# Advanced Topics in Networking

## George Varghese

August 9, 1999

In recent years the Internet has been transformed from the playground of a few hackers to trendy hangout. —— *paraphrased from Scott Shenker*



## Goals

- **1.0** Provide background on more recent Internet protocols.
- **2.0** Describe some Internet Protocol Implementation Bottlenecks.

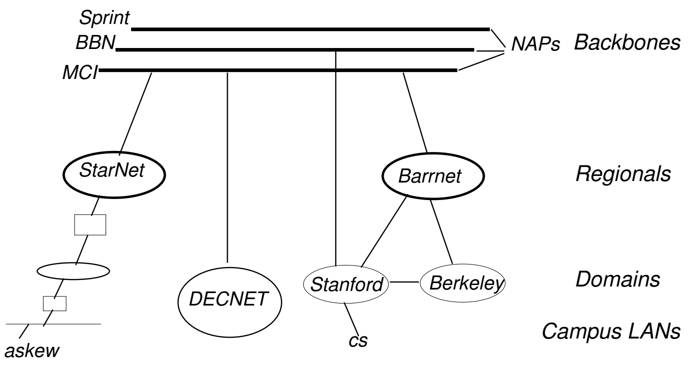
Some interesting research questions are thrown in.



## 1.0 Internet Protocols

- **1.1** Interdomain Protocols and Policy Routing
- *Segue:* Internet Future Requirements and Goals
- **1.2** Multicast Routing
- **1.3** Mobility Support
- **1.4** Providing Service Guarantees
- **1.5** The Next Generation Internet

## Internet Topology






## Basic Internetworking

- Internetworking was a specific goal of IP (unlike say DECNET, SNA etc.). Starts with an implicit hierarchy of physical networks (with network specific routing that IP does not care about) and an internetwork of physical networks. Each physical network has a network number.
- IP's job is to route to the right physical network based on the network number. Offers a datagram service with possible fragmentation and reassembly to deal with different MTU sizes.
- IP always configured with a companion protocol called ICMP for error messages (header checksum failed, TTL reached.) redirect)



## 1.1 Policy Routing between Domains



## Why Interdomain Routing

Why not one happy melting pot of a network:

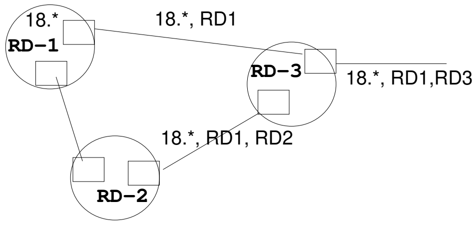
- Multiple providers (see IP evolution) implies need for independence and independent policies.
- Different metrics, trust patterns, different charging policies (hot potato, cold potato), different administrative and legal requirements (e.g., ARPANET only for government business, Canadian traffic stays within Canada).
- Not very well developed. Basic conflict between abstraction and hierarchies (for scaling) and ability to specify arbitrary policies.

## Possible Policies

- Never use Routing Domain X for any destination.
- Minimize number of domains in path.
- Government messages can traverse the ARPANET but not others.
- Use the set of domains whose combined cost is least.



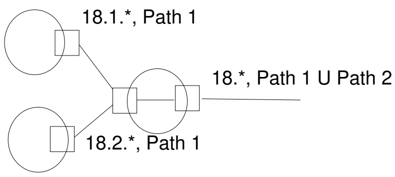
## The Border Gateway Protocol(BGP)




- Path vector scheme allows both policy flexibility and loop free routing.
- Per source domain policy can cause information to be lost.



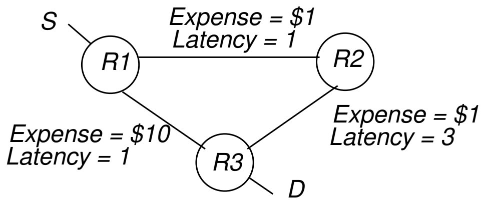
## Some BGP features




- Can aggregrate addresses to improve routing table scalability by taking unions of paths. Uses TCP between BGP neighbors for incremental updates.
- Conclusion: Some loss of flexibility. But does it even *work?*



## Research Issues




Does BGP work?

- Examples galore of policies where BGP Loops (see Govindan, Estrin et al, see Estrin's home page)
- Examples of BGP implementation instability (Liebowitz et al, SIGCOMM 98)

Design a correct scalable policy routing protocols that handles as wide a variety of policies as possible.

## Segue: Aspects of the Future

- Multicasting *MBONE protocols*
- Mobility and Ubiquitous Computing (Laptops, PDAs) *Mobile IP*
- Increased Scale (users, required and available bandwidth) *IP v6*
- Integrated Services (voice, video, data, telemetry) *RSVP, fair queuing.*
- Accounting *TBD*
- Better ways to access information and use the network *TBD*



## Segue: Goals for any new Internet Protocol

- Internet is big: protocol needs to be scalable to large numbers
- Internet topology is always changing: protocol needs to be adaptive
- Internet has many existing components: must work with all existing ones
- Internet is decentralized: must be capable of incremental deployment.
- Internet protocols must change with technology: must be modularly designed so independent parts can be replaced.
- Internet is successful because of diversity: must support heterogeneity.



