Threshold Compression for 3G Scalable Monitoring

Suk-Bok Lee, Dan Pei, MohammadTaghi Hajiaghayi, Ioannis Pefkianakis
Songwu Lu, He Yan, Zihui Ge, Jennifer Yates, Mario Kosseifi

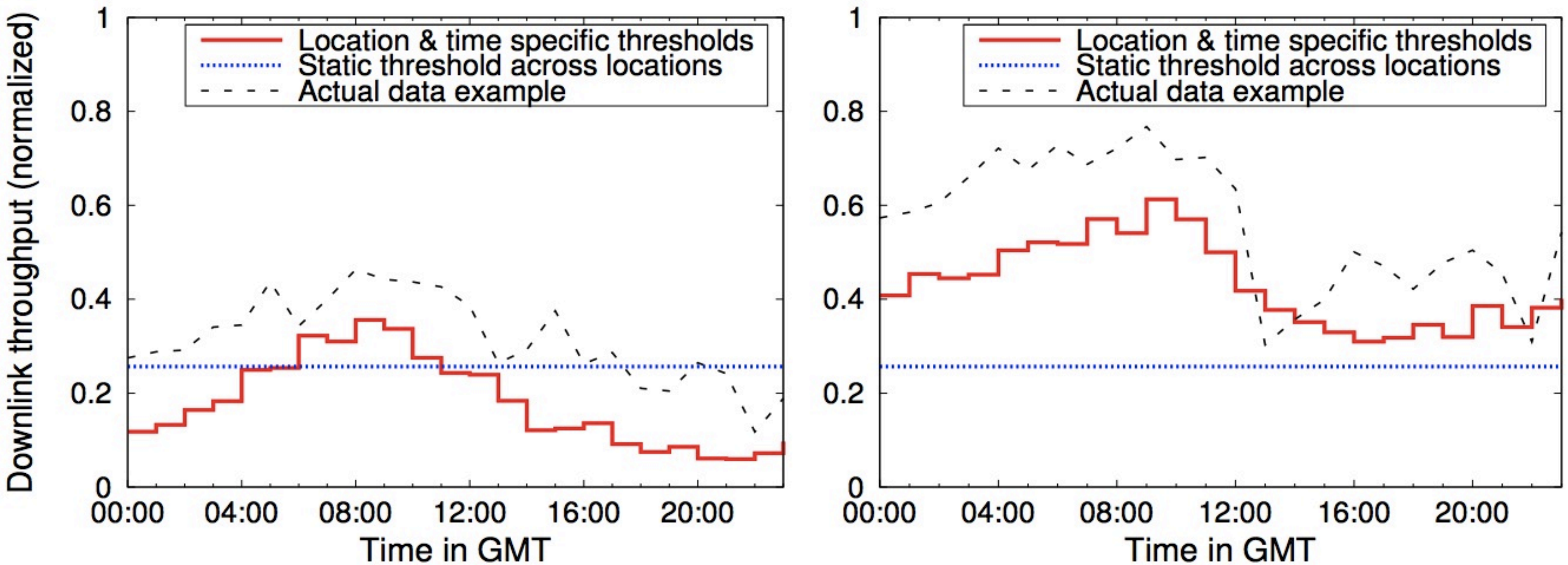
CMU AT&T Labs-Research UCLA
U of Maryland AT&T Network Services

What’s unique challenges on 3G network monitoring?

- A large number of network elements (NEs)
  - E.g. several thousands cell-sites in a single market area
  - Different-types of NEs:
    - GGSN - SGSN - RNC - NodeB – Sector
  - Various KPIs (key performance metrics)

- Dynamics on measurement results
  - Both in time and spatial domains
  - Reflecting:
    - Mobile user’s daily 3G usage pattens
    - Cell-site physical location & network topology

Naïve threshold-based alarming model is *not scalable* to large 3G networks



(a) Location 1 (b) Location 2

- Single static threshold (across locations): poor alarm quality
- Fine-grained threshold (location & time specific): management complexity

