# 1. Introduction

*A* method for performing switch or router system thermal control monitoring, such as that employed for complex system's configuration with a single FAN control and with multi points of temperature measure, is disclosed. The technique, referred as the Optimized Thermal Monitoring Algorithm (OTMA) approach, periodically re-calculates the highest temperature score of the all thermal zones and enforce thermal control to following the thermal zone with the highest score.

# 2. Current Methodology and Problems Being Addressed

The OTMA approach is targeted for the complex multiport any type network switches or routes required thermal management to improve reliability and prevent premature failure and save energy.

Such kind of systems are used to be equipped with big number of temperature sensors and limited or even single cooling device performing temperature monitoring and protection over the system. When multiple sensors are mapped to the same cooling device, the cooling device is should be set according the worst sensor from the sensors associated with this cooling device. The system shall implement cooling control based on thermal monitoring of the critical temperature sensors. The temperature control should keep the system in a desired operation temperature. Along with this requirement a thermal monitoring should not abuse the system resources and should be protected from the situation, when some of system temperature sensors requires to speed up cooling device, while some other requires speed down. Last case could lead to undesirable system thermal behavior, redundant noise and after all such system could get to malfunction behaviour due to bad thermal monitoring.

There is now open source solution addressing the thermal monitoring for such class of systems and there are no any commercial software for that.

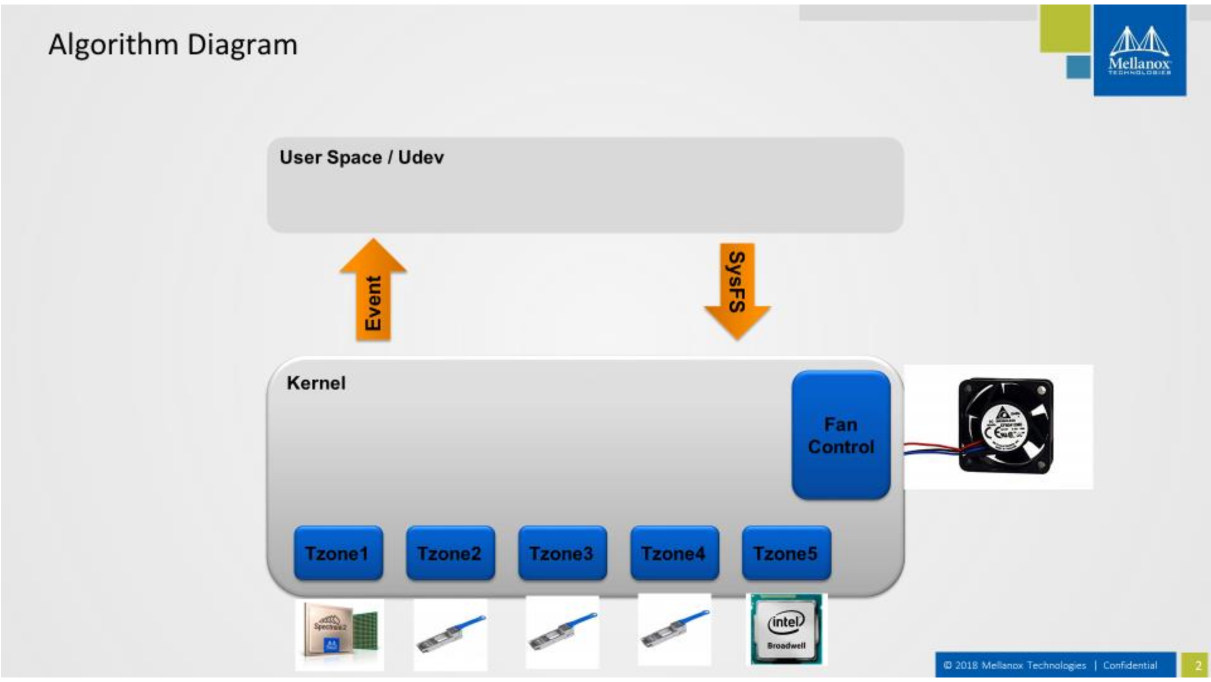
## 3. Solution summary

Currently there is no solution for the third party OS.
The general purpose of the solution is to provide optimal thermal monitoring for Mellanox switches running third party OS, independently of the silicon type (Spectrum-2, Spectum, SX or IB Switch), and the number of ports system equipped with.

