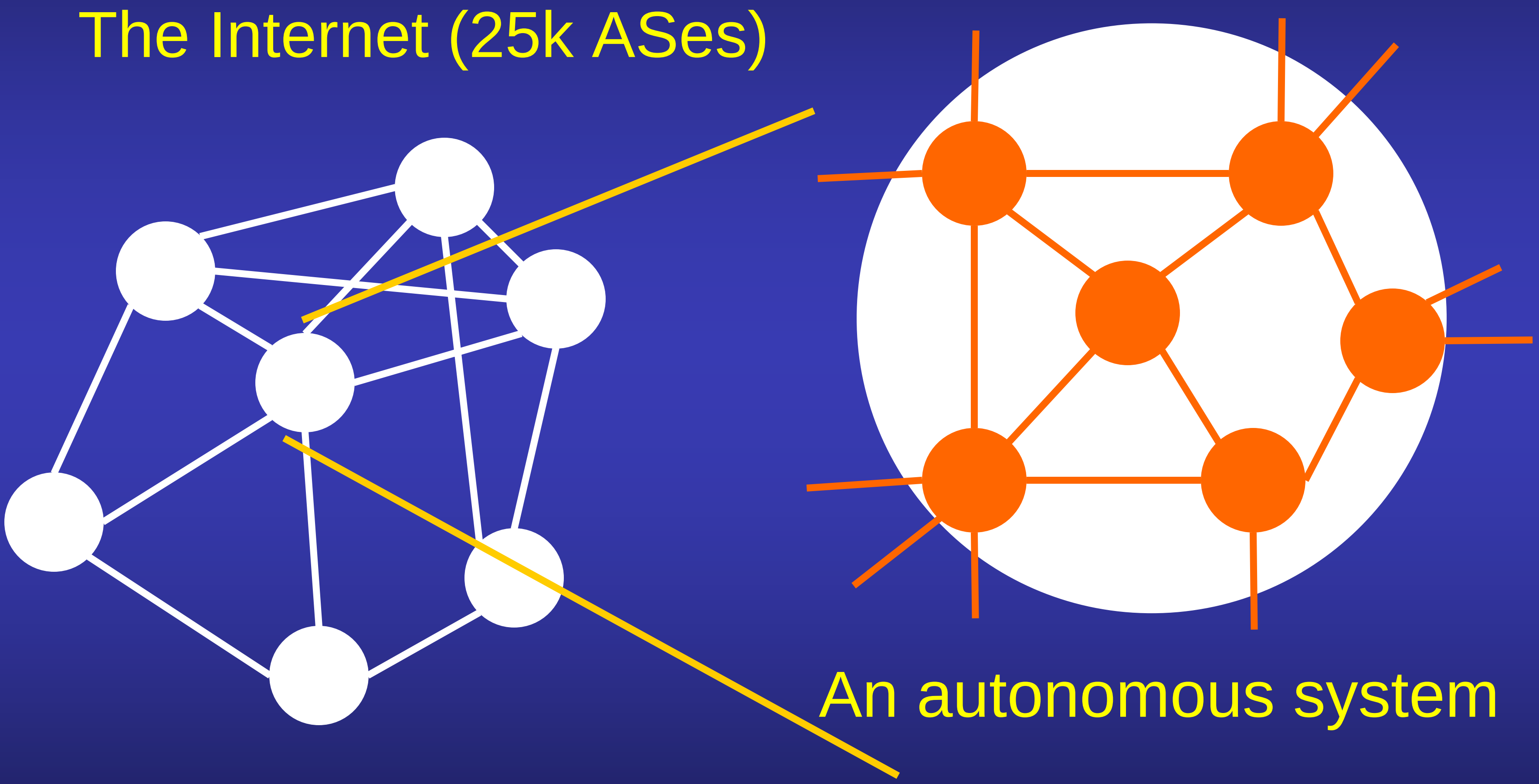
Making an AS Route Like a Single Node

Rui Zhang-Shen
Princeton University
rz@cs.princeton.edu

joint work with Jennifer Rexford and Yi Wang

What Is an AS?



The Internet (25k ASes)

An autonomous system

Example Policy Objectives of an AS

Goals

- Prefer profitable routes
- Avoid known bad routes
- Conserve bandwidth
- Low protocol overhead
- Simple management
- …

Obligations

- Export full customer routes to peers
- Respect MED
- Respect communities
- Export consistent routes
- …

Together they form the *Routing Policy*

Correctness of Policy Realization

- ❑Configuration within a single AS to realize its routing policy
- ❑Today's practice
  - ▪ Define an AS-level policy
  - ▪ Configure all routers with the policy
    - Assume configuration files generated correctly
    - Assume no human errors
  - ▪ Some policy objectives are violated
    - Due to peculiarities of the protocol
    - Hard to detect, diagnose, or fix
- ❑How to correctly realize an AS's policy?

Outline

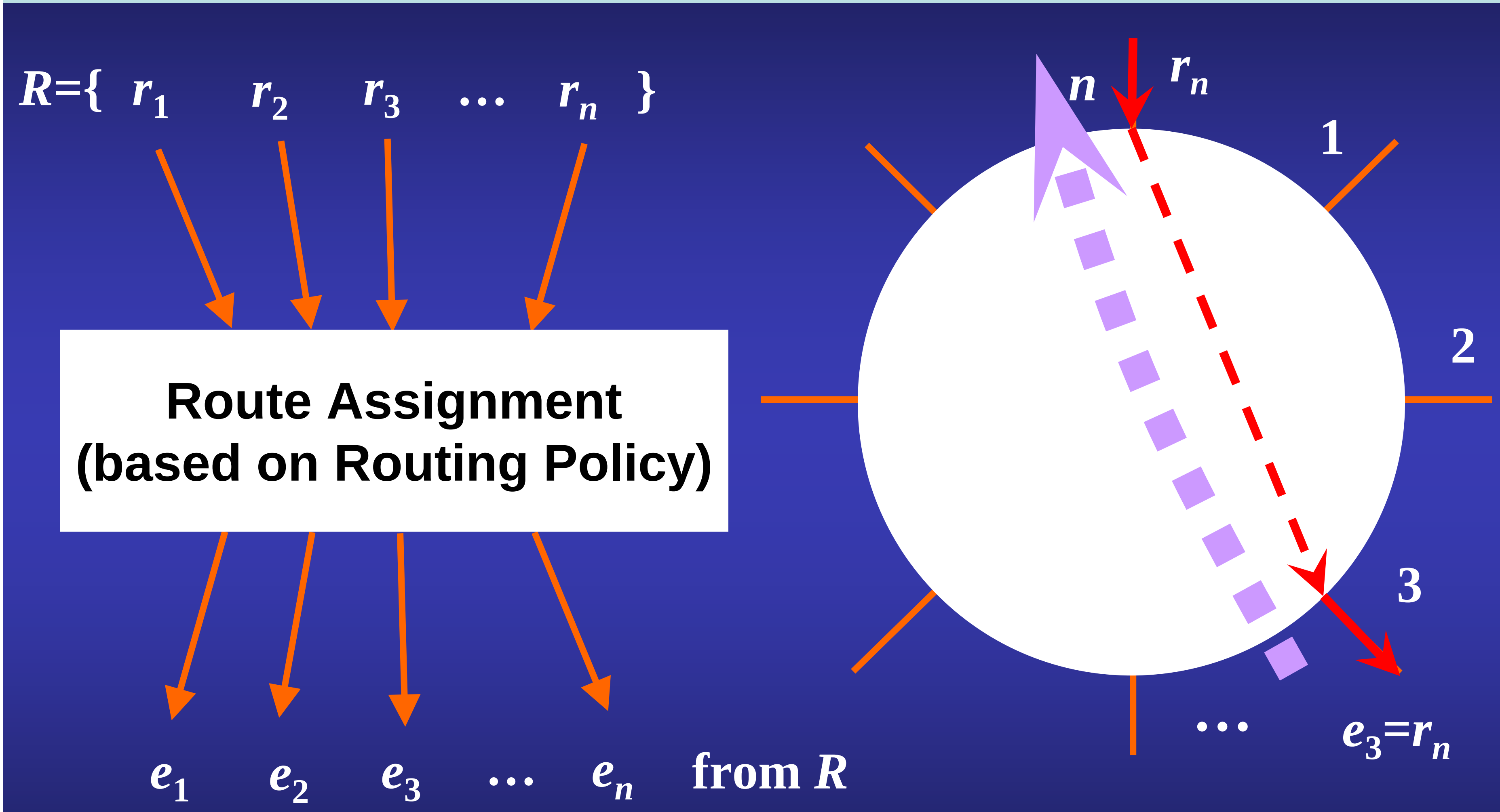
❑Practice

- Policy violation examples and possible fixes
- BGP mechanisms

❑Atomic Routing Theory (ART)

