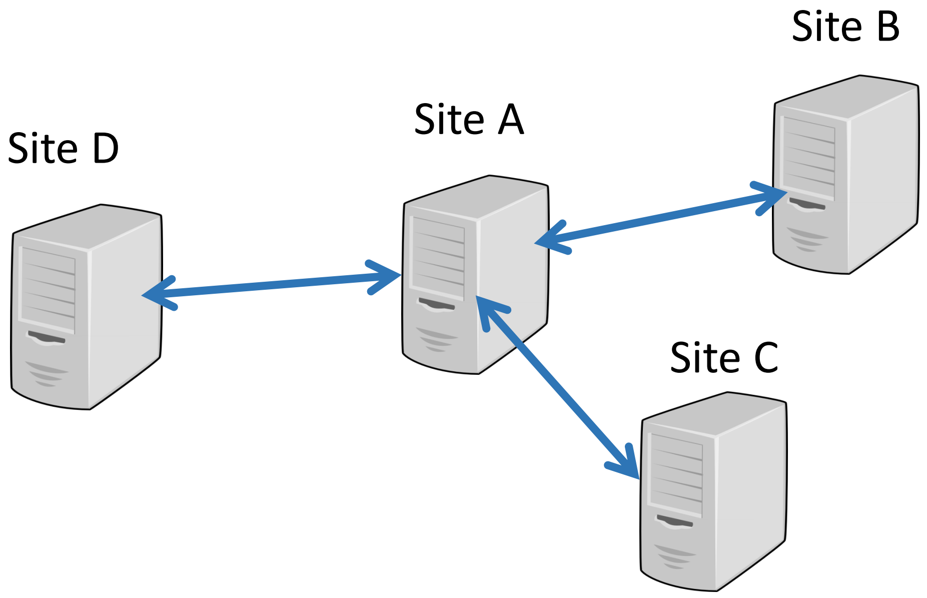
Samba KCC: Saying No to Full Mesh Replication

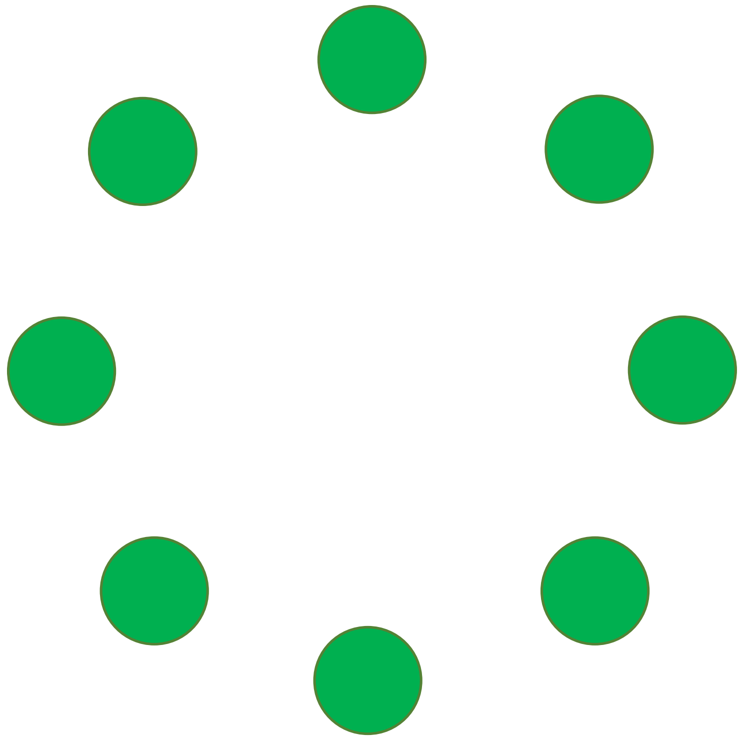
Garming Sam
Catalyst IT, Samba Team

What is the KCC?

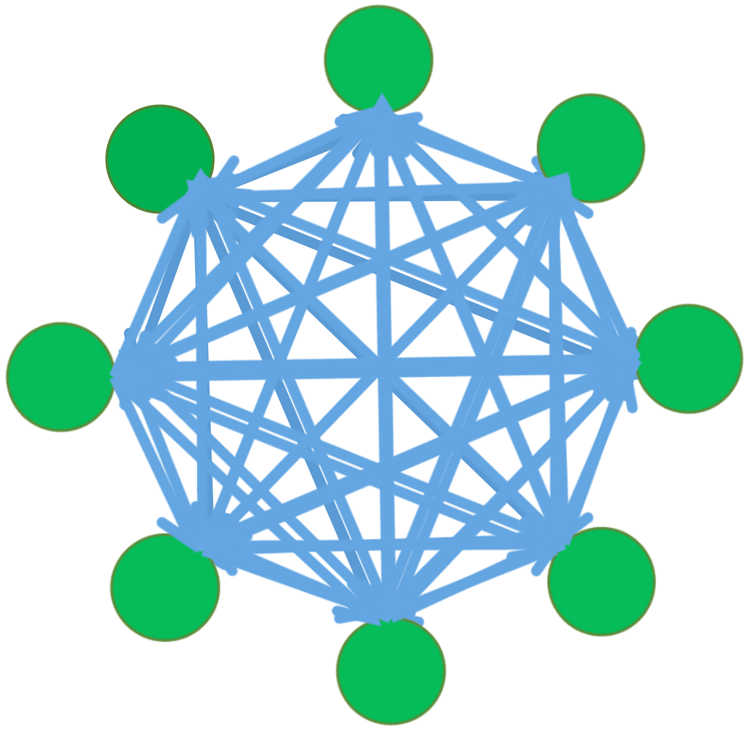
- Knowledge consistency checker
- Used to manage replication connections in AD
- Set of algorithms to produce efficient network topologies



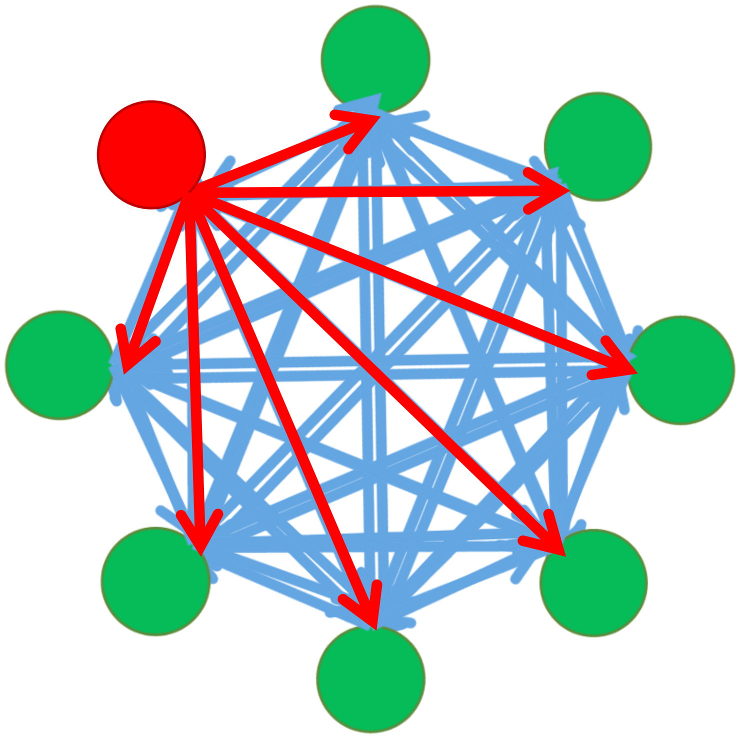
What is the KCC?



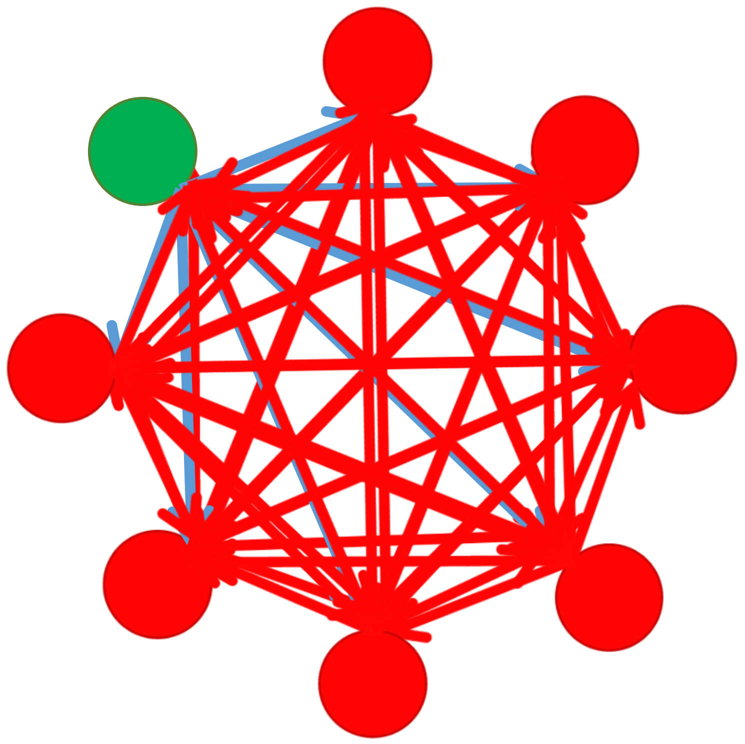
What is the KCC?



What is the KCC?



What is the KCC?



History of the KCC

- Original full-mesh C code
- Attempt at MS-ADTS algorithms in C
- Dave Craft (2011) on Python inter-site algorithms
- Late 2014—Early 2015 Douglas and myself
- Samba 4.3 introduced, Samba 4.5 set as default

Stages of the algorithm

- Intra-site algorithm
- Inter-site algorithm
- Removing unneeded connections
- Translate connections

Although the KCC creates ‘connection’ objects, they may not represent the underlying replication. They are only the implied connections given the current network topology.

