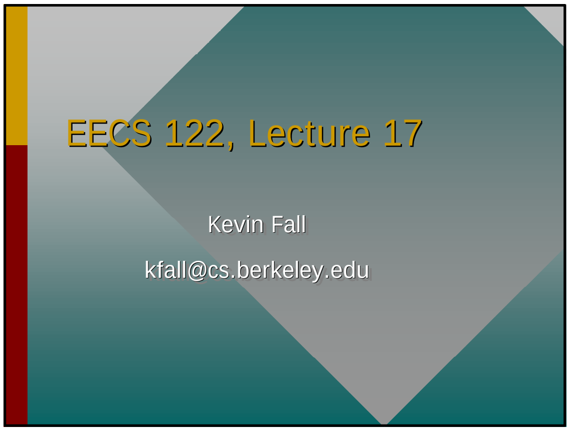
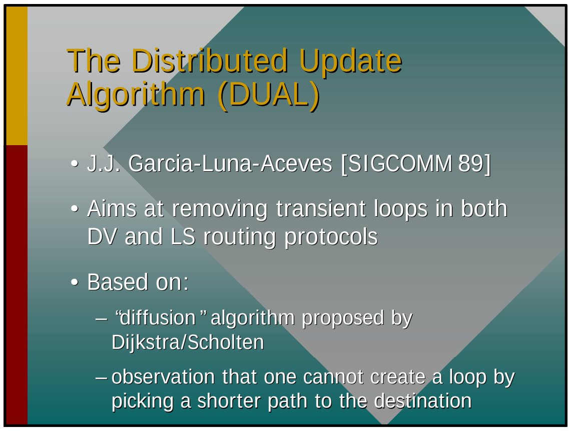


EECS 122, Lecture 17

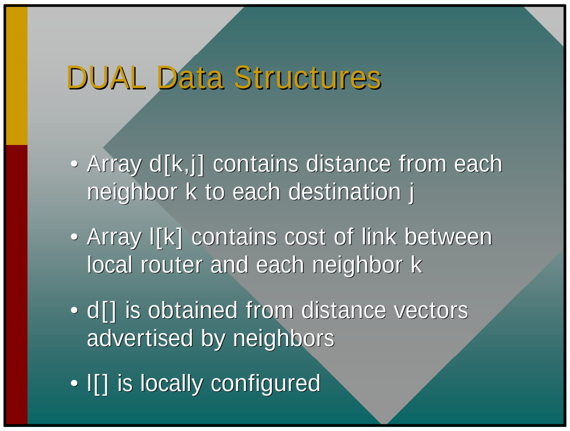
Kevin Fall

kfall@cs.berkeley.edu



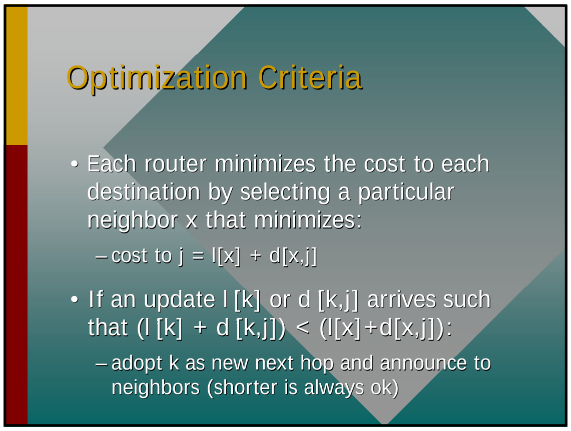
The Distributed Update Algorithm (DUAL)

- J.J. Garcia-Luna-Aceves [SIGCOMM 89]
- Aims at removing transient loops in both DV and LS routing protocols
- Based on:
  - “diffusion” algorithm proposed by Dijkstra/Scholten
  - observation that one cannot create a loop by picking a shorter path to the destination



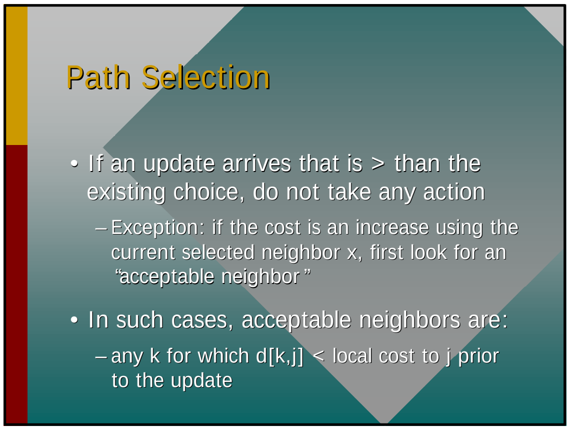
DUAL Data Structures

- Array d[k,j] contains distance from each neighbor k to each destination j
- Array l[k] contains cost of link between local router and each neighbor k
- d[] is obtained from distance vectors advertised by neighbors
- l[] is locally configured