- Provide correctness guarantees for any policy
- Applied to BGP) atomic BGP

The Route Assignment Problem

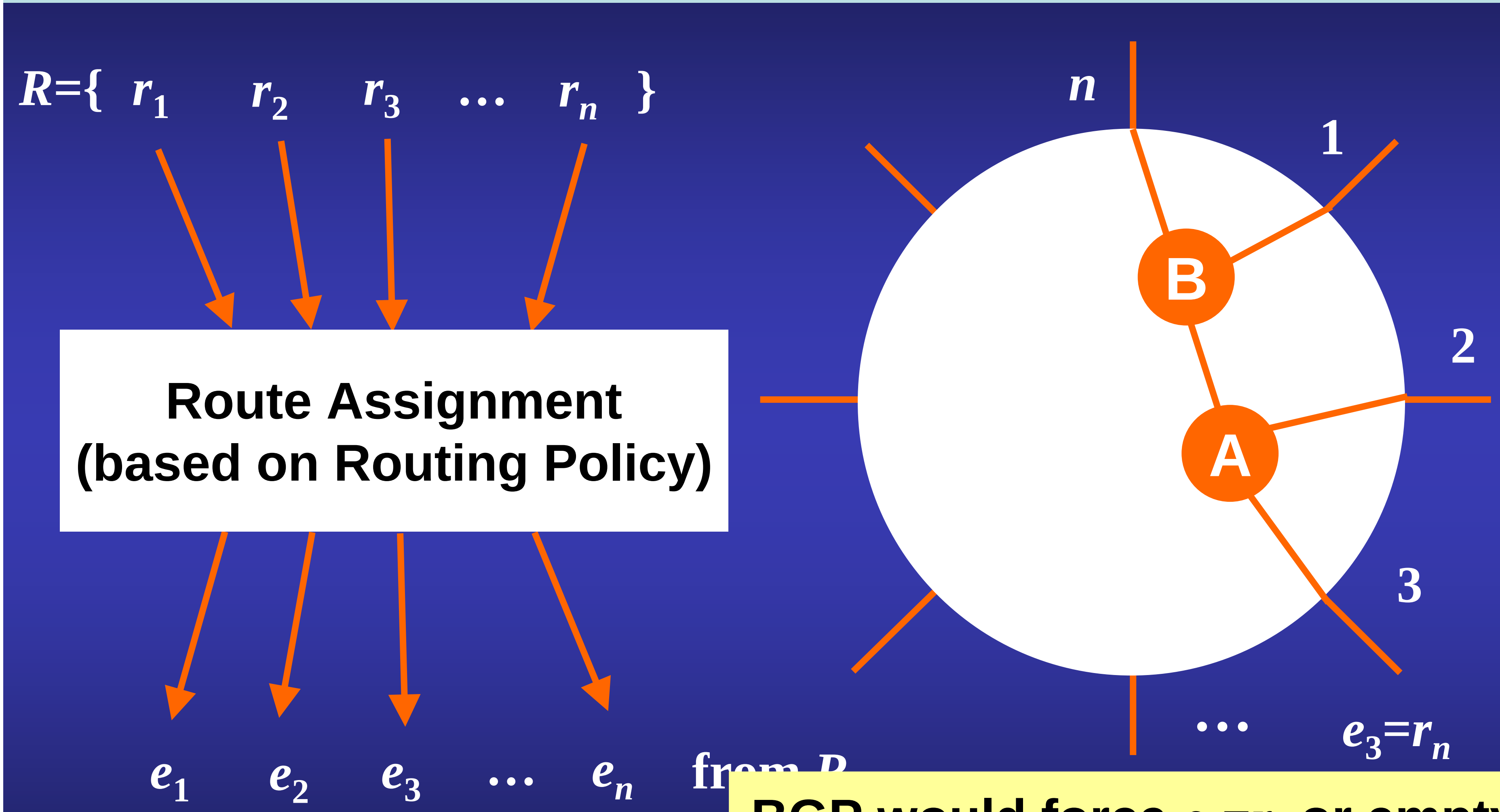


$R=\{\ r_1\quad r_2\quad r_3\quad \dots\quad r_n\ \}$

Route Assignment (based on Routing Policy)

$e_1\quad e_2\quad e_3\quad \dots\quad e_n$ **from** R

Protocol Restrictions



Route Assignment (based on Routing Policy)

$R=\{\ r_1\ \ r_2\ \ r_3\ \ \ldots\ \ r_n\ \}$
$e_1\ \ e_2\ \ e_3\ \ \ldots\ \ e_n$ from R
n
1
2
3
B
A
…
$e_3=r_n$

BGP would force $e_2=r_n$ or empty

No router can pick r_1

Policies BGP Cannot Realize Correctly

Let customers use MED

Us

Us

Do not carry traffic between peers

Equally prefer customer and peer routes

Equally prefer customer and peer routes

Use hot-potato routing

Expo

Let customers use export communities

Use hot-potato routing

Use route reflectors

Use hot-potato routing

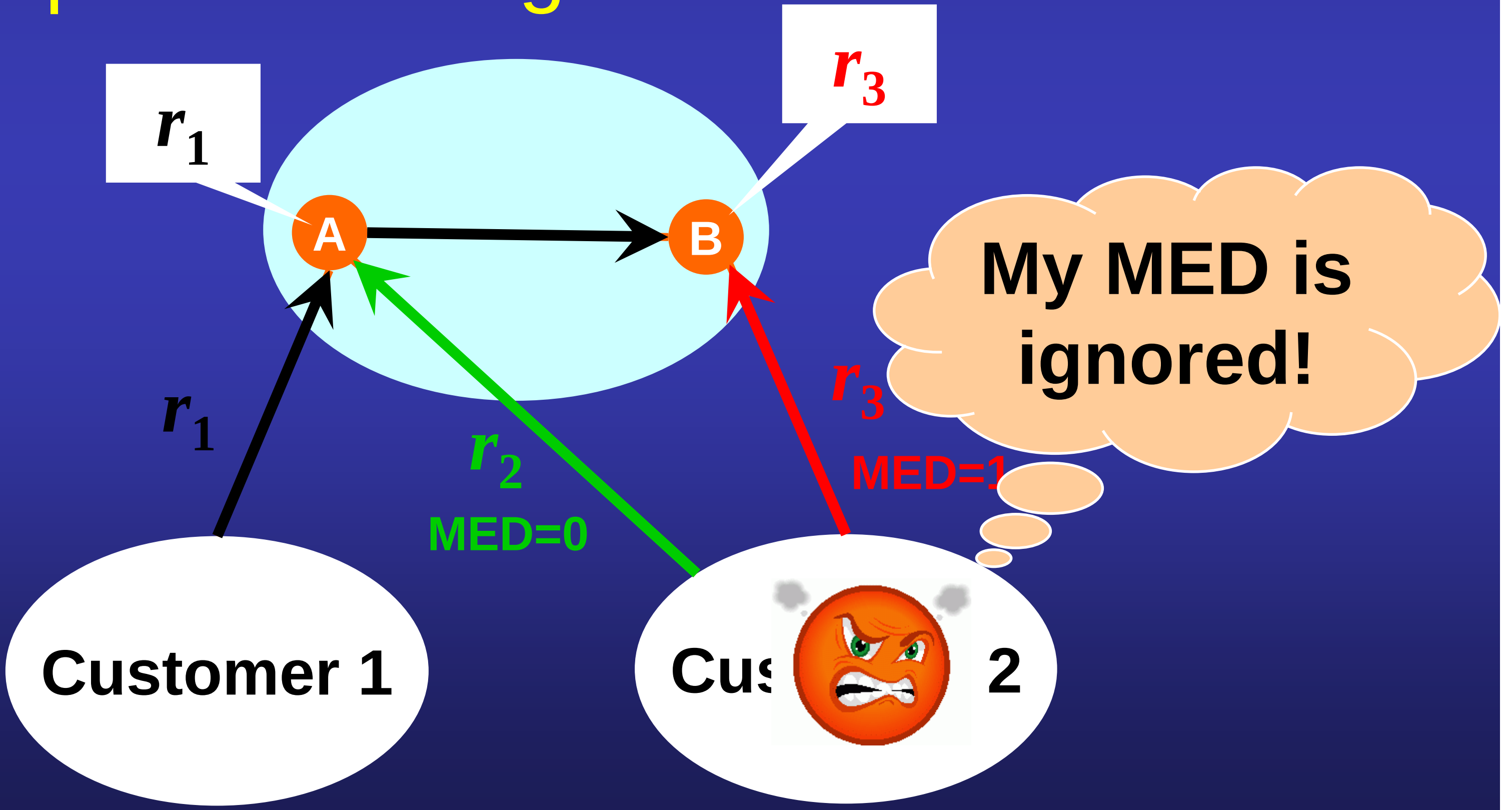
Let customers use MED

Three Possible Solutions

- ❑ Check the policy configuration for conflicts
  - Disallow conflicting policies, or
  - Live with the consequences
- ❑ Change the physical topology
  - Terminate links on different routers
- ❑ Extend the routers
  - Disseminate extra routes inside the AS
  - Export different routes on different links

