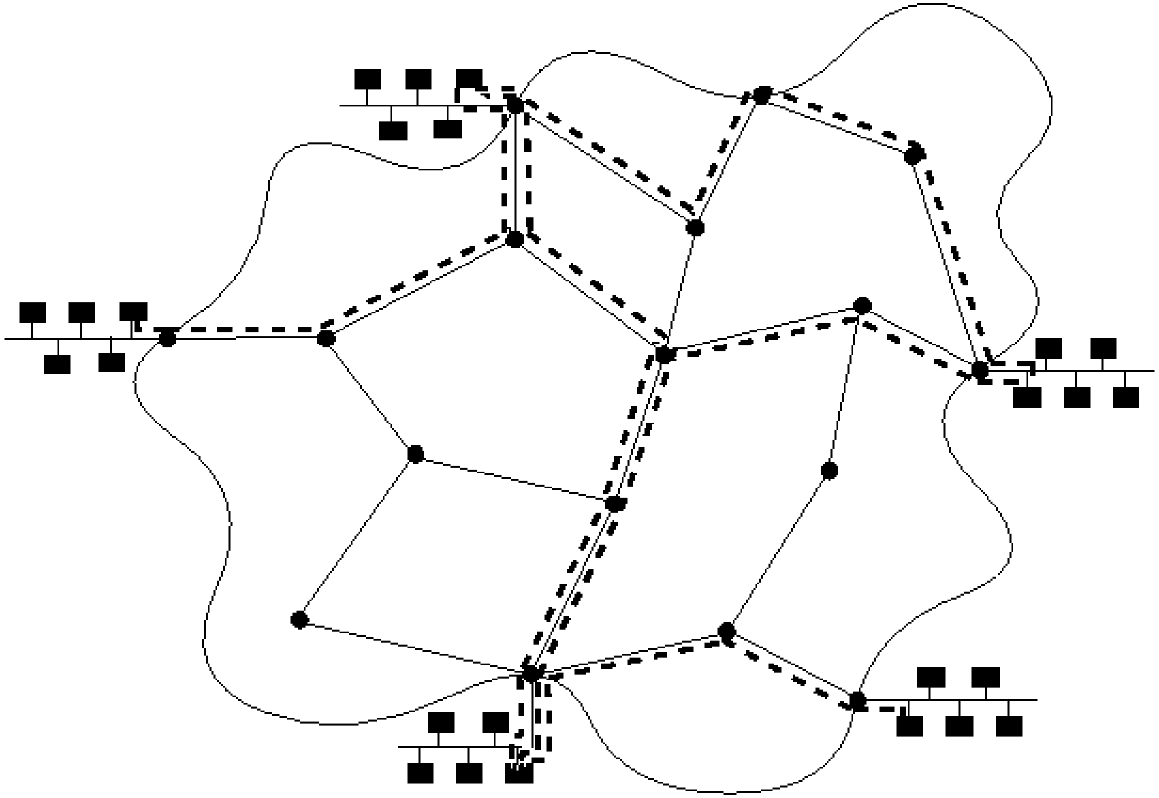
The MBone

- MBone = Multicast backbone
  - Virtual Internet backbone for Multicast IP
  - linked by "tunnels" when native multicast is not possible
  - on top of an unicast topology (overlay network)

