ICNS'09 Tutorial

Capacity Constrained MANETs – Adaptive Policy-driven Fault and Performance Management

Dr. Latha Kant
Telcordia Technologies
Director, Mobile Networking Research
1 Telcordia Drive, Piscataway, NJ 08854
lkant@research.telcordia.com

April 20, 2009

Outline

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - Case Studies
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

MANETs: What are they? – Brief Overview

- Mobile Ad hoc Wireless NETworks (MANETs) are wireless networks that:
  - Do not have fixed infrastructures
    - Contrast with cellular and wireless LANs/WANs where the networks rely on existence of fixed towers or access points for relaying communications
      - ➢ **Implication: Every 'node' in a MANET must be capable of functioning both as a router and as a 'host'**
  - Are usually deployed in 'on-demand' situations or areas that are difficult to 'reach'; e.g.,
    - Military operations such as: Network Centric Warfare – NCW, Future Force Networking
    - Emergency situations: rescue operations, disaster relief, ..
    - Sensory environments: Oil sensing, structural stability sensing, ..
    - Other General: Travelling expos, concerts, rallies, …
    - ➢ **Implication: Unlike cellular nets, LANs/WANs/MANs or telecommunication (wireline) networks, there is not much of a 'planning' stage while deploying MANETs**

MANETs: What are they? - Brief Overview: continued

- Mobile Ad hoc Wireless NETworks (MANETs) are wireless networks that (continued):
  - Require distributed and automated networking and network management operations
    - E.g., Networking operations: Topology creation, maintenance, routing
    - E.g., Network Management Operations: network element configurations, fault management and self-healing, performance adaptations
  - Have widely varying 'network sizes'
    - MANETs can consist of a handful of nodes (local area emergency) or thousands of nodes (NCW-nets)
  - ➢ Implication: Very little structure to network with limited *a priori* knowledge of network; unlike cellular /LANs /WANs /MANs /telecom nets

Unique MANET characteristics & associated management challenges

- Dynamic Topology with no fixed infrastructure
  - All nodes in MANET potentially mobile (with the exception of sensors); 'dynamism' only exacerbated due to unpredictability of node movement
  - No concept of 'relay' nodes that remain 'fixed'

  ➢ ***Example management challenges: Network (Re)Configuration, Fault Diagnosis & Self-healing***

- Power & Processing Constraints
  - MANET nodes are generally constrained by processing and power limitations
    - Notes:
      - Cellular handsets are also constrained somewhat, impact of these limitations is far more severe in MANETs – e.g., cannot re-charge soldier's battery during mission; or cannot add battery power to a remote sensor whereas handsets can be relatively easily re-charged
      - MANET nodes have to perform both routing and end-system functions; unlike LAN/WAN/MAN nodes - imposing heavy power requirements on MANET nodes

  ➢ ***Example management challenges: Network monitoring for fault and performance management (self, QoS, SLA), limitations on security management operations (that are typically CPU intensive) while balancing need for enhanced security (classified/unclassified nature of information transfer)***

Management of MANETs: Unique characteristics & Challenges - continued

- Intermittent Connectivity caused by
  - Stochastics of the environment (foliage, pathloss)
  - Random mobility
  - Limited battery power (thus nodes may 'die' suddenly)
  - Hostile jamming

  Note: Above causes frequent node/network element 'disconnects'

  ➢ ***Example management challenges: network configuration/reconfiguration, fault diagnosis & self-healing, security/trust management, performance assurances (QoS, SLAs)***

- Varying Security Requirements
  - Dependent on mission needs certain nodes can not and should not communicate with others

  Note: This causes 'network' partitions, which when combined with 'dynamic topology', results in 'network wide' implications

  ➢ ***Example management challenges: security management, performance management (due to security management overheads), configuration management (due to need for separate network configurations)***

Management of MANETs: Unique characteristics & Challenges - continued

- Scarce Bandwidth
  - Due to a combination of harsh environment, mobility and hostile jamming, bandwidth in MANETs is typically a very scarce (and dear) quantity.

  - ➢ Management Implication: MANET Management system design becomes a nightmare ☹; Management operations 'consume' bandwidth; bandwidth consumption increases as 'uncertainty' of underlying network increases; how can one afford 'good' network management when the 'bit-pipe' is not available??