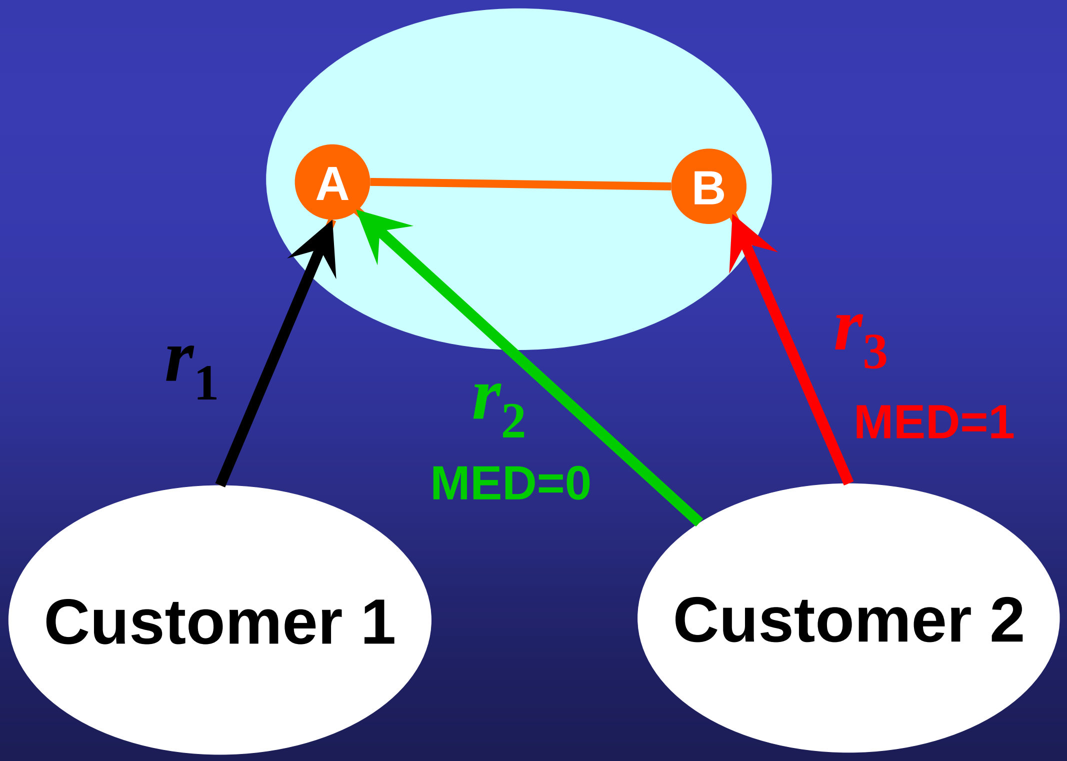
Example I

- ❑Let customers use MED
- ❑Use router ID tie break
- ❑Use hot-potato routing



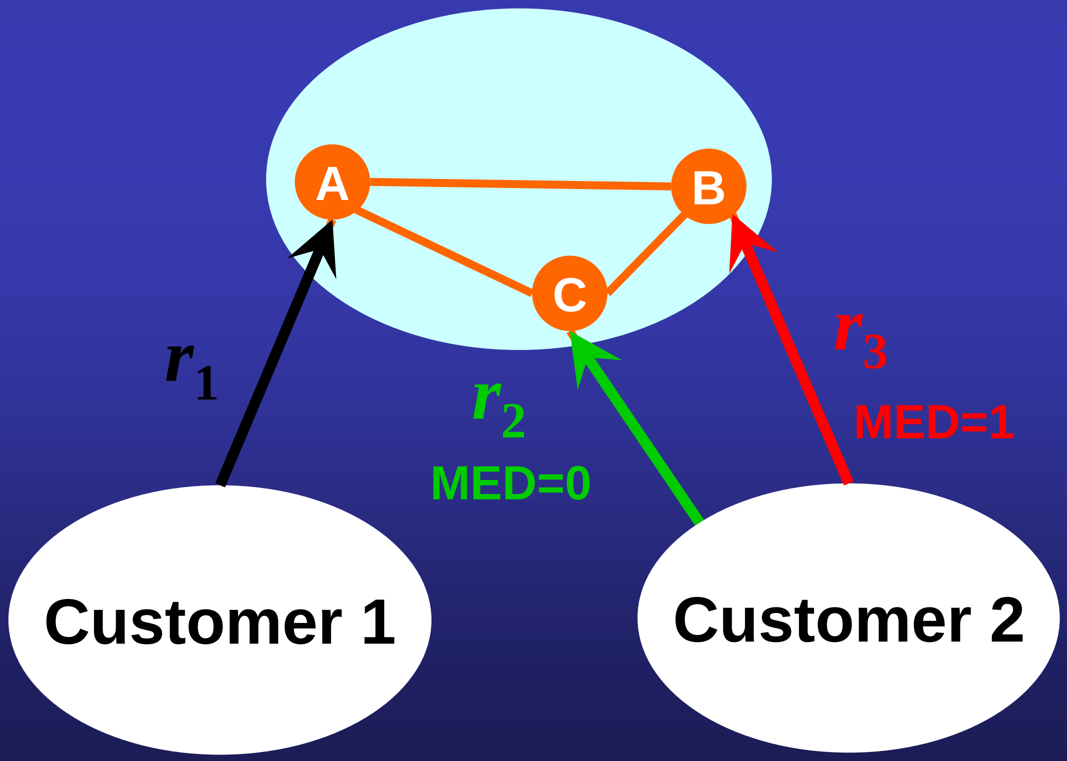
Example I: Solutions

- Ignore MED



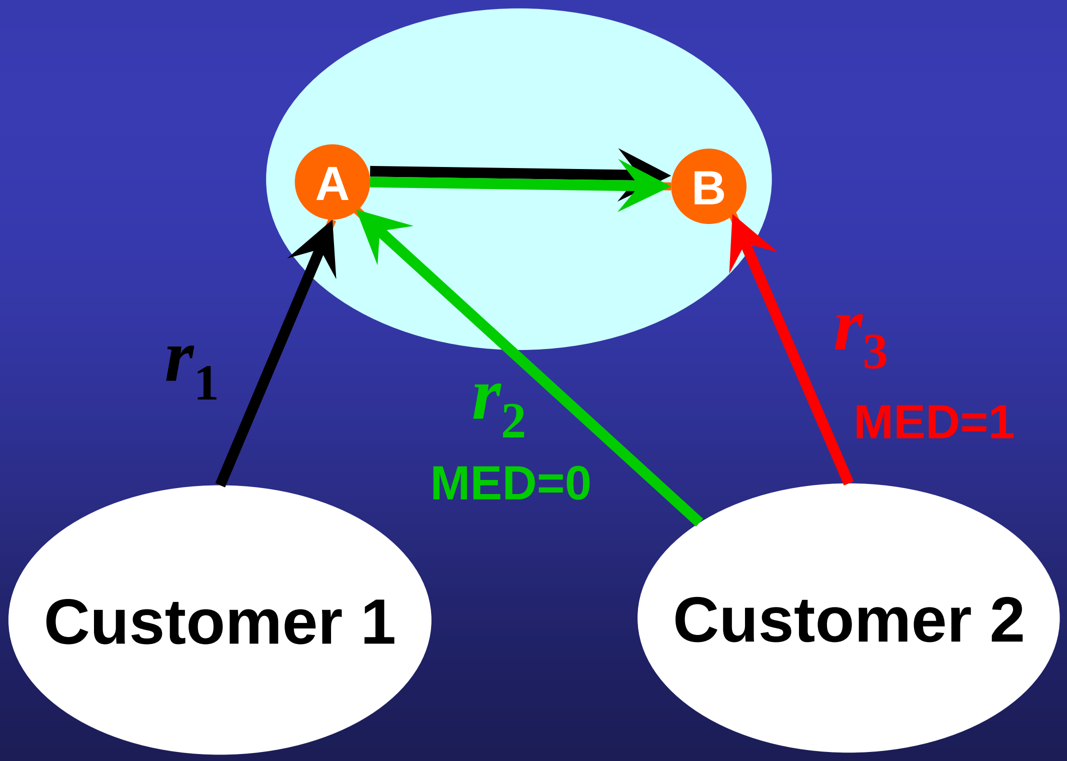
Example I: Solutions

- ❑Ignore MED
- ❑Terminate each “MED” link separately



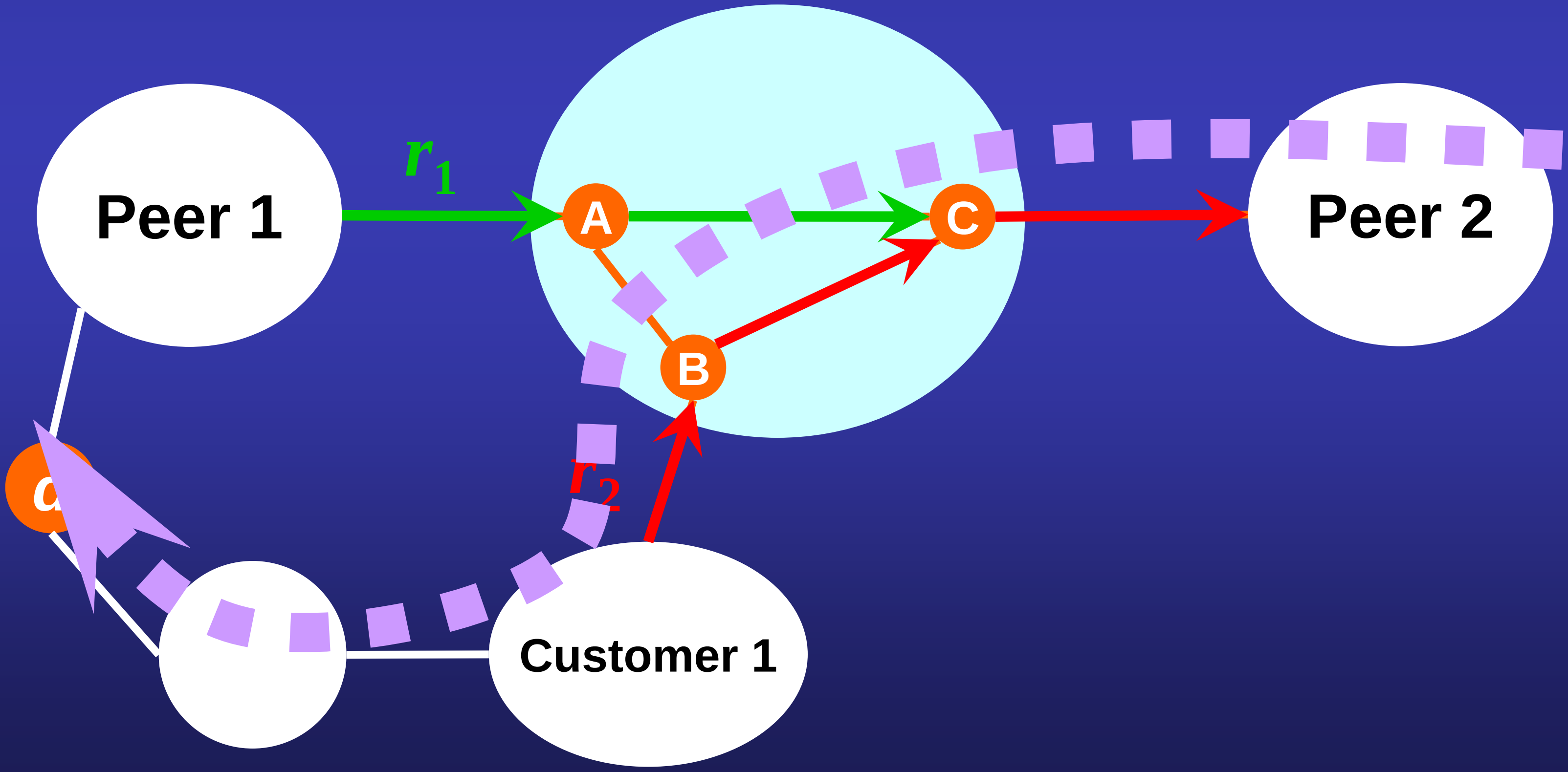
Example I: Solutions

- ❑Ignore MED
