Addressing & Routing on the Internet

Avi Freedman
Ravi Sundaram

Akamai

Outline

- Origins of the Internet
- Protocols and packets
- Addressing – IPv4 vs IPv6
- Routing - overview
- BGP - model
- BGP – convergence and hardness

Introduction

- The Internet is a NETWORK of networks – logically and physically
- Millions of computers capable of communicating with each other in real time
- Packet-based, store and forward
- Addressing – way of identifying computers
- Routing – getting packets from source to destination

Origins

- Academic experiment in 1960s, funded by ARPA – Advanced Research Projects Agency, now called DARPA
- December 1969 – first 4 node network went live using 56kbps links
- 1978 – IP emerges
- 1982 – TCP emerges, ARPANET split into MILNET and Internet
- 1983 – Internet composed of 200 computers

Origins

- 1984 – newsgroups emerge
- 1986 – DNS emerges, motivated by email, replaces host table
- 1988 – worm emerges, CERT formed
- 1989 – 100,000 computers on Internet, TCP retooled to prevent congestion collapse
- 1990 – commercial traffic still banned on Internet's backbone – NSFNET
- 1991 – commercial ban lifted, www emerges

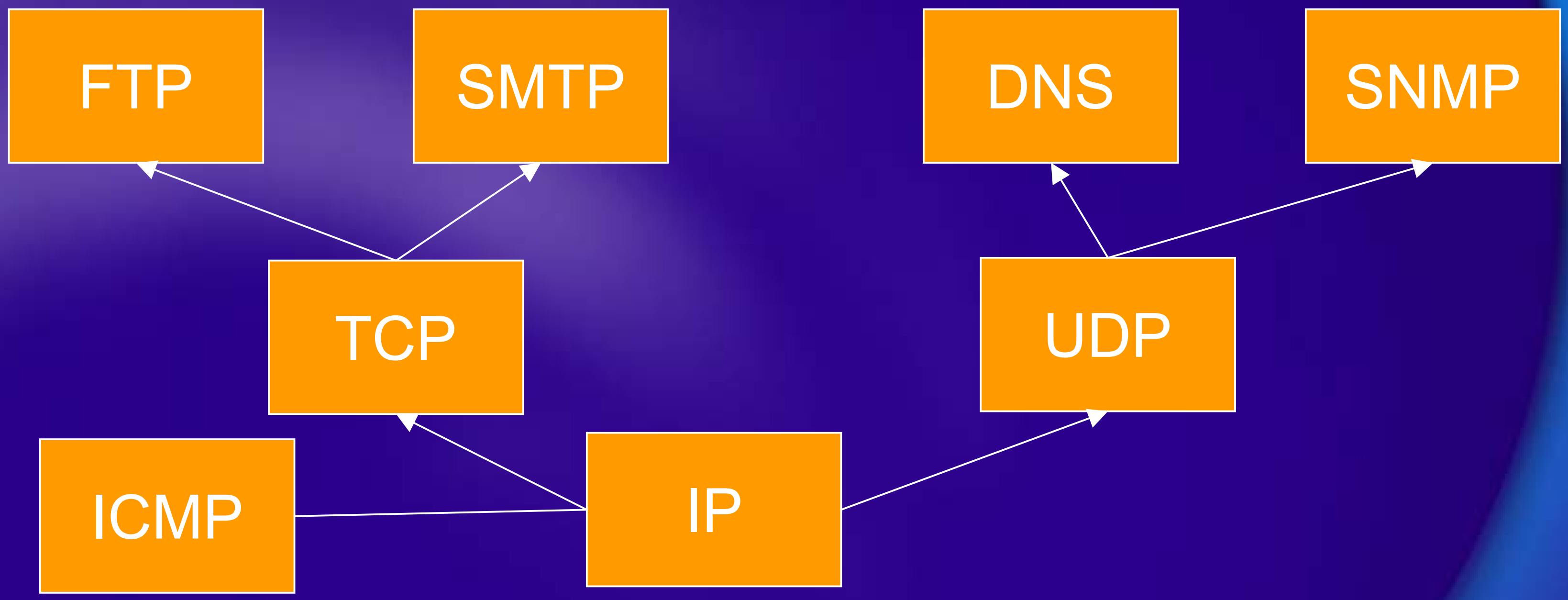
Origins

- May 1993 – last NSFNET solicitation for private NAPs
- 1995 – NSFNET replaced by vBNS – high performance backbone service linking certain universities and research centers at 155Mbps and higher, contract given to MCI (superceded by Abilene 10Gbps?)
- 2002 – 350 million hosts