Possible thresholding schemes with different monitoring granularity

1. Per-NE-hourly (fine-grained location & time dependent)
   - Each NE has its own hourly thresholds
2. Per-NE-static
   - Each NE has a single (aggregating all hours) threshold
3. Per-NEtype-hourly
   - Every NE shares the same hourly (aggregating all NEs) thresholds
4. Per-NEtype-static
   - A single threshold (aggregating all hours and all NEs)

None of them are scalable to 3G monitoring

Threshold scheme	#thresholds	FPR	FNR
per-NE-hourly	**25320**	-	-
per-NE-static	1055	**31.1%**	**51.8%**
per-NEtype-hourly	24	**51.2%**	**47.5%**
per-NEtype-static	1	**53.2%**	**58.0%**

Thresholding on DL-throughput in a single area
(2010/06 - 2010/10)

- Per-NE-hourly
  - ideal for capturing dynamic 3G characteristics
  - threshold size per KPI grows very large with network size
- Aggregate-based threshold schemes
  - small threshold settings
  - high FPR (false positive rate) and FNR (false negative rate)

Fundamental tradeoff: threshold setting vs. alarm quality

- Fine-grained spatial-temporal thresholds
  - Pros: good alarm quality
    - Capture well each NE's location and time specific behavior
  - Cons: large # of thresholds, management complexity
    - E.g. a single area has >30,000 thresholds per KPI

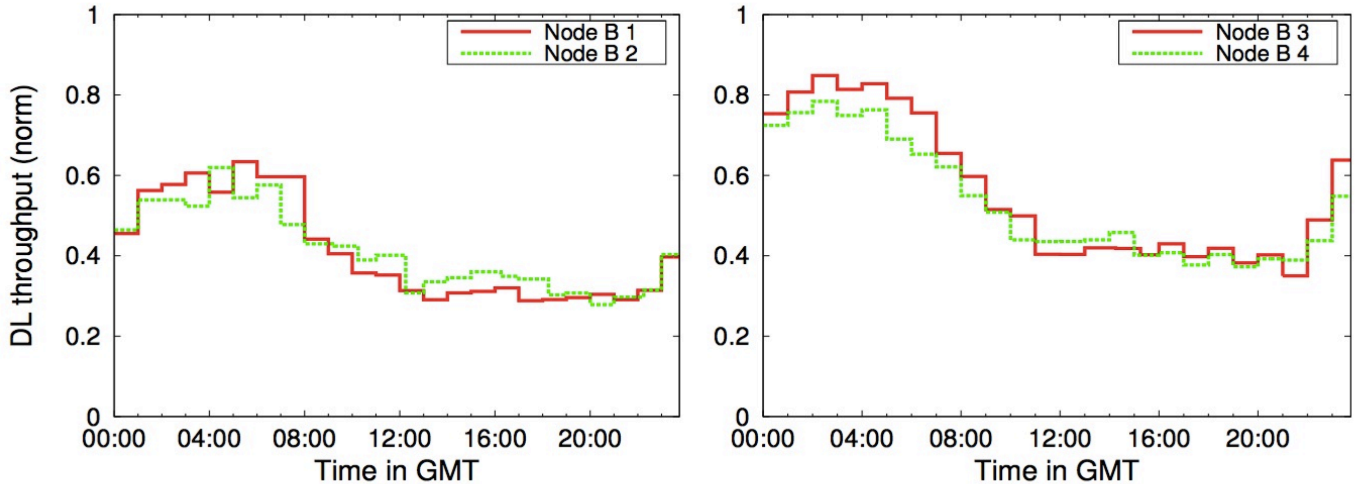
- Aggregate-based thresholds
  - Pros: a single threshold value for all NEs and hours
    - low system management overhead
  - Cons: poor alarm quality
    - E.g. can be observed ~70% false positive/false negatives

- **Can we have both advantages (small threshold settings and good alarm quality) in a large 3G network?**

Our solution: threshold compression

- Intelligent threshold aggregation
  - Observation 1:
    - Some group of NEs show similar threshold behaviors
      → threshold aggregation via **NE grouping**
  - Observation 2:
    - Certain group of hours show similar threshold behaviors
      → threshold aggregation via **hourly grouping**
- Our threshold-compression
  - characterizes the location- and time-specific threshold trend of each NE with minimal threshold setting
  - Maintains acceptable alarm accuracy