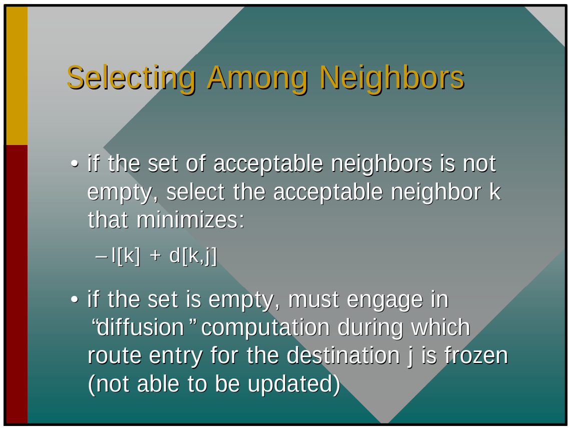
Optimization Criteria

- Each router minimizes the cost to each destination by selecting a particular neighbor x that minimizes:
  - cost to j = l[x] + d[x,j]
- If an update l [k] or d [k,j] arrives such that (l [k] + d [k,j]) < (l[x]+d[x,j]):
  - adopt k as new next hop and announce to neighbors (shorter is always ok)



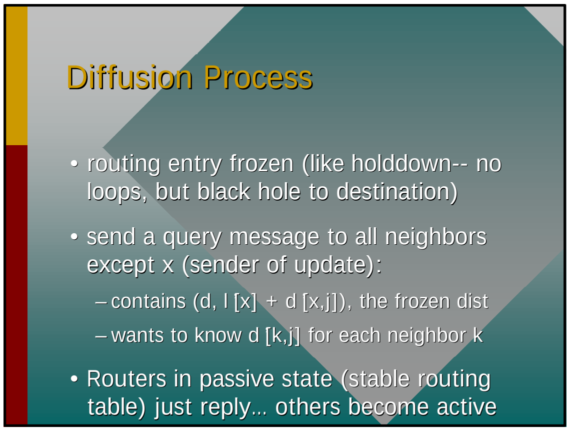
Path Selection

- If an update arrives that is > than the existing choice, do not take any action
  - Exception: if the cost is an increase using the current selected neighbor x, first look for an “acceptable neighbor”
- In such cases, acceptable neighbors are:
  - any k for which d[k,j] < local cost to j prior to the update



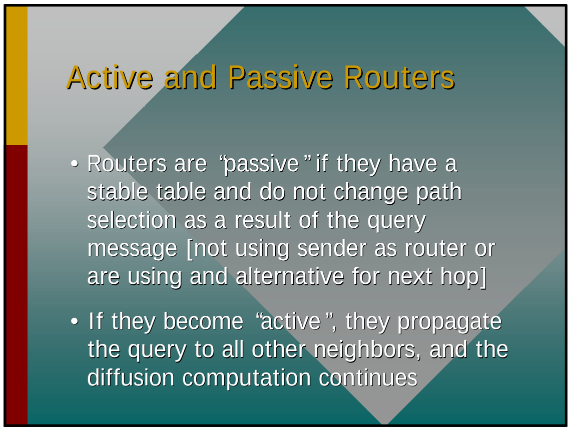
Selecting Among Neighbors

- if the set of acceptable neighbors is not empty, select the acceptable neighbor k that minimizes:
  - l[k] + d[k,j]
- if the set is empty, must engage in “diffusion” computation during which route entry for the destination j is frozen (not able to be updated)



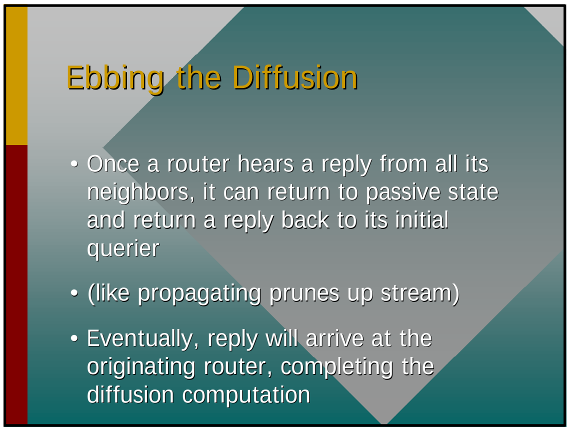
Diffusion Process

- routing entry frozen (like holddown-- no loops, but black hole to destination)
- send a query message to all neighbors except x (sender of update):
  - contains (d, l [x] + d [x,j]), the frozen dist
  - wants to know d [k,j] for each neighbor k
- Routers in passive state (stable routing table) just reply... others become active