The presented solution provides common solution for the thermal monitoring on Mellanox systems for Linux based switch operating systems . Algorithm will incorporated with existing Linux thermal control and will enhance it to allow more advance and accurate thermal control in Mellanox switches. Existing Linux kernel algorithm dose not support multiple sensors on same device . Each device is map to a thermal zone , however each thermal zone can effect fan rotation speed. This will result in collision between different thermal zones about how fast to spin the fans.

The presented approach for performing thermal control monitoring for systems with a single cooling system and multiple temperature sensors:

- introduces the definition of the highest thermal zone. highest thermal zone will be the zone that is currently most critical to the systems.
- provides the flexible approach, based on introducing of the "highest thermal zone" definition, which allows easy extension of the thermal monitoring, for example in the cases, when the additional thermal factors are to be considered (for example, next generation systems with Gearbox).
- allows to Decrease the amount of the context switching within the thermal monitoring. Highest thermal zone calculation is provided through the kernel space thread and user space should be involved only in case of the highest thermal zone re-assignment.



## 8.1. Detailed Explanation

**Table 1 Terminology**

| Definitions | Description |
|---|---|
| OTMA | Optimized Thermal Monitoring Algorithm. |
| TA | Thermal Area - contains the number of the thermal zones (TZ<i>) shared single cooling device, bound to all these zones. |
| TZ<i> | Thermal Zone <i> - node containing thermal measurement sensor, definition of the temperature trip points and cooling device binding. |
| TZD | Thermal Zone Designated - thermal zone assigned for periodical re-calculation of the highest temperature score. It's possible to have a few TZD within the same TA. In such case TZDx will be associated with with TZ<xi> (i form 1 to n1) zones, TZDy with TZ<yj> (j from 1 to n2), etc. |
| TZH | Thermal Zone Highest - thermal zone with the highest temperature score. |
| TZM | Thermal Zone Mode - working mode of a thermal zone, where mode could be enable or disable. When thermal zone is in disabled mode, it is not the subject of monitoring. |
| TZP | Thermal Zone Policy - thermal governor for a thermal zone, where governor is responsible for performing throttling of the cooling device (FAN) associated with a thermal zone. The allowed polices are:<br><br>step wise policy - set a cooling device according to the thermal trends (high temperature - faster cooling device, lower temperature - slower cooling device);<br><br>standby policy - which does not change cooling device setting, until a thermal zone is not assigned as TZH. |
| TZTno<i> | Thermal Zone Trip point normal temperature of TZ<i>. When a thermal zone temperature is below this value - the temperature is below the desired temperature range. |
| TZThi<i> | Thermal Zone Trip point high temperature of TZ<i>. When a thermal zone temperature is below this value and is above of TZTno<i> - the temperature is within the desired temperature range. |
| TZTho<i> | Thermal Zone <i> Trip point hot temperature of TZ<i>. When a thermal zone temperature is below this value and is above of TZThi<i> - the temperature is above the desired range but and cooling device speed should be increased. |
| TZTcr<i> | Thermal Zone <i> Trip point critical temperature of TZ<i>. When a thermal zone temperature is below this value and is above of TZTho<i> - the cooling device should set to the maximum speed. When a thermal zone temperature is below this value - it might cause system damage and thermal shutdown should be initiated. |

### Description

Below is the TA model. It contains n thermal zones (where a thermal zone could be associated with any system component, like CPU, Network processor, memory, ports, PS units or just with some system critical points from thermal perspective).