Summary of MANET management requirements

- Minimal use of bandwidth
  - Given the bandwidth paucity in MANETs, and the fact that the MANET has been deployed to provide a transmission medium for critical mission information transfer, it is imperative that MANET management operations use very little bandwidth.
- Survivability & Self-healing
  - Given the stochastics/unpredictability of MANET environment that in turn cause nodes to ‘disappear’, it is imperative that the management system be robust and resilient to network fluctuations, incomplete and erroneous information.
- Automated Reconfiguration
  - Given the need for frequent reconfiguration in MANETs, the management system must support automated (re)configuration to (a) improve network performance by enabling rapid responses t onetwork problems and (b) reduce amount of human intervention to manage the network

Summary of MANET management requirements - continued

- Quality of Service Management
  - Given the fact that MANETs must support diverse applications ranging from mission critical (platinum), to priority (gold) to routine (bronze), the management system must be capable of differentiating traffic and providing quality of service, based on misison requirements (i.e., ensure most critical traffic get preferential treatment, esp. when bandwidth fluctuates and dwindles)
- Scalability
  - Given that MANET sizes can vary from a handful to thousands of network nodes, the management system must be architected to be scalable

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - Case Studies
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Fault Management: What is it, and what is it's role in MANETs?

- Fault Management – deals with monitoring, diagnosing and recommending solutions to network failures. More specifically, once the MANET has been configured, it is the fault management's responsibility to:
  - Monitor network elements to ensure that network elements and services are functioning correctly
  - Detect any network failures and collect evidence of the malfunction
  - Provide automatic diagnostic techniques to pin-point the root cause of the problem
  - Provide self-healing mechanisms that can restore services in a seamless manner
  - ➢ Fault Management, in essence, is vital to maintaining the 'health' of the MANET
- Role of Fault Management (FM) in MANETs:
  - Heightened due to the unpredictable nature of MANETs
  - The stochastics combined with the fact that MANETs are often deployed in remote or hard-to-reach-into areas, underscore
  - ➢ Require automated fault diagnostic and self-healing mechanisms

Fault Management - Operations Models: Quick Overview

- To assist with the key FM tasks that involve root-cause analyses and self-healing, several operational models have been developed in the Industry
- The TMN model of fault management operations developed for telecommunications networks (as discussed in the next few slides) is the most popular
- Due to the fundamental differences between MANETs and telecom networks, the TMN model can not be used as-is for MANETs
- However, much can be learnt from the TMN models, which can then be 'adapted' for MANETs (as will be discussed next)

Fault Management - Operations Models : Traditional (TMN) view:

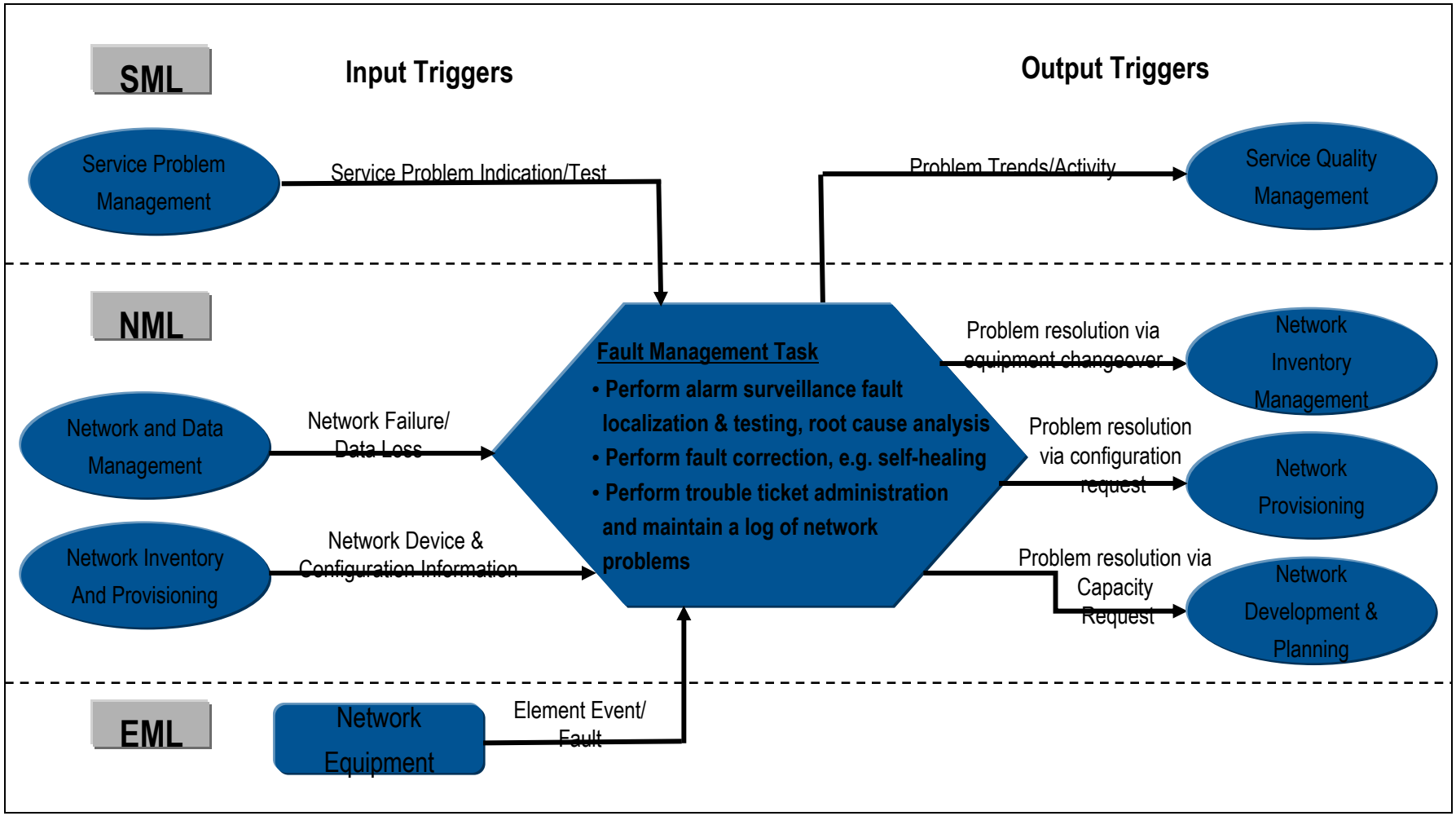

Figure 1: FM operations model – TMN model

Fault Management - Operations Models : Traditional (TMN) view - continued

Salient points

- FM operations organized around the service management layer (SML), network management layer (NML) and element management layer (EML) concept
  - SML: layer at which 'telecom operators' function – e.g., service operators periodically look out for 'service problem indications' and may have to issue periodic 'test' triggers; typically very large scale in terms of geographic scope
  - NML: layer at which 'networking' actions take place; typically moderate scale in terms of geographic scope, i.e., NML scope is across several routers within a 'network domain'
  - EML: layer at which the 'packet forwarding' related actions take place; typically limited scale in terms of geographic scope, i.e., EML scope is within a network element – e.g., within routers, switches, transcievers, ..
- Automated network diagnostics at the 'lower layers' (EML, NML)
- Human-in-the-loop (HITL) to assist with trouble ticket administration and recovery (SML)
- HITL coupling with other management operations

Need for new operations models

- Layer Classification
  - While OK for telecom-type of networks, concept of SML with several ‘operators’ will not translate to MANETs
  - However can still use the “SML” concept to service assurances via self-healing and quality of service guarantees that are ensured via ‘automated’ (vs. HITL) system policies
- Stove-piping of network management operations
  - The network management functions (e.g., FCAPS operations) are not tightly coupled with each other
  - While this may be OK in a telecom environment (due to a high degree of HITL), this is certainly not the case in MANETs, where the FCAPS operations have to link in with each other
    - E.g., Self-healing that requires adaptive network re-configuration, calling for close operations models between fault, performance and configuration management, and potentially security management as well

Need for new operations models - continued

- Response to network faults
  - Notice the heavy involvement of network operators (via trouble ticket and service repair operations) in the telecom (TMN) model; contrast this with the need for automated failure recovery to the maximum possible extent in MANETs
    - Recall, MANETs are often deployed in remote places and/or hostile environments, where it is not feasible to have HITL.
    - The above calls for more autonomous self-healing mechanisms that can discern failures amidst noisy (stochastic, erroneous) environment and initiate self-healing recovery actions, via a policy framework, with very minimal (and ideally, no) HITL.