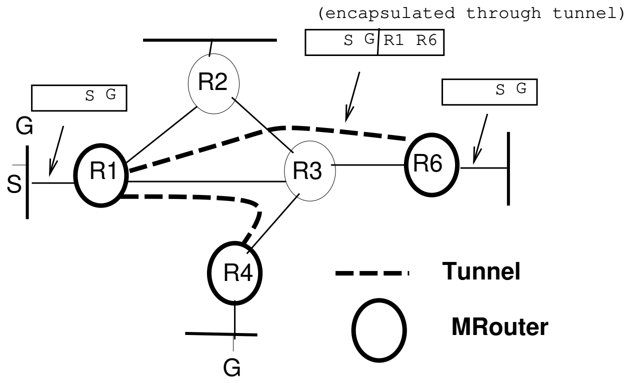
## 1.2 Multicast Routing



## Context

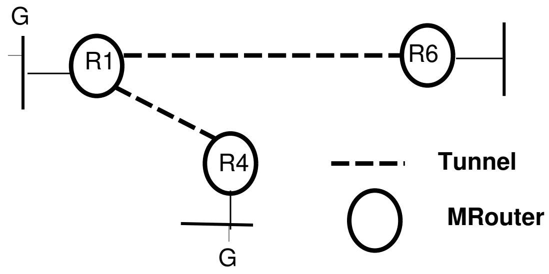
- Multicast needed for efficient data delivery, voice and video broadcast, and for autoconfiguration.
- Works today using a flat overlay network called the MBONE where mrouters are connected by tunnels through which packets travel encapsulated. MBONE using separate routing protocols and is not hierarchical.
- Several protocols (DVMPRP, CBT, MOSPF, and PIM)

## MBONE example





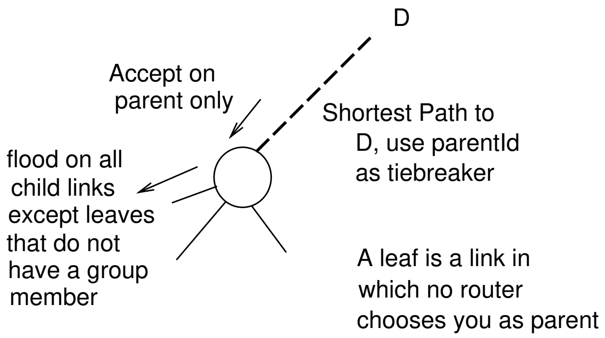

## Logical topology of MBONE example




- Notice how logical topology differs completely from original topology, Requires separate routing protocols, exchanges, and tables.



## RPB with Truncation

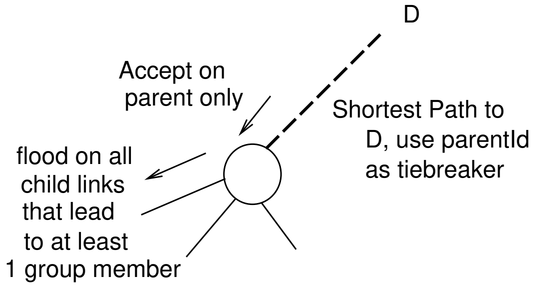





## Finding Children and Leaves

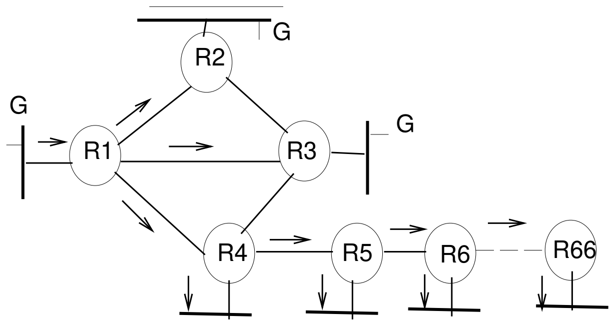
- Each child is responsible for telling its parent that that the link has a child. In DVMRP, which uses Bellman-Ford, this is done by Poison Reverse (reporting infinity to D to parent) which also helps avoid count up.
- Parent can then easily tell whether a link is a leaf if it contains no children.