Tunnelling illustrated

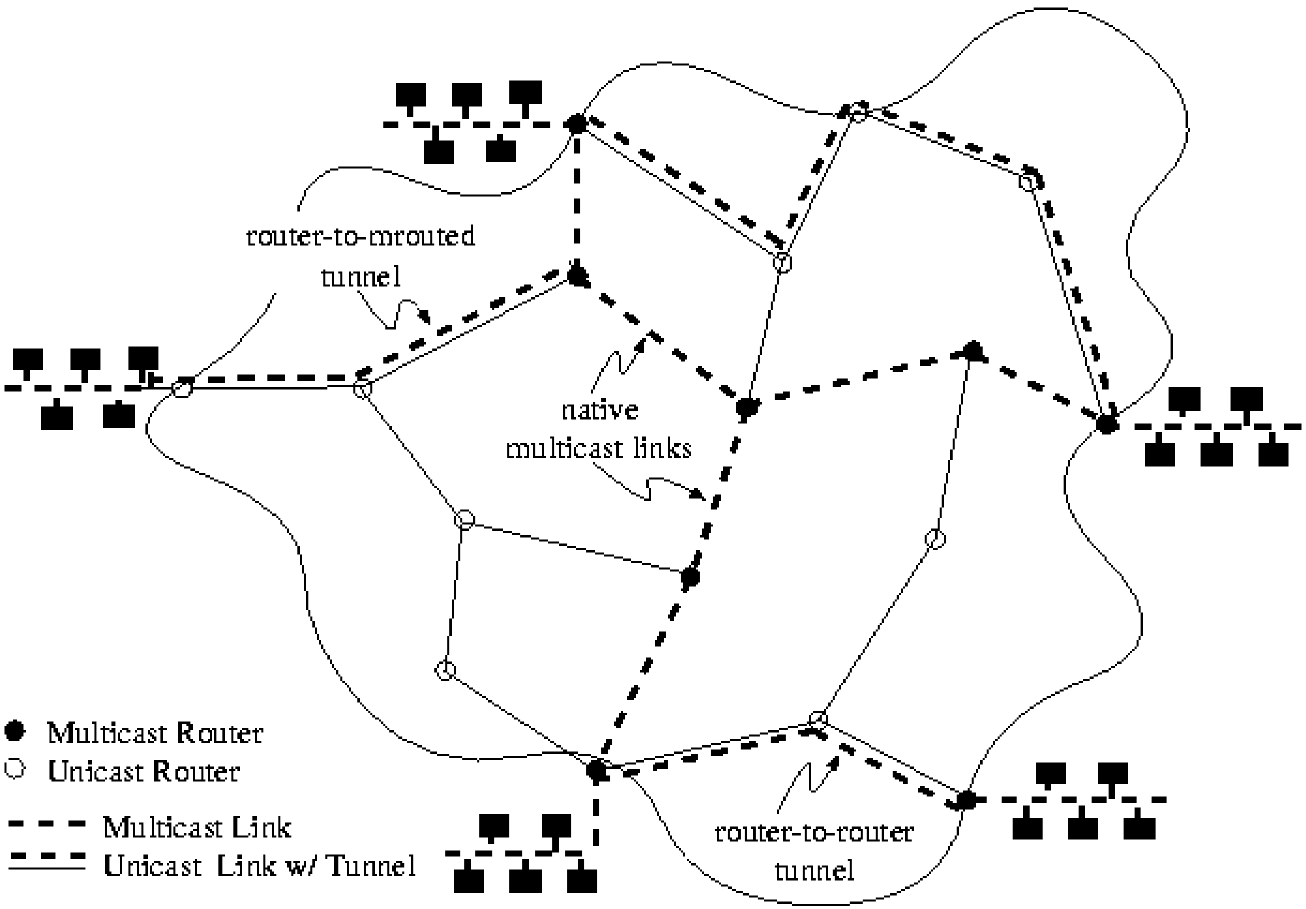


The early MBone with tunnels



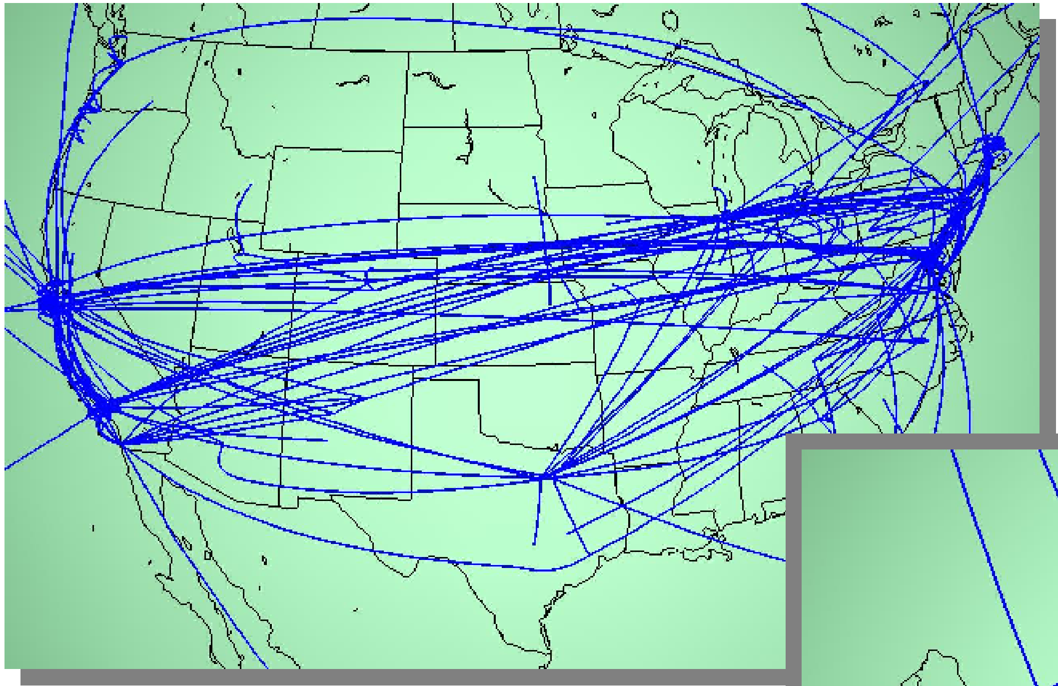
source K. Almeroth's paper. IEEE Networks Magazine, Vol.14(1)

Mixing tunnels and native multicast

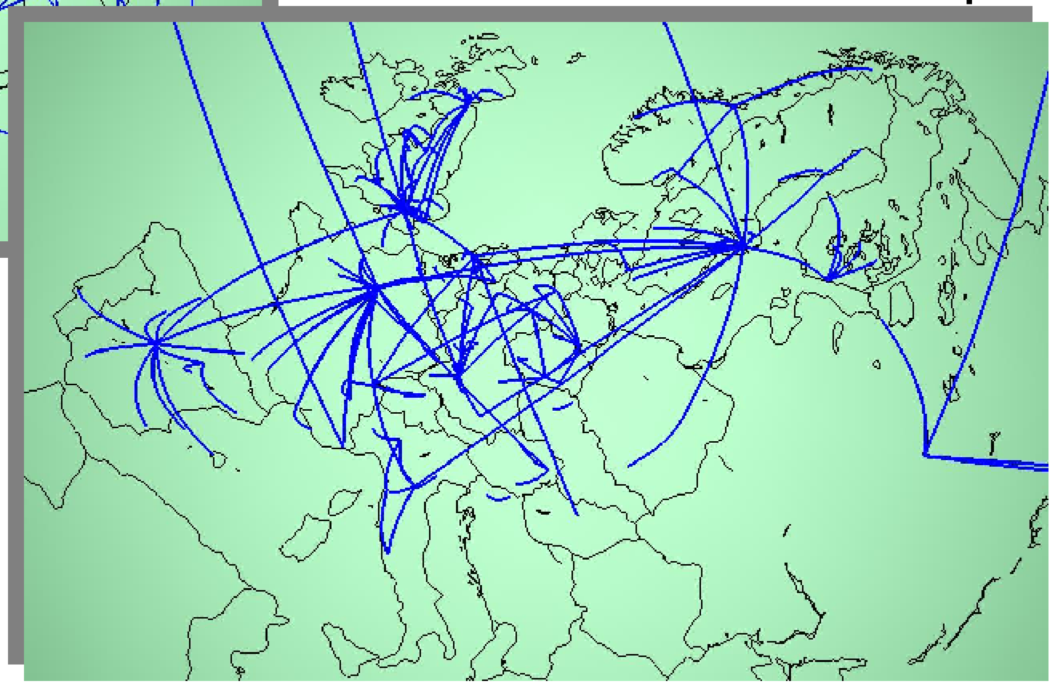


source K. Almeroth's paper. IEEE Networks Magazine, Vol.14(1)