TZD assignment is pre-defined within the TA and this role can't be changed on the fly. It could be a few TZD within the same TA. Any of TZ<i> can be set as TZD, but preferably to pick a thermal zone, which could not be removed. For example, CPU or network processor are good for such assignment, while transceiver or replaceable units are not so good for a such role. TZD is always enabled, since it's responsible for TZH re-calculation.

Only one thermal zone can be set at a given time as TZH. The cooling device manipulation is performed according to TZH state only (step wise policy), while for all others thermal zones the standby policy is applied (which means these thermal zones don't affect cooling device).

Same thermal zone could be TZD and TZH at the same time.

**Table 2 Thermal Area (TA) model with n thermal zones**

| Name | TZTno/TZThi/ TZTho/TZTcr | TZD | TZH | TZP | TZM |
|---|---|---|---|---|---|
| TZ<1> | Variable | No | No | Standby | Disable |
| TZ<i> | Variable | Yes | No | Standby | Enable |
| TZ<k> | Variable | No | Yes | Step wise | Enable |
| TZ<n> | Variable | No | No | Standby | Disable |

Such approach allows to reduce CPU utilization, since thermal control works over the maximum two thermal zones (TZD and TZH), while it services much more number of thermal zones.

It also helps to avoid competing between the thermal zones for the cooling device control and helps to avoid collisions between thermal zones, when one of them could require increasing of the cooling device speed, while another one could require its reducing.

### *Highest thermal zone calculation*

TZH is represented by 32 bits unsigned integer and calculated according to the next formula:

For T < TZ<t><i>, where t from {no = 0, hi = 1, ho = 2, cr = 3}:

TZ<i> score = (T + (TZ<t><i> - T) / 2) / (TZ<t><i> - T) * 256 ** j;

It divides positive dividend T (current TZ<i> temperature read from thermal sensor) by positive divisor (t trip temperature minus TZ<t><i> - T) rounds the result to closest integer and multiply it by 256 in power j.

Following this formula, if TZ<i> is in trip point higher than TZ<k>, the higher score is to be always assigned to TZ<i>, independently of the real temperature.

If TZ<i> and TZ<k> are in the same trip point <t>, the higher score is to be assigned to the TZ, which is "closer" to its related trip point <t + 1> - respectively to TZ<t + 1><i> and TZ<t + 1><k>.

For T > TZcr<i> system thermal shutdown is to be performed.

TZH score = MAX(TZ<i>score);

And TZ with the maximum score is to be assigned as TZH.

***Rationale***

Consider the below trip points, temperatures and scores for a few thermal zones.

**Table 3 Thermal zones scores calculation**

| Name | TZT<>no | TZT<>hi | TZT<>ho | TZT<>cr | Temperature | Score | Position |
|---|---|---|---|---|---|---|---|
| TZ<i> below normal trip | 75 | 85 | 105 | 110 | 51 | 0x00000002 | 5 |
| TZ<j> below normal trip | 60 | 70 | 80 | 90 | 59 | 0x0000003C | 3 |
| TZ<k> below normal trip | 66 | 76 | 88 | 98 | 63 | 0x00000010 | 4 |
| TZ<l> above hot trip | 60 | 70 | 80 | 90 | 82 | 0x000A0000 | 1 |
| TZ<m> above high trip | 60 | 70 | 80 | 90 | 62 | 0x00000900 | 2 |

According to the provided calculation method, the score for the thermal zone is assigned depending on thermal zone trip point location within the below ranges:

| | |
|---|---|
| Tx <= TZ<x>no | 0x00000000 <= Tx Score <= 0x000000FF |
| TZ<x>no < Tx <= TZ<x>hi | 0x000000FF < Tx Score <= 0x0000FF00 |
| TZ<x>hi < Tx <= TZ<x>ho | 0x0000FF00 < Tx Score <= 0x00FF0000 |
| TZ<x>ho < Tx <= TZ<x>cr | 0x00FF0000 <= Tx Score <= 0xFF000000 |
| Tx > TZ<x>cr | Tx Score is 0xFFFFFFFF - subject for the system thermal shutdown |

For two thermal zones located at the same kind of trip point, the higher score will be assigned to the zone, which closer to the next trip point, where distance is calculated as normalized value, between thermal zone real temperature and next trip point temperature.