## RBP with Pruning



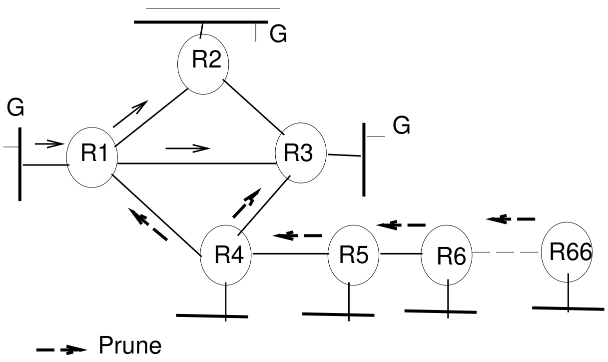



## RPB Example



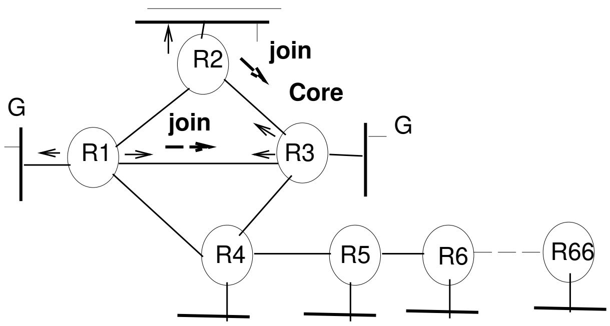



## Pruning Example

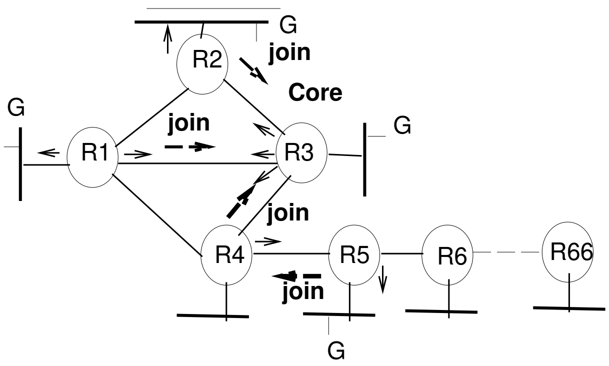





## Core Based Trees Example



## Core Based Trees: joining






## MOSPF

- Extends LSPs to report group membership. Each router has complete map including all group members.
- Like DVMRP calculates per source per group trees. Each can be calculated using Dijkstra but too many possible ones. So calculate trees on demand basis. (Lazy evaluation)
- DVMRP does pruning lazily on a data driven basis; MOSPF does shortes path calculations lazily but prunes "in memory". Bandwidth versus latency tradeoff. MOSPF also sends info about receivers to routers who do not have group members.



| | Basis | Memory | Bandwidth | Optimality | Lookup | Latency |
|---|---|---|---|---|---|---|
| **DVMRP** | Bellman Ford + RPB | O(SG) | Broadcast + prune | Shortest Path | S,G | Low |
| **CBT** | SPT, core as root | O(G) | Periodic joins on SPT | Off by 2? | G | Low |
| **MOSPF** | LSPs Dijkstra | O(SG) | Larger LSPs | Shortest Path | S,G | High on startup |
| **PIM-DM** | Bellman Ford + RPB | O(SG) | Broadcast + prune Worse than DMRP | Shortest Path | S,G | Low |
| **PIM-SM** | Like Core Based Trees with RPs replacing Core Routers | | | | | |



## PIM

PIM (Protocol Independent Multicast) defined by IDMR working group. Goals:

- Distinguishes between dense topologies (group members densely packed and bandwidth is plentiful as in a local campus) and sparse mode (opposite assumption.) Wants to work well in both environments by using DVMRP and CBT.
- Wants to be independent of the Unicast Routing Protocol and wants to reuse as much of that protocol as possible without trying to modify it. Should work over LSP or Bellman-Ford protocols.

## Research Issues

- Aggregation for multicast forwarding. Prefixes aggregate unicast well to reduce backbone entries.
- Combining multicast and policy routing. (See SIGCOMM 98. Estrin et al)
- Multicast address allocation. (See SIGCOMM 98, Handley)
- Correctness for Core Based Trees (counterxamples by Garcia-Luna et al Infocom 97) and PIM.



## 1.3 Routing to Mobile Hosts



## Normal IP

Today without mobility support, a node must be located on the network indicated by its IP address. Thus:

- A node must change its IP address whenever it moves (breaking long lived transport connections) or
- Host specific routes must be propagated by Internet routers (breaks scaling)

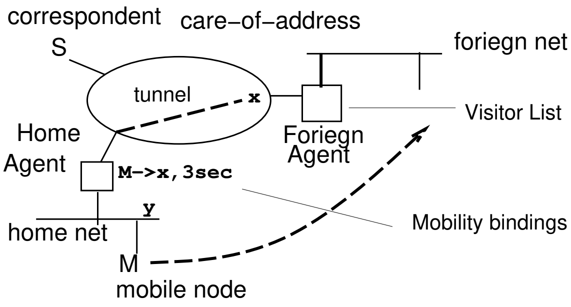
Hence want mobility suport



## Goals

- Mobile node must be able to communicate with other nodes after changing location without changing its IP address.
- Mobile must be able to communicate with other nodes that do not support mobile IP (incremental deployment).
- Security issues involved in redirection must be addressed.
- For low bandwidth, high error rate links with low batteries, must reduce the number of control messages.
- 

## Basic Idea




- Agent advertisements, agent solicitation, agent discovery, registration, security bindings



## Header Compression

Mobile links are small bandwidth (infrared LANs 1 Mbit). Cellular phone techlologies even lower and high error rate. IPv6 headers and tunneling causes lots of overhead. 2 Ipv6 headers plus TCP = 100 bytes

- Telnet packets have very little data (1 byte). Extra headers causes large latency/
- Voice packets need to be transmitted often. Lots of small packets
- Even wide area file transfer (512 bytes) can have 20 percent overhead

Van Jacobson and soft-state header compression. Now being standardized.