The Mbone big picture

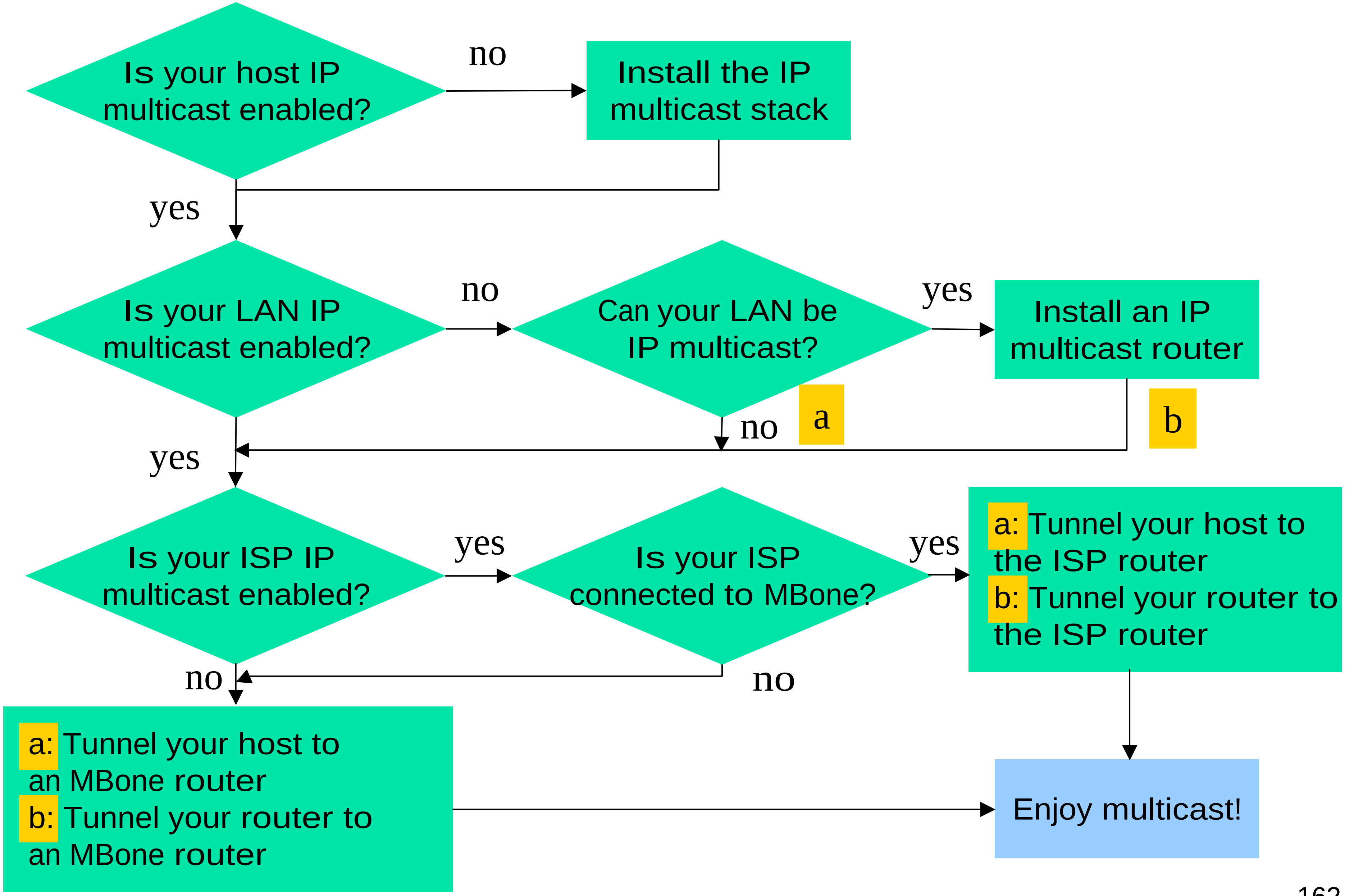

US

Europe



source http://graphics.stanford.edu/papers/mbone/

The MBone HOWTO

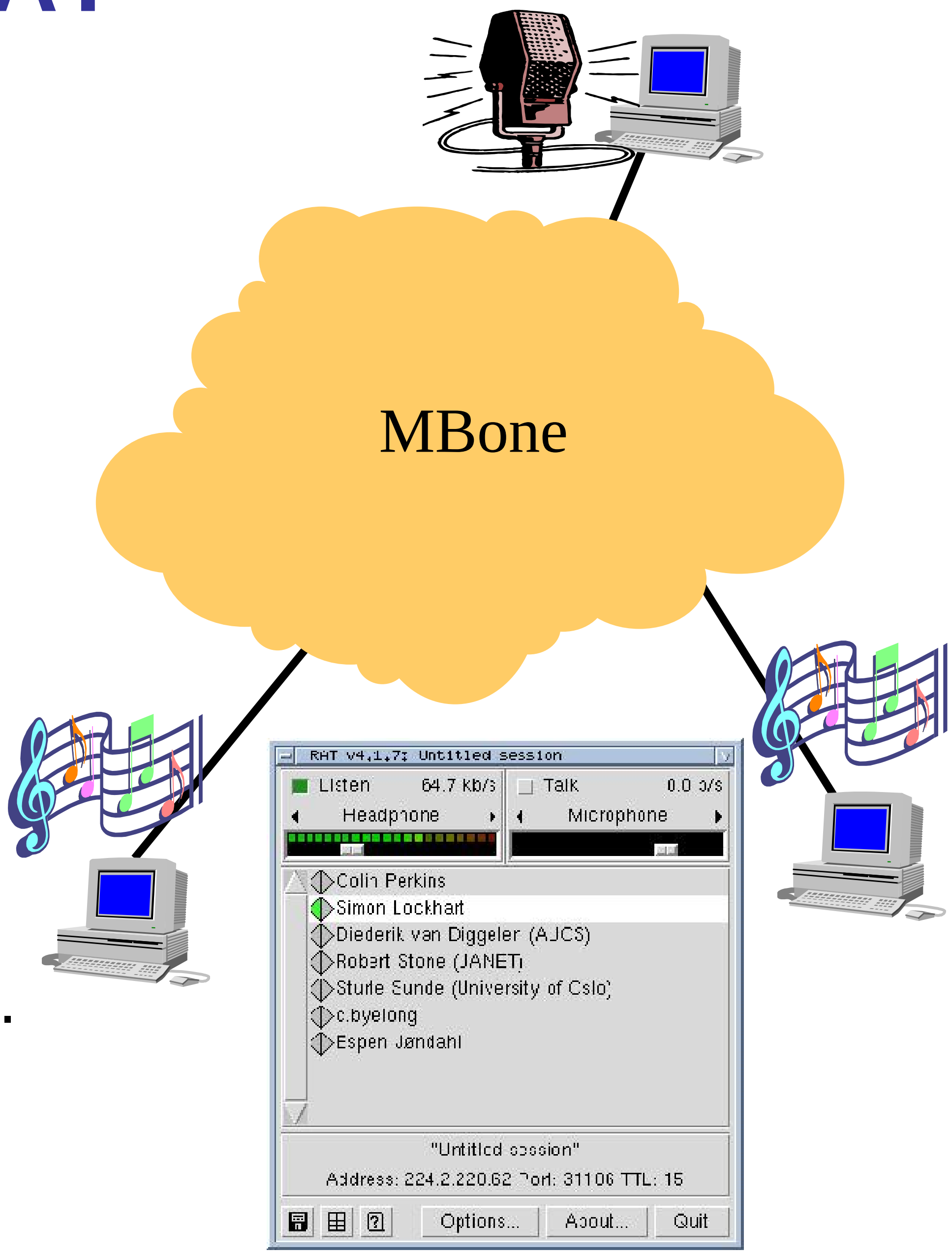


Tunnel connection kit

- use mrouted tunnel (IP-in-IP)
- mTunnel http://www.cdt.luth.se/~peppar/ progs/ mTunnel/
  - tunnels multicast packets over an unicast UDP channel
  - several multicast streams can be sent over the same tunnel while the tunnel will still only use one port (useful if tunneling through a firewall).
  - t he applications primary goal is to allow for easy tunneling of multicast over for instance a modem and/or an ISDN connection