Pre-requisites

- Transport – IP

```
dn: CN=IP,CN=Inter-Site Transports,CN=Sites,CN=Configuration,DC=example,DC=com
objectClass: interSiteTransport
```

- Sites – Default-First-Site

```
dn: CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=com
objectClass: site


dn: CN=NTDS Site Settings,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=example,DC=com
objectClass: nTDSSiteSettings
interSiteTopologyGenerator: CN=NTDS Settings,CN=DC,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=example,DC=com
```

Pre-requisites

- Site-Links – DEFAULTIPSITELINK

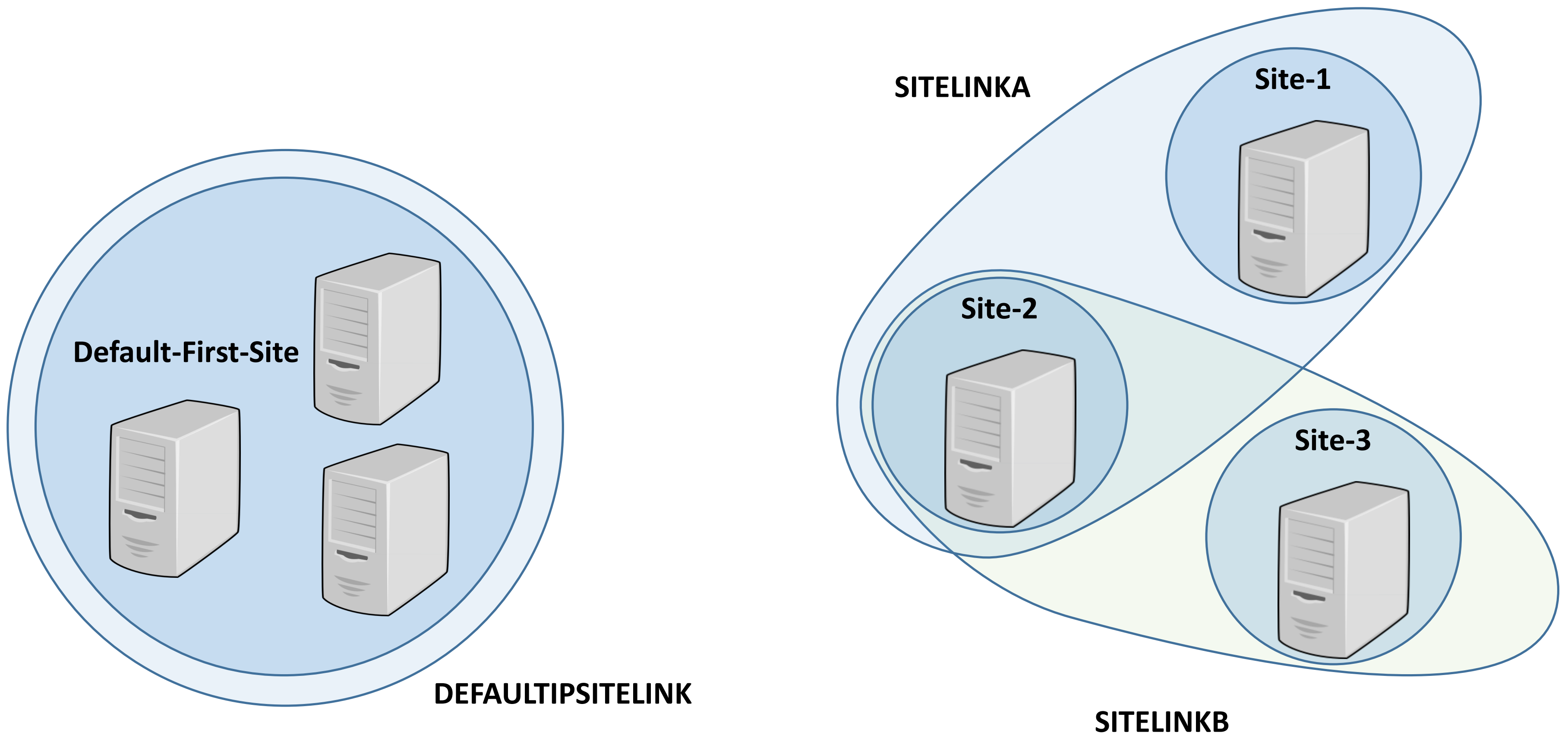
```
dn: CN=DEFAULTIPSITELINK,CN=IP,CN=Inter-Site
Transports,CN=Sites,CN=Configuration,DC=example,DC=com
objectClass: siteLink
cost: 100
siteList: CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=com
```

Site-links define the allowable connections between sites

Site-links represent (hub-like) physical connectivity

Site-links needs to collectively span your entire network

Pre-requisites - Scenarios

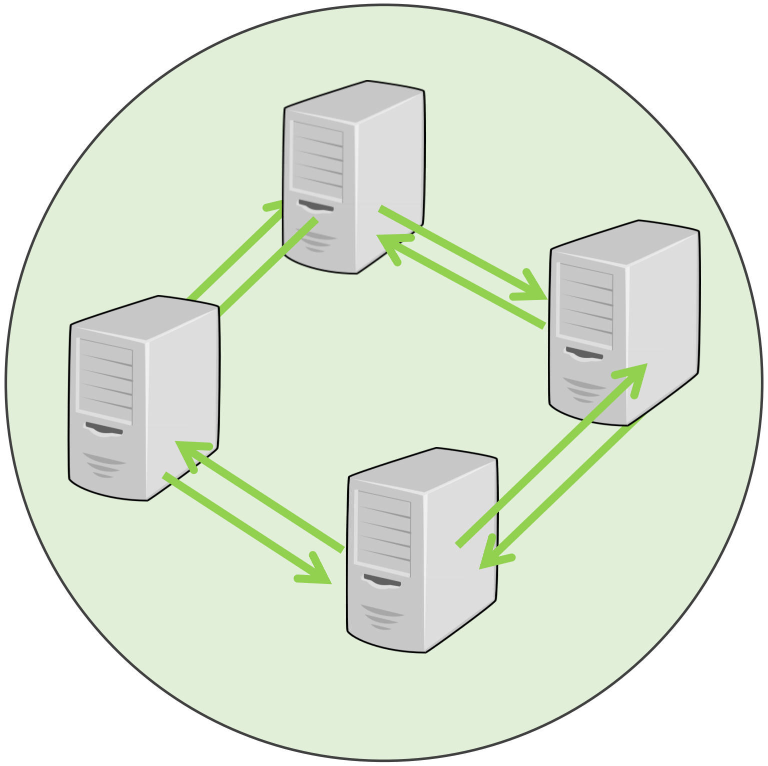


Intra-site algorithm

- Creates connections within a single site
- With just a single server, no work is necessary
- Ring topology, with a few extra connections (n > 7)

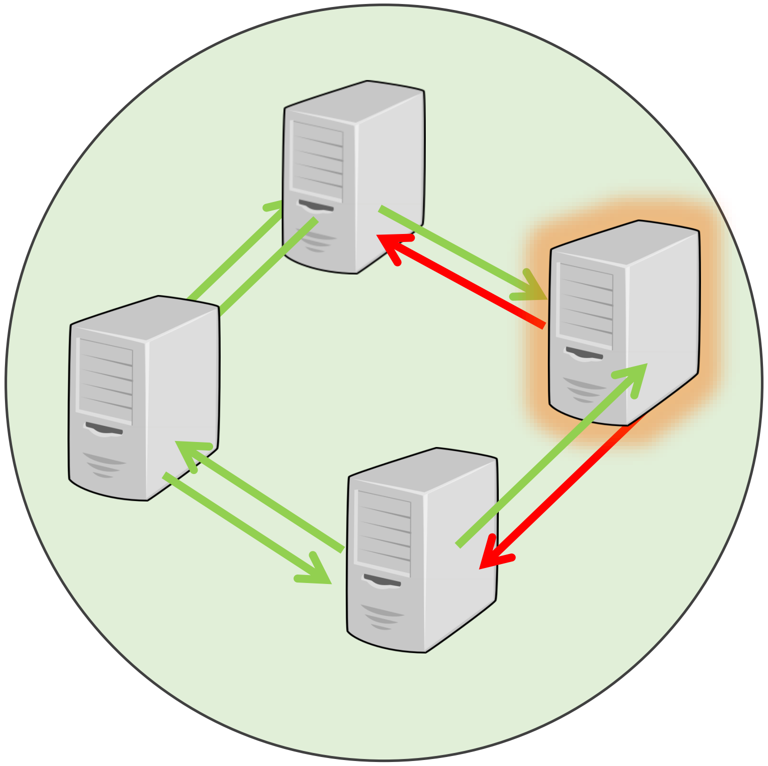
Intra-site algorithm

- Ring topology, with a few extra connections



Intra-site algorithm

- Every DC in the site has a sorted list of site DCs



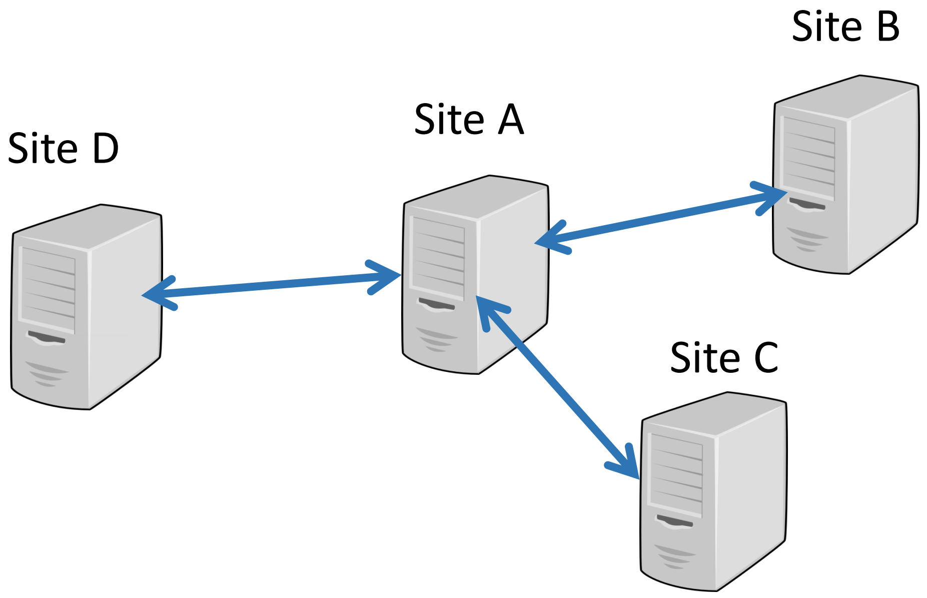
Intra-site algorithm

- Compared to the old KCC, there are fewer connections
- The algorithm is quite reliable, adding additional connections
- Information propagates in a more controlled manner

In a single-site use-case, with not that many DCs, behaviour should be quite similar to the old code.