## VJ TCP + IPv4 Header Compression

- Fields that do not change between two successive packets are not sent. Fields that do (sequence numbers, windows etc.) are encoded by their differences. Full checksum sent.
- Starts with a flag byte that identifies which incremental fields are being sent (e.g., sequence, ack etc.)
- If one packet is lost, all packets in window are encoded incorrectly and so fast retransmission and selective ack are useless. Detect error when compressor sees a retransmission, which prompts a full header transmission.



## Research Issues in Mobility

- Combining Multicasting with all possible schemes (see Mobile IP spec)
- Mobile routers on planes! (see Mobile IP spec)
- Fault tolerance (see Huitema's book)
- Tradeoffs between search and tracking.
- Truly scaleable mobile routing (using aggregation and hierarchies)

## 1.4 Quality of Service



## Context

- With the advent of multicast video and audio, its is important to have routers modified to give guarantees to such stringent applications using reservations.
- RSVP is a mechanism by which receivers in large multipoint conversations (several senders, multiple receivers) send their reservation requests to routers.
- Other modules such as admission control (decide whether to accept a reservation) or scheduling (enforce reservations) interact with RSVP to do the whole job.



## Why Scheduling

- For fairness (large jobs should not block small jobs)
- Provide guarantees for inelastic applications which require a min level of performance (e.g., min latency for video or control device)

Implications:

- Can only choose a subset of inelastic flows to guarantee performance (Admission Control)
- Must schedule admitted flows so that each gets guarantees (Scheduling)



## Single CPU

- Time slices for fairness; different quanta
- Consider say video job requiring 1 msec every 20 msec.
  - Admission Control: $<$ 20 video jobs
  - Scheduling: Round Robin giving priority to video
- Not quite as trivial because video jobs may have different periods and data may arrive asynchronously. Mostly use either Earliest Deadline First (EDD) or Rate Monotonic, shortest period first (RM)

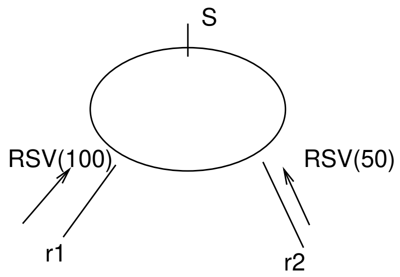
## Network Routers and Network Flows

Precious resource to schedule is no longer CPU but output link bandwidth.

- For fairness, use multiple queues for an output link and a scheduler for each output link. Can't use time slicing. Next lecture for specific schemes.
- A flow may request BW at each link. Admission control may reject new flow is sum of requested plus allocated bandwidth exceeds link bandwidth.
- Traditionally sources set up reservations towards receivers.



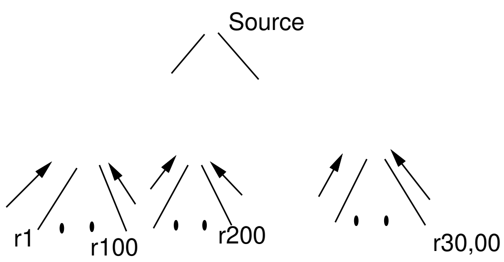
## Designing RSVP




- Since receivers can be heterogeneous, only way to know about their needs is for receivers to "tell" sources by sending a reservation of their unique needs towards source.
- For now, assume we know somehow where "towards the source" is. Will return to this.
- Since we want to deal with lost messages and timeout receivers who are dead, we want to send RSV messages periodically.



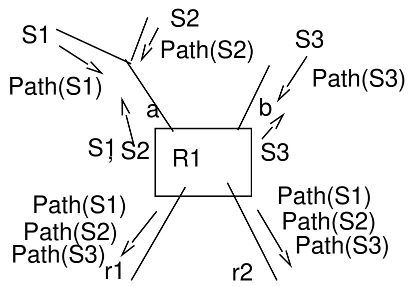
## Receiver aggregation




- Don't want sources to deal with huge numbers of potential resource reservations for scaling.
- Only way is to have RSVP routers merge requests before forwarding uptree.



## Creating source trees by Path Messages




- Path messages from source are forwarded along multicast tree for the group. When arriving at router, router saves Path State that indicates which direction sources are reachable.
- Path state per link is used to send Path state downwards. Path state per link is timed out which allows RSVP to adapt to different topologies.

## Research Issues

- Simpler ideas like Differential Service using Priority being proposed. Is RSVP too complex?
- Granularity of reservations? Interactions with Accounting.
- Bugs already discovered make a proof of correctness desirable.



## 1.5 The next generation IP, IPv6



## Context

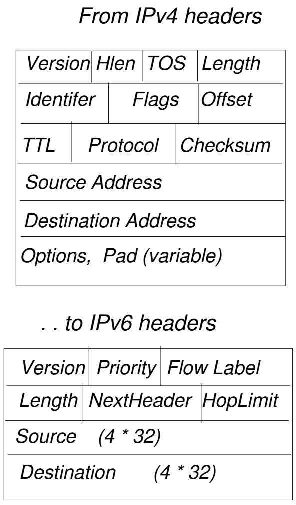
- Subnetting and BGP-4 (which helped CIDR to be deployed) have contained scaling problems. Still 4 billion hosts are not going to enough, especially with address inefficiencies.
- IETF began looking at the need for larger addresses in 1991. Requires a new version. While doing so, might as well fix old problems (e.g., autoconfiguration) and add new features (e.g., real time, security, mobility).
- IETF appointed a committee called IPng and solicited white papers. IPv6 proposal with 128 bits. Unfortunately, temporary hacks like NAT boxes and the complexity of v6 are making the prospects of a quick v6 rollout slim.