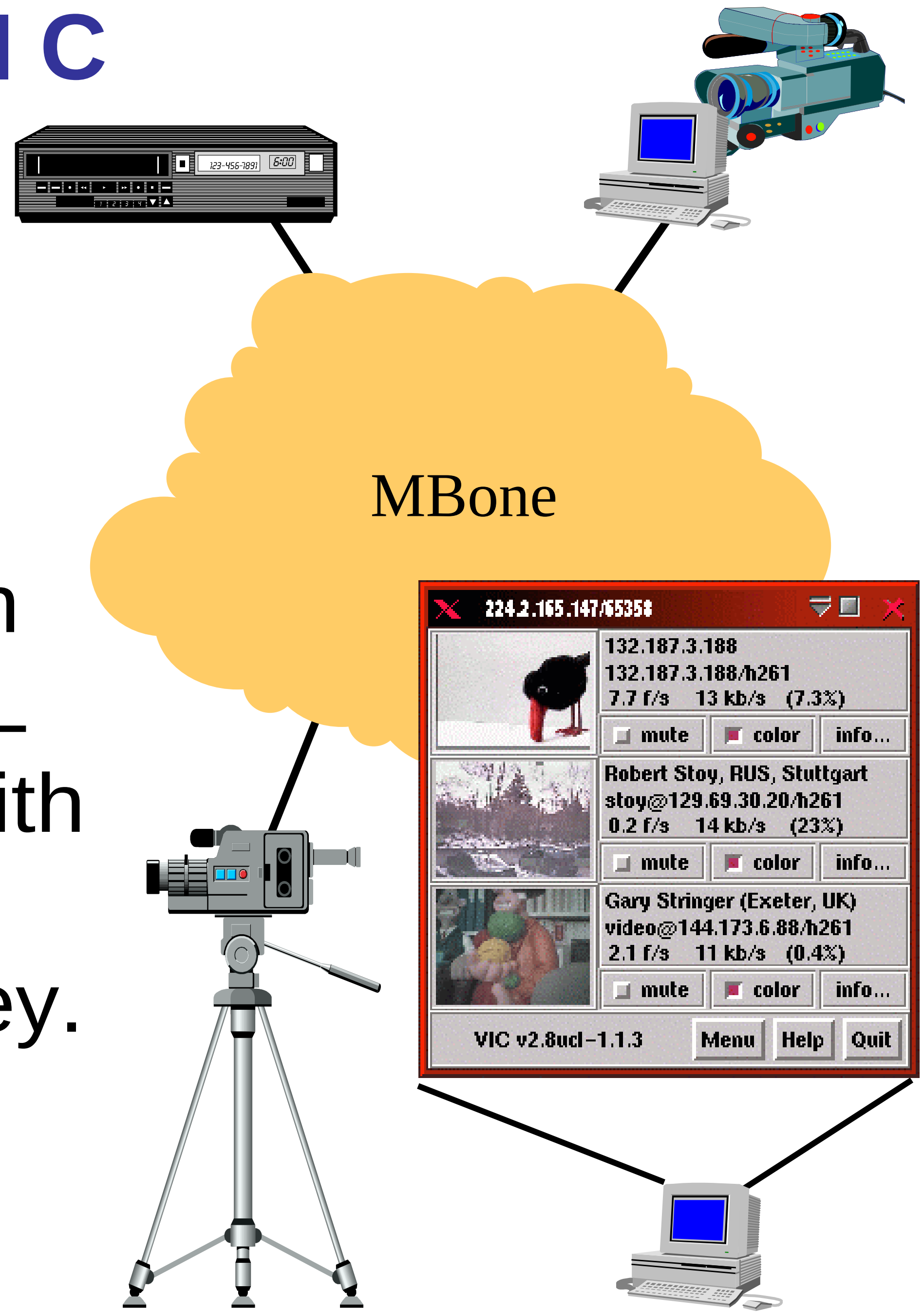
MBone tools - RAT

- The **Robust Audio Tool** (RAT) is a an open-source audio conferencing and streaming application that allows users to participate in audio conferences over the internet. These can be between two participants directly, or between a group of participants on a common multicast group.