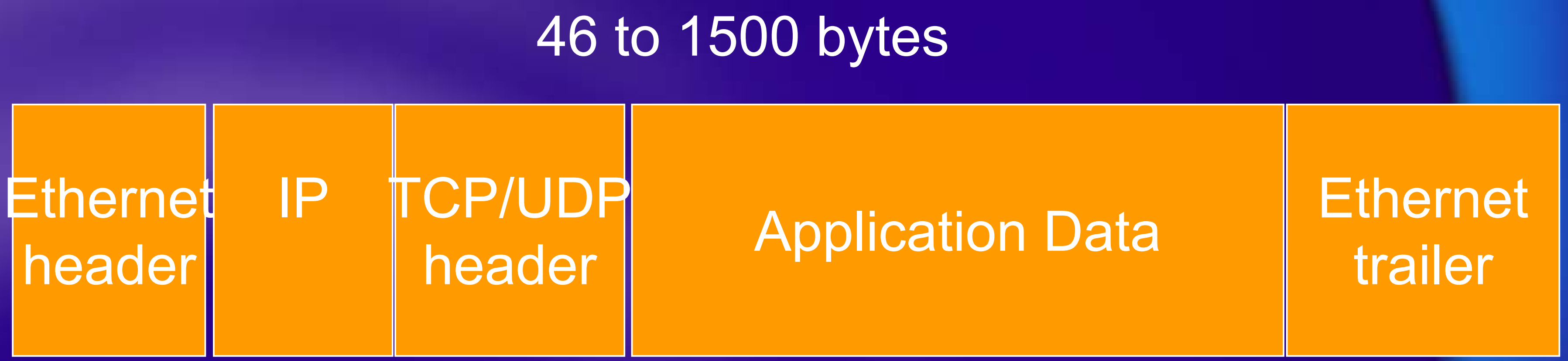
Comments

- Unprecedented growth
- Decentralized control – challenges and opportunities
- Performance
- Reliability
- Accounting
- Security
- Directory
- End-to-end arguments in system design. ACM Trans on Comp systems, Nov 84, 277-288.

Protocols



Packets



46 to 1500 bytes
Ethernet header
IP
TCP/UDP header
Application Data
Ethernet trailer

Addressing

- 32 bit addresses – a.b.c.d
- 4 billion potential addresses
- About 250 million hosts
- IPv4 based on RFC791 in 1981

Addressing

- Classful in early days:
  - Class A – first 8 bits fixed
  - Class B – first 16 bits fixed
  - Class C – first 24 bits fixed
- CIDR – Classless Interdomain Routing
  - a.b.c.d/m – first m bits fixed
  - e.g. 0.0.0.0/29 = 0.0.0.0 to 0.0.0.7
- Most specific match routing rule

Addressing

- Issues with IPv4
    - Address space depletion
    - Control by central registry
    - No network/routing consideration
    - No security consideration
    - No QoS consideration
- Summarized as scalability, security and QoS

Addressing

- IPv6 or IPng
  - 128 bits
  - hierarchical (network-based)
  - secure (uses IPSec)
  - QoS (bits allocated for labeling flows)

Akamai

Addressing

- Will migration happen 4 to 6

  Scalability – CIDR/NAT (not before 2010)

  Secure – IPSec & application level

  QoS – application level

Routing

- Internet – collection of Autonomous Systems
- Autonomous System – set of routers sharing same routing policies, routers in an AS are analogous to post offices in a country
- Routing protocol – collection of rules for forwarding packets