Observation of similar threshold behavior

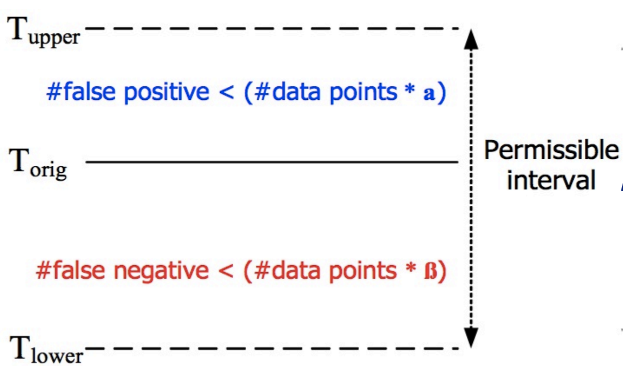


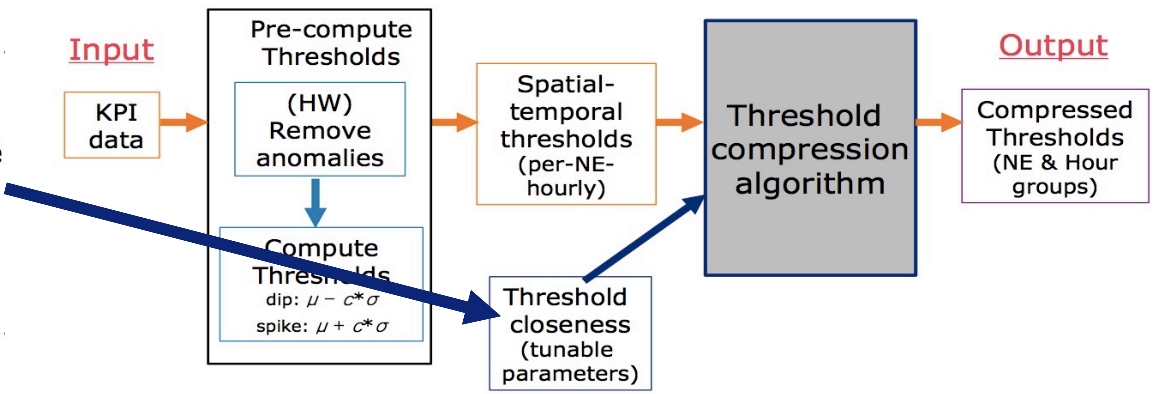
(a) NodeB1 and NodeB2 (b) NodeB3 and NodeB4

- Spatial-domain similarity
    - Geographic locations & user population around NEs
- Time-domain similarity
    - Daily trend of 3G usage pattern

Desirable properties of the solution

1. High compression gain
   - Small threshold setting even with large number of NEs
2. Low false alarm rate
   - Enforced by two input parameters α and β
     - Applying α and β to historical data → permissible interval
3. Management-oriented grouping
   - Each NE belongs to only one NE group, but multiple hour groups within an NE group → two-level hierarchical clustering





Threshold compression problem formulation

- Objective function
  - Find the minimum number of spatial-temporal clusters from a given fine-grained threshold setting (i.e. per-NE-hourly)

- Constraints
  1. Each compressed (aggregated) threshold must be within the permissible threshold interval of each spatial-temporal block which it represents to
  2. NE grouping must be consistent across time

- Hardness result
  - This problem is **not only NP-hard, but indeed inapproximable** as well

Threshold compression algorithm: two-staged approach

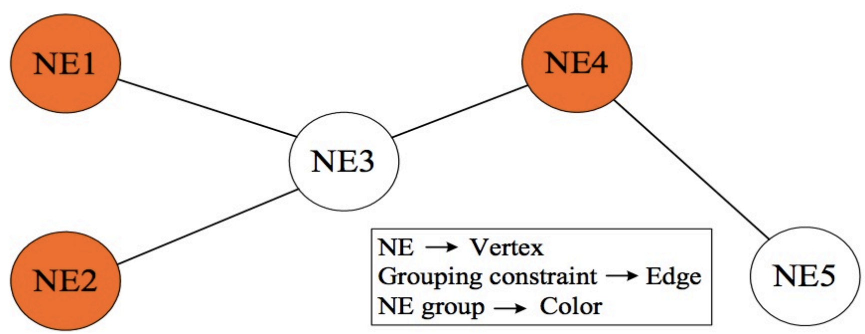
1. Spatial NE grouping
   - Identifies NE groups each showing similar threshold behavior each hour among its members
   - Each NE group consists of 24 hour-groups