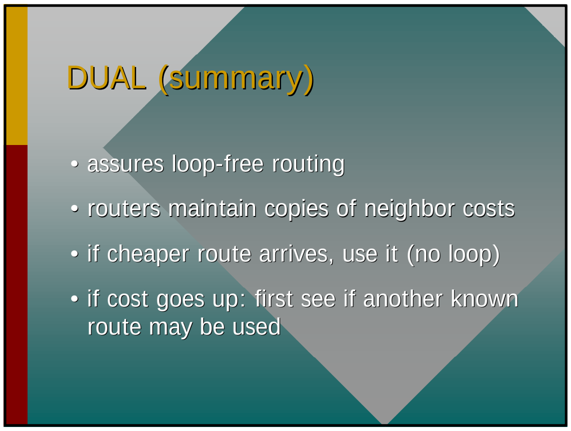
Active and Passive Routers

- Routers are “passive” if they have a stable table and do not change path selection as a result of the query message [not using sender as router or are using and alternative for next hop]
- If they become “active”, they propagate the query to all other neighbors, and the diffusion computation continues



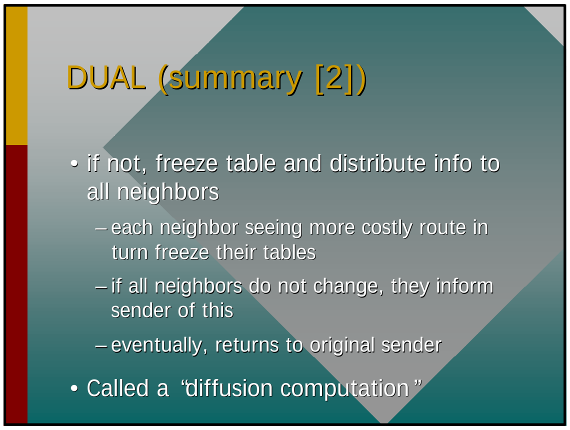
Ebbing the Diffusion

- Once a router hears a reply from all its neighbors, it can return to passive state and return a reply back to its initial querier
- (like propagating prunes up stream)
- Eventually, reply will arrive at the originating router, completing the diffusion computation



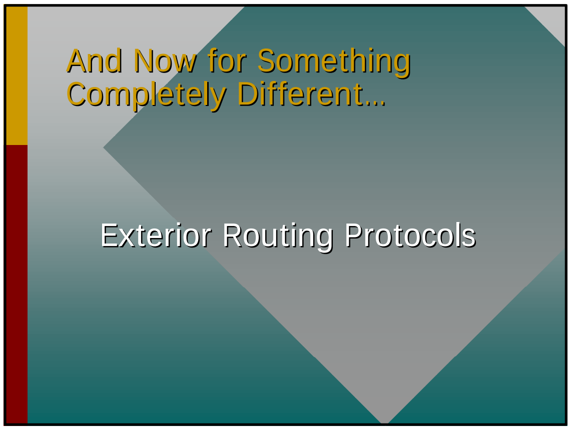
DUAL (summary)

- assures loop-free routing
- routers maintain copies of neighbor costs
- if cheaper route arrives, use it (no loop)
- if cost goes up: first see if another known route may be used



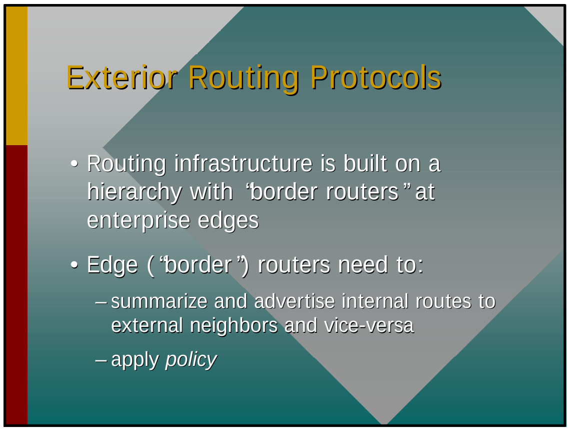
DUAL (summary [2])

- if not, freeze table and distribute info to all neighbors
  - each neighbor seeing more costly route in turn freeze their tables
  - if all neighbors do not change, they inform sender of this
  - eventually, returns to original sender
- Called a “diffusion computation”



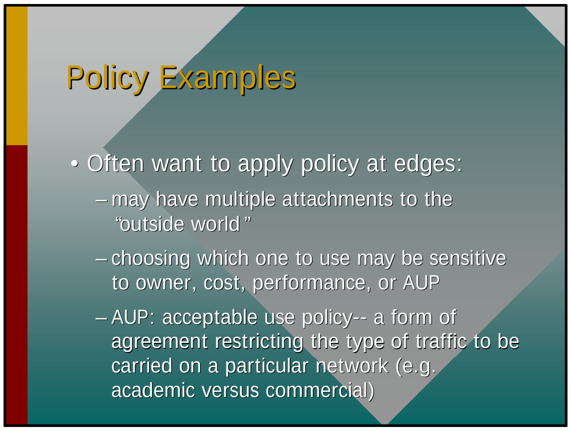
And Now for Something Completely Different...

