CAP for Networks

OR: HOW TO STOP WORRYING AND EMBRACE FAILURE

Aurojit Panda, Colin Scott, Ali Ghodsi, Teemu Koponen, Scott Shenker
UC Berkeley, KTH, VMware, ICSI

Keshav raps about SDN

CAP Theorem

In the presence of network **P**artitions pick one of

- Service **C**orrectness
- Service **A**vailability

CAP Theorem: Impact

Divides the database community (even today)

SQL
Correctness above all

PostgreSQL

NoSQL
Availability above all

How does the CAP theorem apply to networks?

What about Networks?

What about Networks?

Traditionally connectivity was the only concern

- **Correctness**: Deliver packets to destination
- **Availability**: Deliver packets to destination
- **Correctness** is the same as **Availability**

The move to SDN

The move to SDN

SDN provides more sophisticated **functionality**:

- Tenant isolation (ACL enforcement)
- Fine grained load balancing
- Virtualization

The move to SDN

SDN provides more sophisticated **functionality**:

- Tenant isolation (ACL enforcement)
- Fine grained load balancing
- Virtualization

Control plane partitions no longer imply data plane partitions

- Control traffic often does not use data plane network

Availability ≠ Correctness

During control plane partitions

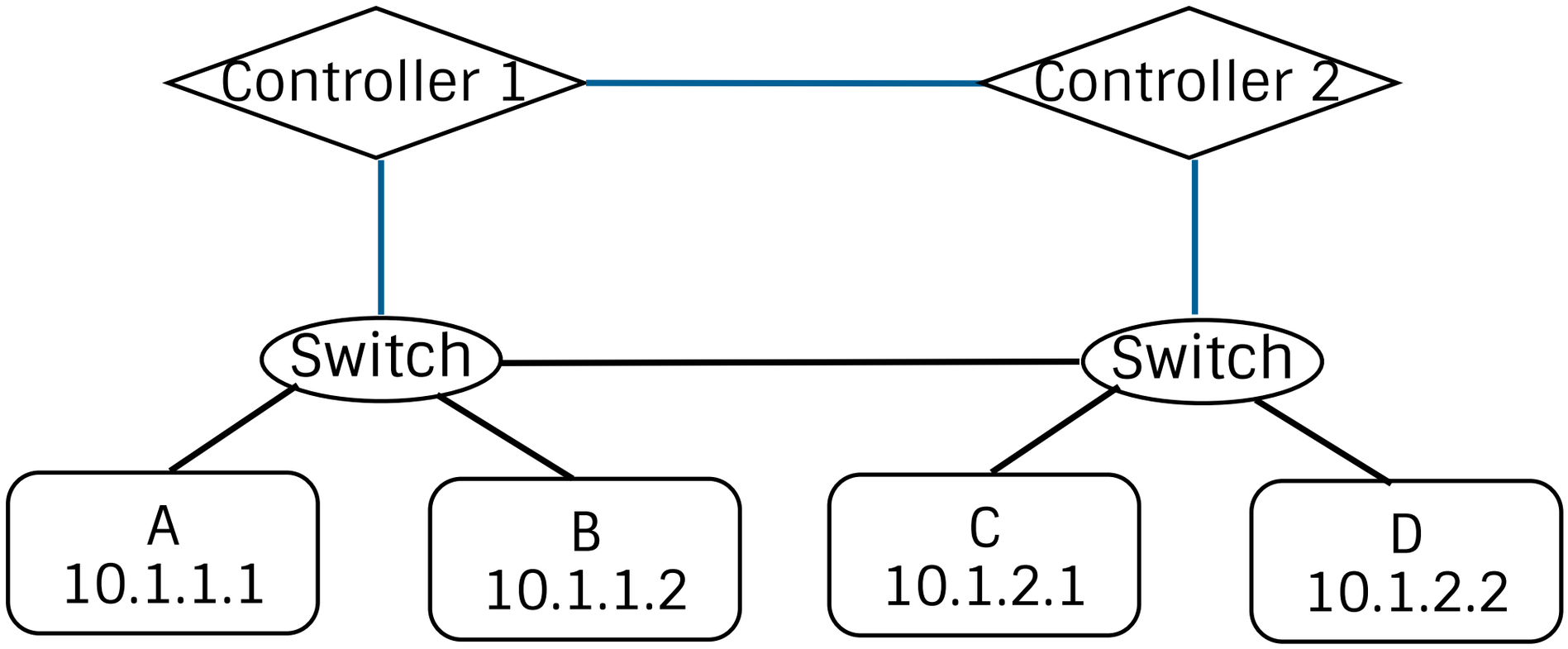
- Data plane connected => Deliver packets (Availability)
- Inconsistent control plane data (~~Correctness~~)
- **Availability** does not imply **Correctness**