2. Temporal-domain clustering within each NE group
   - Takes the NE grouping result as input to perform hour grouping for each identified NE group

- Strategy for clustering
  - Combine spatial-temporal blocks if they
    1. have <u>common intersection</u> in their permissible intervals
    2. Meet the consistent NE grouping rule

NE grouping: greedy coloring approach

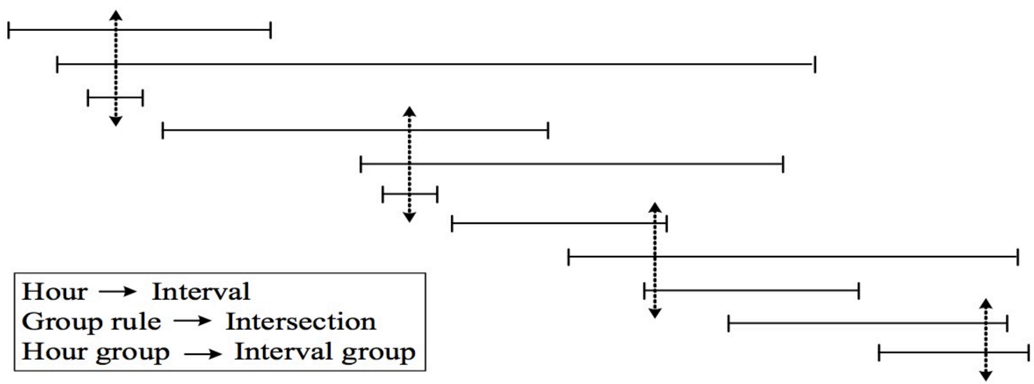
1. Convert to graph
   - Each NE → vertex
   - Put edge between two NEs, if they have disjoint permissible interval in any hour
2. Apply graph coloring
   - Minimum number of colors (NE groups) assignable to each vertex (NE) such that no edge (common intersection) connects two identically colored vertices (NE group members)
     - We apply the Welsh-Powell coloring algorithm that uses at most one more than the maximum degree of the graph