Inter-site algorithm

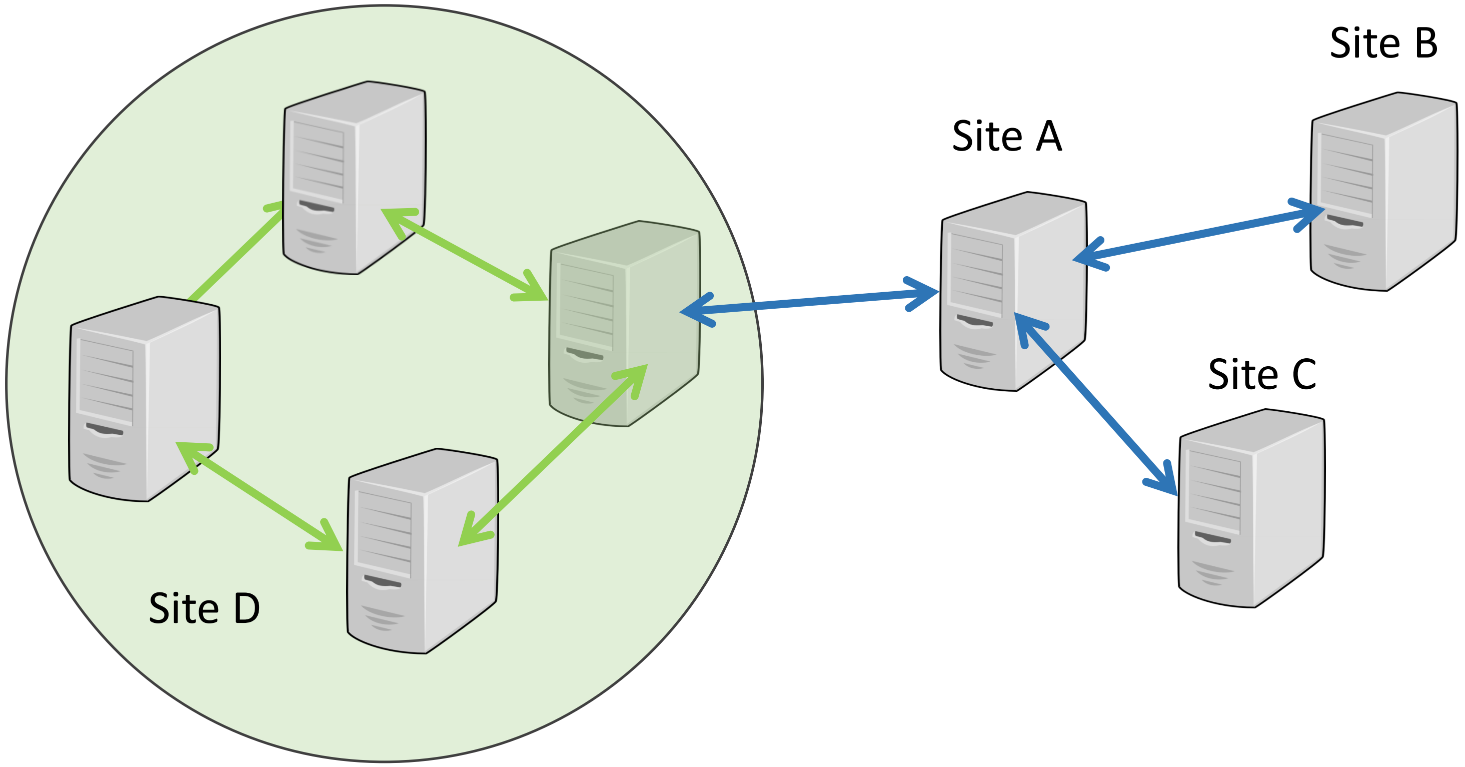
- Each site elects an inter-site topology generator (ISTG)
- Re-election attempts to occur if the ISTG is not responding
- Attribute: interSiteTopologyFailover



Inter-site algorithm

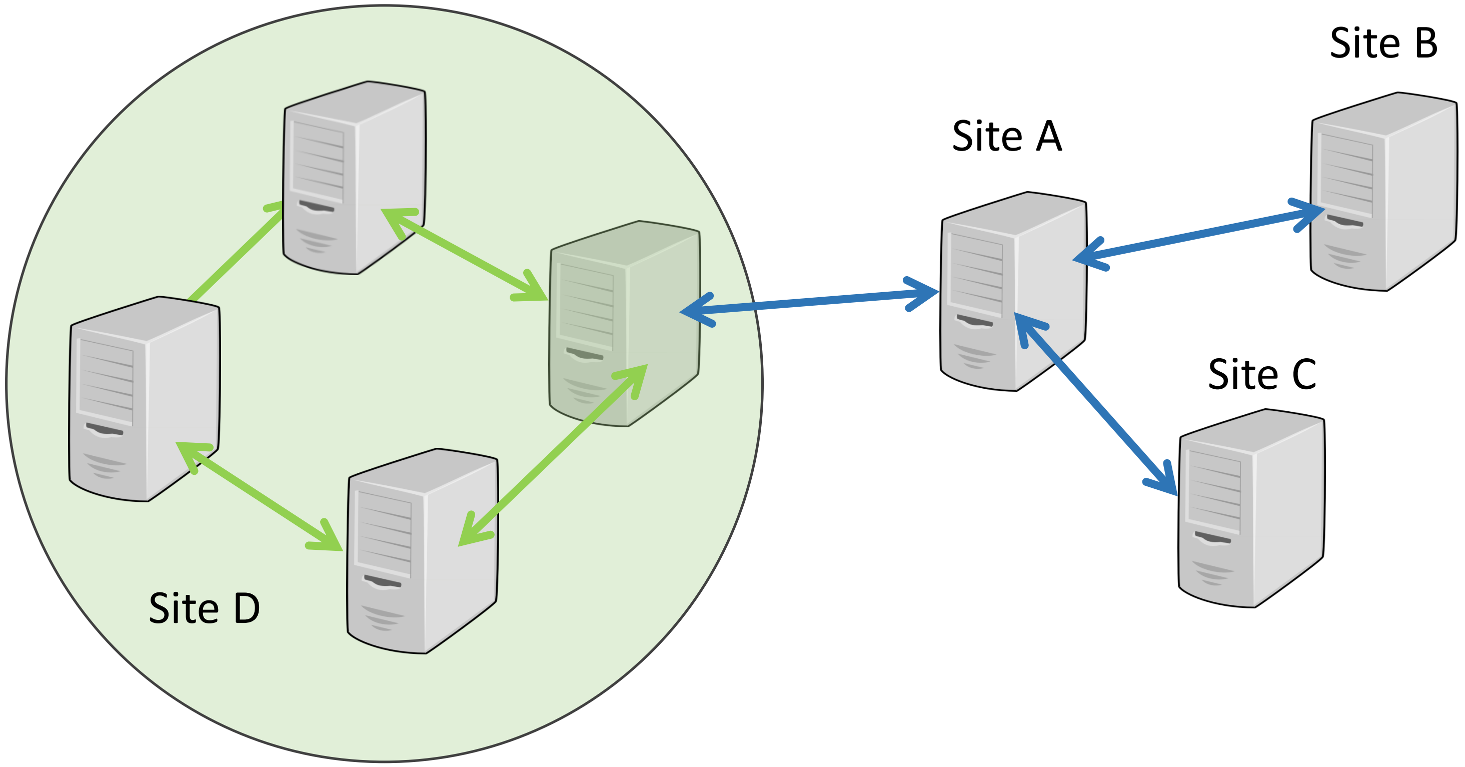
- Stable answer across entire DC network
- One DC per site managing inter-site connections
- Needs to be as fault tolerant as possible
- Must produce topology optimizing cost and schedules