Exterior Routing Protocols



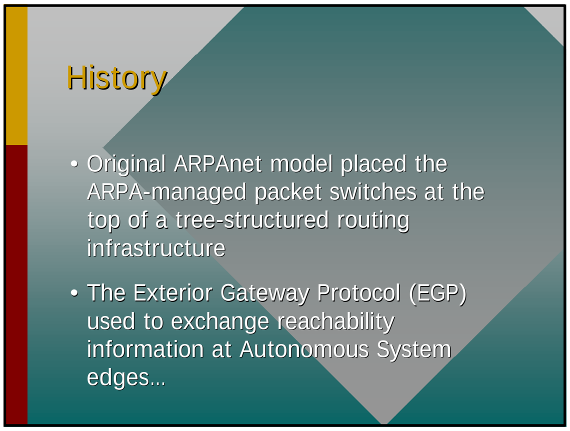
Exterior Routing Protocols

- Routing infrastructure is built on a hierarchy with "border routers" at enterprise edges
- Edge ("border") routers need to:
  - summarize and advertise internal routes to external neighbors and vice-versa
  - apply *policy*



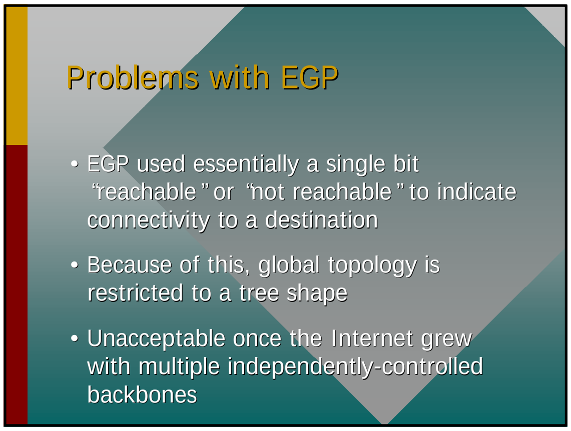
Policy Examples

- Often want to apply policy at edges:
  - may have multiple attachments to the "outside world"
  - choosing which one to use may be sensitive to owner, cost, performance, or AUP
  - AUP: acceptable use policy-- a form of agreement restricting the type of traffic to be carried on a particular network (e.g. academic versus commercial)



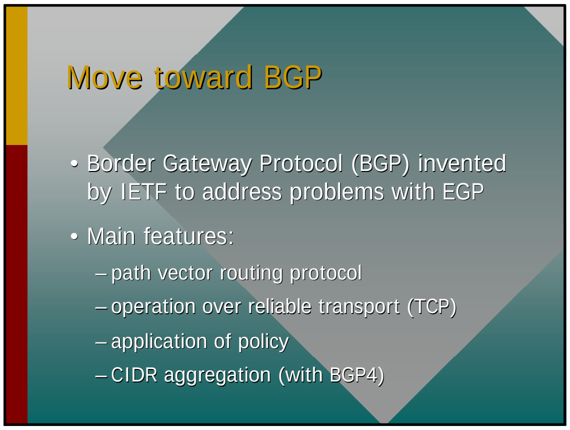
History

- Original ARPAnet model placed the ARPA-managed packet switches at the top of a tree-structured routing infrastructure
- The Exterior Gateway Protocol (EGP) used to exchange reachability information at Autonomous System edges…



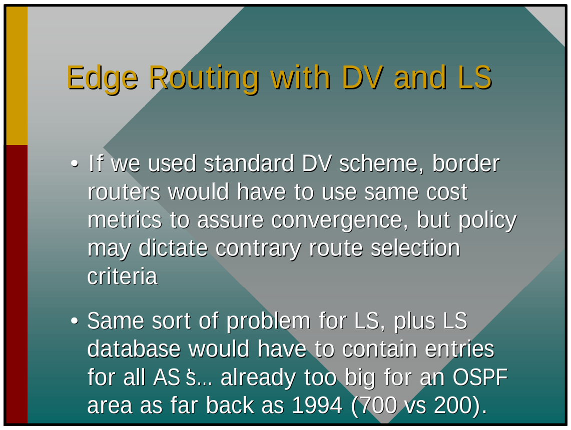
Problems with EGP

- EGP used essentially a single bit "reachable" or "not reachable" to indicate connectivity to a destination
- Because of this, global topology is restricted to a tree shape
- Unacceptable once the Internet grew with multiple independently-controlled backbones