Thus, the highest score will always be assigned objectively to the hottest thermal zone and the control of the cooling device will always be stuck to the thermal zone with the highest score.

### *Initial state*

At initial state TZD is always set as enabled with active step wise policy, while all others thermal zones are set as disabled and with standby policy. TZD is set according to the system configuration, for example to TZ<0>. The thermal zone assigned as TZD will keep this role while the system is running.

**Table 4 Thermal zones initial state**

| Name | Initial mode | Initial policy |
|---|---|---|
| TZD (TZ<0>) | Enable | Step wise |
| TZ<i> (i from 1 to n) | Disable | Standby |

### *Dynamic re-assignment of TZH*

When TZD during highest score re-calculation detects new TZH (for example TZ<y> should be assigned as TZH instead of TZ<x>), the next transition of mode and policy is performed:

**Table 5 TZH transition from non TZD to non TZD**

| Name | Current mode | Current policy | Next mode | Next policy |
|---|---|---|---|---|
| TZ<x> | Enable | Step wise | Disable | Standby |
| TZ<y> | Disable | Standby | Enable | Step wise |

**Table 6 TZH transition from TZD**

| Name | Current mode | Current policy | Next mode | Next policy |
|---|---|---|---|---|
| TZ<x> | Enable | Step wise | Enable | Standby |
| TZ<y> | Disable | Standby | Enable | Step wise |

**Table 7 TZH transition to TZD**

| Name | Current mode | Current policy | Next mode | Next policy |
| --- | --- | --- | --- | --- |
| TZ<x> | Enable | Step wise | Disable | Standby |
| TZ<y> | Enable | Standby | Enable | Step wise |