Inter-site algorithm



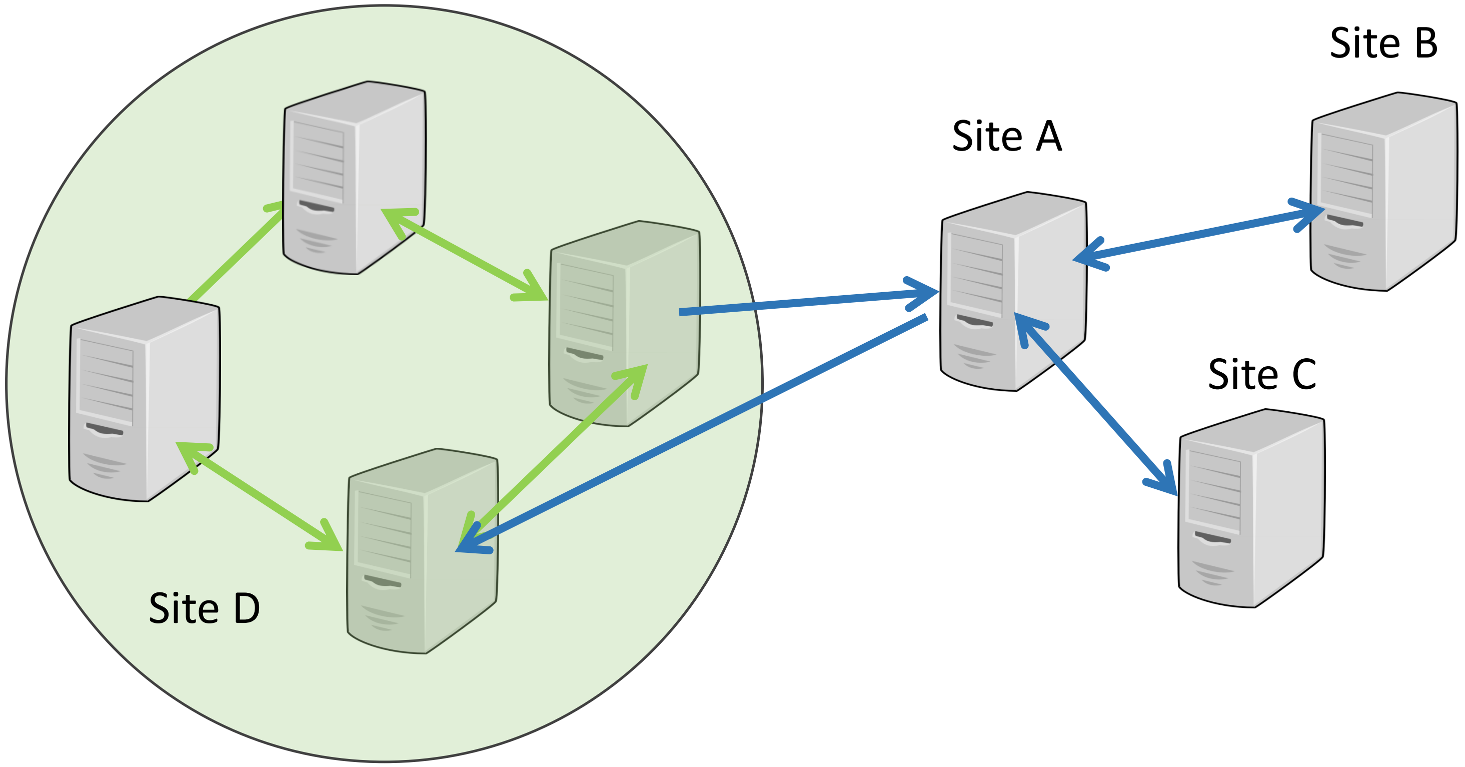
Bridgehead servers are the end-point connections between sites.

Inter-site algorithm



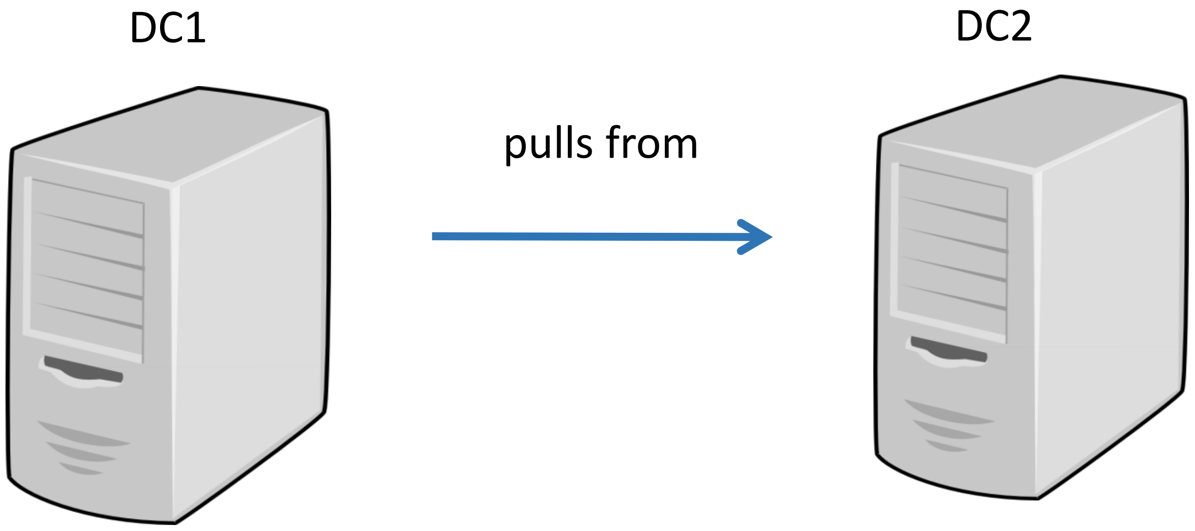
Being a bridgehead does not imply being an ISTG.

Inter-site algorithm



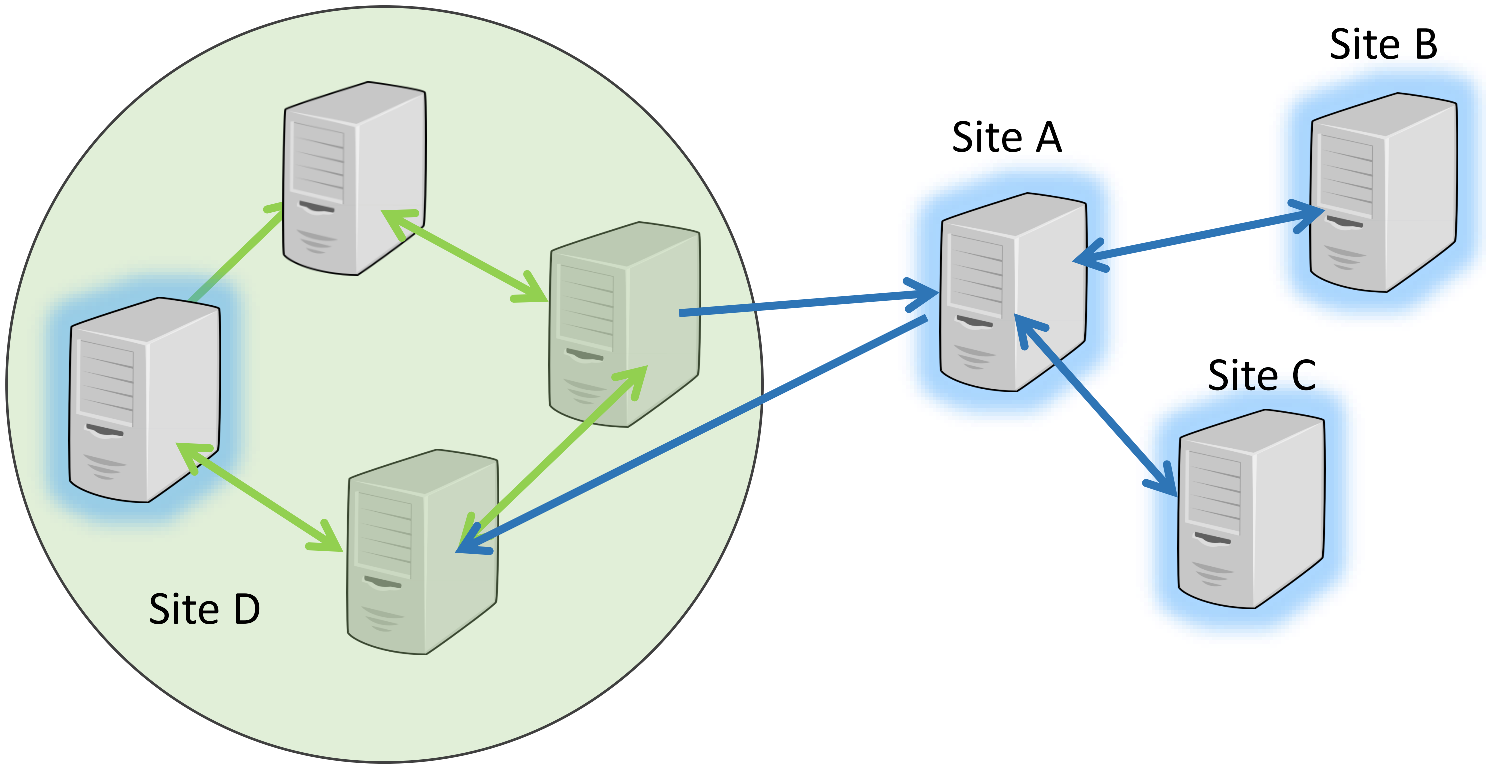
There is not necessarily a single bridgehead server.

Inter-site algorithm



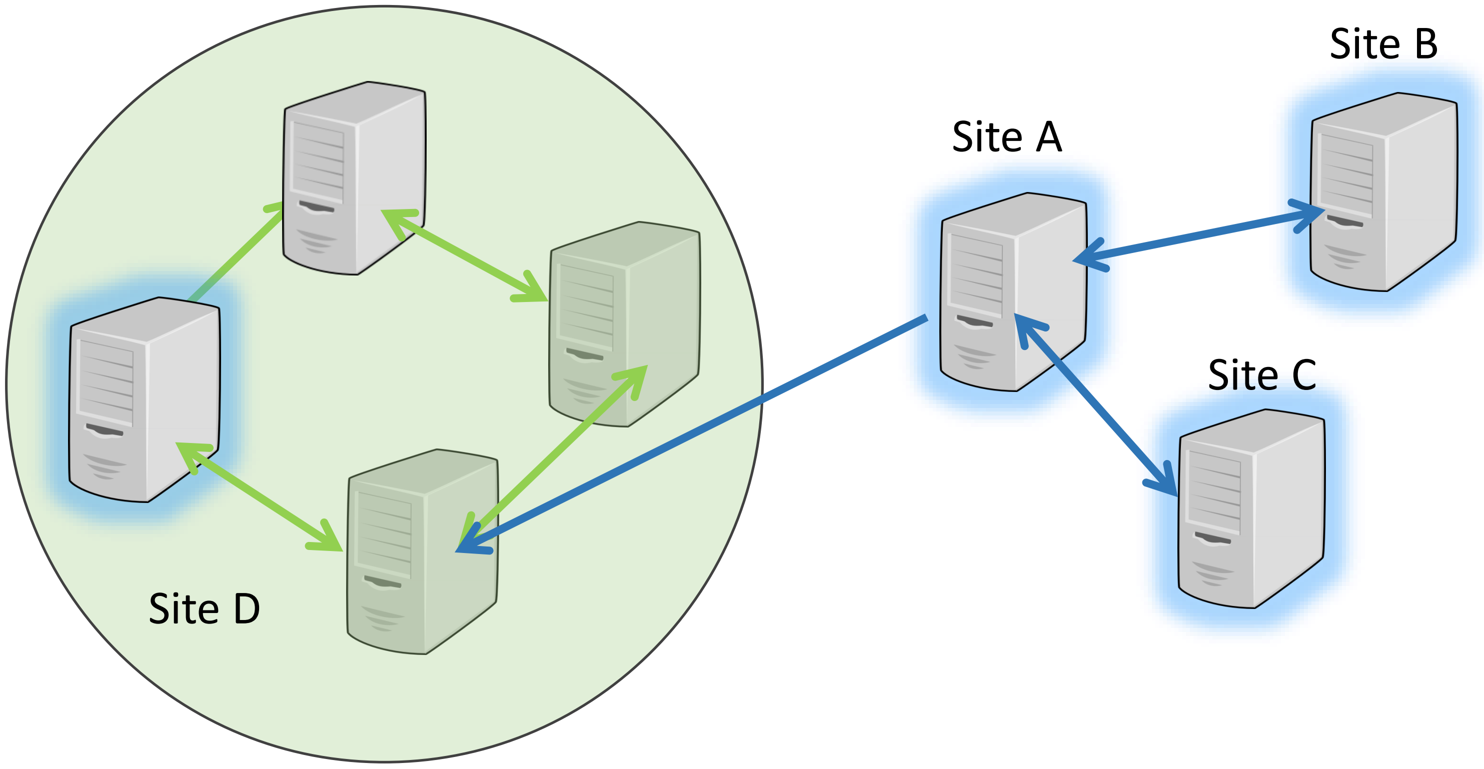
There is only pull replication.
Bi-directional replication must be done with two distinct connections.