- ❑Terminate each “MED” link separately
- ❑Disseminate more than one route



Example II

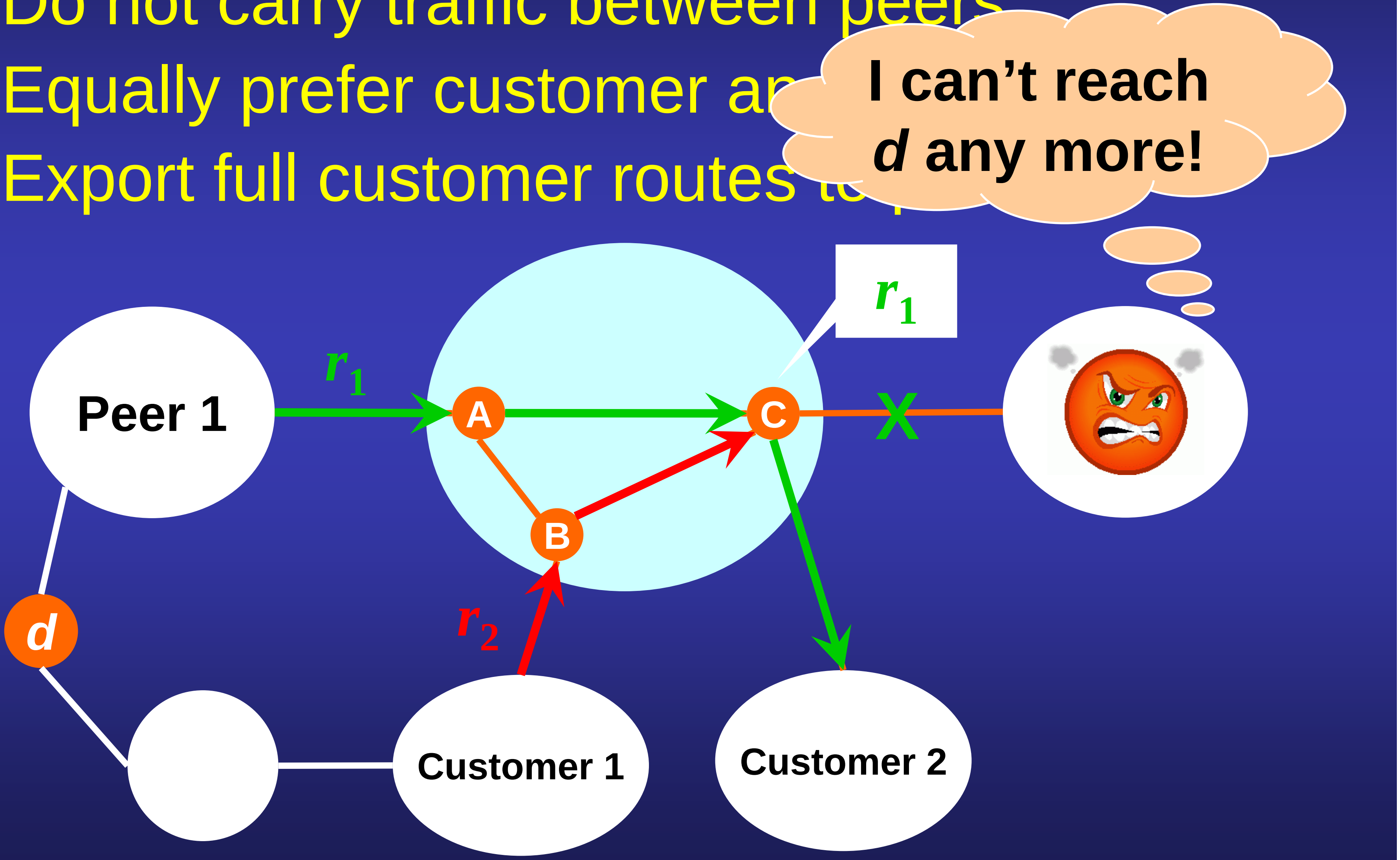
- ❑Do not carry traffic between peers
- ❑Equally prefer customer and peer routes
- ❑Export full customer routes to peers



R. Zhang-Shen

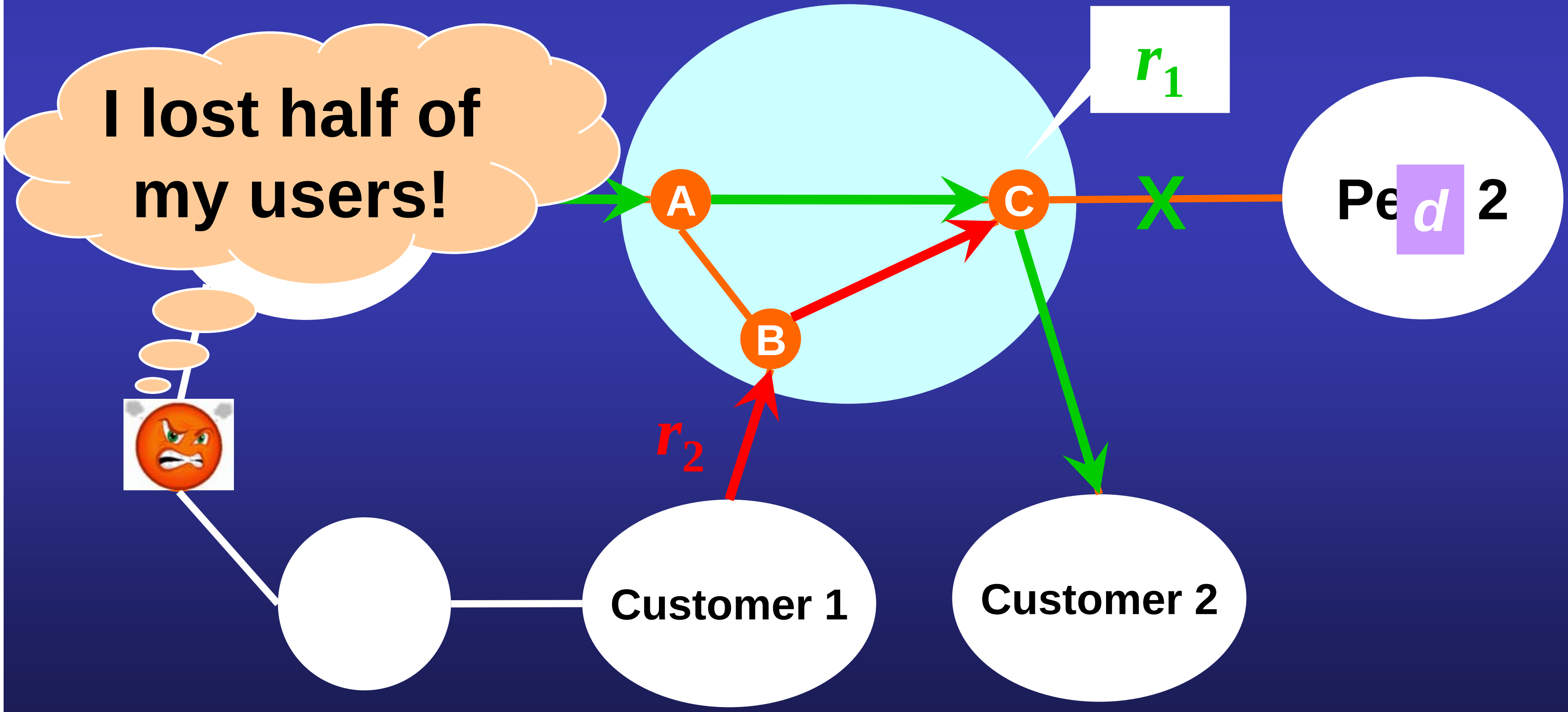
Example II

- ❑Do not carry traffic between peers
- ❑Equally prefer customer an
- ❑Export full customer routes to