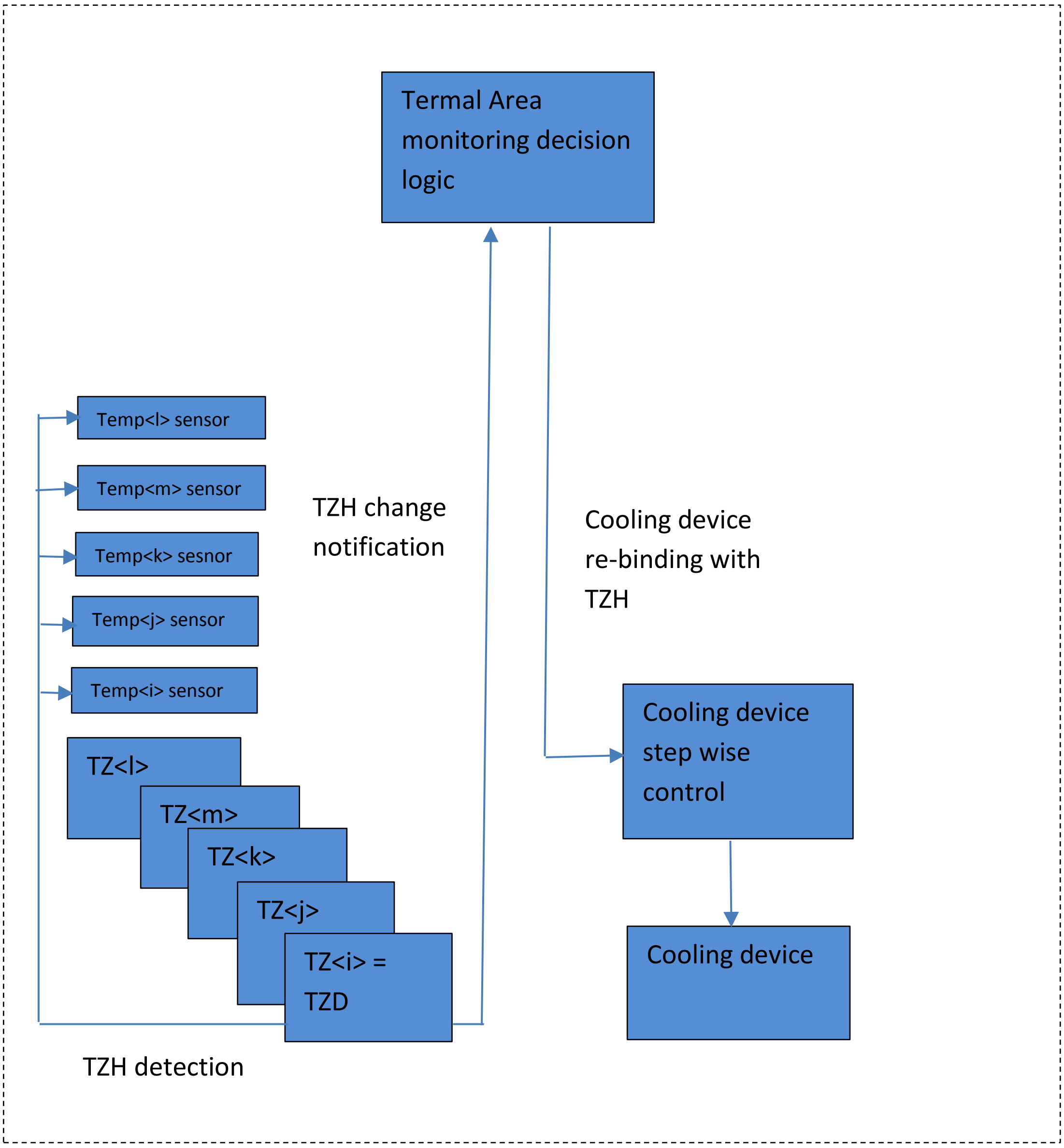
### *Dormant state*

The dormant state means that cooling device speed is set to some constant value. System can get to the dormant state according to some system particular criterias. For example it could be requirement to set cooling device to maximum speed in case some system unit, for example power suppply, is absent, or some of tacometers is faulty. In such case the system will be in this state until power unit is not inserted back or until tacometer fault is not cleared. On exit from the dormant state system is always moved to the initial state.