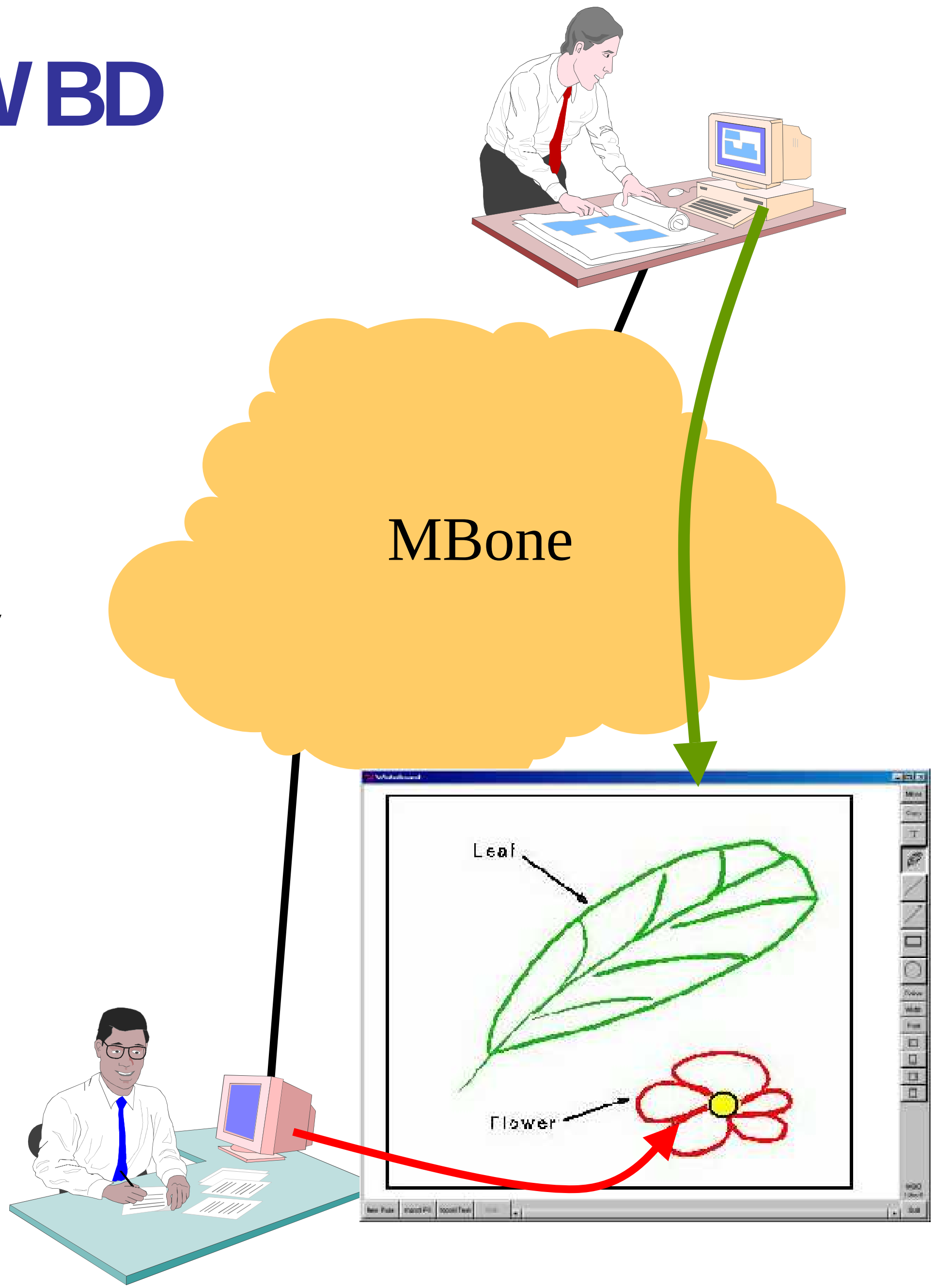
MBone tools - VIC

- **VIC** is a video conferencing application developed by the Network Research Group at the LBNL in collaboration with the University of California, Berkeley.



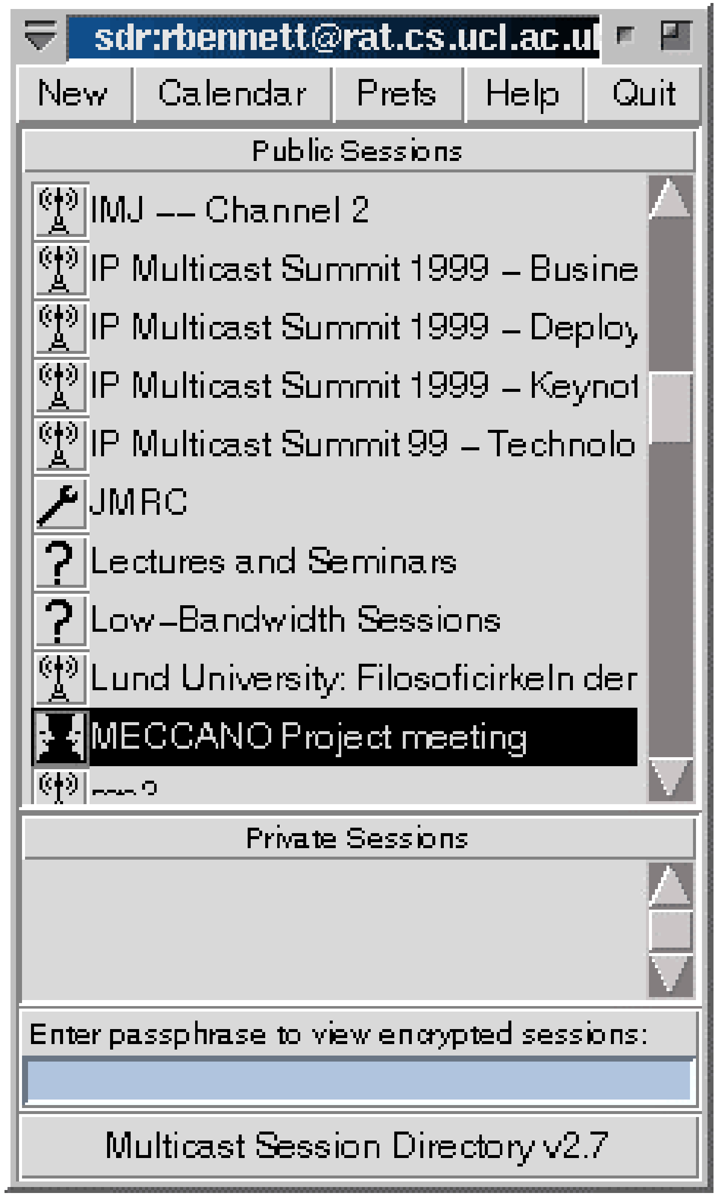
MBone tools - WBD

- WBD is a shared whiteboard compatible with the LBL whiteboard, WB. It was originally written by Julian Highfield at Loughborough University and has since been modified by Kristian Hasler at UCL.



MBone - Advertising sessions

- **SDR** is a session directory tool designed to allow the advertisement and joining of multicast conferences on the Mbone. It was originally modelled on **sd** written by Van Jacobson at LBNL.



MBone ressources

- MBone:

  http://www.lbl.gov/ web/MBONE.html

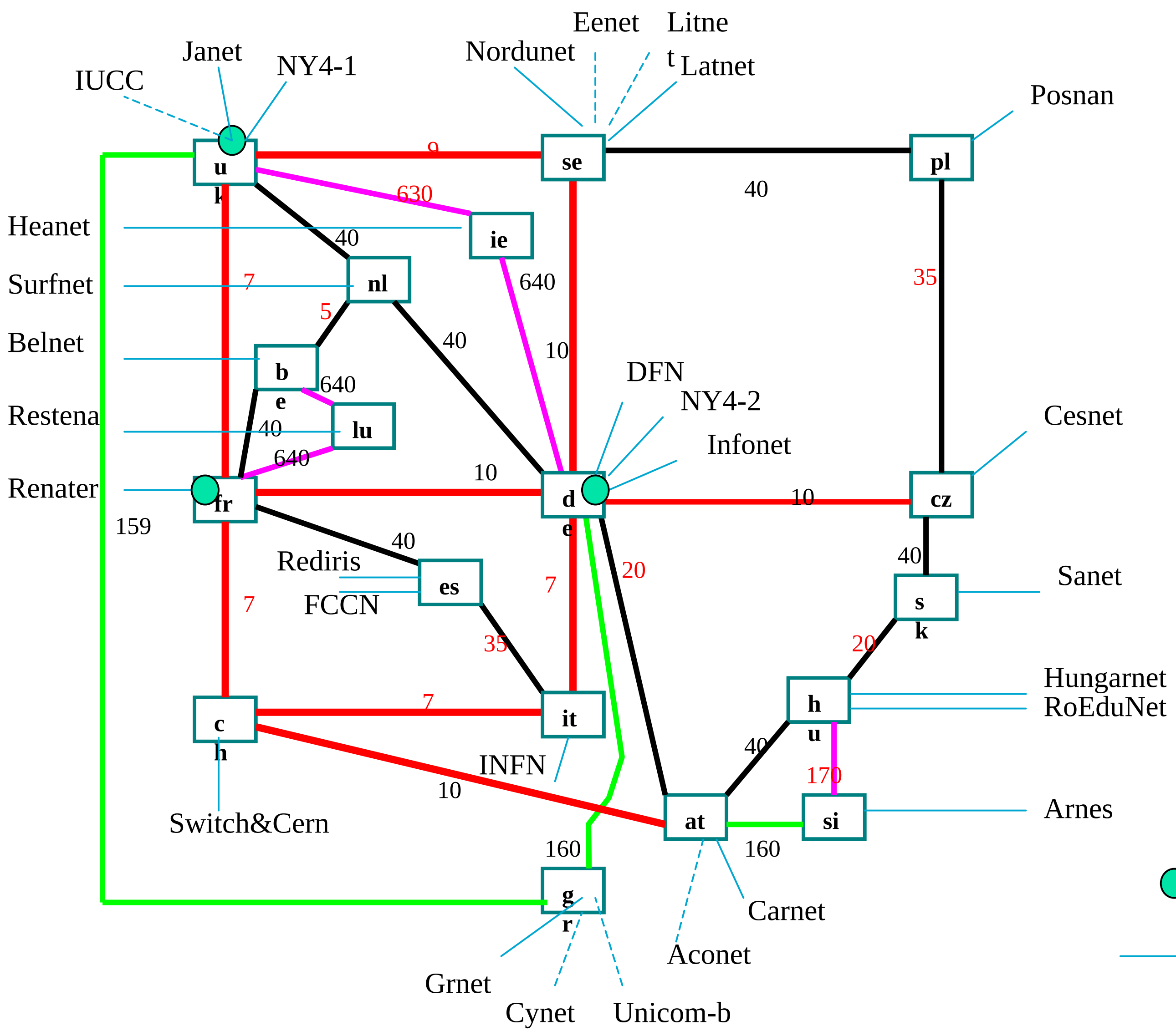
- MBone software:

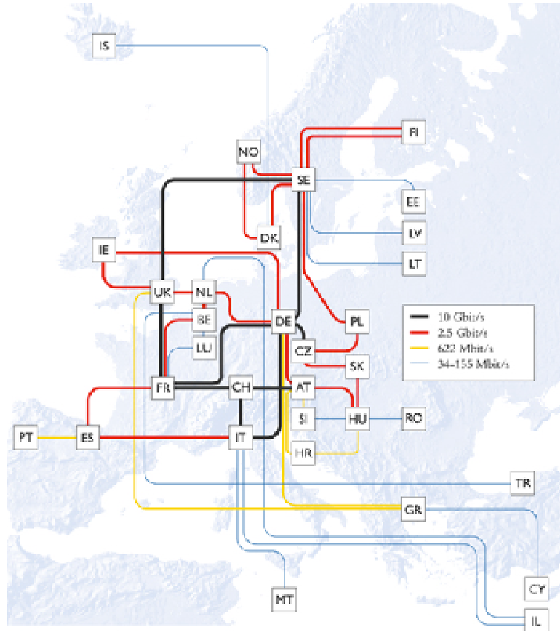
  http://www-mice.cs.ucl.ac.uk/ multimedia/software/

- MBone topology, statistics

  http://www.multicasttech.com/ status

2003 - Multicast on GEANT network





Rendez-vous Point

Multicast access

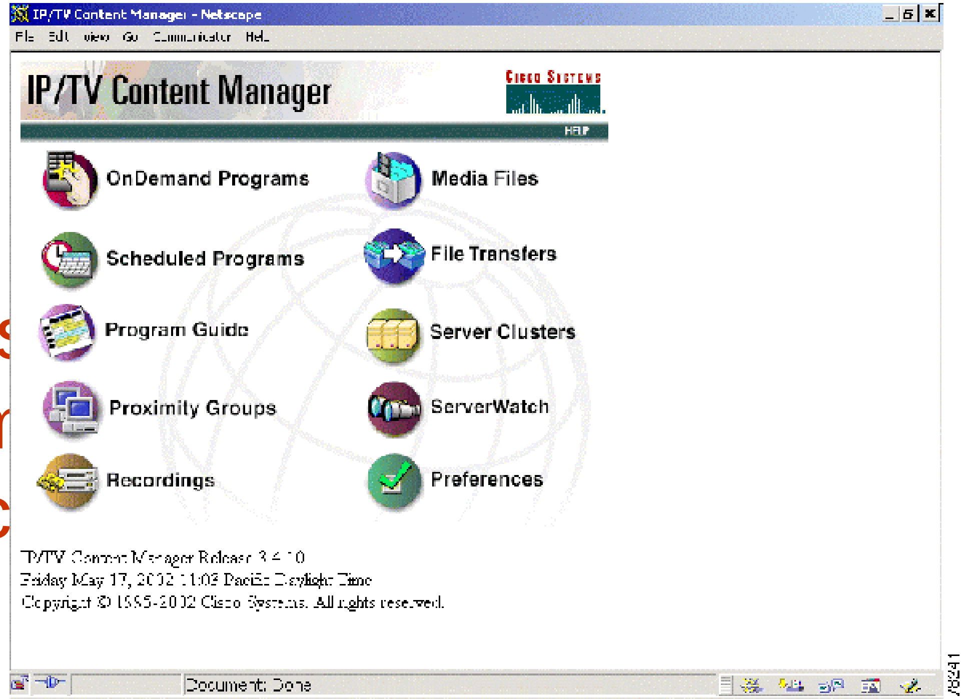
source http://www.dante.net/nep/GEANT-MULTICAST/

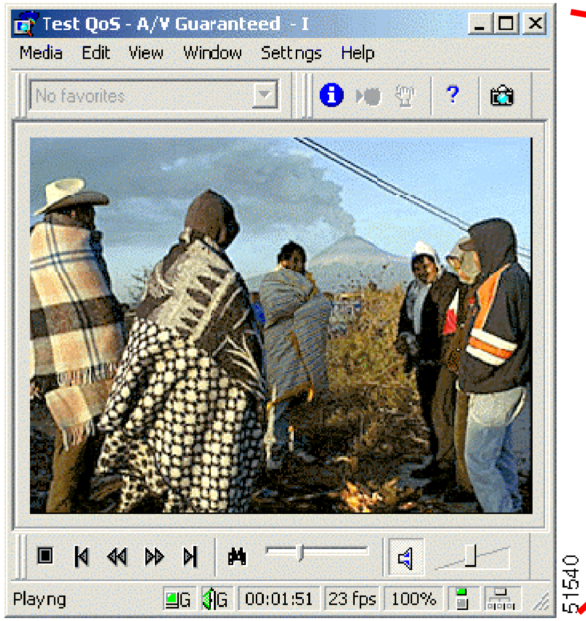
Selection of other commercial/prototype products

