Distributed Traffic Monitor Health Monitoring

September 22, 2021

COMCAST

ABOUT ME

- Comcast 15 years, CDN team 6 ½ years
- Lead the CDN Application Engineering/Ops team at Comcast
- Member of ATC PMC and ASF Member
- Have developed in all ATC components…..but not in a while
- Still contribute a lot of troubleshooting, debug, and SRE
- Enjoy coaching my son’s baseball team and my daughter’s softball team

COMCAST

A Golang application that implements the CDN health protocol. Every cache in the CDN is checked via HTTP for vital stats, and based on these stats, caches are declared healthy or unhealthy. This information is then used by Traffic Router to make routing decisions.

CORE FUNCTIONALITY

Polls all REPORTED and ADMIN_DOWN caches

Basic problem detection

- TCP (connection, reset, timeouts)
- Application (port, astats content)

Configurable thresholds

- System (NIC throughput, load average)
- Application (remap stats)

Health (2s) and Stat (6s) polling intervals

Optimistic Health Protocol

- Raw health state polled from all peers
- All peers must agree on negative state

Proxies configuration to Traffic Router

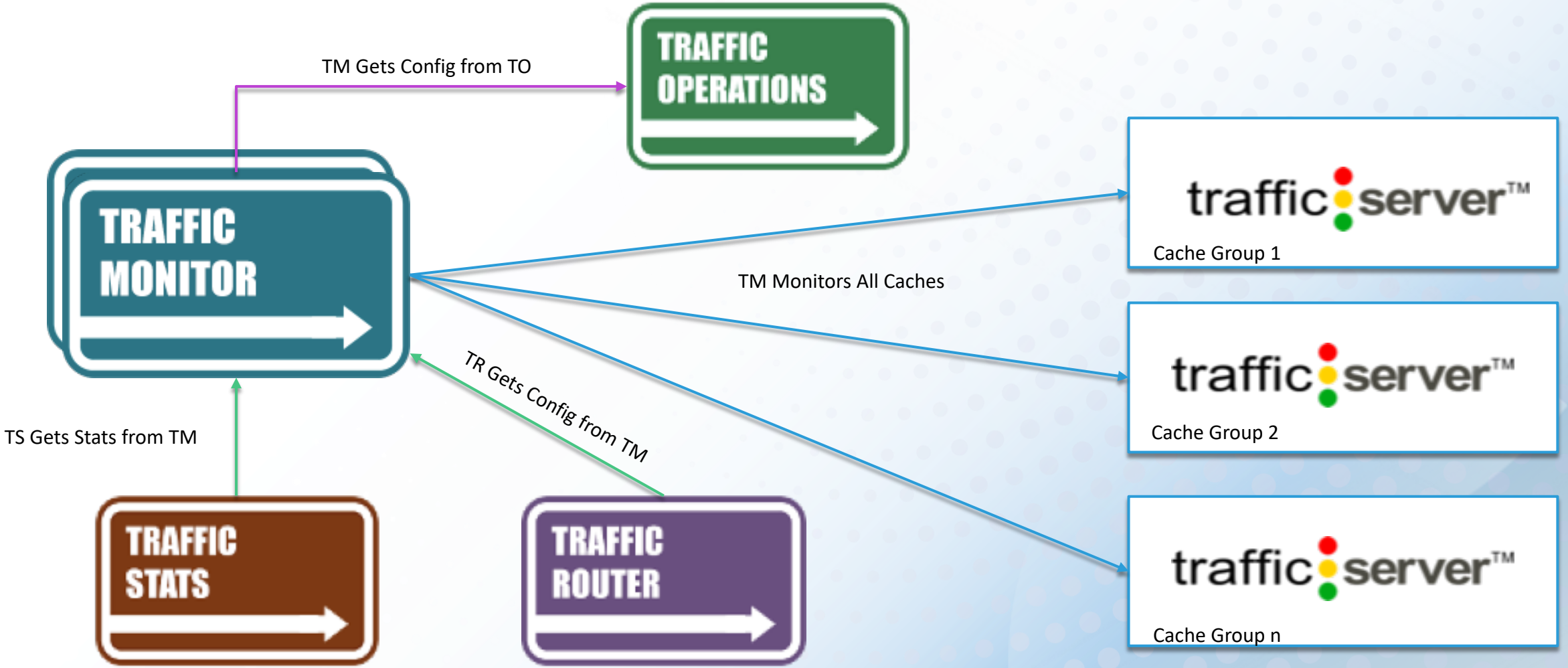
- CrConfig (TR app config)
- Cache health config

Exposes data in JSON format via API

Provides basic UI to see events, stats, APIs, etc

COMCAST

TODAY'S TRAFFIC MONITOR



COMCAST

MOTIVATIONS

Currently Traffic Monitor can only monitor an entire CDN.