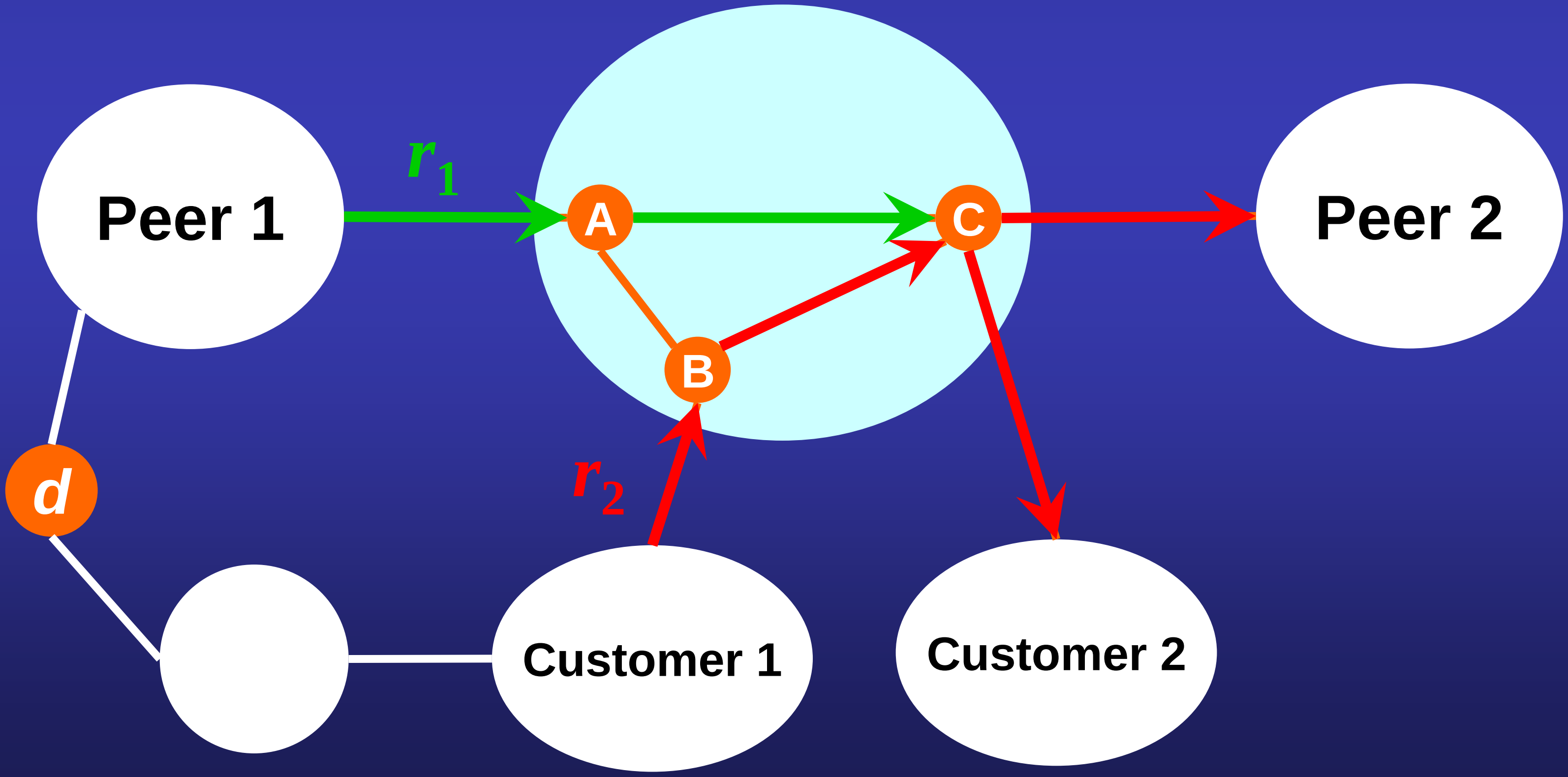
Example II

- ❑ Do not carry traffic between peers
- ❑ Equally prefer customer and peer routes
- ❑ Export full customer routes to peers