## Perspective on IPv6 (RFC 1883

- First, its mostly header formats and addressing modes. No protocol because that is assumed to be extensions of existing protocols (RIP, BGP etc.) and forwarding mechanisms (Best Matching Prefix etc.)
- Want to streamline packet formats to allow routers forwarding vanilla data packets to be able to forward packets at Gigabit rates
- Want to allow use of old features like source routing (*backward compatibility*) and use of new features being worked on and yet to be added (*future compatibility*) without having simple data forwarding to be greatly affected.

## Comparing Headers






## Streamlining for Fast Router Processing

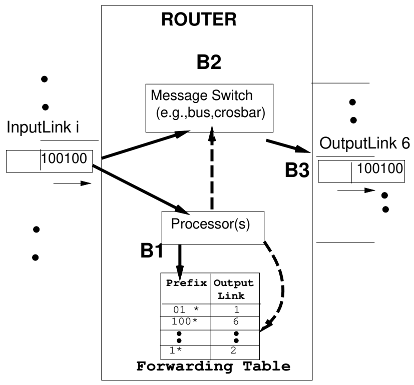
- Most extensibility features hidden in extension headers and extensions. Extension headers unified with demultiplexing field.
- Extension headers are ordered (most of the time) and separated into destination and hop-by-hop for further streamlinined processing by routers.
- Features moved to different points to retain fast processing while retaining backward compatibility (e.g., fragmentation: from real time to compile time for Path MTU discovery, and larger min MTU size achieved by requiring transparent hop-by-hop fragmentation if necessary.
- Fixed length headers and 32/64 bit alignment wherever possible. Handle exceptions at more cost (variable length options by padding, escape hatch for jumbograms).
- Ability to add new options and have sensible policies be implemented by old routers that do not understand new options (much like BGP attributes).
- One new feature (real time, quality of service) is given two fields in based header (flowId and priortity). flowId can be also used for efficient lookup.
- Header checksum eliminated. TTL becomes hop count.
- In some sense, optimizing expected case and moving features to different time scales (time) and nodes (space).



## 2.0 Implementation Bottlenecks



## Router Model



## On Algorithms in Systems

I don't do algorithms, son — *Famous system designer, to Maurice Herlihy.*

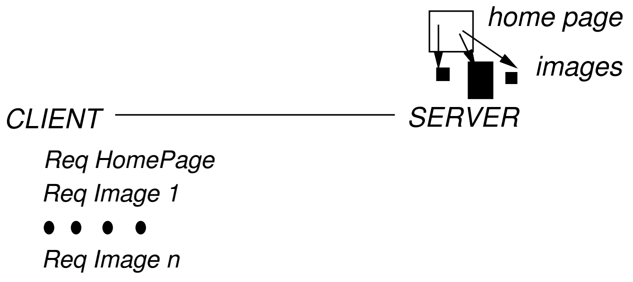


## Efficiency Principles

- Avoid Waste in Sequences of operations
- Shift computation in space and time
- Relax subystem requirements
- Leverage off other system components
- Pass information between rouines and in messages
- Specialize general purpose routines
- Optimize expected Case
- Use or invent efficient data structures and algorithmic paradigms.



## Why Optimizations Can be Hard




- Simple ideas interact with artifacts of rest of system. Complexity in breadth not depth.
- Crucial to first identify bottlenecks and build prototypes to validate that improvements are real.



## Some important bottlenecks

- **2.1** Best Matching Prefix Lookups: *Lulea, Wash U Schemes*
- **2.2** Fair Queuing *Virtual Clock, Worst Case Weighted Fair Queuing, Deficit Round Robin*
- **2.3** Filter Matching: *Rectangle Search*

## 2.1 IP Lookups

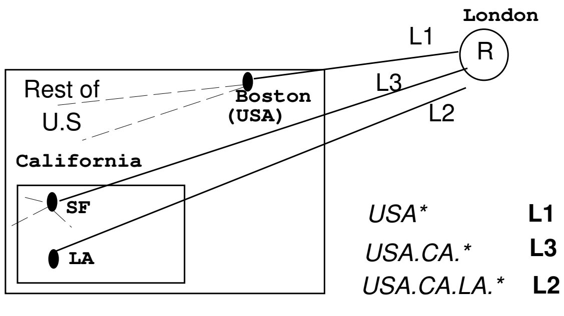


## The need for Increased Router Performance

- Internet Hosts doubling every two years, traffic increasing exponentially, new applications, transition from 32 bit IPv4 to 128 bit IPv6.
- Requires faster links (e.g., 45 Mbit to 155 Mbit and faster) and and *faster routers*. 50 percent of packets are 40 byte TCP ACKs.
- Fastest Routers today forward 100,000 to 500,000 packets per second. Need to process millions of packets to keep up with a Gigabit link.



## Is Prefix Matching Fundamental?




- Reduction in database size (from 1 billion to 40,000 prefixes) but at cost of longest matching prefix.