Akamai

Routing

- Distance(path)-vector protocols

  routing updates include vector of distances(paths)

  each node has a (policy-based)shortest

  path tree

  examples RIP, BGP4

Routing

- Link-state protocols

  routing updates include state of links and others' updates

  each node has the entire graph

  examples OSPF

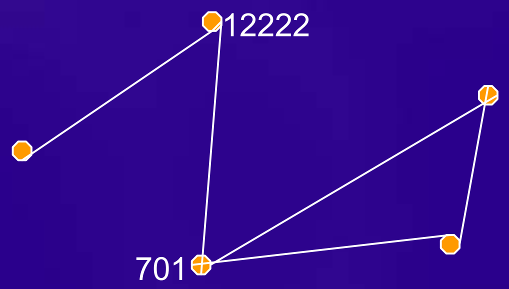
Traceroute

```
[koods@koods-desktop ~]$ traceroute www.berkeley.edu
traceroute to arachne.berkeley.edu (169.229.131.109), 30 hops max, 40 byte packets
1  172.24.80.1 (172.24.80.1)  0.401 ms  0.308 ms  0.291 ms
2  corp2-primary.kendall.akamai.com (172.24.8.2)  0.411 ms  0.334 ms  0.331 ms
3  akafire.kendall.akamai.com (172.24.44.4)  0.280 ms  0.208 ms  0.368 ms
4  65.202.32.3 (65.202.32.3)  0.608 ms  1.651 ms  0.923 ms
5  65.202.33.246 (65.202.33.246)  0.754 ms  0.664 ms  0.832 ms
6  serial4-0-2.hsipaccess1.Boston1.Level3.net (166.90.184.53)  0.912 ms  0.888 ms  0.881 ms
7  unknown.Level3.net (64.159.3.141)  1.349 ms  1.696 ms  2.018 ms
8  so-2-0-0.mp2.SanJose1.Level3.net (64.159.0.218)  85.658 ms  85.287 ms  84.278 m
 9  gige9-1.hsipaccess1.SanJose1.Level3.net (64.159.2.103)  84.682 ms  84.666 ms  84.404 m
10  unknown.Level3.net (209.247.159.110)  80.145 ms  80.630 ms  80.860 m
11  ucb-gw--qsv-juniper.calren2.net (128.32.0.69)  83.634 ms  84.703 ms  110.922 m
12  vlan196.inr-201-eva.Berkeley.EDU (128.32.0.74)  83.906 ms  87.205 ms  85.161 m
13  vlan209.inr-203-eva.Berkeley.EDU (128.32.255.2)  138.753 ms  141.608 ms  142.004 m
14  arachne.Berkeley.EDU (169.229.131.109)  140.416 ms  128.705 ms  143.716 ms
```

Akamai

BGP - model

- Modeled as collection of Autonomous Systems with Peering Relationships between one another.
- Can be thought of as a graph G=(V,E) with Autonomous Systems represented by vertices v in V, and Peering Relationships by edges e in E.



BGP – Border Gateway Protocol

- Path-vector protocol – each vertex maintains a shortest-path tree rooted at itself
- “shortest” – combo of policy and distance based metrics
- Each Autonomous System selects its routes based on its own policy and the best routes of its neighbors.

Akamai

BGP – idealized model

- The Internet is modeled as an undirected graph G=(V,E), whereV corresponds to the Autonomous Systems and E corresponds to the peering relationships.
- Each vertex learns a set of route announcements from its neighbors.
- A route announcement is a record with the following attributes:

  nlri: network layer reachability info, e.g. 1.2.3.4

  as_path: ordered list of vertices starting with next hop, e.g. 701 12222

  loc_pref: local preference with dlp used to denote default value

Akamai

BGP – idealized model

- Each vertex selects the best route to a given destination. If it has many routes r_1, r_2 … r_k with the same destination, i.e. r_i.nlri = r_j.nlri, then it selects first based on highest local_pref then on shortest as_path, with ties being broken arbitrarily.
- Route transformations:
  - Local_prefs are not communicated
  - No loops: v never accepts routes r where v ε r.as_path
  - The set of routes selected at v is passed onto v's neighbors with v prepended to the as_path
  - Import and export policies

BGP – idealized model

- Import and Export Policies

- If all import and export rules are “true => allow” then BGP reduces to a pure distance vector protocol

BGP – idealized model

- Dynamic behavior.