How does the CAP theorem apply to ~~networks~~ SDN?

How does the CAP theorem apply to ~~networks~~ SDN?

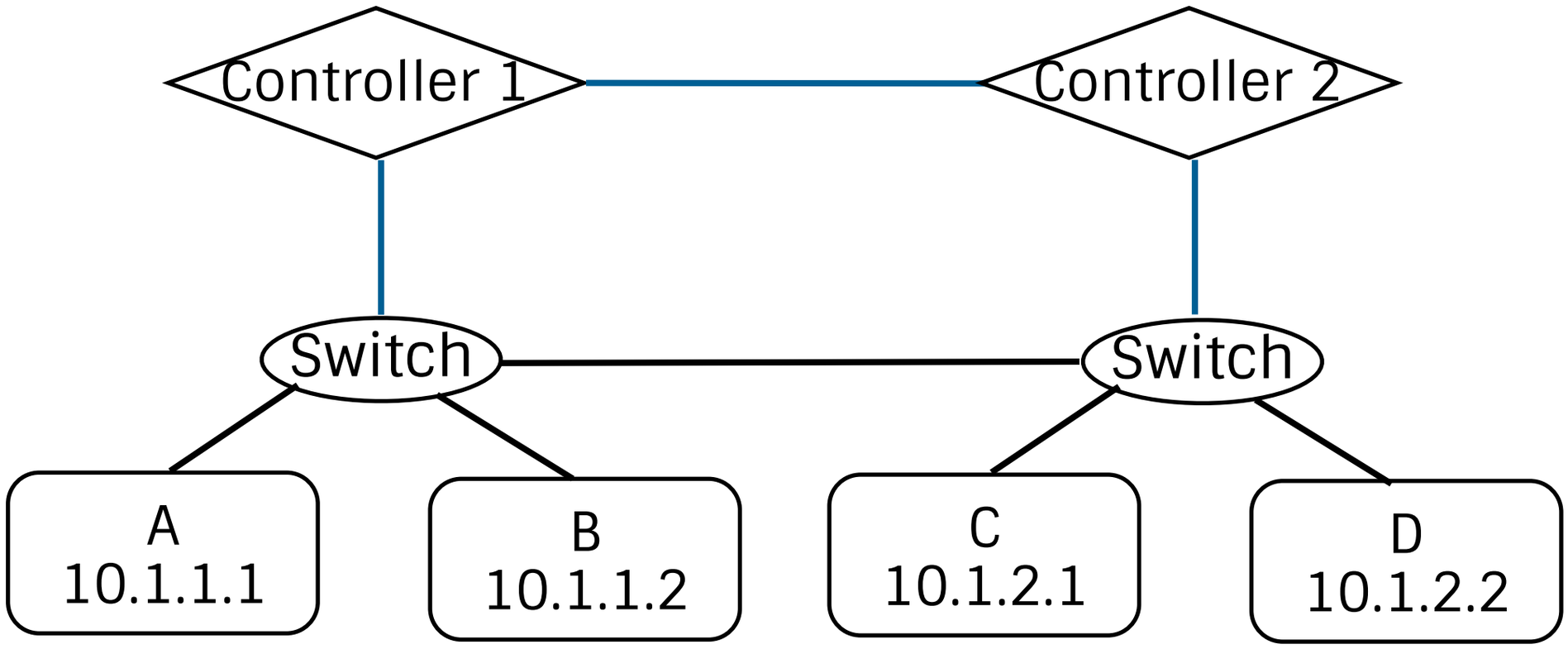
Can one provide correct isolation and availability in the presence of link failures?