## Routing Environment

- Backbone routers: 45,000 prefixes, all lengths from 8 to 32; density at 24 and 16 bits. Unstable BGP implementations may require route updates in 10 msec. May need programmability. Bimodal packet sizes (65, 576). 250,000 concurrent flows, caching works poorly.
- Enterprise routers: (1000 prefixes), route changes in seconds, bimodal (64, 1519).
- IPv6 plans heavy aggregation but multihoming, Anycast etc, make large backbone databases likely.

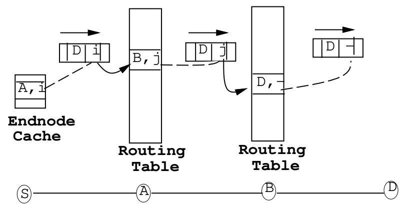
## Avoiding Router Lookups by Protocol Changes

Basic idea: change protocol to reduce best matching prefix to a memory lookup.

- ATM: do a set up protocol to replace IP address with a Virtual circuit index. **Problem: extra Round trips, drastic changes**
- Source hashing: add a random index to each packet address to allow faster cache lookups. **Problem: depends on locality, no worst-case improvement**
- Tag switching: add an index to each packet that is an index into next hop's forwarding table. **Problem: does not work well at hierarchy boundaries, requires changes.**
- IP switching: pass a VCI to previous hop **Problem: as in tag switching.**



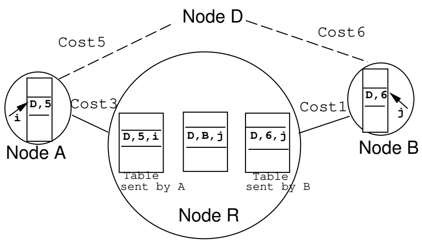
## Threaded Indices (Chandranmenon-Varghese 95)




- Like VCI lookup for datagram networks. (Sigcomm 95). However:
- Only one index per *destination* and not one index for every active *source-destination* pair. Now being proposed in CISCO Tag Switching.



## Computing Threaded Indices




- Unlike VCIs, indices are set up at *topology change* time. No round-trip start up delays.
- Easy to modify Bellman-Ford and LSP routing as well.



## Older approaches to best matching prefix

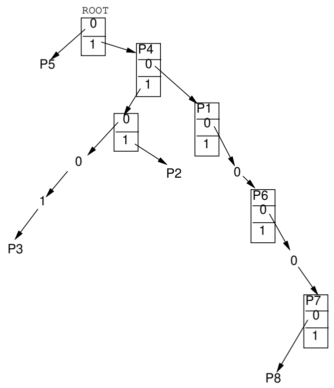
- Patricia tries. **: Problem**: requires up to 32 memory accesses, backtracking in Patricia, excess memory.
- Binary Search: By expressing a prefix like 10* as 10..0 and 10F..F we can use binary search. **Problem:** requires $\log_2 2N$ time, 16 probes to memory for 30,000 prefix table.
- Use caches keyed by full IP address: **: Problem**: only 80 percent hit rate observed in practice. (5 packet web exchanges?)

## Three New Approaches

- Expanded Multibit Tries (SIGMETRICS 98) Srinivasan, Varghese
- Lulea scheme based on compressed multibit tries. (SIGCOMM 97) M. Degermark et al
- Wash U Binary Search on Prefix Lengths. (SIGCOMM 97) Waldvogel et al



## Compressed Trie using Sample Database





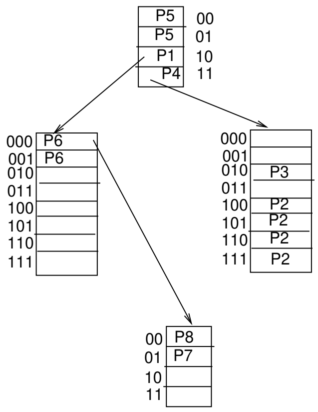

## Controlled Prefix Expansion

Reduce a set of prefixes with $N$ distinct lengths to equivalent prefixes with $M < N$ distinct lengths. Recursively apply two simple ideas:

- **Expansion:** Expand prefix $P$ of length $L$ into two equivalent prefixes $P0$ and $P1$ of length $L+1$.
- **Capture:** When a prefix expanded from Level $L$ to $L+1$ has same value as a prefix at Level $L+1$, discard expanded prefix.

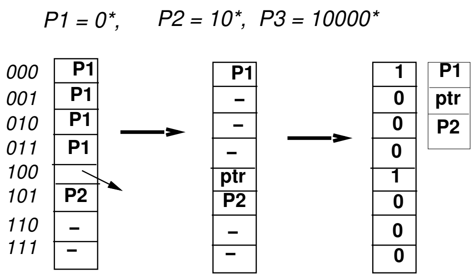


## Expanded Version of Trie




- Reduces search time from 7 to 3 memory accesses, but needs more memory. Can reduce memory by choosing optimal tries (Sigmetrics 98).

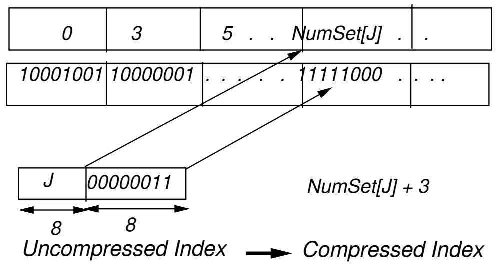
## Lulea Scheme: Node Compression of Multibit Tries




- Expansion can lead to a lot of filled entries in each multibit trie array.
- If we (effectively) do linear search for the earliest array entry with a valid prefix, we can use bit map compression.