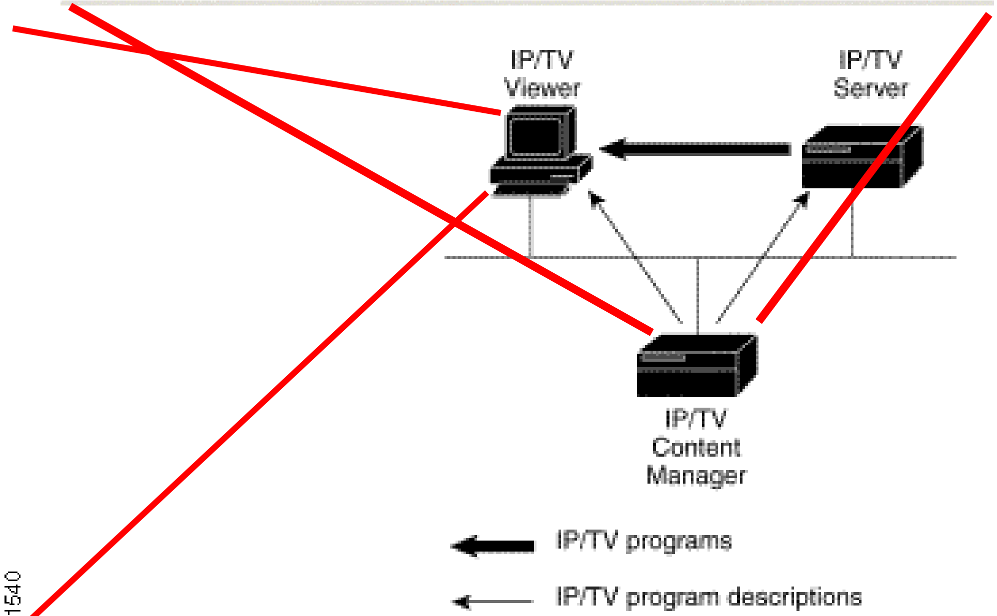
- CISCO IP/TV, CISCO IP/VC
- XtremeCast from mPulse
- Digital Fountain
- Multicast Monitor
- much more
  - RendezVous, Freephone,
  - MASH, CMT, MultiMon, NTE
  - MPOLL

CISCO IP/TV,

- Usages
  - Training, Busines
    Corporate Comm
    Learning, Videoc





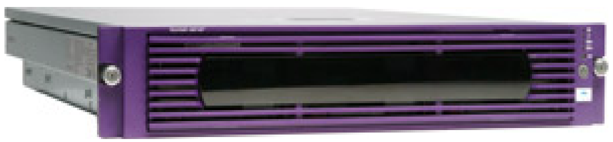


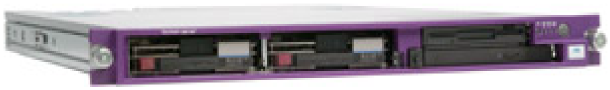
XtremeCast from mPulse

- Usage
  - Used by financial firms for stock quotes broadcasting
  - Chat server
- Reliable multicast implementation with the JRMS (from SUN) library

Digital Fountain products

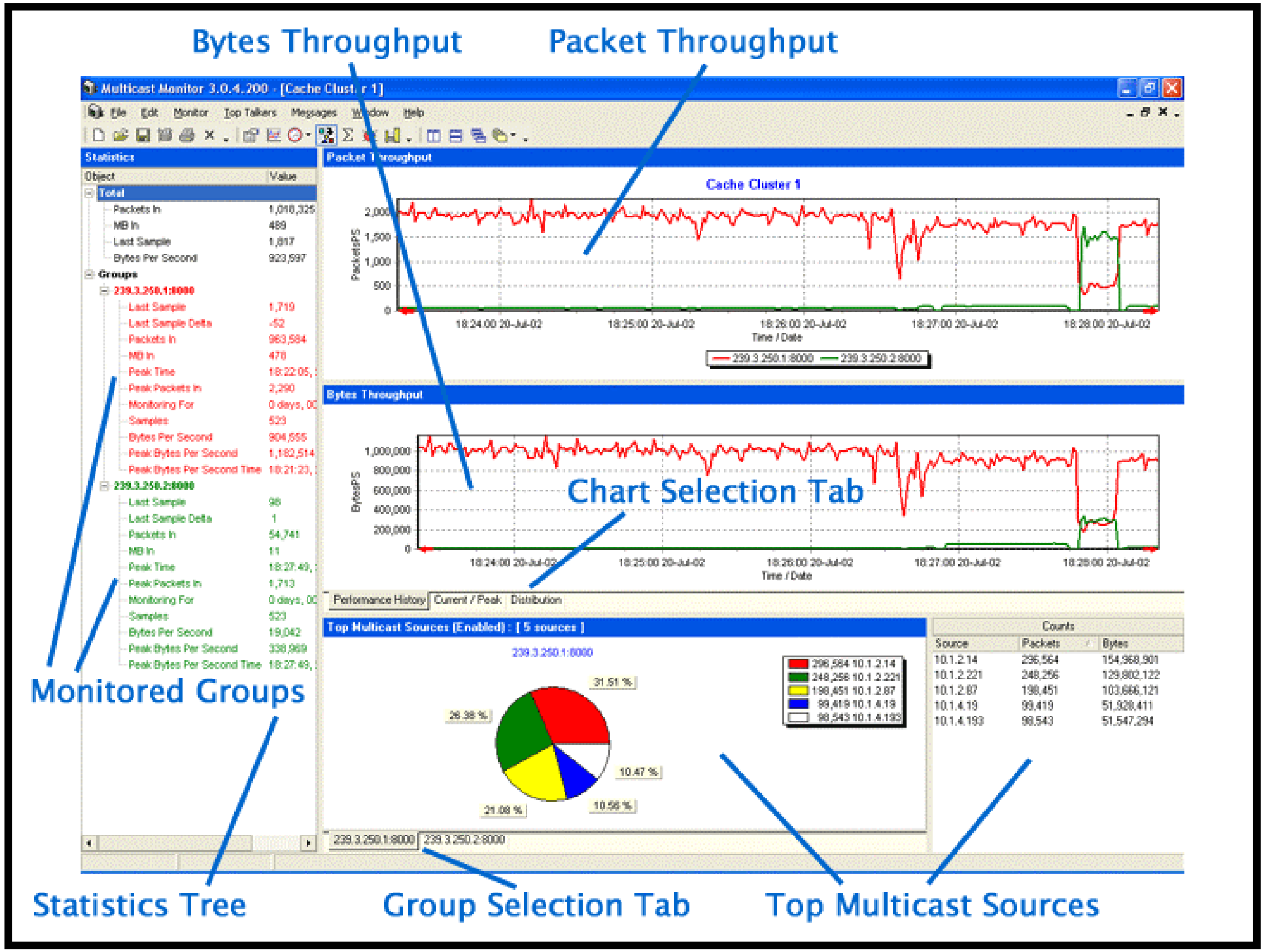
- Implement ALC/LCT/WEBRC and rely on two highly efficient large block FEC codecs
  - http://www.digitalfountain.com
  - high implication in the IETF RMT standardization process