Inter-site algorithm

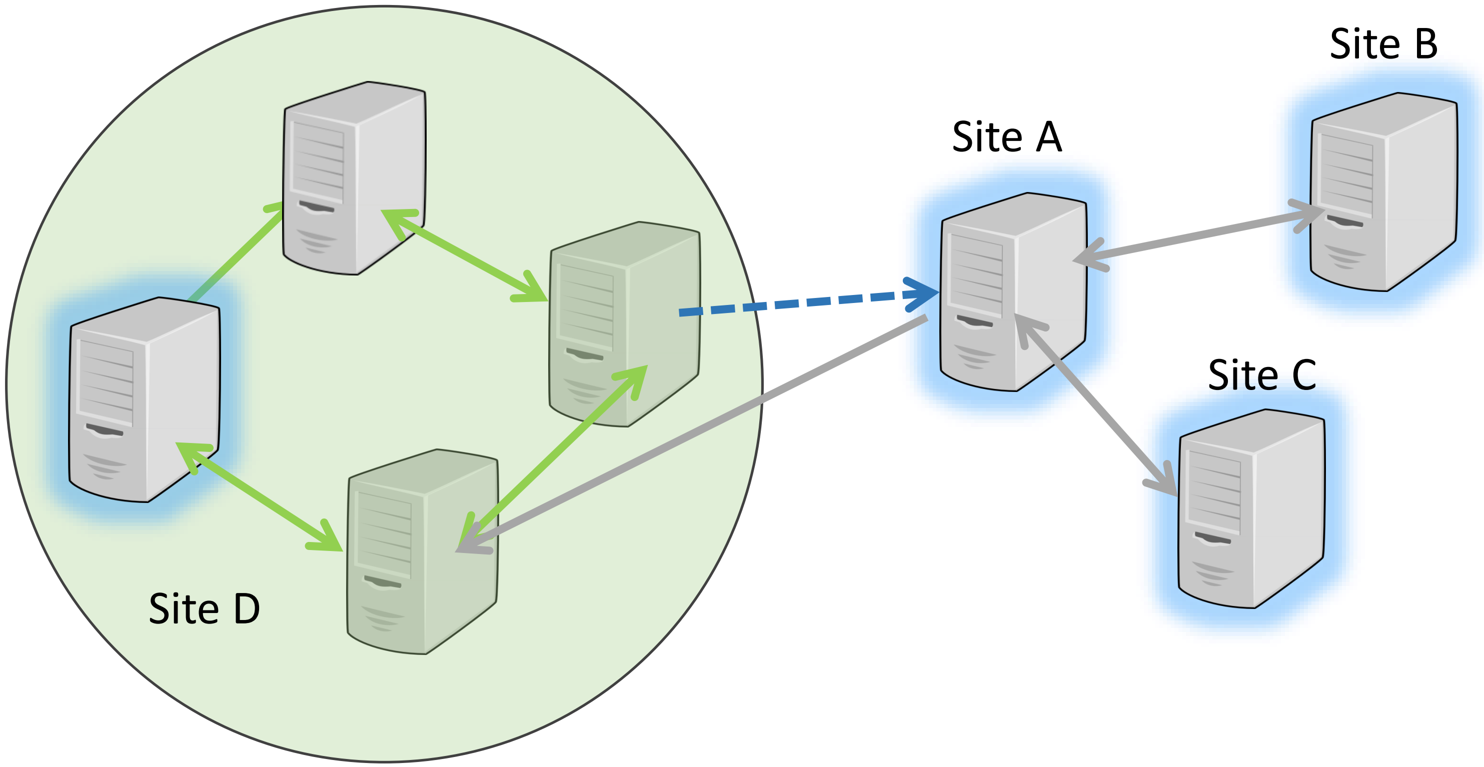


The inter-site algorithm only runs on the ISTG.