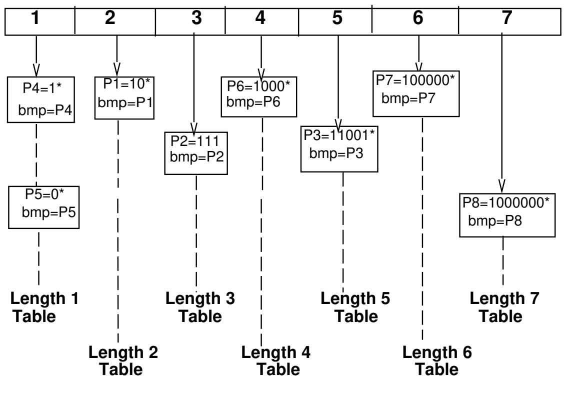
## Why Lulea Speed Loss is not too bad




- For large arrays of bits we store with each word $j$, $NumSet[j]$, the number of bits set in words $< J$. High order bits of index yield $j$ and counting bits in Word $j$ give compressed Index.
- Takes 3 memory accesses in software and 2 in hardware.



## Binary Search of Prefix Lengths and Markers






## Comparing Lookup Schemes

| | Average (24 bit prefix (nsec) | Worst case (nsec) | Memory required for Mae East database (KBytes) |
|---|---|---|---|
| Patricia trie | 1500 | 2500 | 3262 |
| Binary search on prefixes | 1200 | 1500 | 1280 |
| Binary search on hash tables | 250 | 650 | 1600 |
| Lulea scheme | 349 | 409 | 160 |
| Multibit trie | 180 | 236 | 640 |

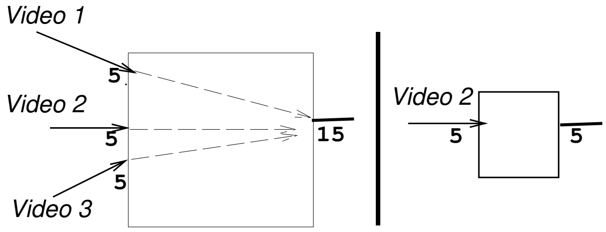
Table 1:

- 300 Mhz Pentium, Mae East Database with around 40,000 prefixes,
- Multibit trie is the only scheme with fast insertion and deletion times (worst case 2.5 msec). Without that, needs only 450 kbytes memory is possible.

## 2.2 Fair Queuing



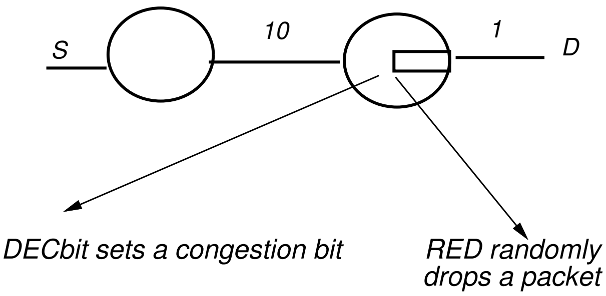
## Performance Metrics




- **Work**: Worst-case work to enqueue or dequeue a packet.
- **Throughput Fairness:** During *any* period of time when two flows $F$ and $S$ are both active, the ratio of bits sent for $F$ and $G$ should roughly be the ratio of their assigned rates.
- **Delay Bounds:** If flow $F$ is assigned rate $r$, then the delay encountered by flow $F$ should be at least as good as if were running on a dedicated router that processed bits at rate $r$.



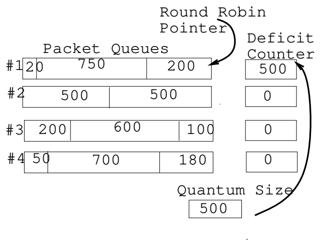
## Today's Router Queues: FIFO or RED

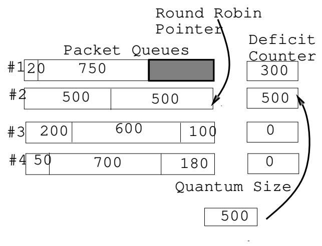



- Today's IP has no room for a congestion bit, so they randomly drop packets with small probability when queue size passes a threshold (RED).
- Proposal on table for a ECN bit for IPv6.



## Throughput Fairness: Multiple Queues and DRR



## Delay Bounds: Basic Virtual Clock

- Suppose a flow $F$'s rate is $r$. What is the delay of a packet $p$ arriving at a server that always serves at rate $r$?
- If $p$ arrives before the previous packet from flow $F$ ($prev$) is served, then $p$ has to wait for $prev$ to be served; otherwise $p$ gets served right away.
- Thus in ideal system, packet $p$ will be served by: Maximum (Arrival Time($p$), Service Time ($prev$)) + Length($p$)/ $r$.
- To guarantee service by ideal deadline use earliest deadline scheduling (Leland,Liu 74). Called Virtual Clock scheme. Requires sorting at high speeds.