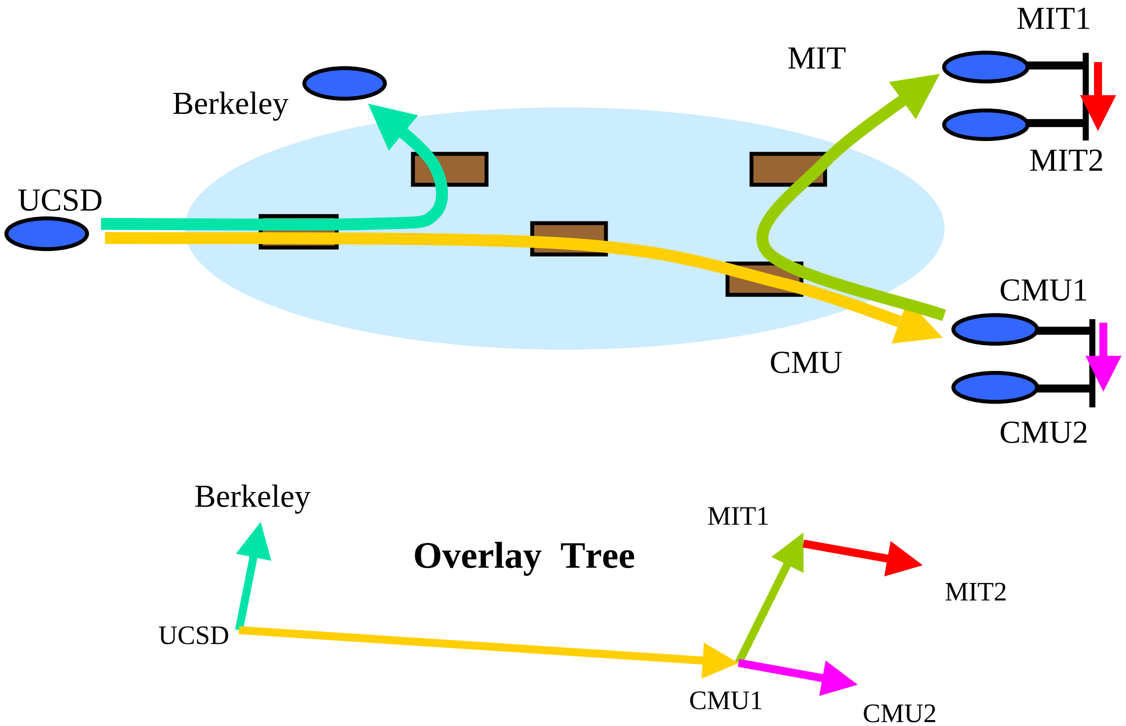
Multicast Monitor

monitor multicast traffic in the entreprise network



Last solution…

- **if** you don't have access to IP Multicast you could **try** using:
  - Overlays, End-system Multicast, Host-level, Application-level Multicast



source Yang-hua Chu

Conclusions

Conclusions (1)

- Multicast: a technology with high potential…
  - … but also awfully complex !
- Technology starts to be mature:
  - problems are well known and some protocols are already standardized (ALC family)
  - ACK/NACK protocols are on the way to standardization (takes more time as problems are tougher)
  - does not prevent the use of private reliable multicast solutions

Conclusions (2)

- Deployment is mainly driven by academic networks…
  - where are the killing applications ?
  - video and popular content distribution to clients… yes
  - high performance computing over datagrids... yes
- Where should we go?
  - More specific models (i.e. SSM),
  - More security, more control

Slides will be available at
www.ens-lyon.fr/~cpham
www.inrialpes.fr/planete/people/roca

Part IV

Short bibliography

Short Bibliography

- IETF RMT working group

  http://www.ietf.org/ html.charters/rmt-charter.html

- ALC, layered CC, FEC documents

M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, ``**Asynchronous Layered Coding (ALC) protocol instantiation**'', RMT Working Group, rfc3450.txt, December 2002.

M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, M. Handley, ``**Layered Coding Transport (LCT) building block**'', RMT Working Group, rfc3451.txt, December 2002.

M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, J. Crowcroft, ``**Forward Error Correction (FEC) building block**'', RMT Working Group, rfc3452.txt, December 2002.

- M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, J. Crowcroft, ``**The use of Forward Error Correction (FEC) in Reliable Multicast**'', RMT Working Group, rfc3453.txt, December 2002.
- M. Luby, V. Goyal, ``**Web and Equation Based Rate Control building block**'', RMT Working Group, draft-ietf-rmt-bb-webrc-04.txt, December 2002.

Short Bibliography…(cont’)

- NORM, single layer CC documents

B. Adamson, C. Bormann, M. Handley, J. Macker, ``**NACK- Oriented Reliable Multicast (NORM) Protocol Building Blocks**”, RMT Working Group, draft-ietf-rmt-bb-norm-04.txt, July 2002.

B. Adamson, C. Bormann, M. Handley, J. Macker, ``**NACK- oriented reliable multicast (NORM) protocol**’’, RMT Working Group, draft-ietf-rmt-pi-norm-05.txt, November 2002.

L. Rizzo, G. Iannaccone, L. Vicisano, M. Handley, ``**PGMCC single rate multicast congestion control: Protocol Specification**”, RMT Working Group, draft-ietf-rmt-bb-pgmcc-01.txt, June 2002

J. Widmer, M. Handley, ``**TCP- Friendly Multicast Congestion Control (TFMCC): Protocol Specification**’’, RMT Working Group, draft-ietf-rmt-bb-tfmcc-01.txt, November 2002

Short Bibliography… (cont’)

- TRACK documents

  B. Whetten, D. M. Chiu, M. Kadansky, S. J. Koh, G. Taskale, B. Levine,

  ``**Reliable Multicast Transport Building Block: Tree Auto- Configuration**'', draft-ietf-rmt-bb-tree-config-03.txt, November, 2002

  B. Whetten, D. M. Chiu, M. Kadansky, S. J. Koh, G. Taskale, ``**Reliable Multicast Transport Building Block for TRACK**'', RMT Working Group, draft-ietf-rmt-bb-track-02.txt, November 2002.

  T. Speakman, L. Vicisano, ``**Reliable Multicast Transport Building Block Generic Router Assist - Signaling Protocol Specification**'', RMT Working Group, draft-ietf-rmt-bb-gra-signalling-01.txt, January 2003.

- Reliable multicast

  S. Floyd, V. Jacobson, C. Liu, S. McCanne, L. Zhang, "**A Reliable Multicast Framework for Light- weight Sessions and Application Level Framing**", IEEE/ACM Transactions on Networking, December 1997, Volume 5, Number 6, pp. 784-803.

  M. Maimour, C. Pham, "**Dynamique Replier Active Reliable Multicast (DyRAM)**", Proceedings of ISCC'02, Taormina, Italy.

  MCL project web page (ALC implementation, NORM implementation soon)
  http://www.inrialpes.fr/planete/people/roca/mcl/
  includes pointers to related documents/other RM implementations