Inter-site algorithm

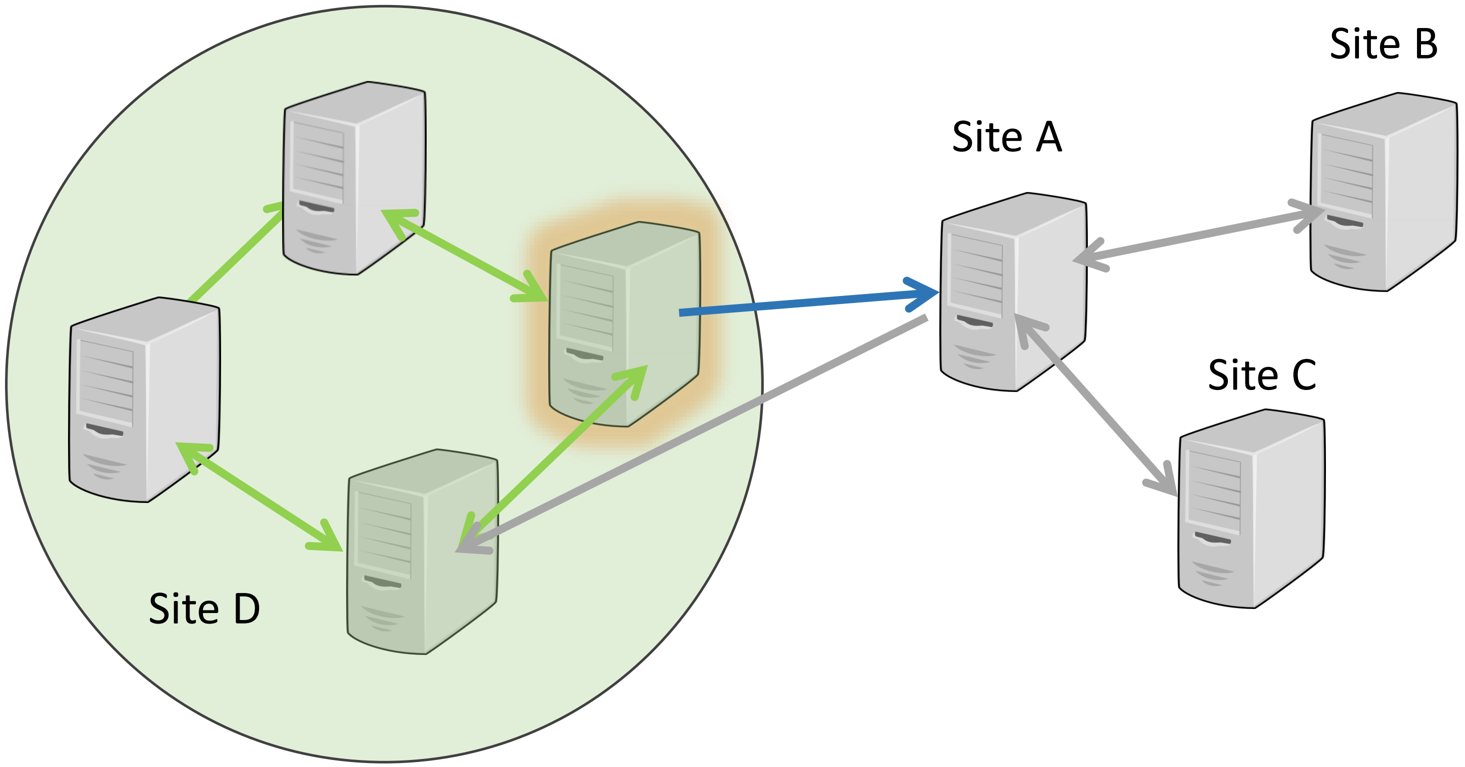


Inter-site algorithm



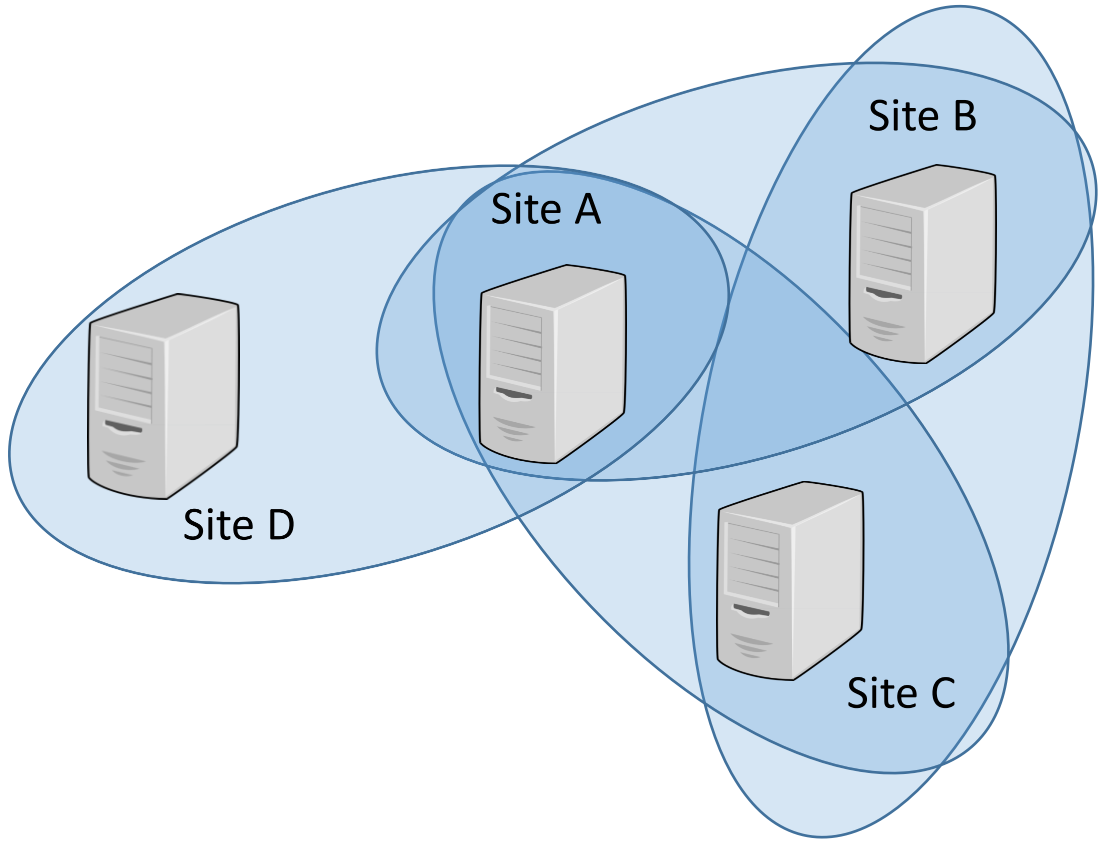
A new connection will be created in the database pointing to a randomly chosen bridgehead in Site A. Intra-site replication will propagate this to the necessary bridgehead in Site D.

Inter-site algorithm

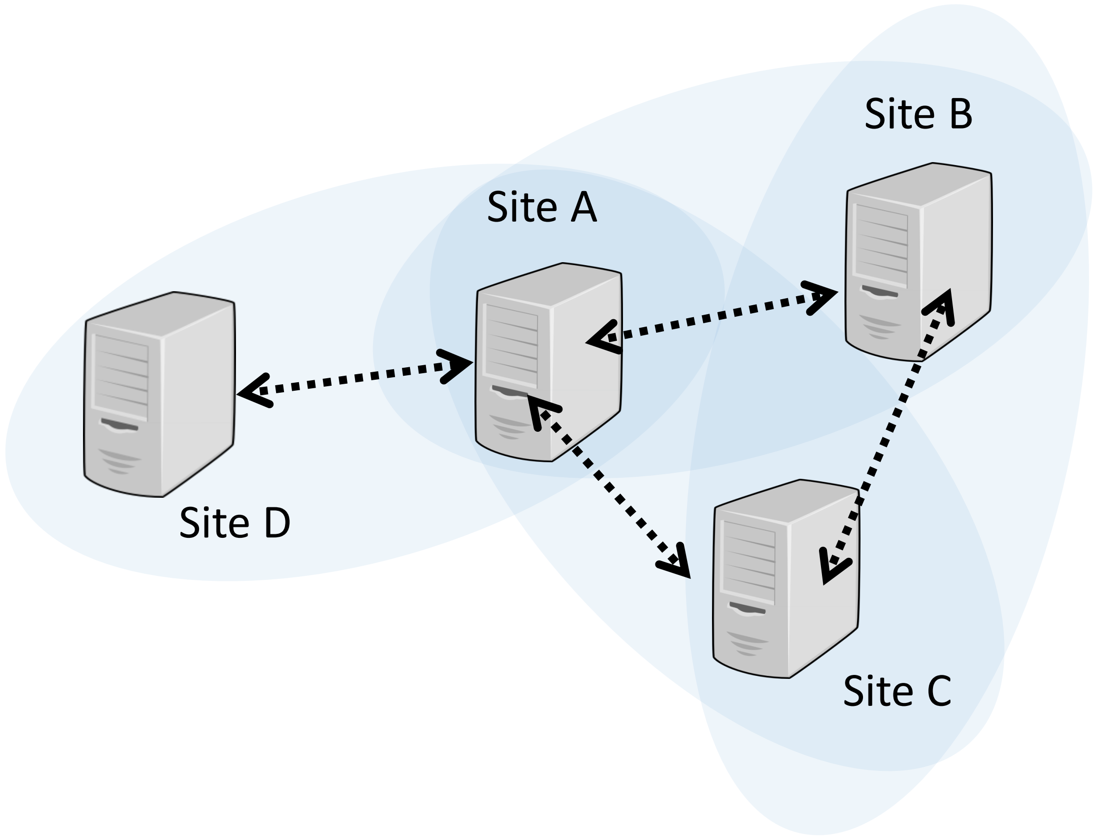


The incoming bridgehead runs the KCC and notices the new connection.
It has no idea why it connects to the DC, that’s the role of the ISTG.