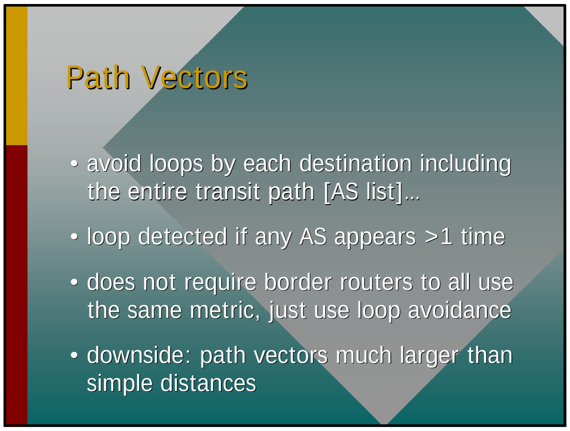
Move toward BGP

- Border Gateway Protocol (BGP) invented by IETF to address problems with EGP
- Main features:
  - path vector routing protocol
  - operation over reliable transport (TCP)
  - application of policy
  - CIDR aggregation (with BGP4)



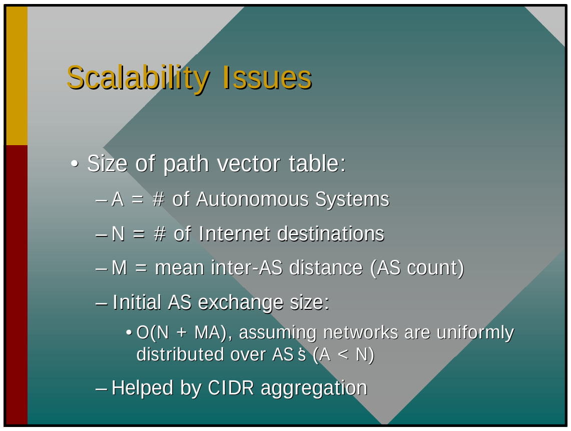
Edge Routing with DV and LS

- If we used standard DV scheme, border routers would have to use same cost metrics to assure convergence, but policy may dictate contrary route selection criteria
- Same sort of problem for LS, plus LS database would have to contain entries for all AS's… already too big for an OSPF area as far back as 1994 (700 vs 200).



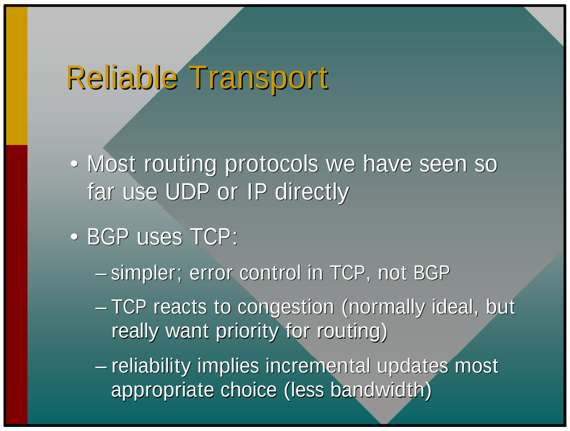
Path Vectors

- avoid loops by each destination including the entire transit path [AS list]…
- loop detected if any AS appears >1 time
- does not require border routers to all use the same metric, just use loop avoidance
- downside: path vectors much larger than simple distances



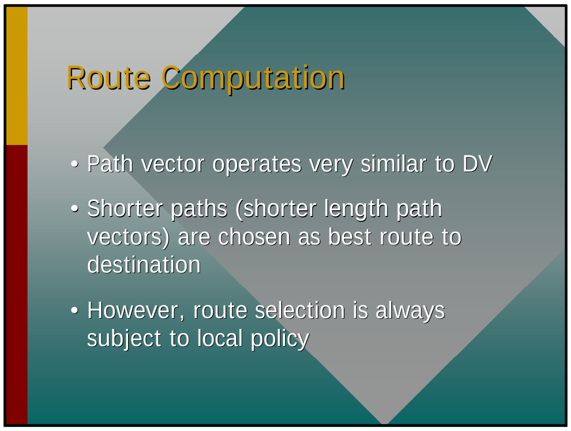
Scalability Issues

- Size of path vector table:
  - A = # of Autonomous Systems
  - N = # of Internet destinations
  - M = mean inter-AS distance (AS count)
  - Initial AS exchange size:
    - O(N + MA), assuming networks are uniformly distributed over AS’s (A < N)
  - Helped by CIDR aggregation



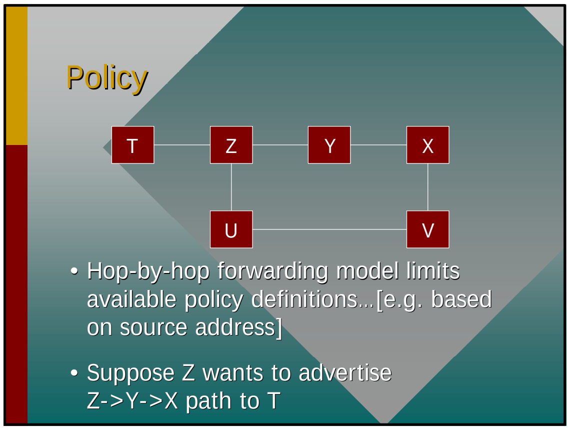
Reliable Transport

- Most routing protocols we have seen so far use UDP or IP directly
- BGP uses TCP:
  - simpler; error control in TCP, not BGP
  - TCP reacts to congestion (normally ideal, but really want priority for routing)
  - reliability implies incremental updates most appropriate choice (less bandwidth)