Network Model



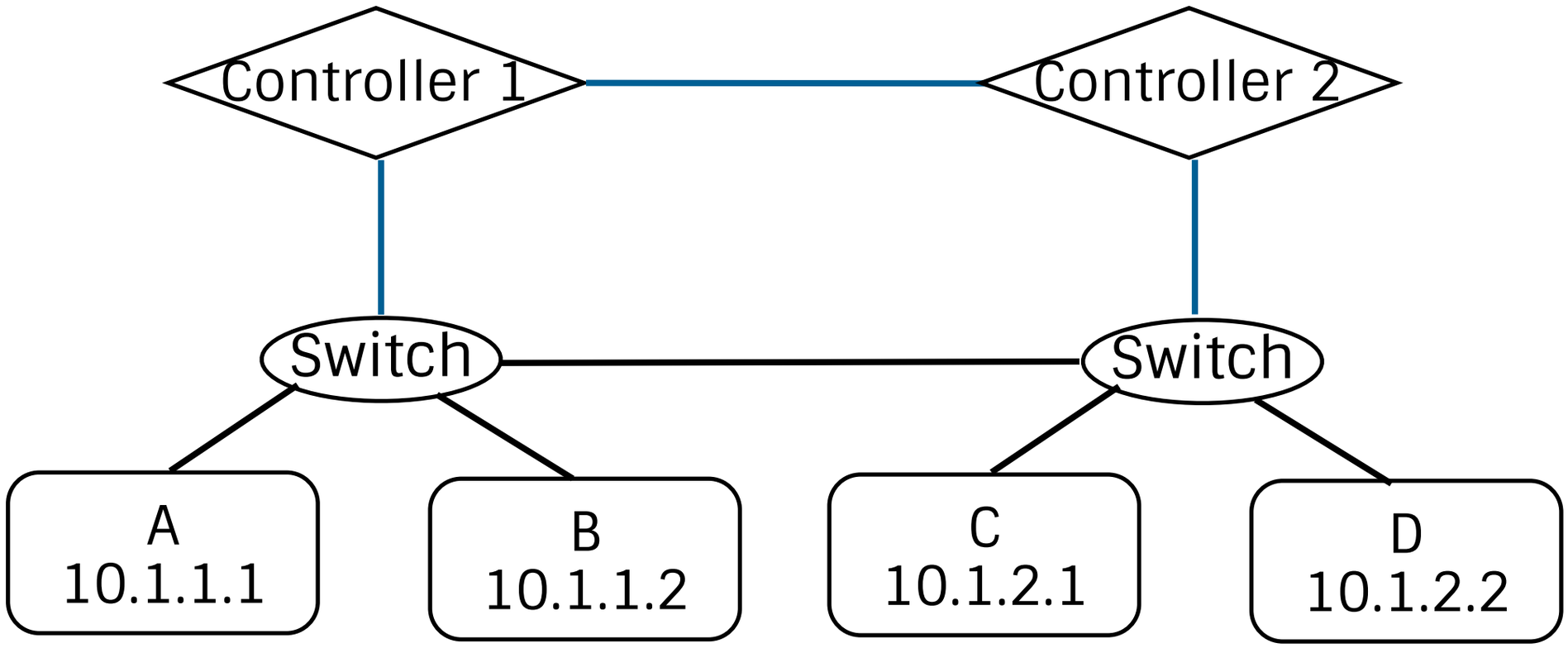
- Out-of-band control network.

