What does the KCC do?

Douglas Bagnall

douglas.bagnall@catalyst.net.nz   dbagnall@samba.org
The Knowledge Consistency Checker creates a replication graph linking Domain Controllers.

- barely changed since Windows 2000.
- we have samba_kcc since Samba 4.3 (2015).
Minimal connected graph
Minimal connected graph with shortcuts
double linked graph
two sites
KCC does this
intersite graph is a *tree*
Site Links

Nantes can only see Inverness
Inverness can see
Nantes and Suva
Suva, Wellington, and Lima have free reign to form a tree.
Site links can have a cost

“I exist but pretend I don’t”
Intra-site links

double-linked ring in GUID order
With more DCs, add random cross-links
Aiming for a probable maximum of 3 hops
*Intersite* is based on Dijkstra’s Algorithm and Kruskal’s Algorithm

*Intrasite* is the double ring + increasing random links

Some DCs are nominated as intersite bridgeheads

Some DCs are InterSite Topology Generators (ISTGs)
Samba’s KCC

The old KCC made fully connected graphs written in C

samba_kcc is written in Python, run as a subprocess

started in 2011 by Dave Craft “finished” in 2015 by Garming Sam and me.

follows [MS-ADTS] 6.2.*
KCC mechanics

The KCC *thinks* in NTDSConnections, produces RepsTo and RepsFrom links

Each DC:
- periodically pulls from DCs in its RepsFrom list
- asks its RepsTo partners to pull when necessary

only the RepsTo and RepsFrom really matter.
KCC mismatch

In a mixed domain, if Samba and Windows have different intersite trees, the network could split.
Samba KCC problems

KCC is
- complex and poorly specified, showing its age
- practically untestable without a large network and iterated runs

samba_kcc
  developed in disconnected spurts
  by novices
  over many versions of Python
  using buggy Python bindings
working around 2011 Python bugs

```python
# WARNING:
#
# There is a very subtle bug here with python
# and our NDR code. If you assign directly to
# a NDR produced struct (e.g. t_repsFrom.ctr.other_info)
# then a proper python GC reference count is not
# maintained.
#
# To work around this we maintain an internal
# reference to "dns_name(x)" and "other_info" elements
# of repsFromToBlob. This internal reference
# is hidden within this class but it is why you
# see statements like this below:
#
# self._dict_['nrd_blob'].ctr.other_info = 
#     self._dict_['other_info'] = drsblobs.repsFromTo1OtherInfo()
#
# That would appear to be a redundant assignment but
# it is necessary to hold a proper python GC reference
# count.

if ndr_blob is None:
    self._dict_['nrd_blob'] = drsblobs.repsFromToBlob()
    self._dict_['nrd_blob'].version = 0x1
    self._dict_['dns_name1'] = None
    self._dict_['dns_name2'] = None

    self._dict_['nrd_blob'].ctr.other_info = 
        self._dict_['other_info'] = drsblobs.repsFromTo1OtherInfo()
else:
    self._dict_['nrd_blob'] = ndr_blob
    self._dict_['other_info'] = ndr_blob.ctr.other_info

    if ndr_blob.version == 0x1:
        self._dict_['dns_name1'] = ndr_blob.ctr.other_info.dns_name
        self._dict_['dns_name2'] = None
    else:
        self._dict_['dns_name1'] = ndr_blob.ctr.other_info.dns_name1
        self._dict_['dns_name2'] = ndr_blob.ctr.other_info.dns_name2

def str (self):
```
Things samba_kcc does not do

- correctly check the liveness of links
- handle SMTP transport

Things samba_kcc does do that are absolutely pointless

- calculate replication schedules that are unused
- calculate all kinds of unused flags

  NTDSCONN_OPT_TWOWAY_SYNC
  DRSUAPI_DRS_DISABLE_AUTO_SYNC
  DRSUAPI_DRS_DISABLE_Periodic_SYNC
  NTDSCONN_OPT_DISABLE_INTERSITE_COMPRESSION
  DRSUAPI_DRS_USE_COMPRESSION
  NTDSSettings_OPT_IS_TOPL_DETECT_STALE_DISABLED
  NTDSCONN_OPT_OVERRIDE_NOTIFY_DEFAULT
  NTDSSiteLink_OPT_TWOWAY_SYNC
  DS_NTDSSettings_OPT_IS_RAND_BH_SELECTION_DISABLED
Unexpected things samba_kcc does

- lots of multi-coloured debug messages, not controlled by -d
- lots of self-testing (graph verification)
- writes .dot graph files (like samba-tool visualize reps --dot)
- loads fake domains via LDIF for testing
- leaves extra repsTo and repsFroms lying around
samba_kcc

- could be 50% smaller with no loss
- or 90% smaller and work better

it could carefully diverge from Windows topology, in interoperable ways (we know accidental divergence works OK).
More reassuring repsFrom graph