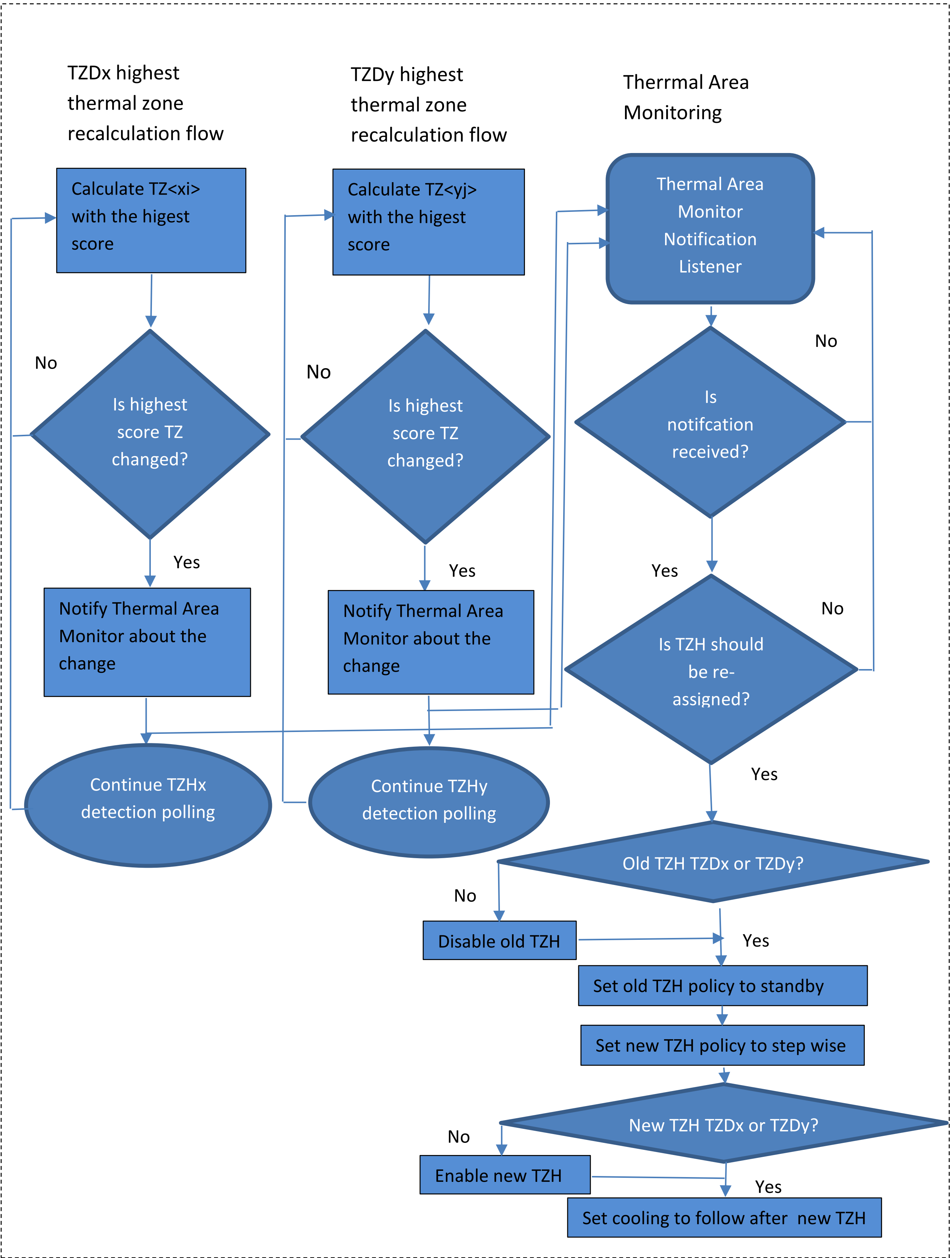
**Table 8 Dormant state**

| Name | Dormant mode | Dormant policy |
| --- | --- | --- |
| TZ<i> (i from 0 to n) | Disable | Standby |

### *(1) System Block Diagram:*



### *(2) Flow Chart*


TZDx highest thermal zone recalculation flow
TZDy highest thermal zone recalculation flow
Therrmal Area Monitoring
Calculate TZ<xi> with the higest score
Calculate TZ<yj> with the higest score
Thermal Area Monitor Notification Listener
No
Is highest score TZ changed?
No
Is highest score TZ changed?
No
Is notifcation received?
Yes
Yes
Yes
Notify Thermal Area Monitor about the change
Notify Thermal Area Monitor about the change
No
Is TZH should be re-assigned?
Continue TZHx detection polling
Continue TZHy detection polling
Yes
Old TZH TZDx or TZDy?
No
Disable old TZH
Yes
Set old TZH policy to standby
Set new TZH policy to step wise
New TZH TZDx or TZDy?
No
Enable new TZH
Yes
Set cooling to follow after new TZH

## PWM full speed policy

This package provides additional functionally to the thermal control, which contains the following polices:

- Setting PWM to full speed if one of PS units is not present (in such case thermal monitoring in kernel is set to disabled state until the problem is not recovered). Such events will be reported to systemd journaling system.
- Setting PWM to full speed if one of FAN drawers is not present or one of tachometers is broken (in such case thermal monitoring in kernel is set to disabled state until the problem is not recovered). Such events will be reported to systemd journaling system.

In the above cases all thermal zones are moved to the dormant state and PWM is to be at a full speed untill all the states (absent PS unit or fan drawer, faulty fan) caused PWM full speed setting are not cleared. After all of them are cleared thermal zones are moved to initial state.

## Dynamic minimal PWM speed policy

The dynamic setting depends on fan direction and cable type. For system with copper cables only or/and with trusted optic cable minimum PWM setting could be decreased according to the system definition. Such events will be reported to systemd journaling system.