Network Model



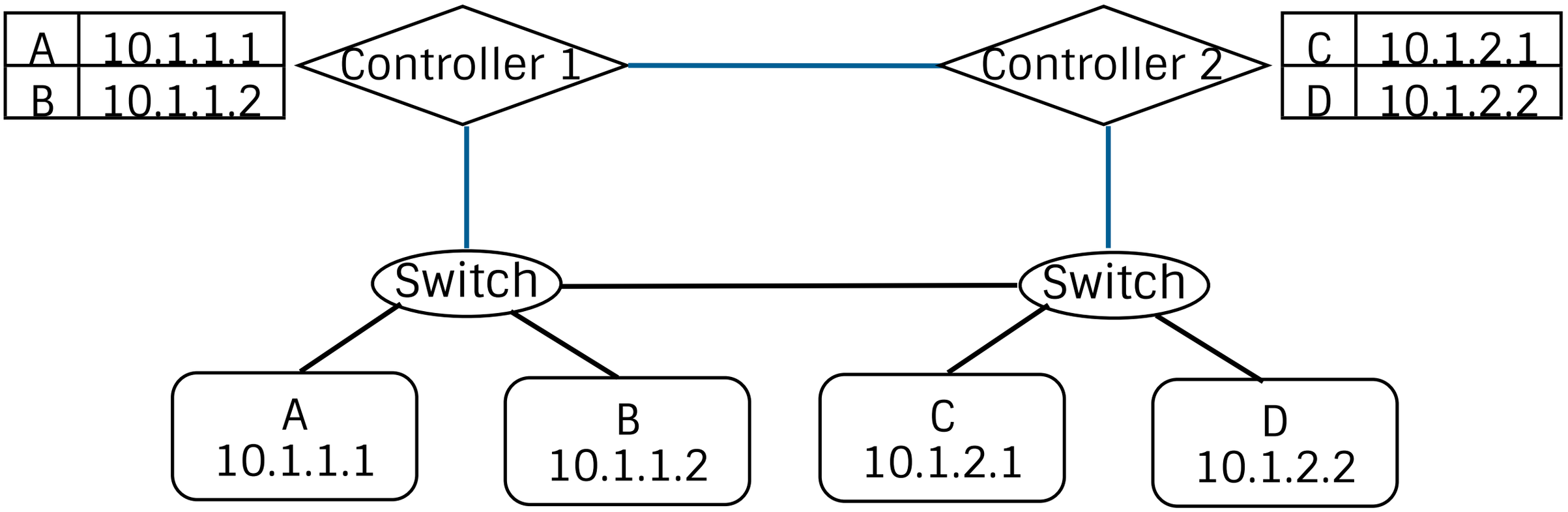
- Out-of-band control network.
- Routing and forwarding based on addresses.