  Informally a BGP system S = <G, Policy(G), S0>, comprising an AS graph G= (V,E), containing import and export policies for every v_j in V and initial state S0 = (c0_1,c0_2,…c)_n) where

  c0_j is the destination originated by v_j

- If v_j is activated then it gets route announcements from its immediate neighbors and selects its best routes.

BGP – question of convergence

- State graph.
  - Directed graph of all states with S_j => S_k if there exists a v whose activation causes the change
  - A state S is said to be final if S => S on activation of any v.
  - A BGP system is said to be solvable if it has a final state
  - A BGP system is said to be convergent if ends up in a final state independent of the activation sequence

BGP – question of convergence

- Can locally well configured policies give rise to global routing anomalies?
- Can the protocol diverge, i.e. cause a collection of Autonomous Systems toexchange messages forever without converging?

BGP – question of convergence

- Does BGP diverge in practice? There are horror stories of networks accidentally setting themselves up as sinks for all the traffic but to date no evidence of large sclae flaps.
- But there are frequent and numerous occurrences of delayed convergence, as high as 50 minutes. In "Delayed Internet Routing Convergence" C. Labovitz, A. Ahuja, A. Bose & F. Jahanian, Proceedings of Sigcomm 2000, pp 175-18, they conduct experiments where they withdraw a route and replace it with another and see how long before it washes through the Internet as observed from a number of vantage points.

BGP – question of convergence

- In addition to various vendor specific anomalies, the main reason for long convergence is that path vector protocols consider multiple paths of a given length as opposed to distance vector protocols that consider only one path of a given length. In Labovitz et al they construct an example where every loop free path in the complete mesh is considered – given that there are an exponential number of such paths it is not surprising that convergence is delayed.

BGP – question of convergence

- The following example is from:

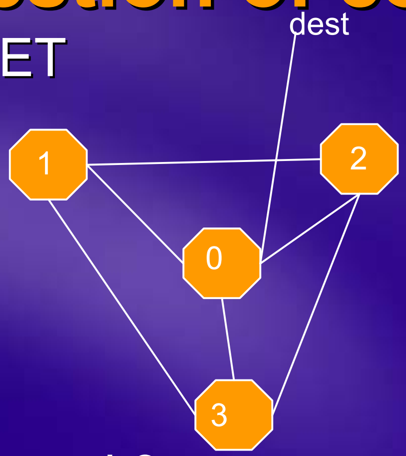
Persistent route oscillations

K. Varadhan, R. Govindan & D. Estrin

ISI TR 96-631

BGP – question of convergence

BAD GADGET



All rules are mod 3

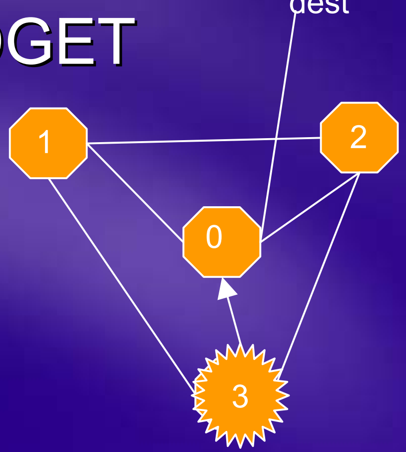
Export Rules: nlri=dest => allow

Import Rules: if i+1 => i then nlri=dest & as_path=[I+1,0] => loc_pref = dlp +1; nlri=d => loc_pref=dlp

if i-1 => I then nlri=dest => allow

BGP – question of convergence

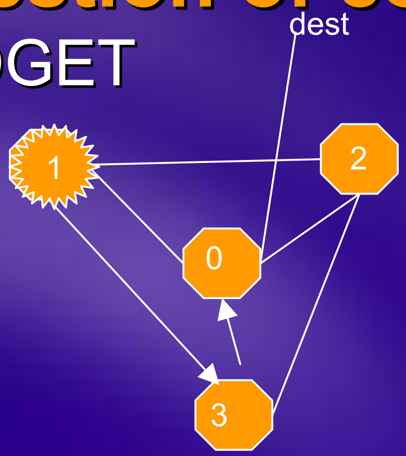
BAD GADGET



Does BAD GADGET have a solution?