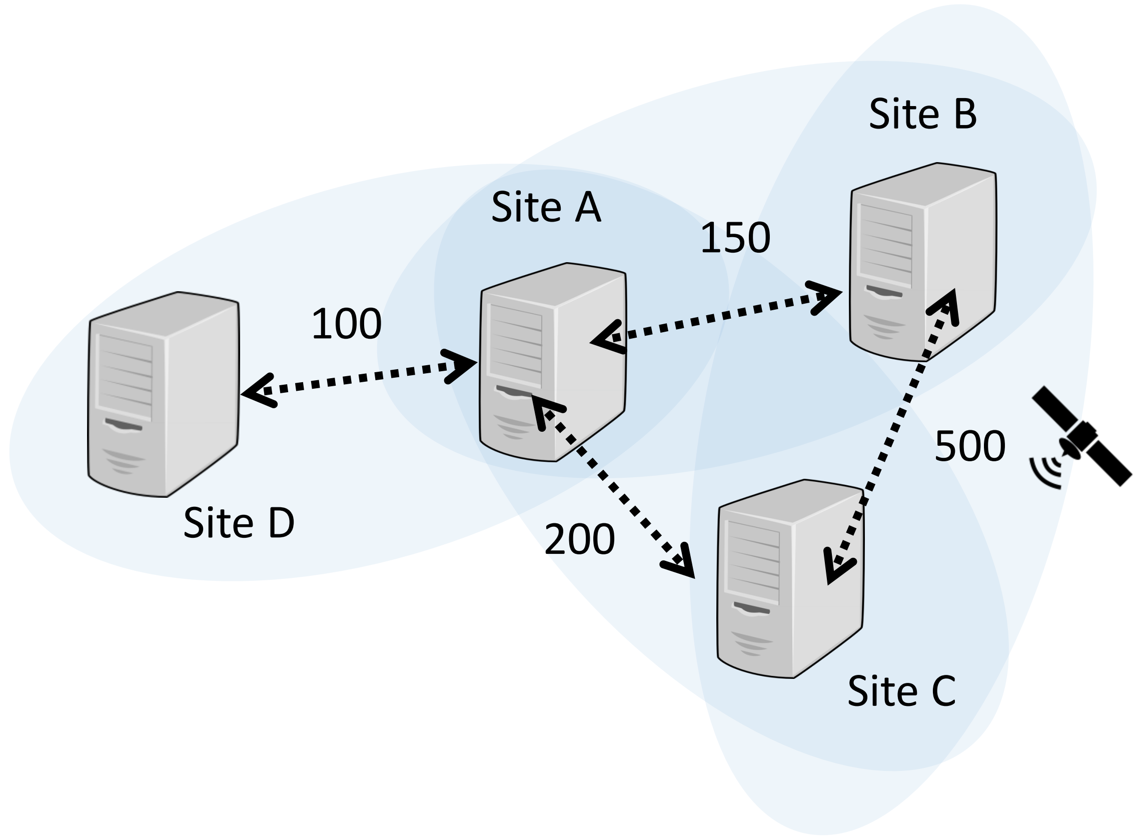
Inter-site algorithm



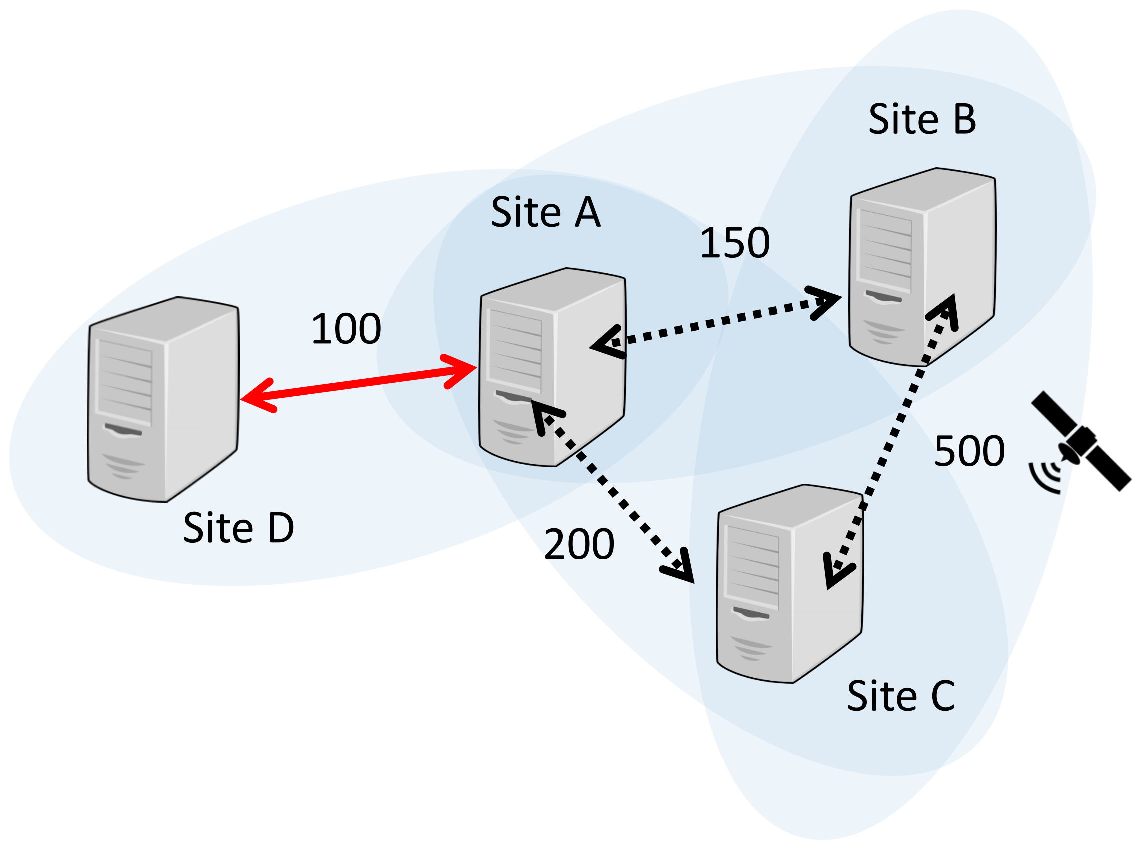
Inter-site algorithm



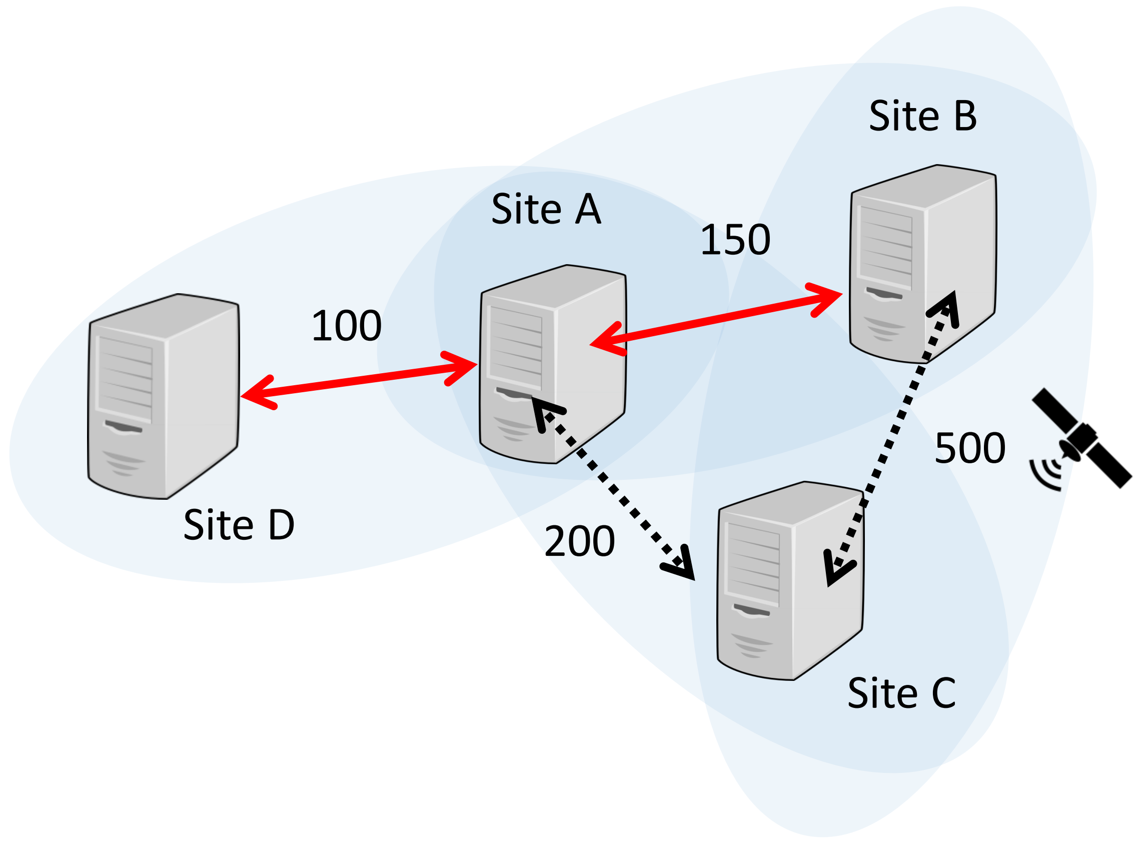
Inter-site algorithm



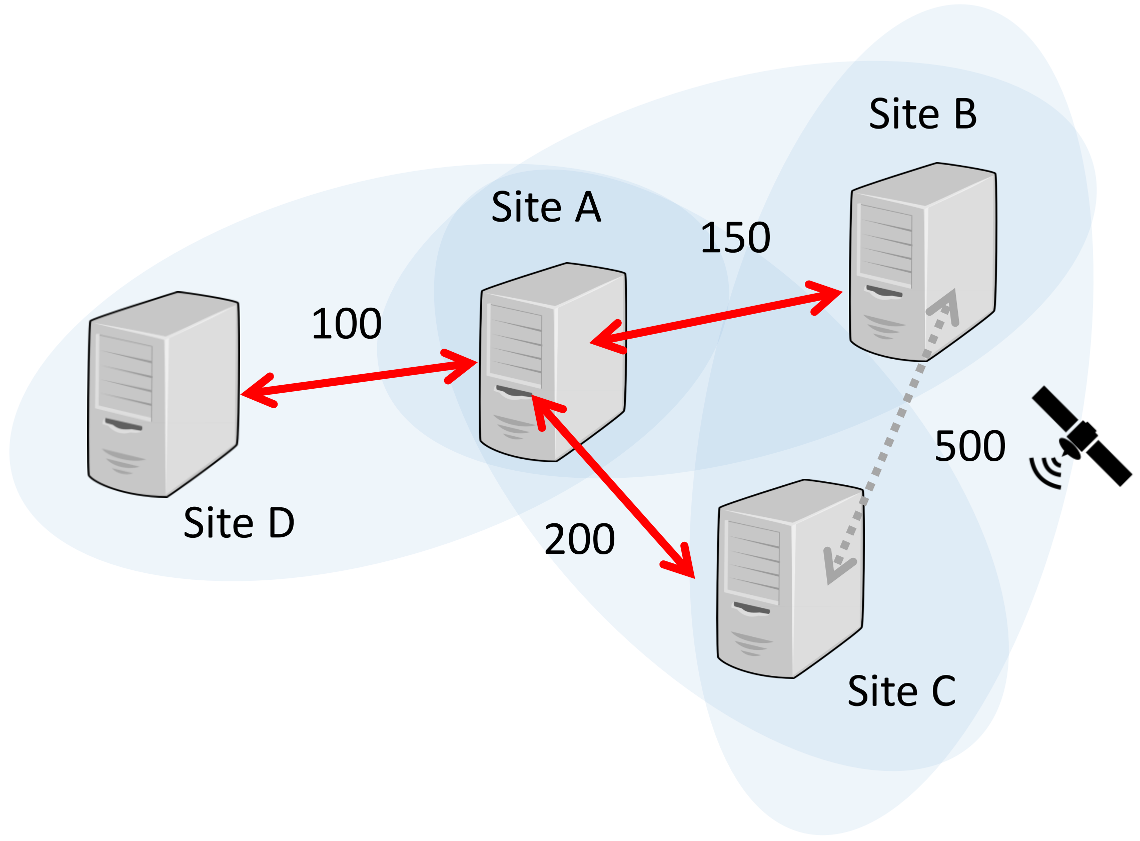
Inter-site algorithm



Inter-site algorithm

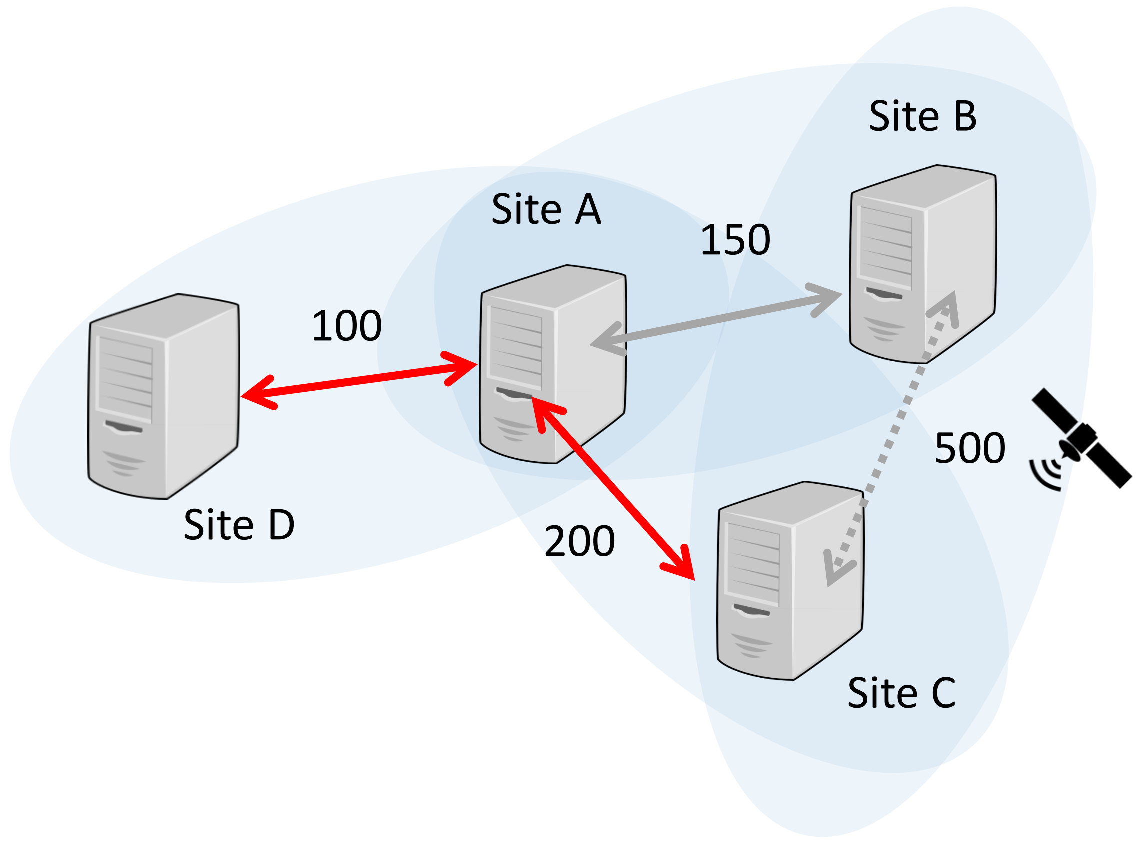


Inter-site algorithm

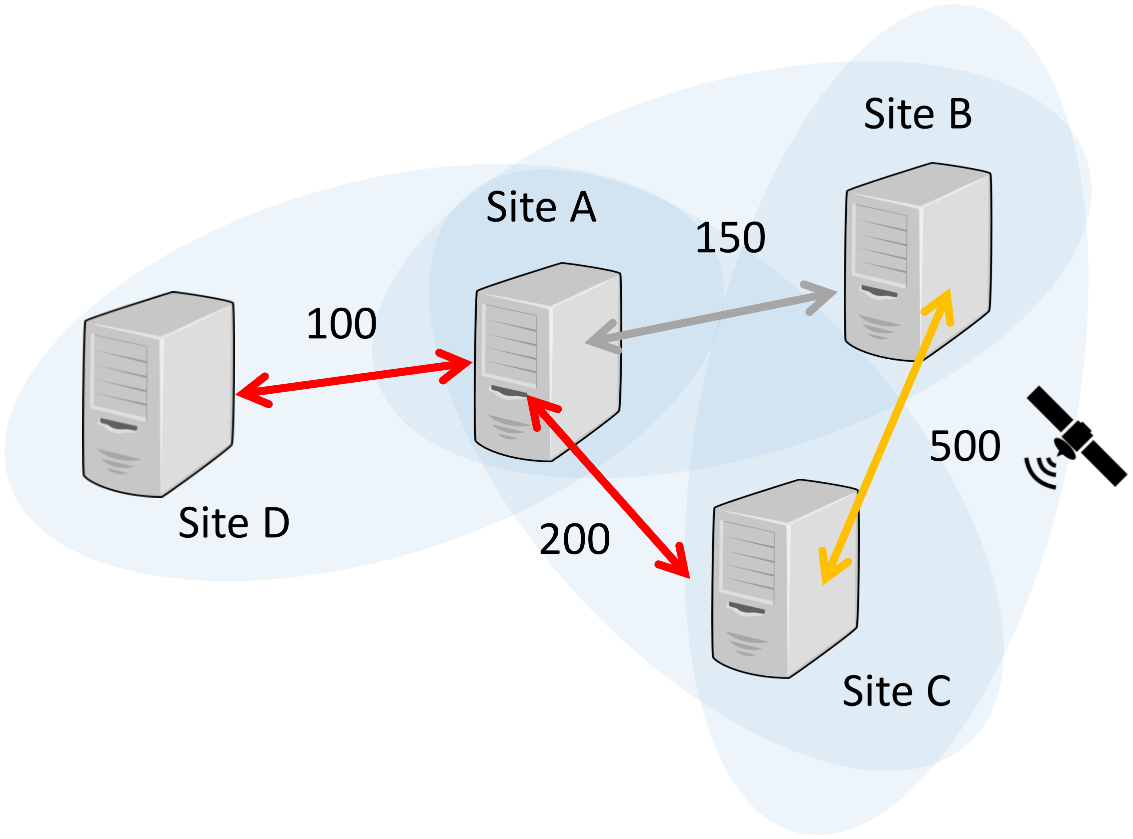


Total cost: 450

Inter-site algorithm - Failover



Inter-site algorithm - Failover



On network connectivity failure, the KCC attempts to overlay a second redundant topology.

Remove unneeded connections

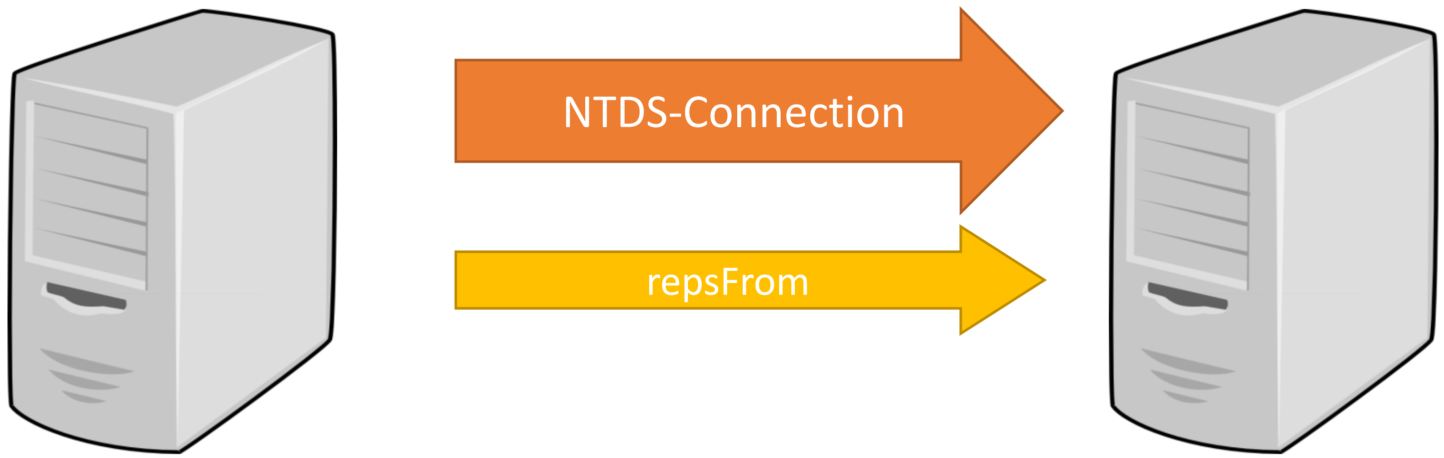
Removes connections:

- which are duplicated (removing the oldest)
- which exceed redundancy limit (intra-site)

Area still needs some work, however, removing too aggressively may cause connectivity issues.

Translate connections

- Of the connections the KCC deems necessary, they are translated into repsFrom (non-replicated attributes)



Two independent tasks running

- KCC running periodically
  - Creating NTDS Connection objects (ISTG or intra-site)
  - Translating NTDS Connections to repsFrom

- DREPL server
  - Reading repsFrom and pulling from the target
  - Reading repsTo and telling target to pull

This means it can take some time to propagate, particularly repsTo which are deferred created by replication on repsFrom.

Translate connections

- Of the connections the KCC deems necessary, they are translated into repsFrom (non-replicated attributes)

- repsFrom flags are set (particularly important for RODC)
- Stale repsFrom SHOULD be deleted
- Stale repsTo SHOULD be deleted

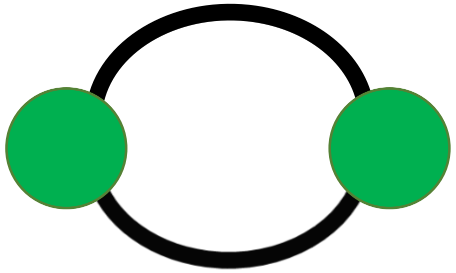
The end result