Network Model



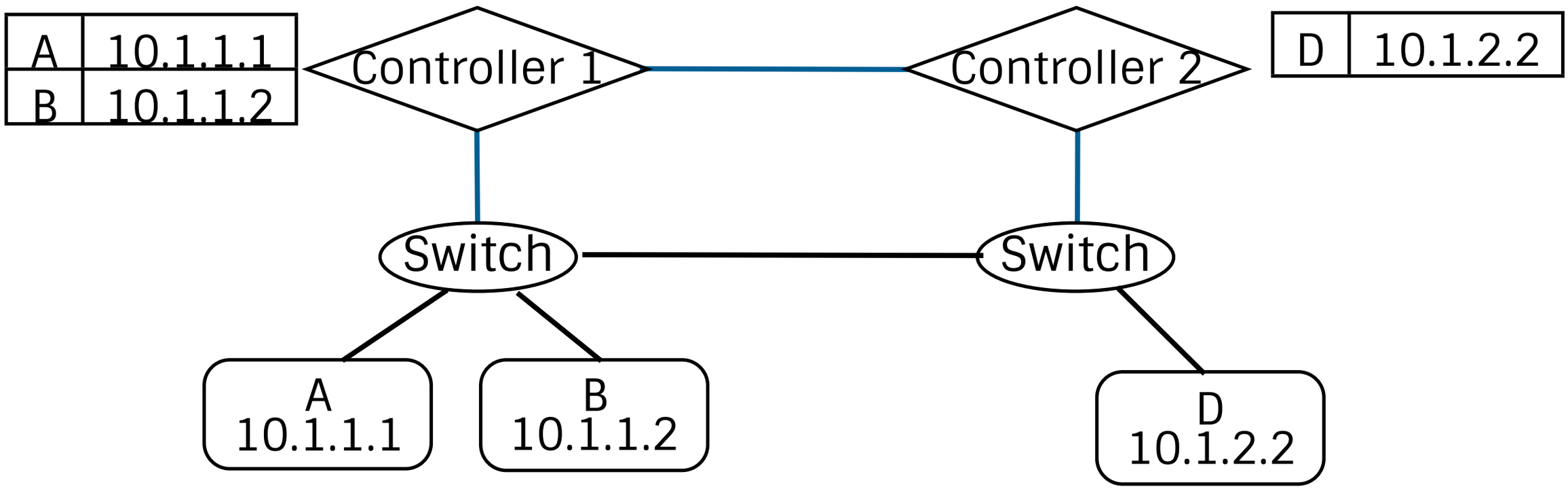
- Out-of-band control network.
- Routing and forwarding based on addresses.
- Policy specification using end-host names.

Network Model



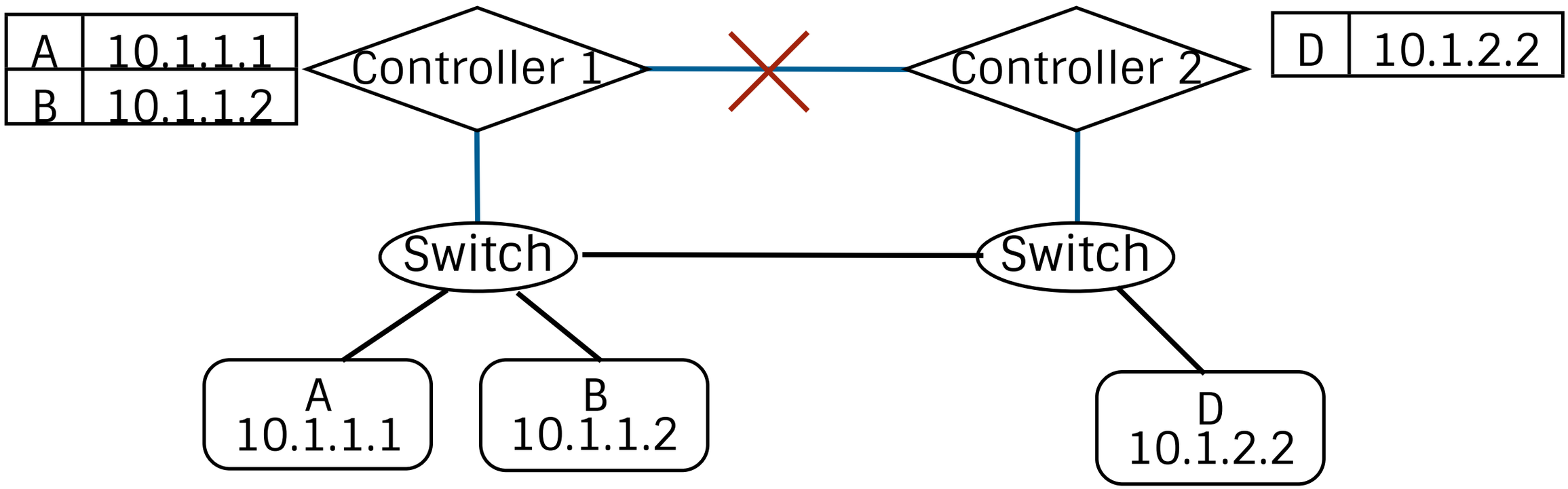
- Out-of-band control network.
- Routing and forwarding based on addresses.
- Policy specification using end-host names.
- Controller only aware of local name-address bindings.