BGP – question of convergence

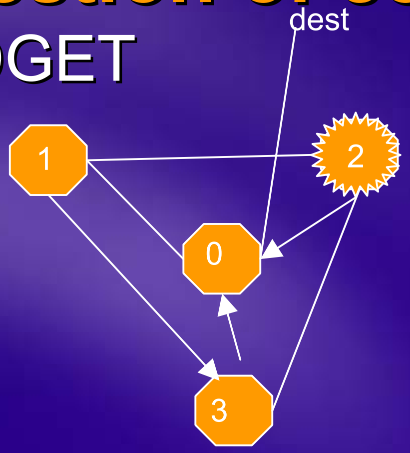
BAD GADGET



Does BAD GADGET have a solution?

BGP – question of convergence

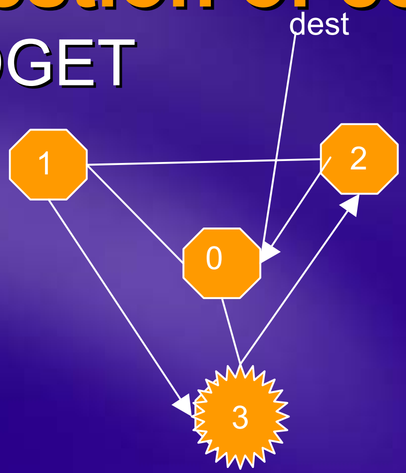
BAD GADGET



Does BAD GADGET have a solution?

BGP – question of convergence

BAD GADGET



Does BAD GADGET have a solution?

BGP – question of convergence

- Does BAD GADGET have a solution?
  - For BAD GADET to have a solution it must have a final state.
  - It is easy to see for single destination systems that in a final state the graph induced by the as_path at every vertex to a destination is a tree rooted at the destination, and that this final state is reachable by activating all the nodes of the tree in breadth-first order.
  - BAD GADGET does not have a final state and this can be checked by looking at all the (6) trees rooted at 0 and verifying that none of them work.

BGP – question of convergence

- The following results are from:

An Analysis of BGP Convergence Properties

T. Griffin & G. Wilfong

Proceedings of Sigcomm 99, pp 277-288

BGP – another problem

- REACHABILITY: Given a system S, vertices v and w and destination d originated by w does there exist a final state in which d is reachable from v?

- REACHABILITY is in NP

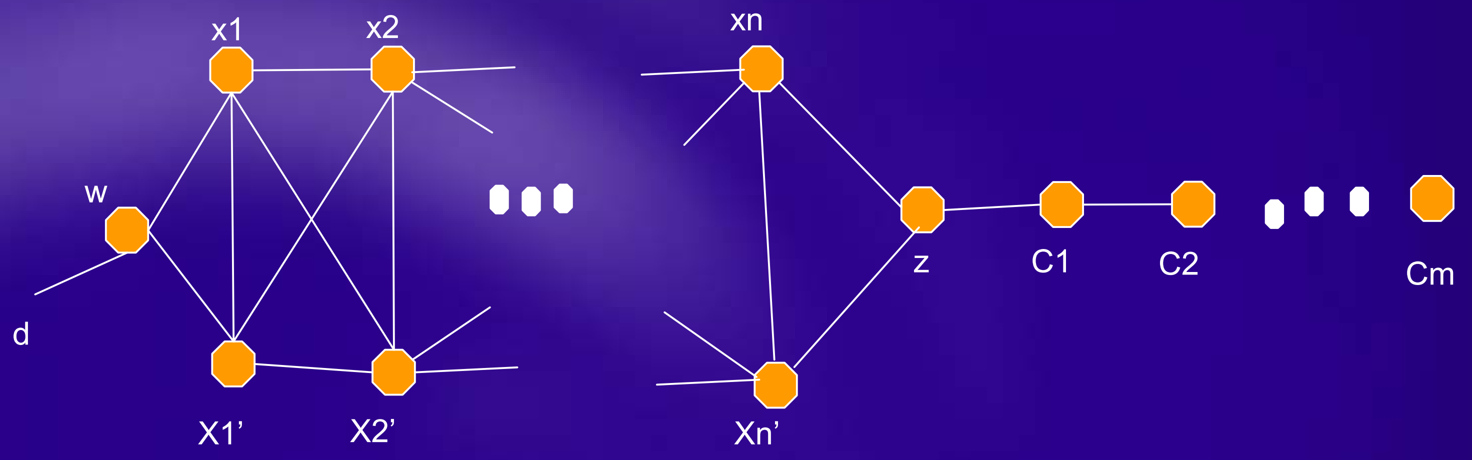
  Pf: Guess a final state and check reachability (and finality).