## Differentiated Services

- Gives up quantatitive QoS for simplicity and immediate deployment. Two bits stolen from header: one for premium and one for assured service.
- Premium service packets get a separate queue at routers and priority service. Traffic must be shaped and within contract and will be charged more.
- Assured service is a "better best effort". Shares same queue as best effort but has preferential treatment in RED algorithm.
- Each company network or ISP has a bandwidth broker that parcels out bandwidth. Bandwidth brokers may communicate dynamically using RSVP or statically using bilateral agreements.



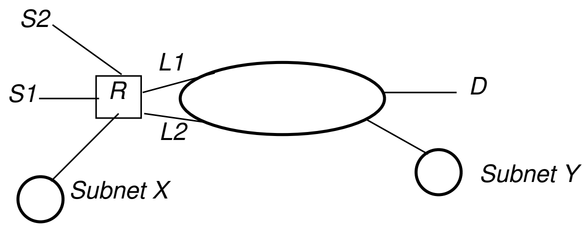
## Research Questions

- Need to have a scheme implementable by routers at Gigabit speeds. *Chao proposes hardware priority queues; Bennett-Zhang describe a simple scheme if packet sizes are constant; Suri-Varghese-Chandranmenon describe an approximate solution that takes* $O(\log\log(n))$
- Need to reduce state in backbone routers that can potentially have hundreds of thousands of concurrent flows. *see SIGCOMM 98 paper on core-stateless fair queuing by Stoica-Shenker-Zhang*



## 2.4 Level 4 Routing

## Use of filters




**Database at Router R**

| To | From | TrafficType | Forwarding Directive |
|---|---|---|---|
| D | S1 | Video | Forward via L1 |
| * | S2 | * | Drop all traffic |
| Y | X | * | Reserve 50 Mbps |

- Unifies three trends: firewall processing, traffic reservation filters, and service differentiation.



## Key Message Fields

| Destination (32) | Source (32) | Protocol (8) | Dest Port (16) | Source Port (16) |
|---|---|---|---|---|

- Filters consist of prefixes or ranges on each of 5 fields.



## Sample Firewall Database

| *Destination* | *Source* | *Destination Port* | *Source Port* | *Protocol* | *comments* |
|---|---|---|---|---|---|
| M | * | 25 | * | * | allow inbound mail |
| M | * | 53 | * | UDP | allow DNS access |
| M | S | 53 | * | * | DNS access to S |
| M | * | 23 | * | * | incoming telnet |
| TI | TO | 123 | 123 | UDP | NTP time info |
| * | Net | * | * | * | outgoing packets |
| Net | * | * | * | TCP ack | return ACKs OK |
| * | * | * | * | * | block everything else |

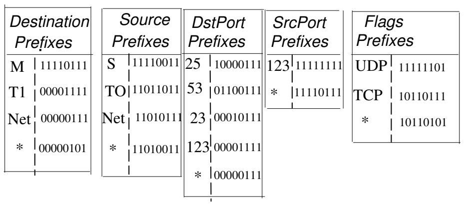
M **is the mail gateway/bastion host**
TI, TO **are the internal and external NTP sources**
S **is the external secondary DNS server**
Net **is the prefix of the corporate network**



## Lucent Bit Vector Scheme

- Divide and conquer. With each possible value $M$ of field $F_i$, we store the set of filters $S(M)$ that match $M$ in field $F_i$ as a bit map
- We once again (like cross-producting) do a lookup in each field $F_i$ to get matches $M_i$ and the sets $S(M_)$. We then take the intersection of all sets $S(M_i)$ and find the lowest cost element of intersection set.
- If filters are arranged in non-decreasing order of cost and all sets are bitmaps, this is indeed an $O(N)$ algorithm but needs only $N/W * k$ memory accesses where $k$ is the number of fields, $W$ is width of memory access (can be 1000 in hardware), and $N$ is number of filters.

## Lucent Scheme in Action




- Example: to $M$ from $S$ with UDP dest port 53, source port 1029. Matches filters 2, 3, and 8; earliest match is Filter 2.
- $M$ match gives us 11110111; $S$ gives 11110011; 53 gives 01100111, $*$ gives 11110111; $UDP$ gives 11111101. AND of 5 bitmaps is 01100001, index of first bit set is 2.



## More information on Packet Filtering

- **Grid-of-Tries and Cross-producting:** (Srinivasan, Varghese, Suri, SIGCOMM 98): Efficient 2D filtering scheme and a general scheme that requires a lot of memory

  **Tuple Space Search:** (Srinivasan, Suri, Varghese, SIGCOMM 99): General scheme using hashing
- **Recursive Flow Classification:** (Gupta, Mckeown, SIGCOMM 99): Improved version of crossproducting that reduces memory needs considerably.



## Conclusions

- The complexity of all systems research is *breadth*: learning to be a systems thinker, understanding system interactions, finessing hard problems.
- This tutorial has attempted:
  - to give you some background to understand Internet routing as a system (actually it interacts with TCP and DNS too!)
  - to give you some of the design philosophy behind IP
  - to stimulate you to work on some problems of interest to the Internet community in protocols and implementation.
- A chance to make an impact, make some money, even start your own company . . .