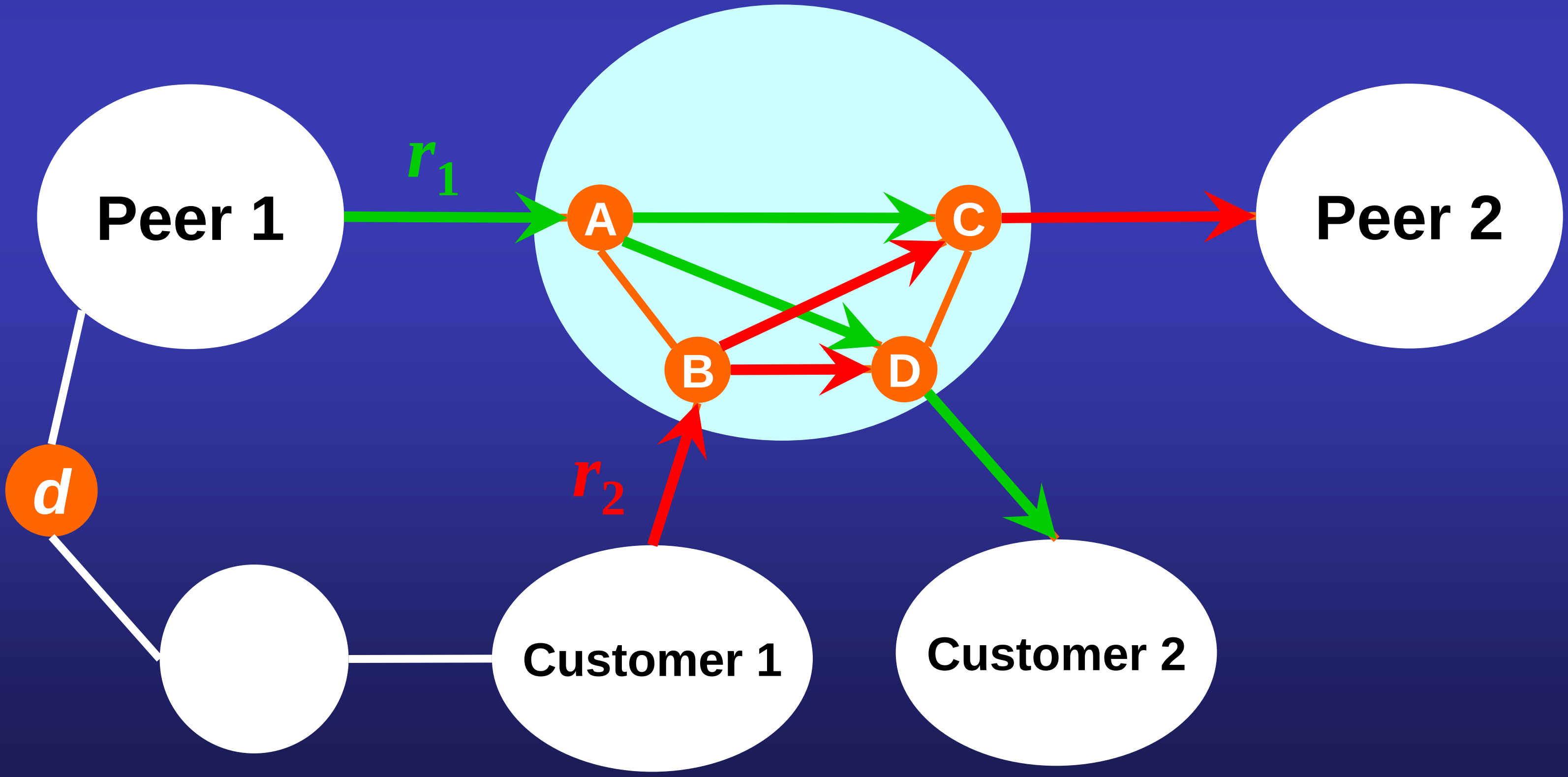
Example II: Solutions

- Do not provide premium service



Example II: Solutions

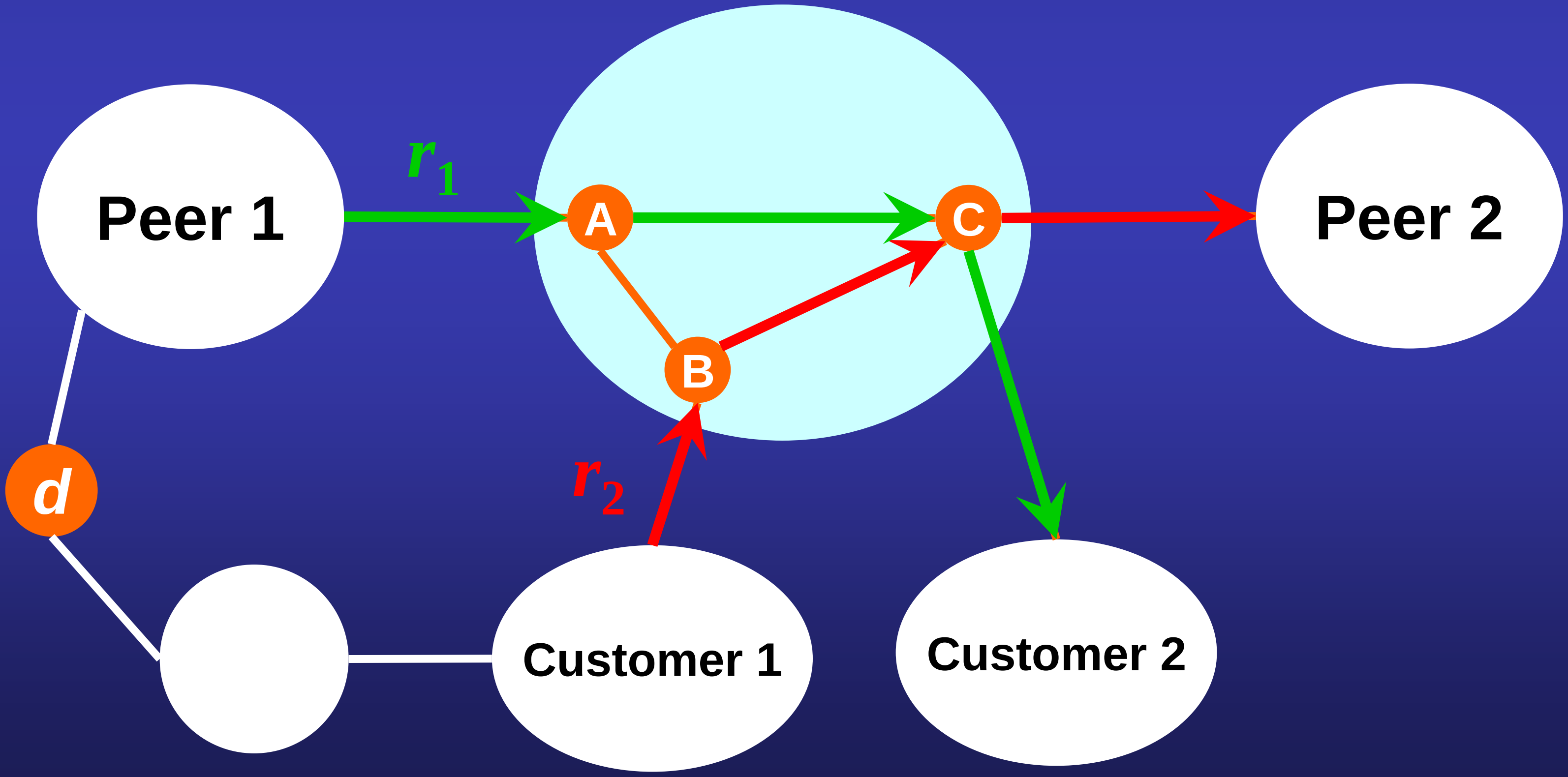
- ❑Do not provide premium service
- ❑Host peers and customers separately



R. Zhang-Shen

Example II: Solutions

- ❑Do not provide premium service
- ❑Host peers and customers separately
- ❑Let a router select and export multiple routes



R. Zhang-Shen

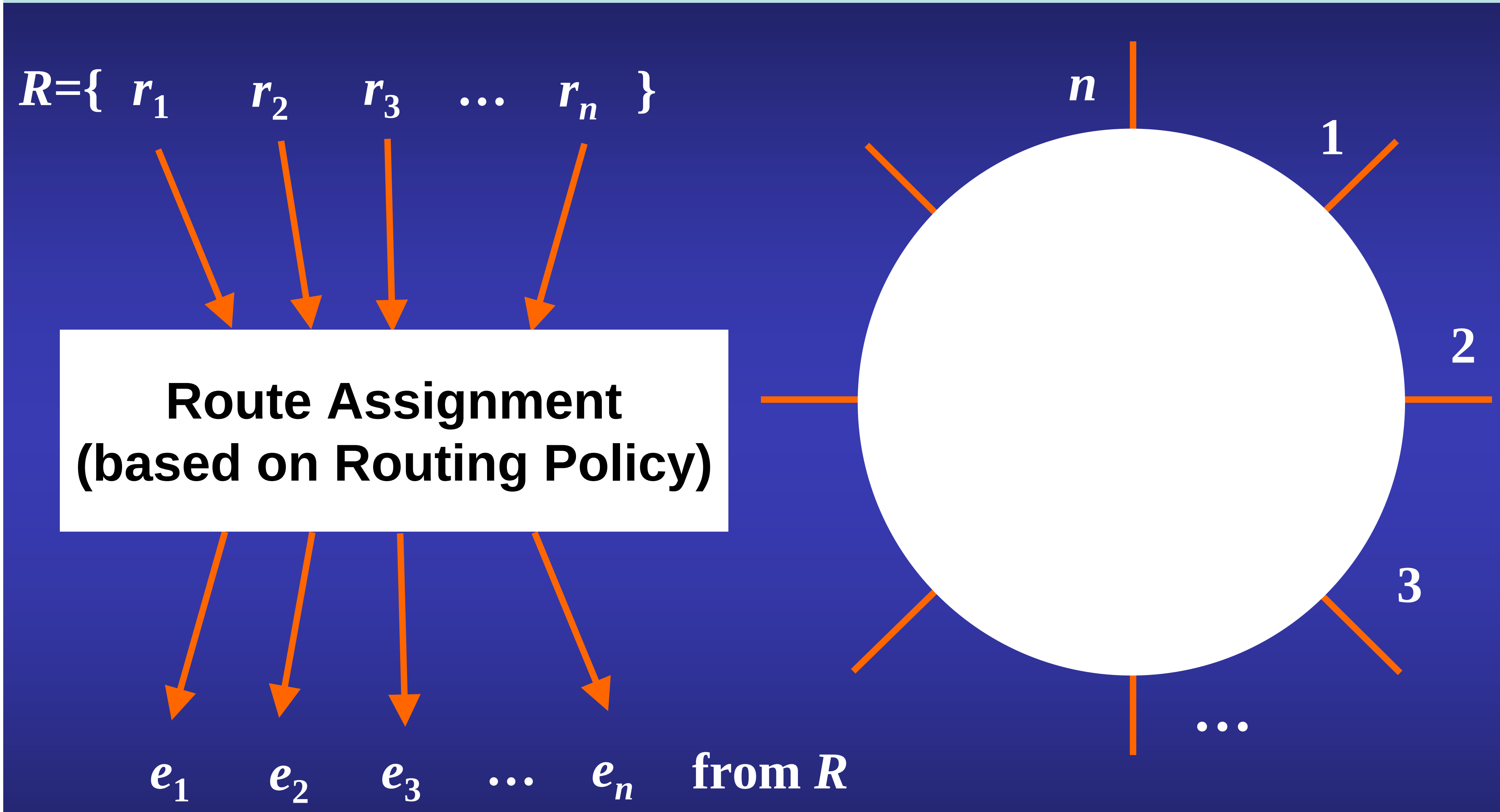
What will be the next example?

- Many more examples today
- BGP is constantly evolving
  - Will more examples be discovered?
  - In fact, this is what has happened
- Need to be proactive
  - Make sure there are no more surprises
- Need a theoretical model
  - Atomic Routing Theory (ART)

ART Is a Practical Theory

- ❑When realizing a policy
  - ▪ Find out potential policy violations
- ❑When introducing new features to BGP
  - ▪ Ensure no new violations are introduced
- ❑When proposing new policy-based routing protocols
  - ▪ Ensure the desired policies can be realized
- ❑When analyzing multiple AS interactions
  - ▪ Safely model an AS as a single node

The Route Assignment Problem



$R=\{\ r_1 \quad r_2 \quad r_3 \quad \ldots \quad r_n\ \}$

Route Assignment (based on Routing Policy)

$e_1 \quad e_2 \quad e_3 \quad \ldots \quad e_n$ **from** R

Defining Policy Correctness

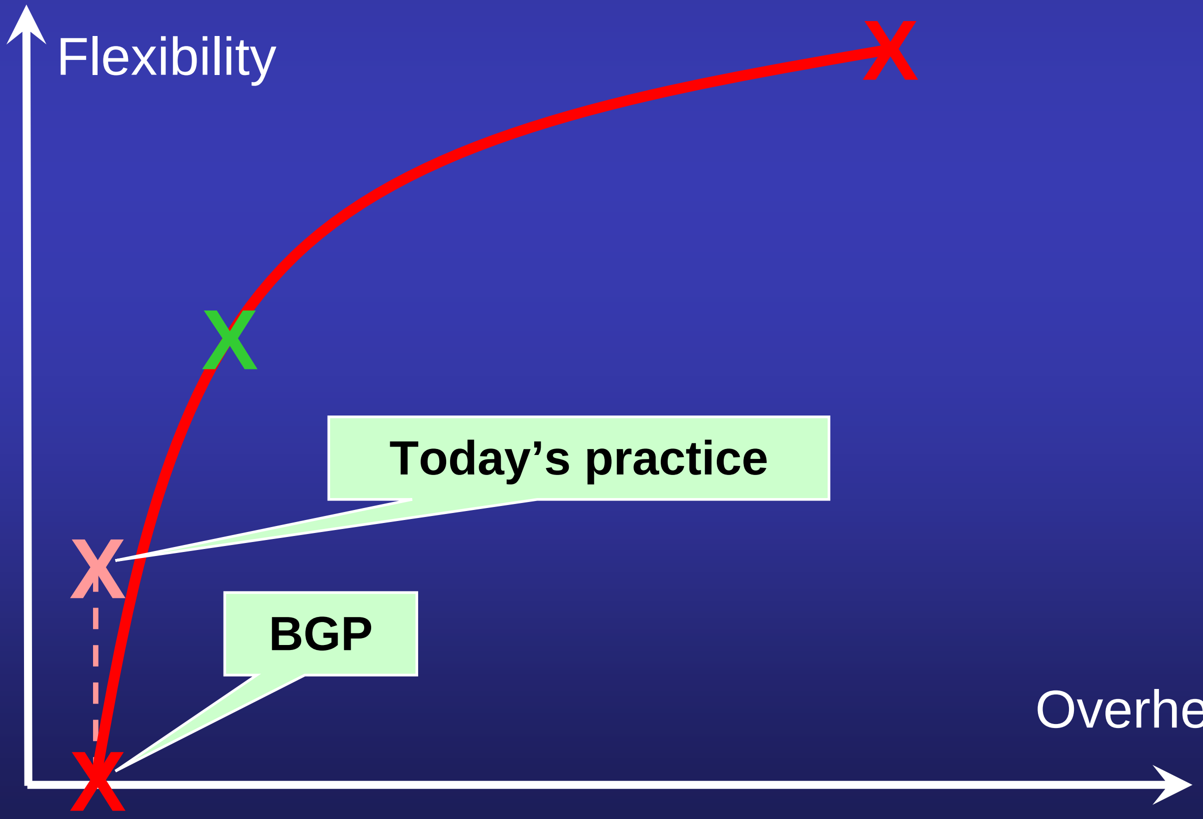
- Route assignment E_i for neighbor AS i
  - Set of routes exported to neighbor AS i
- AS-wide preference function $B_i()$
  - $B_i(R)$ is set of best routes from our perspective
- Consistency requirement $C_i()$
  - $C_i(S)$ returns TRUE if routes in S are considered consistent by neighbor i
- Realizing a policy correctly means
  - E_i ½ $B_i(R)$
  - $C_i(E_i)=\mathrm{TRUE}$