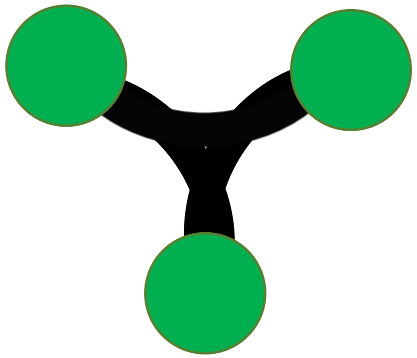
- Single path from any site to any site (property of a tree)
- Changes should not bounce around significantly
- Significantly reduced replication traffic
- Ability to customize who should talk to who

- Small networks (n <= 4) should have no visible effect
- Larger networks with varying connectivity shows huge effect

Challenges

- Verbose documentation
- Site-Link: ‘Multi-edge’, hyper-edge?
- White, red, black vertices?





More challenges

- Logical inconsistencies, ambiguities and omissions
- Pseudo-code vs textual description
- Easy to debug your own bugs
    - Testing?
    - --dot-file-dir
    - --readonly --exportldif, --importldif

Incomplete features

- Trusted domains and global catalog replication
- RODC self-management
- Site-Link-Bridge Topologies
- Respecting schedules and other AD attributes
    - Preferred bridgehead servers
    - Replication frequency?

Incomplete features

- Failed connection and failed DC failover
- Better stale connection clean-up
  - MS-DS-Replicates-NC-Reason
  - Use normal replication to propagate failure info
- Better debugging and failure information
- Better defaults for modern networks

Alternative topology strategies

- What is the best topology for various networks?

- Ring algorithm from intra-site for inter-site
- Minimum cost spanning tree plus additional connections
- Fully connected bridge-head servers

Alternative topology strategies

- What is the best topology for various networks?

- Ring algorithm from intra-site for inter-site
- Minimum cost spanning tree plus additional connections
- Fully connected bridge-head servers

Questions?

Email: garming@catalyst.net.nz