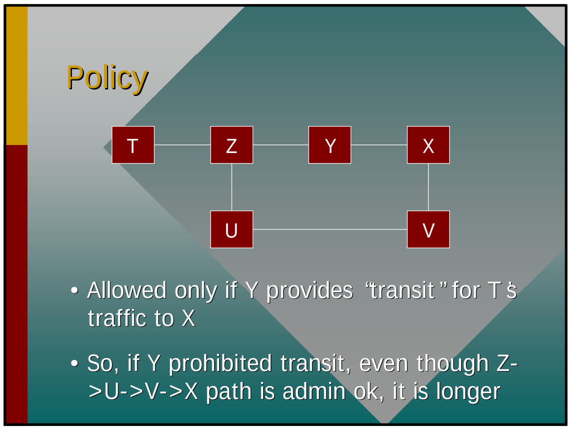
Route Computation

- Path vector operates very similar to DV
- Shorter paths (shorter length path vectors) are chosen as best route to destination
- However, route selection is always subject to local policy



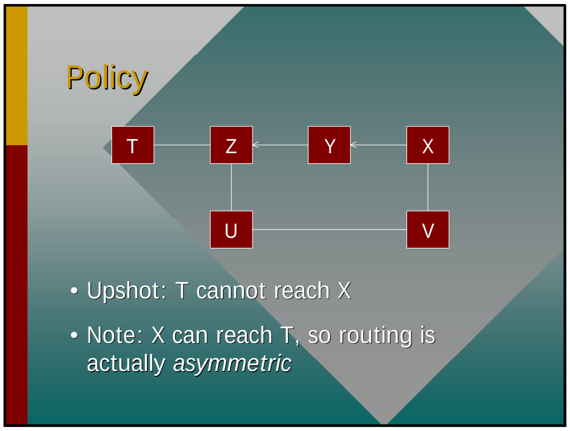
Policy

- Hop-by-hop forwarding model limits available policy definitions…[e.g. based on source address]
- Suppose Z wants to advertise Z->Y->X path to T



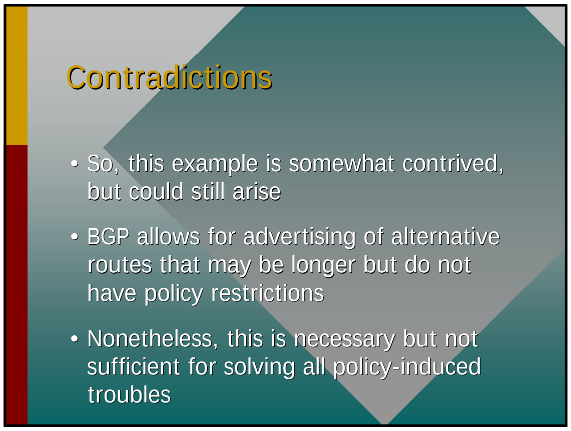
Policy

- Allowed only if Y provides “transit” for T’s traffic to X
- So, if Y prohibited transit, even though Z->U->V->X path is admin ok, it is longer



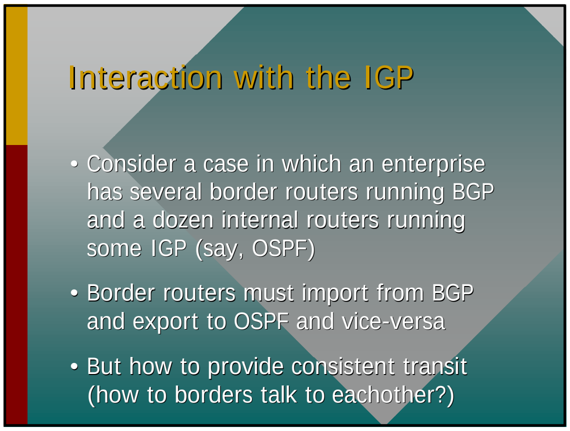
Policy

- Upshot: T cannot reach X
- Note: X can reach T, so routing is actually *asymmetric*



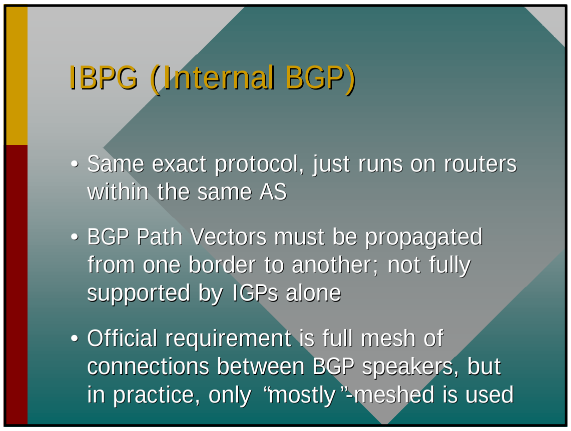
Contradictions

- So, this example is somewhat contrived, but could still arise
- BGP allows for advertising of alternative routes that may be longer but do not have policy restrictions
- Nonetheless, this is necessary but not sufficient for solving all policy-induced troubles