There are per system type dynamic minimum tables, which define default minimum FAN The cooling device bound to the thermal zones operates over the ten cooling logical levels. The default vector for the cooling levels is defined with the next PWM per level speeds:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 20% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |

In case system dynamical minimum is changed for example from 20% to 60%, the cooling level vector will be dynamically updated as below:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 60% | 60% | 60% | 60% | 60% | 60% | 70% | 80% | 90% | 100% |

In such way the allowed PWM minimum is limited according to the system thermal requirements.

Per system thermal tables for the minimum PWM setting contain entries with the ambient temperature threshold values and relevant minimum speed setting. All Mellanox system are equipped with two ambient sensors: port side ambient sensor and fan side ambient sensor. The fan direction can be read from its EEPROM data, in case it is equipped with EEPROM device. If it dpesn't eqipped with EEPROM, the direction is read from CPLD fan direction register. Or for the common case it can be calculated according to the next rule:

- if port side ambient sensor value is greater than fan side ambient sensor value - the direction is power to cable (forward);
- if port side ambient sensor value is less than fan side ambient sensor value - the direction is cable to power (reversed);

- if these value are equal: the direction is unknown.

For each system the following six vectors are defined:

- p2c_dir_trust_tx: all cables with trusted sensors or with no sensors, FAN direction is power to cable (forward).
- p2c_dir_untrust_tx: some cable sensor is untrusted, FAN direction is power to cable (forward).
- c2p_dir_trust_tx: all cables with trusted sensors or with no sensors, FAN direction is cable to power (reversed).
- c2p_dir_untrust_tx: some cable sensor is untrusted, FAN direction is cable to power (reversed).
- unk_dir_trust_tx: all cables with trusted sensors or with no sensors, FAN direction is unknown.
- unk_dir_untrust_tx: some cable sensor is untrusted, FAN direction is unknown.

Below is an example dynamic PWM setting for the different systems type.

**Table 9 Dynamic PWM minimum speed table sor system type 1 (Panther, Spider)**

| | PWM minimum [%] | | | | | |
|---|---|---|---|---|---|---|
| Direction | P2C | | C2P | | Unknown | |
| Cable reliability | trusted | untrusted | trusted | untrusted | trusted | untrusted |
| Abmbient [C] | | | | | | |
| <0 | 30 | 30 | 30 | 30 | 30 | 30 |
| 0 - 5 | 30 | 30 | 30 | 30 | 30 | 30 |
| 5 - 10 | 30 | 30 | 30 | 30 | 30 | 30 |
| 10 - 15 | 30 | 30 | 30 | 30 | 30 | 30 |
| 15 -20 | 30 | 30 | 30 | 30 | 30 | 30 |
| 20 - 25 | 30 | 30 | 40 | 40 | 40 | 40 |
| 25 -30 | 30 | 40 | 50 | 50 | 50 | 50 |
| 30 - 35 | 30 | 50 | 60 | 60 | 60 | 60 |
| 35 - 40 | 30 | 60 | 60 | 60 | 60 | 60 |
| 40- 45 | 50 | 60 | 60 | 60 | 60 | 60 |

**Table 10 Dynamic PWM minimum speed table sor system type 2 (Bulldog)**

| | PWM minimum [%] | | | | | |
|---|---|---|---|---|---|---|
| Direction | P2C | | C2P | | Unknown | |
| Cable reliability | trusted | untrusted | trusted | untrusted | trusted | untrusted |
| Abmbient [C] | | | | | | |
| <0 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0 - 5 | 20 | 20 | 20 | 20 | 20 | 20 |
| 5 - 10 | 20 | 20 | 20 | 20 | 20 | 20 |
| 10 - 15 | 20 | 20 | 20 | 20 | 20 | 20 |
| 15 -20 | 20 | 30 | 20 | 20 | 20 | 30 |
| 20 - 25 | 20 | 30 | 20 | 20 | 20 | 30 |
| 25 -30 | 20 | 40 | 20 | 20 | 20 | 40 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 30 - 35 | 20 | 50 | 20 | 20 | 20 | 50 |
| 35 - 40 | 20 | 60 | 20 | 20 | 20 | 60 |
| 40- 45 | 20 | 60 | 30 | 30 | 30 | 60 |