- To show REACHABILITY is NP-hard we demonstrate a reduction from 3-SAT.

Akamai

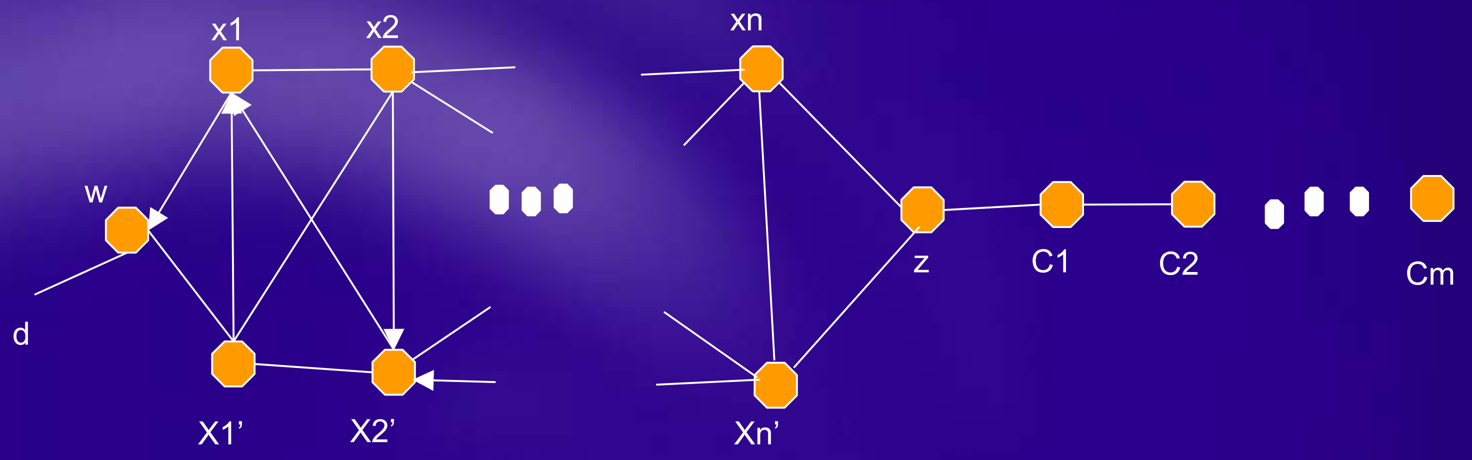
REACHABILITY is NP-hard

3-SAT example: (x1 V x2’ V x3) & (x1’ V x2’ V x3’) …



REACHABILITY is NP-hard

X1=true; x2=false; x3=false…



REACHABILITY is NP-hard

- Export policies: true => allow.
- Import policies: enforce that only one of xj or xj' is in the as_path of a route to d and oncethe route is chosen then a lock-in is forced. Example xj → xj': nlri=d => loc_pref = dlp + 1;

  xj-1 → xj : nlri=d & xj-1' not in as_path => loc_pref = dlp;

  For clause Cj = xk V xl V xm: xk in as_path or xl in as_path or xm in as_path => loc_pref = dlp.

REACHABILITY is NP-hard

- Satisfiable => REACHABLE

  Pf: activate along the literals that are set to true.

- REACHABLE => satisfiable

  Pf: Follows trivially from the way the policies work to ensure a unique path.

Other Problems and Implications

- ASYMMETRY
- SOLVABILITY
- ROBUSTNESS

- RADB and centralized vetting

Research

Consider a path vector protocol such as BGP – at each step a node gets information from its neighbors and uses its (local) policy to update its table of routes. A topology and collection of policies is satisfiable if there exists a state where updates do no changes. A system is said to converge if it reaches such a state.

The problem is to try and characterize the behavior of these systems – when do they diverge, can they converge to more than one satisfiable state.

Reference: www.acm.org/pubs/citations/proceedings/comm/316188/p277-griffin/

Questions?