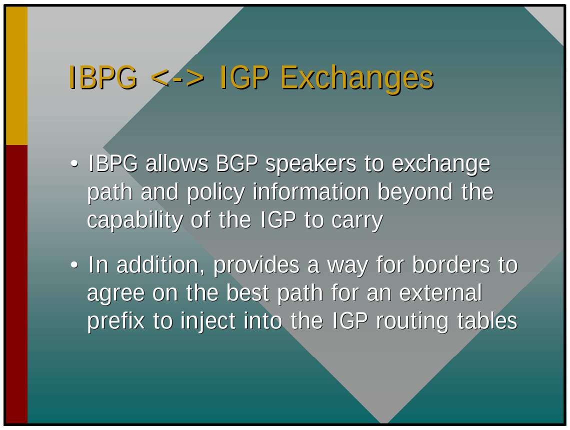
Interaction with the IGP

- Consider a case in which an enterprise has several border routers running BGP and a dozen internal routers running some IGP (say, OSPF)
- Border routers must import from BGP and export to OSPF and vice-versa
- But how to provide consistent transit (how to borders talk to eachother?)



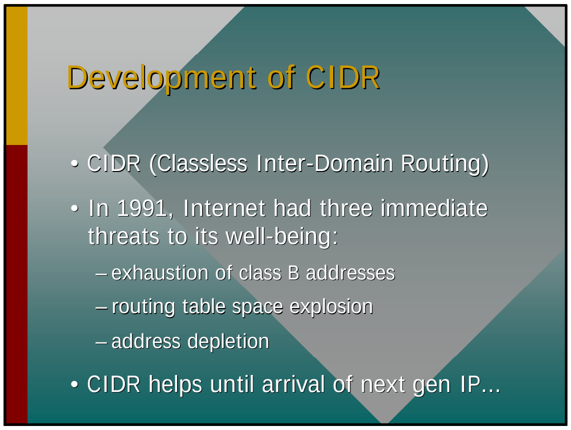
IBPG (Internal BGP)

- Same exact protocol, just runs on routers within the same AS
- BGP Path Vectors must be propagated from one border to another; not fully supported by IGPs alone
- Official requirement is full mesh of connections between BGP speakers, but in practice, only "mostly"-meshed is used



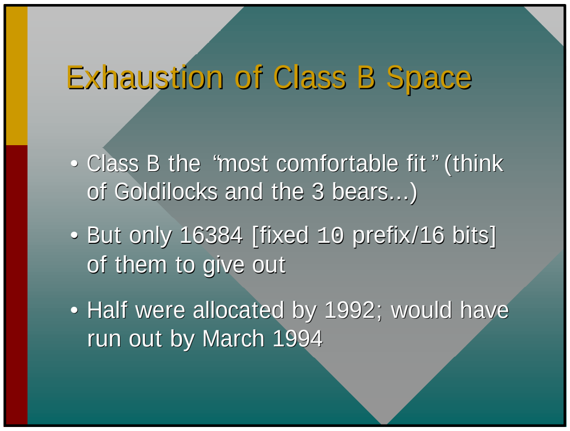
IBPG <-> IGP Exchanges

- IBPG allows BGP speakers to exchange path and policy information beyond the capability of the IGP to carry
- In addition, provides a way for borders to agree on the best path for an external prefix to inject into the IGP routing tables



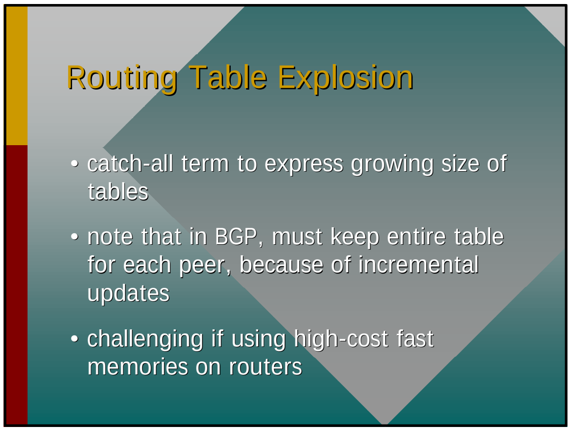
Development of CIDR

- CIDR (Classless Inter-Domain Routing)
- In 1991, Internet had three immediate threats to its well-being:
  - exhaustion of class B addresses
  - routing table space explosion
  - address depletion
- CIDR helps until arrival of next gen IP…