Centralized Atomic Routing

- ❑Learn all the routes R
- ❑For each neighbor AS i
  - Calculate best set $B_i(R)$
  - Find a subset E_i ½ $B_i(R)$ such that $C_i(E_i)$=TRUE
  - Assign routes in E_i to links connected to neighbor i
- ❑In practice, only a few classes of neighbors:
  - Customers
  - Peers
  - Providers

Distributed Atomic Routing

- ❑Disseminate all routes to all routers
  - Any policy



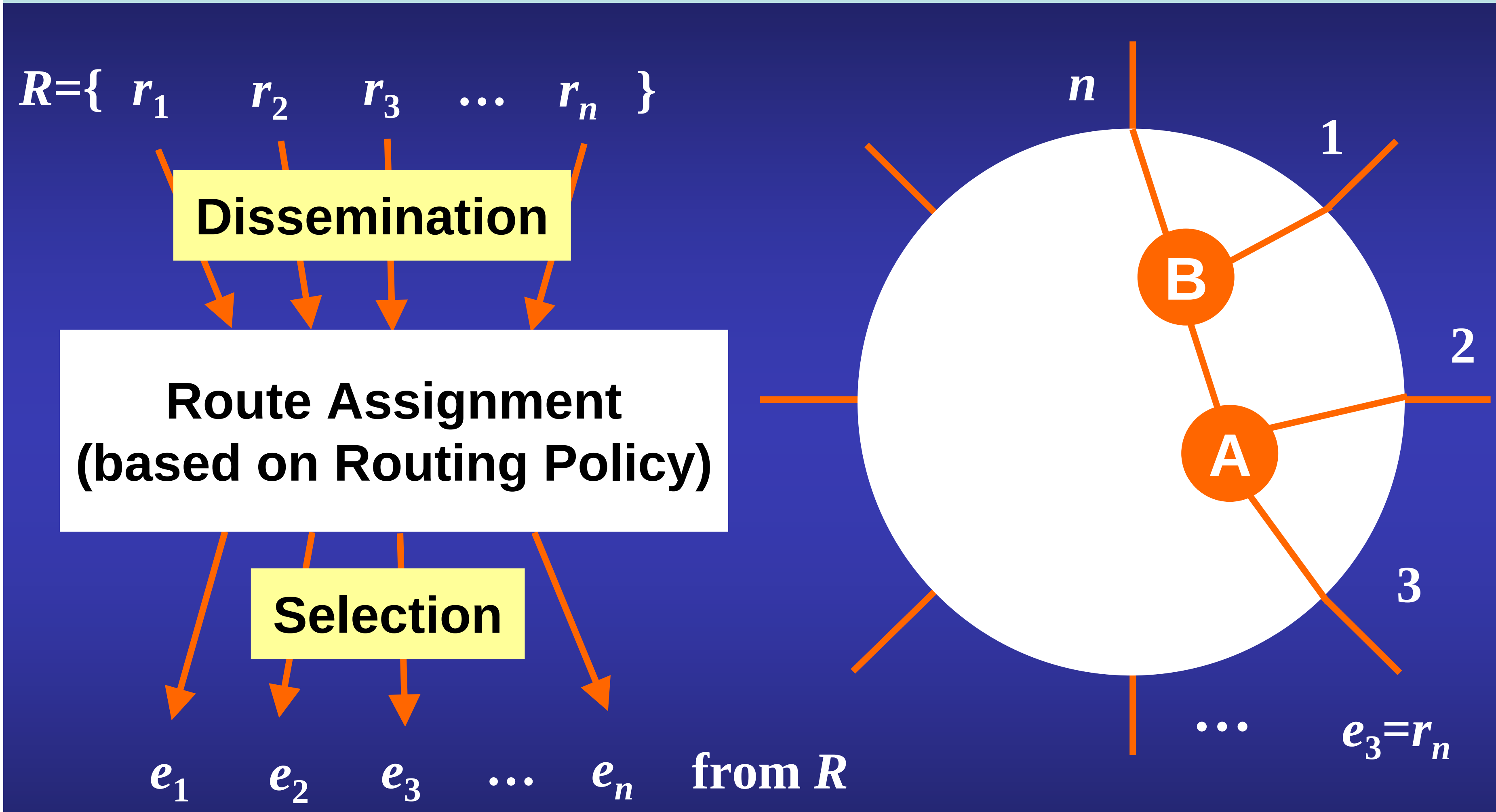
The Sweet Spot: Flexibility

- Realizes all policies based on comparing route attributes
  - Supports today’s common policies correctly
  - And any new attributes to be added
- Mathematically
  - If $S = Y_k S_k$, then $B_i(S) = B_i(Y_k B_i(S_k))$
  - If $C_i(S)=\mathrm{TRUE}$ and T ½ S, then $C_i(T)=\mathrm{TRUE}$
  - $C_i(B_i(S))=\mathrm{TRUE}$ for any S

Example: Today's BGP

- ❑ Policy is defined by AS-wide preference $B_i()$
  0. Import actions
  1. Highest local_pref
  2. Shortest AS path
  3. Lowest origin type
  4. Lowest MED among routes from the same neighbor AS
  8. Export actions
- ❑ Consistency check $C_i()$
  - E.g., AS-path length

How fare are we from atomicity?



$R = \{\ r_1 \quad r_2 \quad r_3 \quad \ldots \quad r_n\ \}$

Dissemination

Route Assignment
(based on Routing Policy)

Selection

$e_1 \quad e_2 \quad e_3 \quad \ldots \quad e_n$ from R

The Sweet Spot: Implementation

- Among set of best routes at the router, disseminate
  - At least one route
  - At least one route from a neighbor using MED
  - E.g., use the new ADD-PATH feature
- Select routes for each neighbor class independently
  - Separate forwarding table per neighbor class
  - E.g., a different forwarding table on each linecard (VRF)
- Forward traffic b oute assignment
  - Tunneling fro gress link