Traffic Monitor is limited in several areas:

- TM must poll every cache in a CDN
- Traffic Monitors are deployed in central locations where they can get to all caches
- Traffic Monitor can only scale vertically, not horizontally
- As a CDN grows, Traffic Monitor requires more HW resources to efficiently poll all caches

In order to solve our impending scaling issues as well as improve our ability to make better and faster health decisions, Traffic Monitor needs to run in a distributed fashion instead of an “all or nothing” fashion.

KEY HIGH LEVEL REQUIREMENTS

Traffic Monitor MUST be capable of being configured to monitor all OR a portion of a CDN

Traffic Monitor MUST provide an API to get the health status of ALL caches in the CDN – TR can get status of all cache from any TM

Traffic Monitor MUST provide an API to get statistics (from e.g. astats data) generated by ALL caches in the CDN. This does not include any statistics generated by external monitoring systems.

Traffic Monitor MUST provide an API to get the status of caches it monitors

Traffic Monitor MUST provide the ability to have more than 1 Traffic Monitor monitor the same cache and come to consensus on the health of the cache.

Traffic Monitor MUST ensure all caches are monitored upon failure of any TM server(s) or physical location. (i.e. no SPoF of TMs for polling/aggregation).

DESIGN DECISIONS

Traffic Monitors will be organized into one or more Traffic Monitor groups

Traffic Monitors will determine which cache groups to poll based on the number of cache groups divided by the number of TM groups

TM will determine which cache groups to poll based on geo-location provided on the cache group object; profile/param will provide override functionality

Traffic Monitor will gain at least two new config options

- `distributed_polling_enabled` (default: false) - when set to true, TM will run in distributed mode. When set to false, TM will run in its legacy, normal mode.
- `stat_polling_disabled` (default: false) - when set to true, TM will not do stat polling for caches. When set to false, TM will do stat polling for caches -- legacy, normal behavior. Must be set to true if distributed_polling_enabled is also set to true in Phase I

DESIGN DECISIONS

All TMs participating in the health protocol should be in the same mode (distributed or legacy)

TMs will peer with all other TMs within their TM group as well as at least 1 TM from all other TM groups – TM will use round robin to poll TMs in other groups

If TM determines that a cache group is unpolled, it will poll it

- An error will be logged, which can be used by monitoring/alarming systems
- The closest TM group distance wise should be the first to poll the unpolled cache group
- An algorithm will be developed for TM to determine if cache groups are unpolled and how to poll them.

For safety, a profile parameter will allow for an override of the number of cache groups to poll.

If TM cannot poll a TM in a different TM group, it should log an error and try a different TM in the same TM group before failing