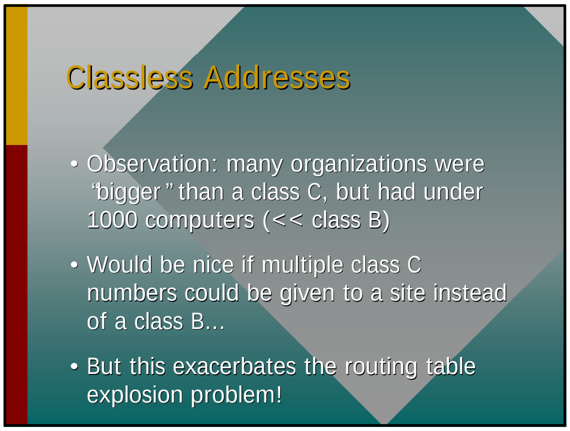
Exhaustion of Class B Space

- Class B the "most comfortable fit" (think of Goldilocks and the 3 bears…)
- But only 16384 [fixed 10 prefix/16 bits] of them to give out
- Half were allocated by 1992; would have run out by March 1994



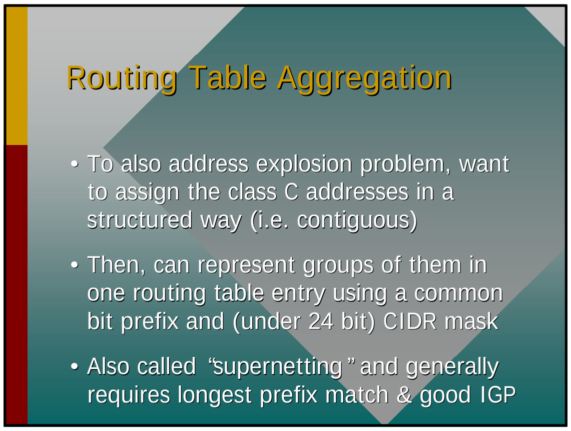
Routing Table Explosion

- catch-all term to express growing size of tables
- note that in BGP, must keep entire table for each peer, because of incremental updates
- challenging if using high-cost fast memories on routers



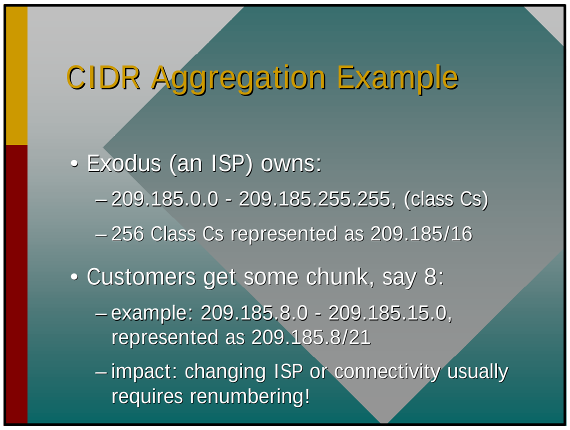
Classless Addresses

- Observation: many organizations were "bigger" than a class C, but had under 1000 computers (<< class B)
- Would be nice if multiple class C numbers could be given to a site instead of a class B…
- But this exacerbates the routing table explosion problem!



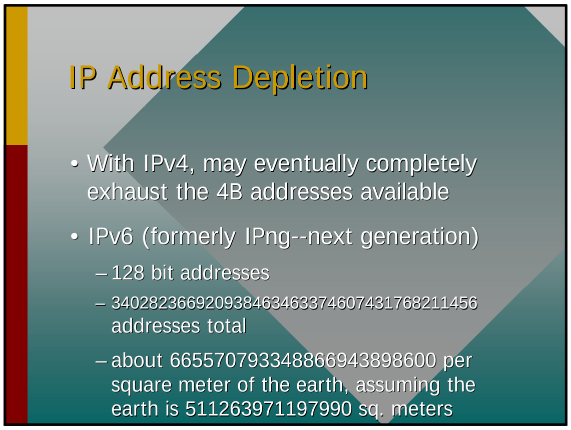
Routing Table Aggregation

- To also address explosion problem, want to assign the class C addresses in a structured way (i.e. contiguous)
- Then, can represent groups of them in one routing table entry using a common bit prefix and (under 24 bit) CIDR mask
- Also called "supernetting" and generally requires longest prefix match & good IGP



CIDR Aggregation Example

- Exodus (an ISP) owns:
  - 209.185.0.0 - 209.185.255.255, (class Cs)
  - 256 Class Cs represented as 209.185/16
- Customers get some chunk, say 8:
  - example: 209.185.8.0 - 209.185.15.0, represented as 209.185.8/21
  - impact: changing ISP or connectivity usually requires renumbering!



IP Address Depletion

- With IPv4, may eventually completely exhaust the 4B addresses available
- IPv6 (formerly IPng--next generation)
  - 128 bit addresses
  - 340282366920938463463374607431768211456 addresses total
  - about 665570793348866943898600 per square meter of the earth, assuming the earth is 511263971197990 sq. meters