Atomic BGP

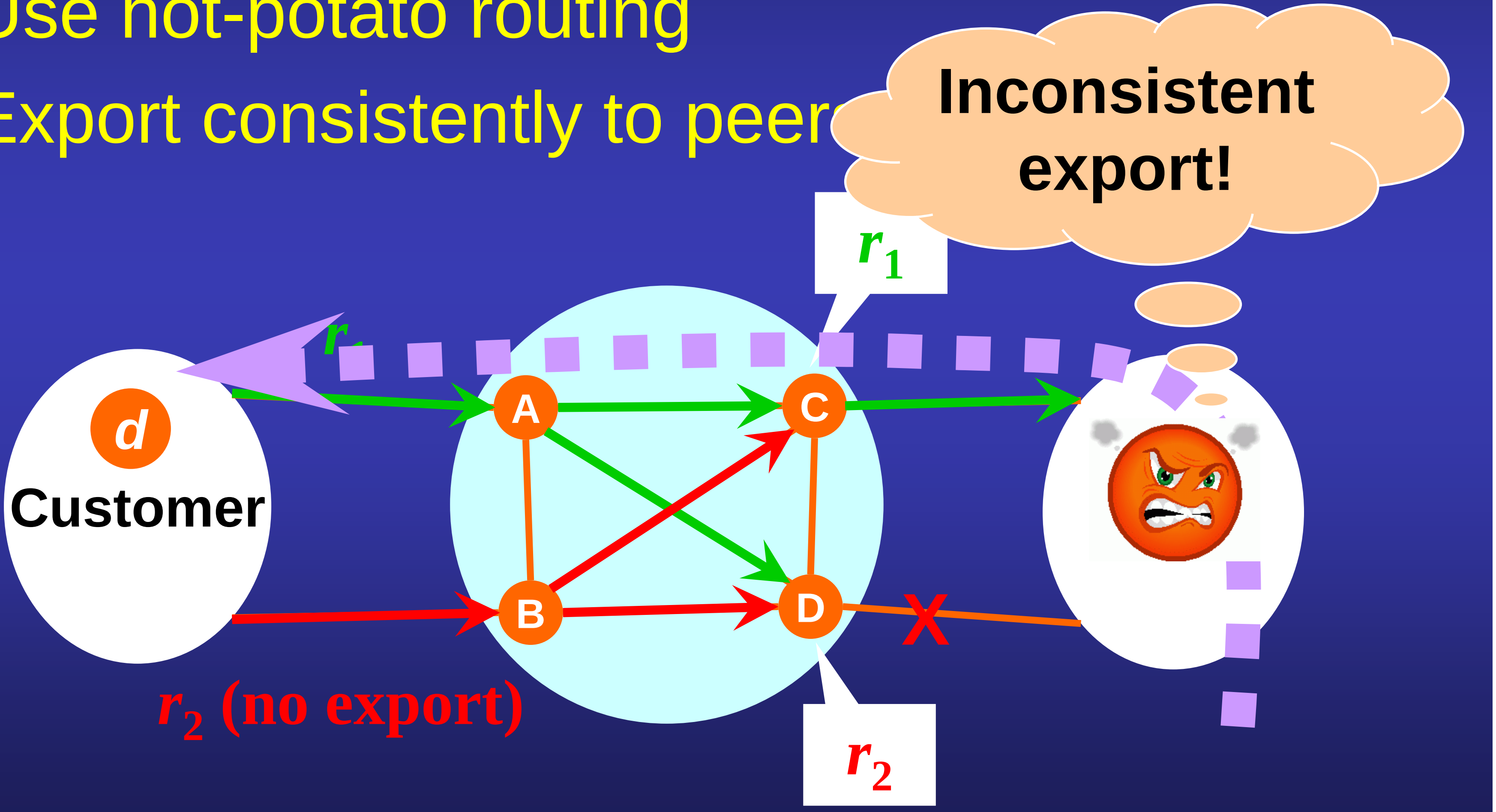
Atomic BGP Offers

- ✓ Correct policy realization
  - Define the AS-wide policy
  - Configure each router with it
- ✓ Simplified router configuration
  - No restrictions on the physical topology
  - All routers configured with the same policy
- ✓ Low protocol overhead
  - Dissemination of routes within the AS
  - Storage for routing tables
- ✓ Incremental deployability
  - Changes local to a single AS
  - Only modest changes to the routers

Backup Slides

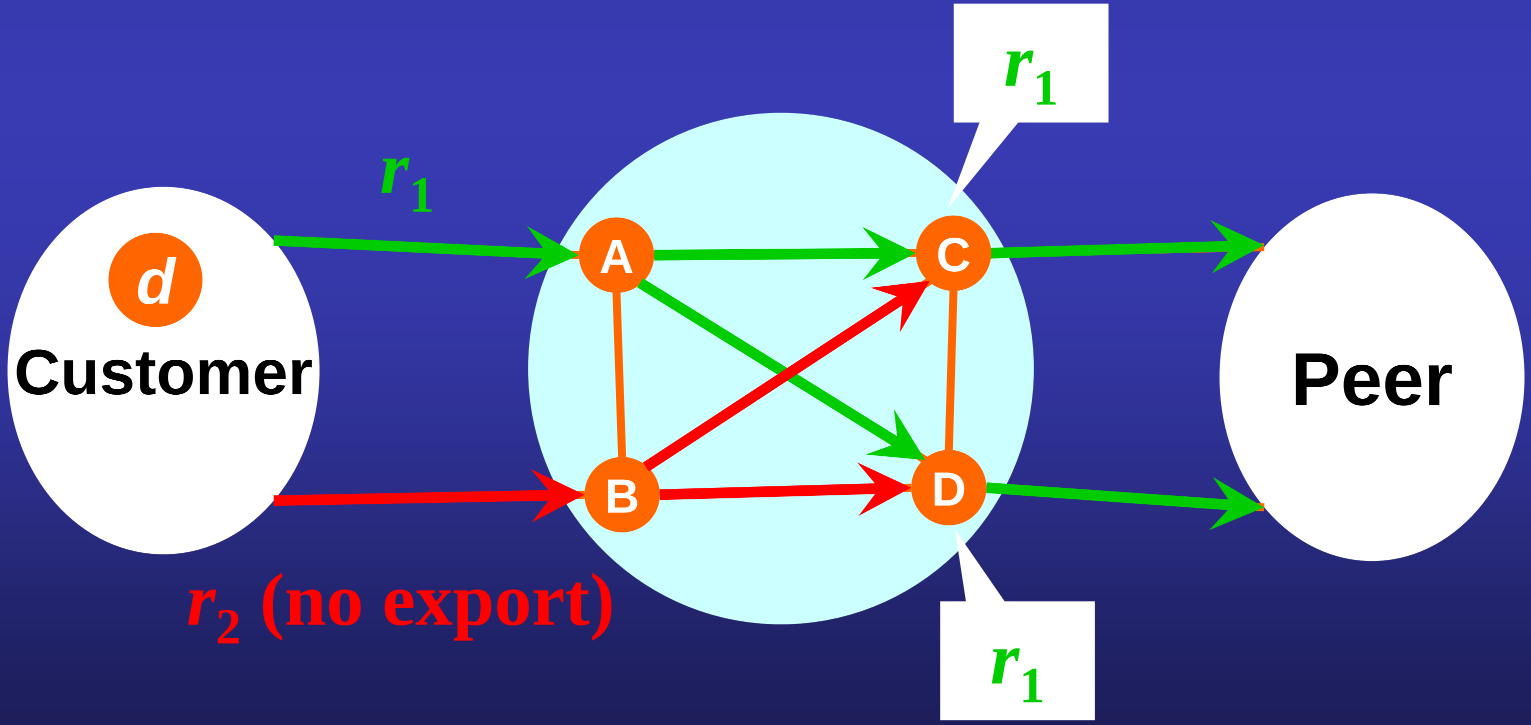
Example I

- □Let customers use export communities
- □Use hot-potato routing
- □Export consistently to peers



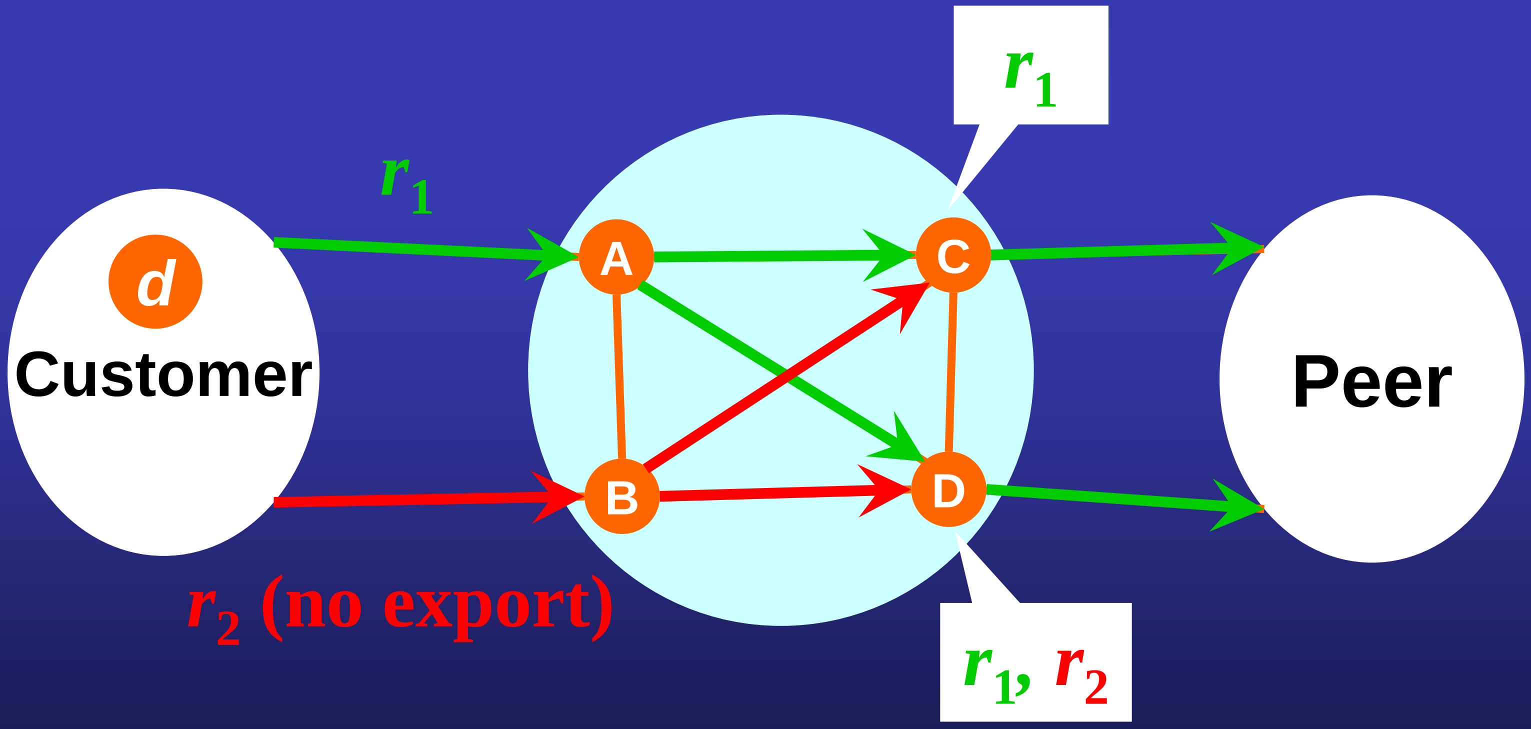
Example I: Solutions

- ❑Disallow export communities
- ❑Change how communities are handled



Example I: Solutions

- ❑Disallow export communities
- ❑Change how communities are handled
- ❑Allow a router to select multiple routes



Sufficient Condition I

1. Strictly prefer customer-learned routes
2. Do not use MEDs
3. Do not use communities
4. Use full-mesh iBGP
5. Use hot-potato routing

Sufficient Condition II

1. Connect customers to different routers than peers and providers
2. Connect neighbors with MED (who may advertise the same prefix) to different routers
3. Handle communities that dictate export actions before and during the decision process
4. Use full-mesh iBGP
5. Use hot-potato routing