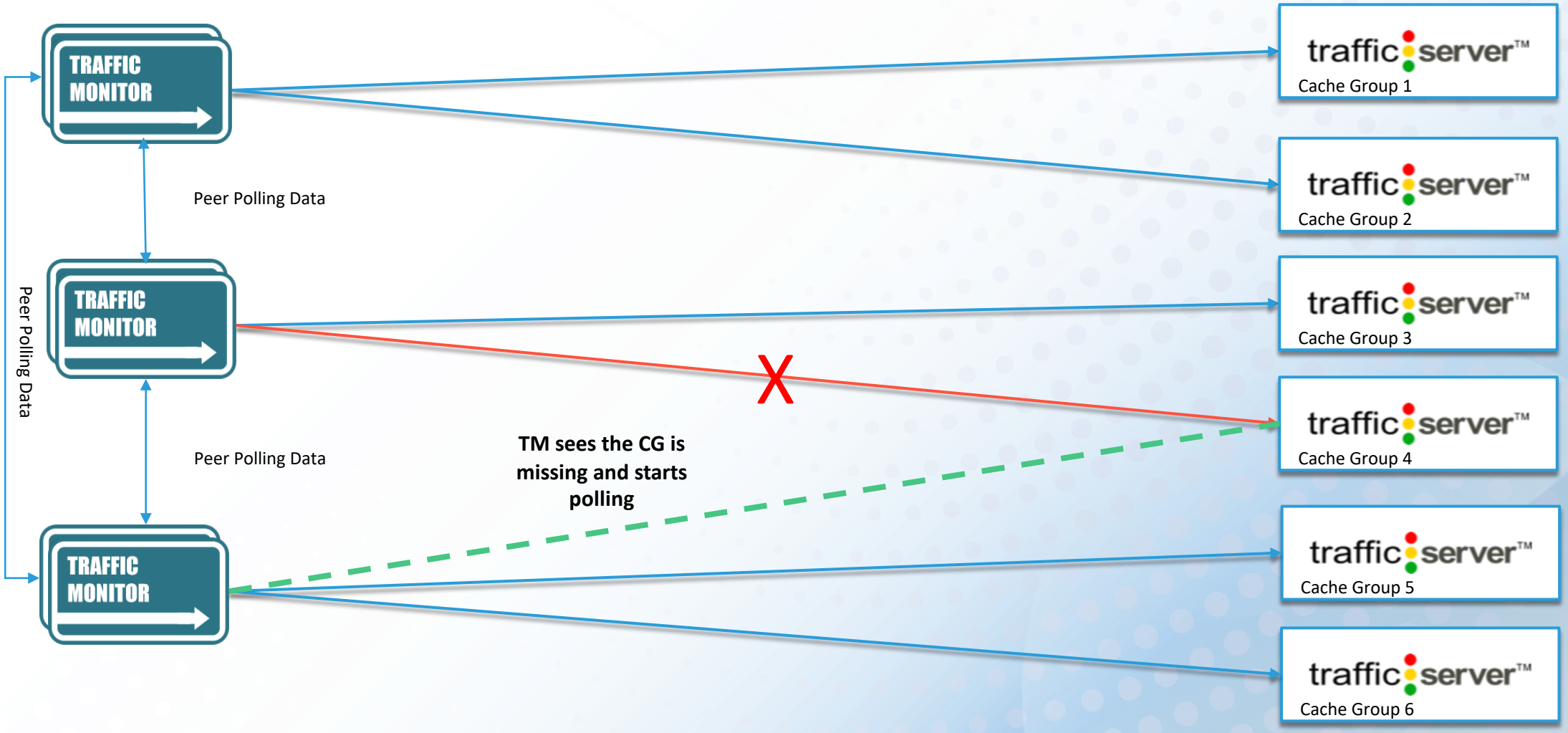
DISTRIBUTED TM POLLING

COMCAST

DISTRIBUTED TM POLLING

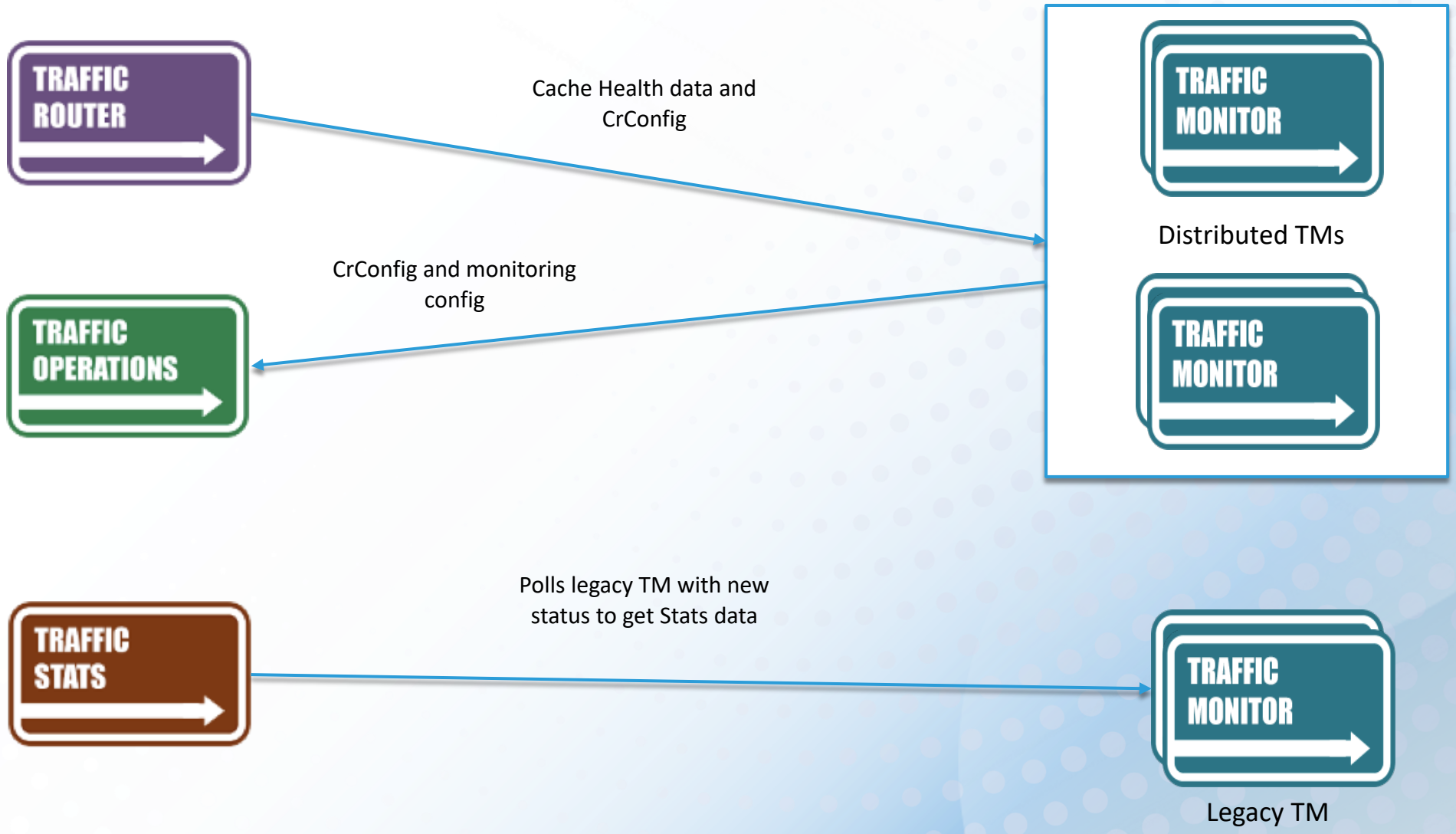
COMCAST

DISTRIBUTED TM POLLING - FAILOVER



COMCAST

DISTRIBUTED TM CONTROL PLANE COMMUNICATION



COMCAST

COMPROMISES

Multiple Phases

- Phase I – Health Polling only
- Phase II – Distributed Stats Polling

Delivery Service thresholds will NOT work with phase I of Distributed polling – e.g. total client connections for a DS, overall DS bandwidth – since these require stats polling information

TM will use round robin when polling TMs in other TM groups as opposed to a load balancer or using a different algorithm

Traffic Ops will continue to be the source of truth for distributed configuration

MIGRATION STRATEGY

Assign TMs to TM Groups

- Create new TM groups (Cache Groups)
- Assign by location (e.g. tm-west, tm-east, tm-central)

Upgrade OFFLINE TMs to latest version of TM

- turn ON distributed mode
- turn OFF stats collection

Monitor legacy and distributed TMs to ensure health status parity

- Could be done by writing a script to compare CRStates output at common interval
- Could be done by using a logging and analytics system to compare cache state changes
- Also need to ensure that CRStates on both systems contain all caches with proper admin status

Create a new status for legacy TMs so that TS can poll them (e.g. STATS_POLL)

- Set Distributed TMs to ONLINE
- Set Legacy TMs to new status
- Only subset need to be new status; Others can be OFFLINE

COMCAST

TRAFFIC MONITOR ROADMAP

Distributed Stat Polling

- Should we break that out into its own thing?
- Provide a flag to do health, stats, or both?

Health scores vs Boolean Health status

- Ability to be more granular with health data
- Rank cache health

Ability to integrate with external health monitoring tools

Improved default polling metrics – deprecate Load Avg?

More decoupling between TO and TM specifically around config?

RESOURCES

Distributed Traffic Monitor Blueprint

High Level Requirements Mailing List Discussion

Distribute Traffic Monitor design mailing list discussion

#traffic-control-traffic_monitor channel on the ASF slack

COMCAST


COMCAST