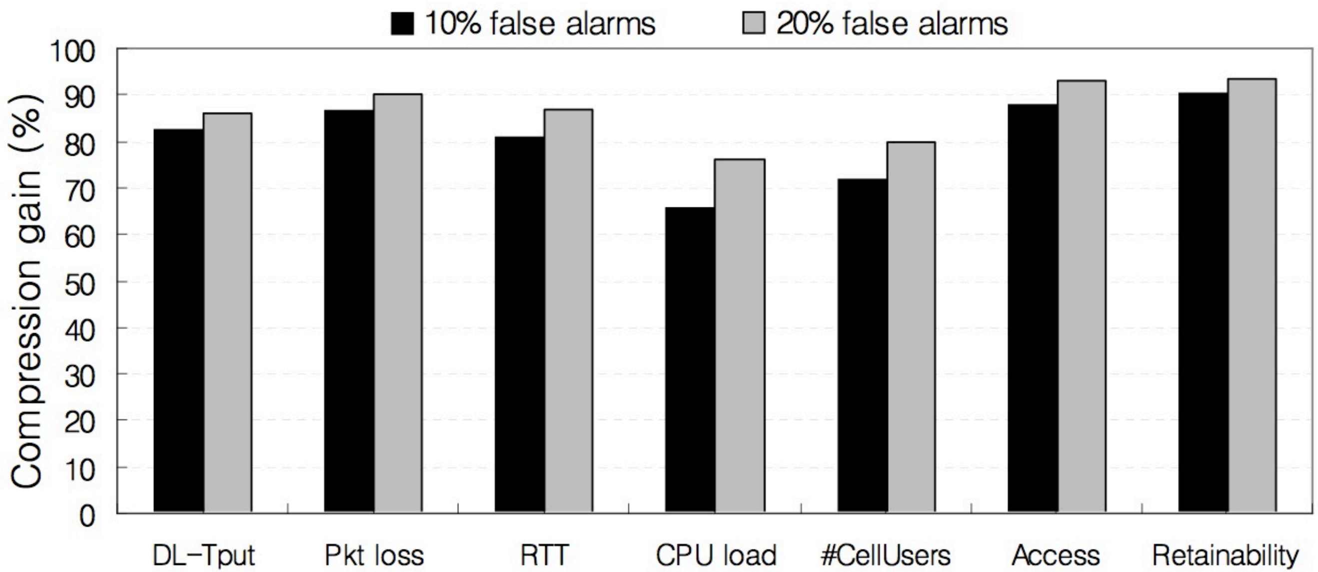
Hour grouping: minimum cover selection

- Convert to intervals
  - Each hour → its threshold (permissible) interval
- Do minimum cover
  - Find the minimum number of interval groups such that there is common intersection in each interval group

1. Sort all the interval endpoints
2. Scan until first encountering an upperbound point
3. Put all intervals containing this point in to a new interval group
4. Repeat from step 2



Evaluation: compression gain & alarm quality

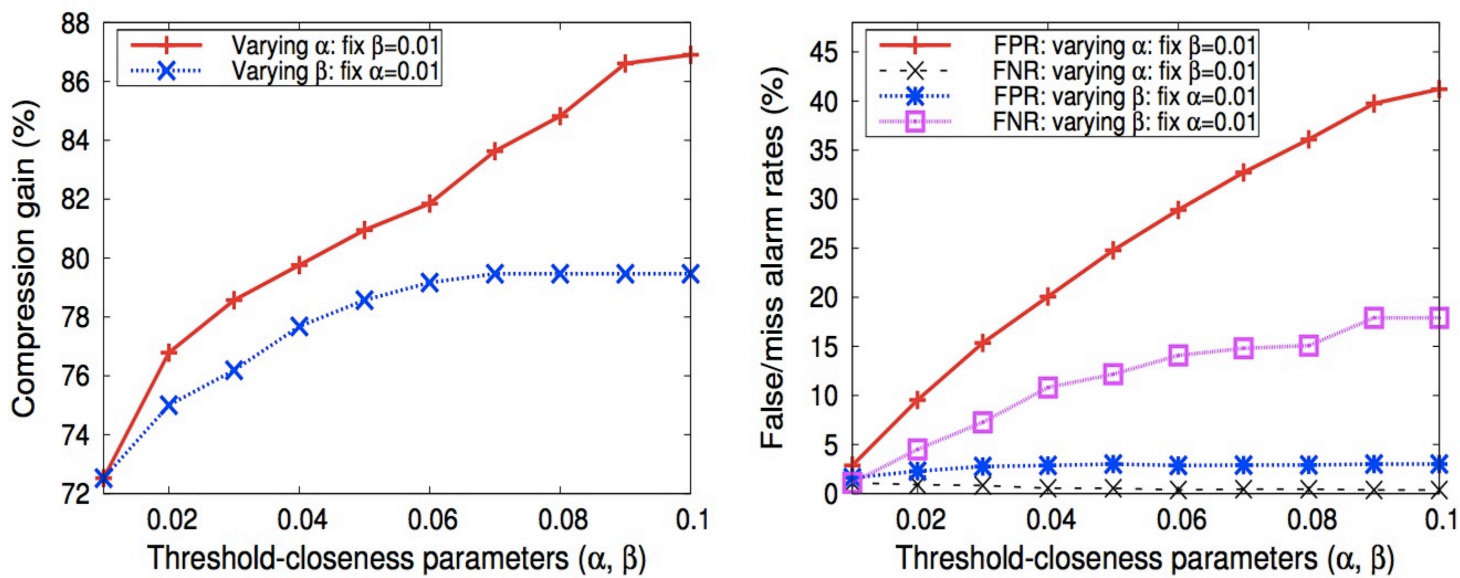


Compression gain on various KPIs
(Training dataset 2010/06 – 2010/08)