Isolation Result



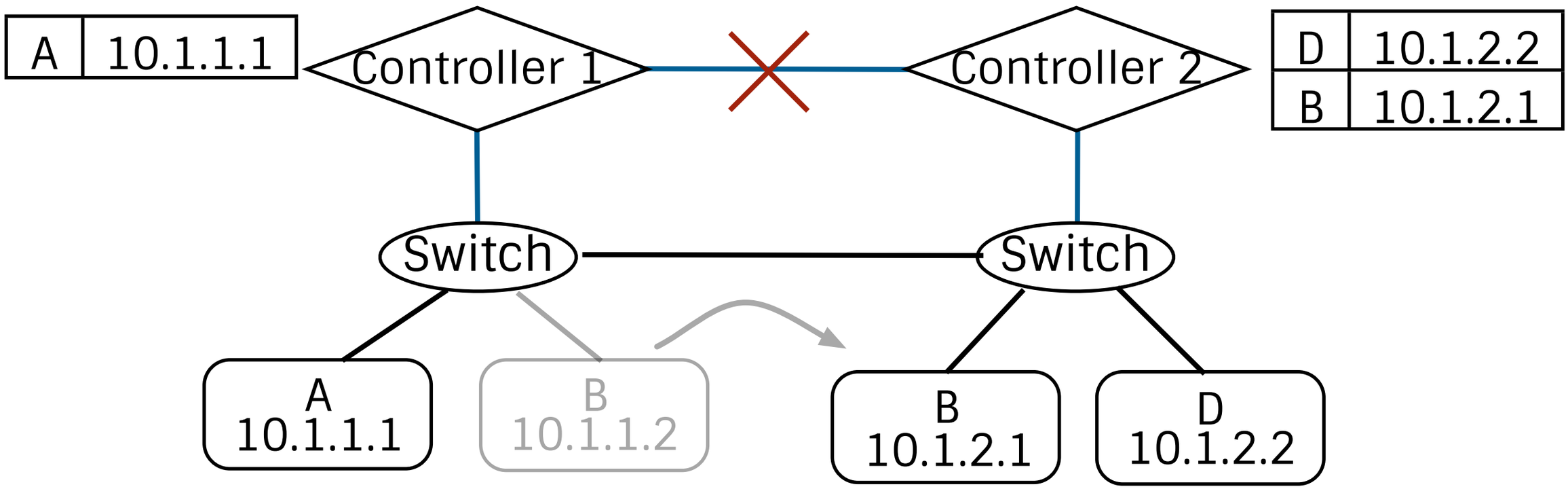
- Consider policy isolating A from B.

Isolation Result



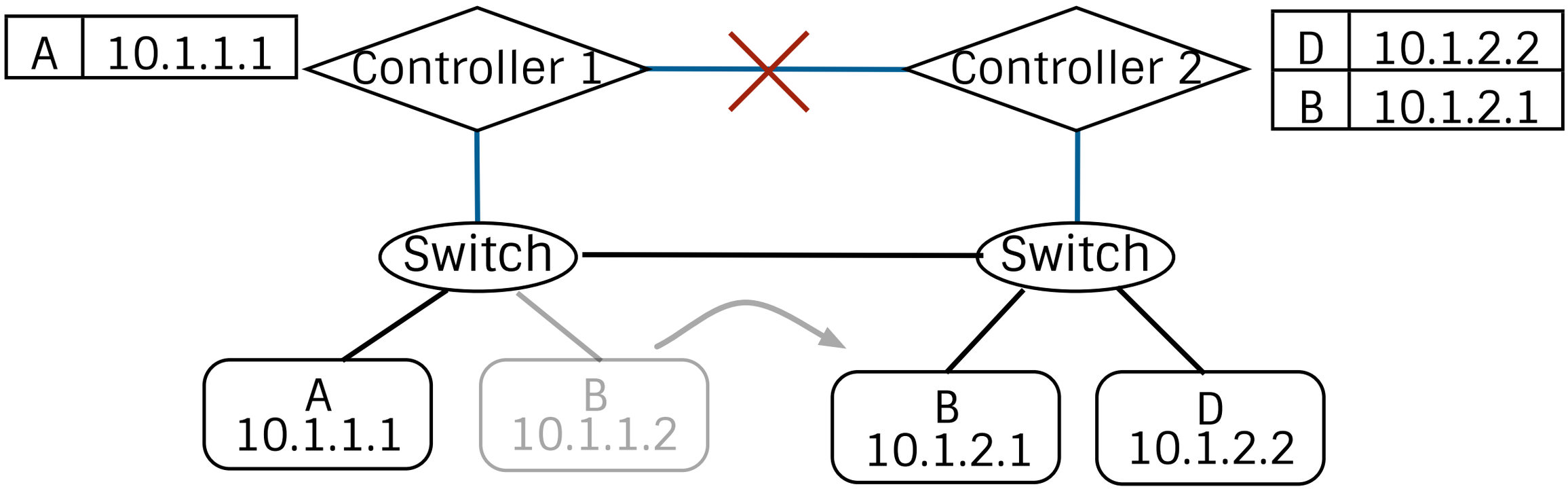
- Consider policy isolating A from B.
- A control network partition occurs.