**Table 11 Dynamic PWM minimum speed table sor system type 3 (Panther Steetfighter)**

| | PWM minimum [%] | | | | | |
|---|---|---|---|---|---|---|
| Direction | P2C | | C2P | | Unknown | |
| Cable relability<br>Abmbient [C] | trusted | untrusted | trusted | untrusted | trusted | untrusted |
| <0 | 30 | 30 | 30 | 30 | 30 | 30 |
| 0 - 5 | 30 | 30 | 30 | 30 | 30 | 30 |
| 5 - 10 | 30 | 30 | 30 | 30 | 30 | 30 |
| 10 - 15 | 30 | 30 | 30 | 30 | 30 | 30 |
| 15 -20 | 30 | 30 | 30 | 40 | 30 | 40 |
| 20 - 25 | 30 | 30 | 30 | 40 | 30 | 40 |
| 25 -30 | 30 | 30 | 30 | 40 | 30 | 40 |
| 30 - 35 | 30 | 30 | 30 | 50 | 30 | 50 |
| 35 - 40 | 30 | 40 | 30 | 70 | 30 | 70 |
| 40- 45 | 30 | 50 | 30 | 70 | 30 | 70 |

**Table 12 Dynamic PWM minimum speed table sor system type 4 (Boxer)**

| | PWM minimum [%] | | | | | |
|---|---|---|---|---|---|---|
| Direction | P2C | | C2P | | Unknown | |
| Cable relability<br>Abmbient [C] | trusted | untrusted | trusted | untrusted | trusted | untrusted |
| <0 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0 - 5 | 20 | 20 | 20 | 20 | 20 | 20 |
| 5 - 10 | 20 | 20 | 20 | 20 | 20 | 20 |
| 10 - 15 | 20 | 20 | 20 | 20 | 20 | 20 |
| 15 -20 | 20 | 30 | 20 | 20 | 20 | 30 |
| 20 - 25 | 20 | 40 | 20 | 30 | 20 | 40 |
| 25 -30 | 20 | 40 | 20 | 40 | 20 | 40 |
| 30 - 35 | 20 | 50 | 20 | 50 | 20 | 50 |
| 35 - 40 | 20 | 60 | 20 | 60 | 20 | 60 |
| 40- 45 | 20 | 60 | 20 | 60 | 20 | 60 |

## Thermal zone configuration

Each thermal zone is configured with the thermal ranges, defined in the below table. Each zone, excepted last one, is configured with 5 Celsius hysteresis trip (th = 5C). For ASIC thermal zone these ranges are set to the pre-defined value. Fot the QSFP modules thermal zones, these ranges are set according to the particular module warning and critical temperature thersholds: tw – warning temperature threshold and tc – critical temperature threshold.

**Table 10 Thermal zone definition**

| Thermal zone state | ASIC thermal zone | Module thermal zone | Thermal action |
|---|---|---|---|
| Cold | t < 75 C | t < (tw – 2 * th) | Do nothing, keep PWM at minimal speed |
| Normal | 75 C <= t < 85 C | (tw – 2 * th) < t < tw | Perform hot algorithm |
| High | 85 C <= t < 105 C | tw <= t < tc | Set PWM at full speed |
| Hot | 105 C <= t < 110 C | tc <= t < (tc + 2 * th) | Keep PWM at full speed and produce warning message |
| Critical | t >= 110 C | t >= (tc + 2 * th) | System thermal shutdown |

### Referneces

Reference code with implementation can be found at https://github.com/MellanoxBSP/thermal-control/tree/for-next.

It contains set of thermal control scripts and kernel patch for kernel v.21.