- Within desired 10% false/miss alarm*, nearly 70-90% compression gain

*In this study, we use slightly different definition of FPR = FP/(FP+TP) and FNR = FN/(FN+TP), to adapt them to the context where TP is much smaller than TN

Evaluation: compression gain & alarm quality by tuning input parameters



DL-throughput KPIs

- These give us a clear idea of how α and β should be chosen
  - E.g., setting α=0.03 and β=0.04 meets the target FPR (<15%) and FNR (<10%), which leads to compression gain of 82%

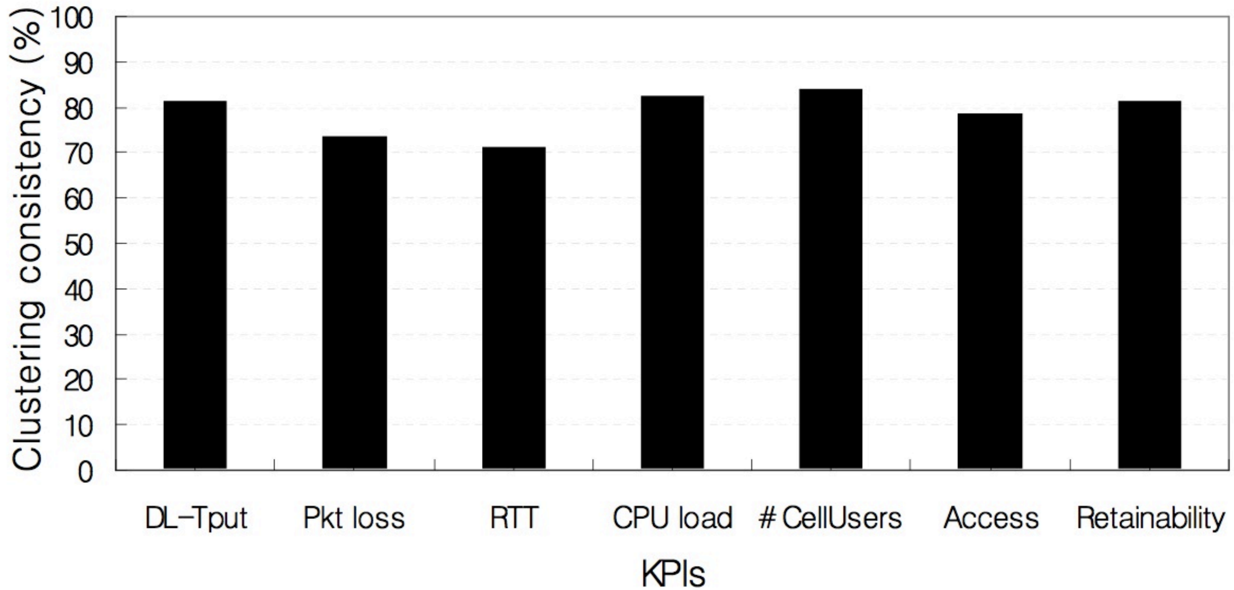
Validation: operational experience

KPI name	Comp.Gain	FPR	FNR
DL-throughput	75.2	15.6	9.4
Packet-loss	84.0	10.5	4.3
RTT	82.5	9.1	8.8
CPU-load	65.1	17.4	12.8
Cell-user-count	71.3	16.8	11.9
Iub-throughput	73.9	15.1	8.8
MAC-throughput	74.6	14.7	11.5
Accessibility	83.0	13.6	7.1
Retainability	81.6	13.4	8.5
Call-drop-rate	80.3	12.8	7.3

Validation results on various KPIs
(Applying the compressed threshold setting
to real data 2010/08 – 2010/10)

- The resulting FPR and FNR are within our target 10-15%

Validation: clustering stability over time



Spatial-temporal clustering consistency between training data and monitoring data on different KPIs

- All KPIs show above 70% consistency → robustness of the solution
  - Similar behavior across locations are consistent over time
  - Members in each identified cluster behave very closely one another across time, just like one single entity → key idea of our solution

Conclusion

- 3G monitoring is challenging due to its large scale and strong dynamics in both in time and spatial domains
  - Tradeoff: threshold setting vs. alarm quality
- We propose an intelligent threshold aggregation solution
  - Characterizes the location- and time-specific threshold trend of each individual NE with minimal threshold setting
- Operational experience with applying our solution has been very positive
  - Threshold setting reduction up to 90% with less than 10% false/miss alarm rates

Backup slides

Common practice for monitoring for a large-scale network

- Pre-defined (compute offline) thresholds is preferable
- Why not use a more sophisticated realtime-based dynamic thresholding? (e.g. exponential smoothing, regression analysis)
  - If applied to each individual node in the network, it will create excessive computational burden on the monitoring system.