Isolation Result



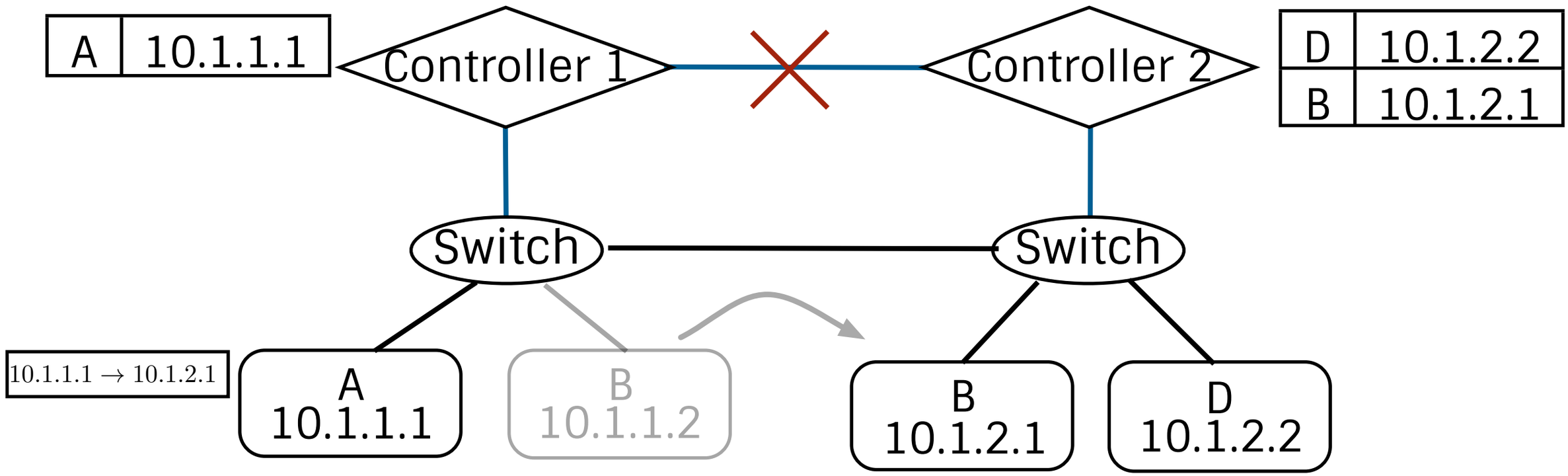
- Consider policy isolating A from B.
- A control network partition occurs.