Pre-computing thresholds

- First pass: remove anomalies.
  - Holt-Winters algorithm
    - taking into consideration diurnal, weekly pattern etc and individual network elements
- Second pass: compute thresholds.
  - compute the mean and standard deviation based on the data without anomalies, and then compute thresholds:
    - Yellow threshold: (mean – std) for dip KPIs (e.g. throughput), (mean + std) for spike KPIs (e.g. loss)
    - Red threshold: (mean – 2*std) for dip KPIs (e.g. throughput), (mean + 2*std) for spike KPIs (e.g. loss)

Observation of similar threshold behavior

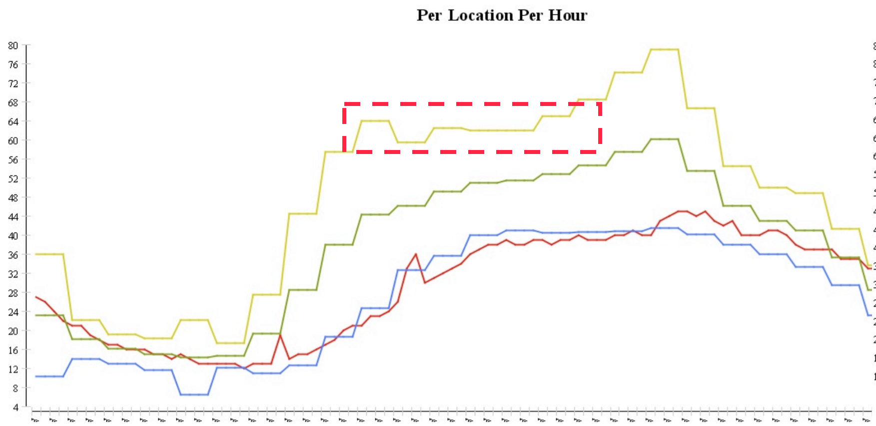
(Optima KPI: RNC CPU load)