Isolation Result



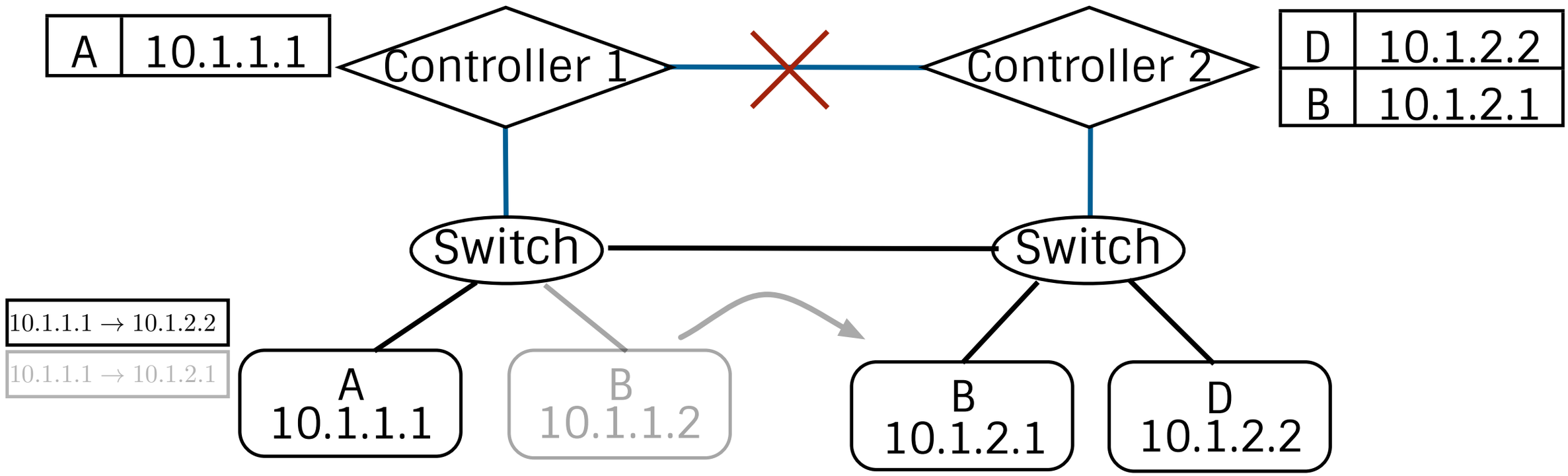
- Consider policy isolating A from B.
- A control network partition occurs.
- Only possible choices

Isolation Result



- Consider policy isolating A from B.
- A control network partition occurs.
- Only possible choices
  - Let all packets through (including from A to B) (~~Correctness~~)

Isolation Result



- Consider policy isolating A from B.
- A control network partition occurs.
- Only possible choices
  - Let all packets through (including from A to B) (~~Correctness~~)
  - Drop all packets (including from A to D) (~~Availability~~)

Workarounds for Isolation

- Identity-Address disconnect underlies isolation result

Workarounds for Isolation

- Identity-Address disconnect underlies isolation result
- Network can label packets with sender's identity

Workarounds for Isolation

- Identity-Address disconnect underlies isolation result
- Network can label packets with sender's identity
- Route based on identity instead of address

Workarounds not General

Edge Disjoint Traffic Engineering

- Two flows must traverse disjoint links
- Requires consistent topology across controllers

Can one provide correct isolation and availability in the presence of link failures?

Can one provide correct isolation and availability in the presence of link failures?

Not in general

In the Paper

- More policies and proofs
- More details on workarounds
- Other ways to model the network

CAP for Networks?

Choices for network architects

Correctness above all	Availability above all
Security Policies?	Traditional Routing?
ICING?	BGP
	NOX Routing

Backup Slides

Host Migration

- Our model assumes host migrations without controller involvement.
- In part this is because host migrations are surprisingly common
  - Soundararajan and Govil 2010: **6** migrations/day/VM
  - In a datacenter **~480,000** migrations/day
  - **5.5** migrations per second
- Controller involvement is too expensive in datacenters
  - NVP and Floodlight work in a similar manner
- In enterprises controller involvement complicated by mobility.