- NRCSGAJTCR0R03:ATLNGAUYRNC001|9:10:11:12:13:14:15|14|64.56

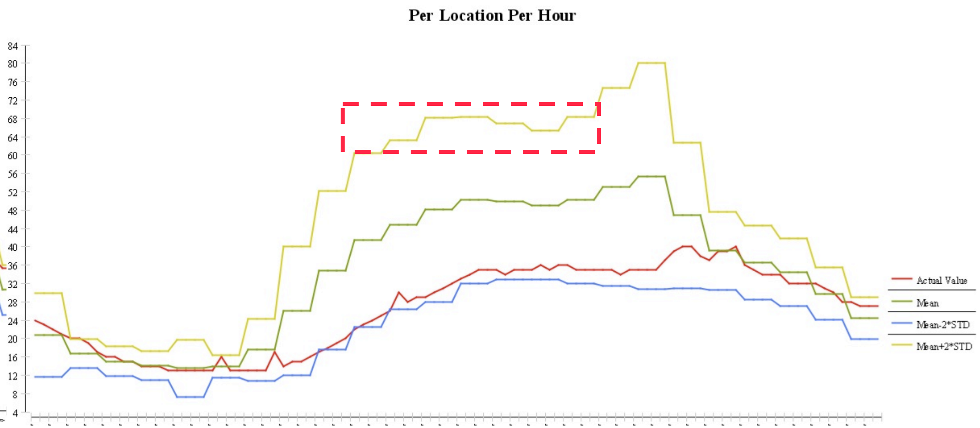
  Grouping across NEs   across hours

- **Previously 14 thresholds can become one threshold**

RNC1: NRCSGAJTCR0R03



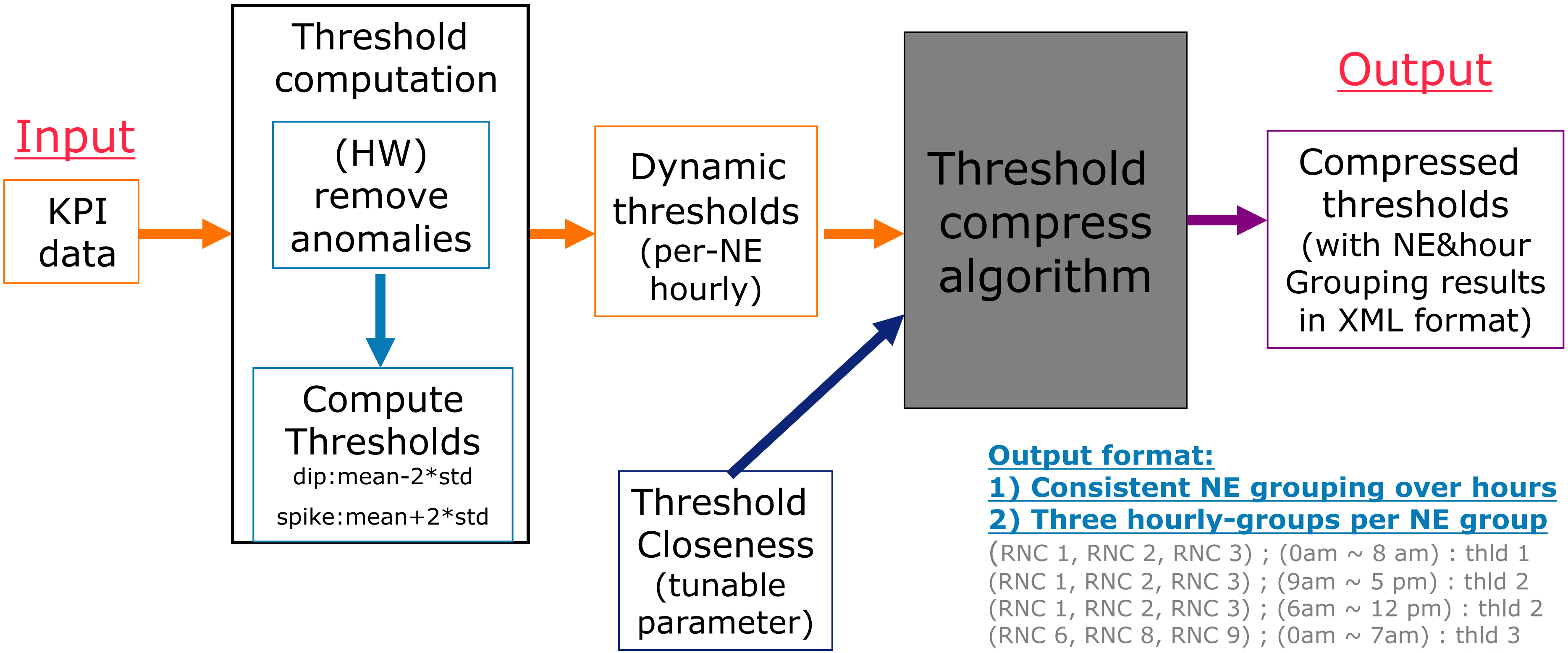
RNC2: ATLNGAUYRNC001



These two RNCs are under the same SGSN…

Overall picture

- For each KPI, the algorithm outputs the compressed thresholds with NE & hour grouping results



Output format:
1) Consistent NE grouping over hours
2) Three hourly-groups per NE group

(RNC 1, RNC 2, RNC 3) ; (0am ~ 8 am) : thld 1
(RNC 1, RNC 2, RNC 3) ; (9am ~ 5 pm) : thld 2
(RNC 1, RNC 2, RNC 3) ; (6am ~ 12 pm) : thld 2
(RNC 6, RNC 8, RNC 9) ; (0am ~ 7am) : thld 3
(RNC 6, RNC 8, RNC 9) ; (8am ~ 7pm) : thld 4
(RNC 6, RNC 8, RNC 9) ; (8pm ~ 12pm) : thld 4