Fault Management - Operations Models: Adapted for MANETs

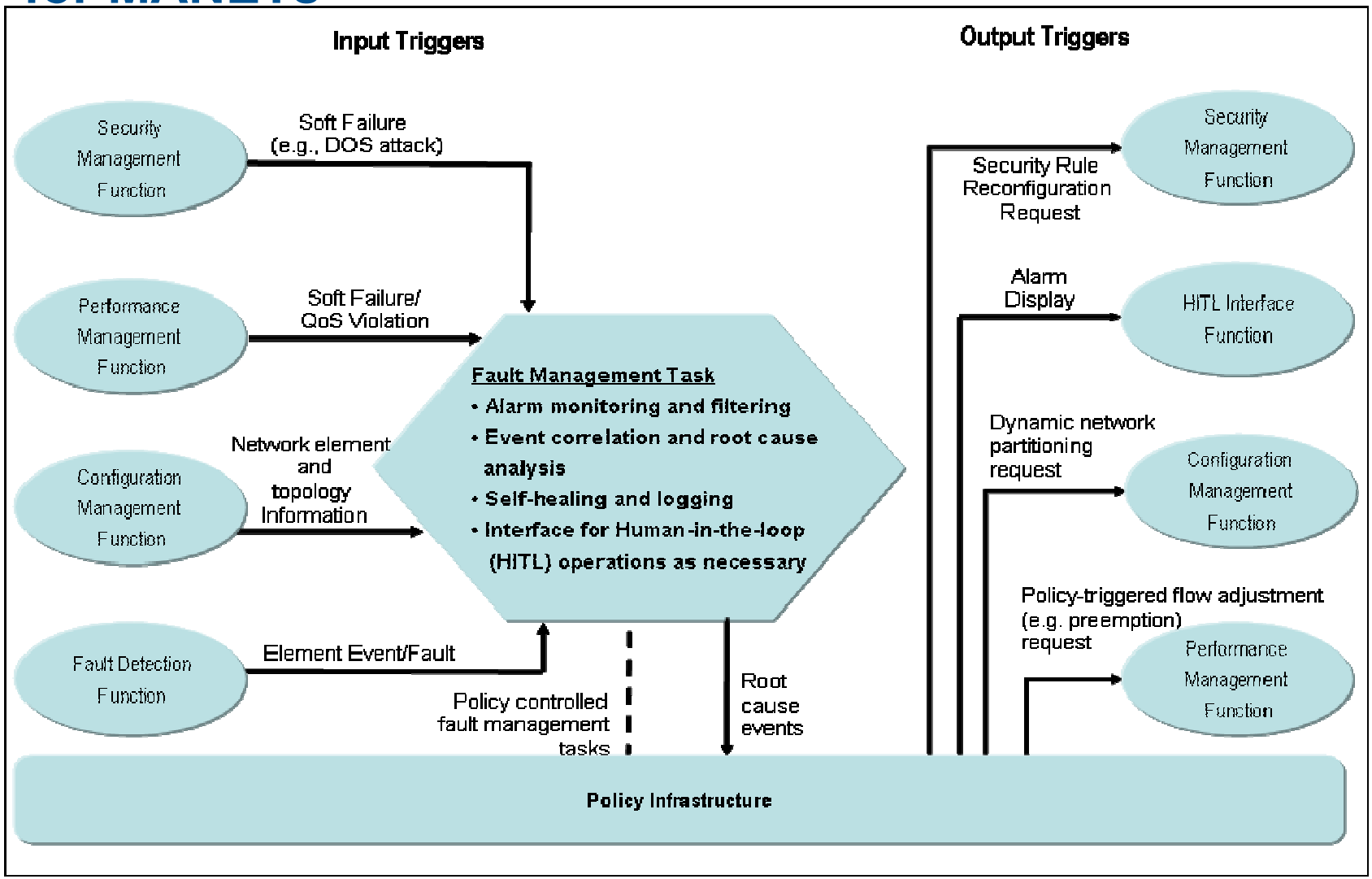

Figure 2: FM operations model – MANET model

Fault Management - Operations Models: Adapted for MANETs (continued)

Salient Points

- The fault management operations model has strong dependencies and is integrated with the operations models for configuration, performance and security operations
- A policy framework 'knits' together various network management functions (fault, performance, configuration and security)
  - Input triggers (shown in preceding slide)
- The self-healing operations are automated via policies, to enable the critically needed rapid responses in the dynamic MANET environment.
  - The policy framework also provides for HITL-knobs, bearing in mind the need to have a human over-ride capability in any automated system☺
- Sample 'output' triggers (shown in Figure 2 - preceding slide)
  - E.g., the output trigger 'dynamic network partitioning request' to the configuration management system is generated via the policy framework in response to a root-cause event, to provide self-healing via 'healing' around a failed/fractured network segment; or even perhaps, an 'un-trusted' network segment

Automating Fault Management Operations in MANET via Policies

- Policies used to couple fault management with other management functions and provide an extensible framework for the critically needed automation in MANETs

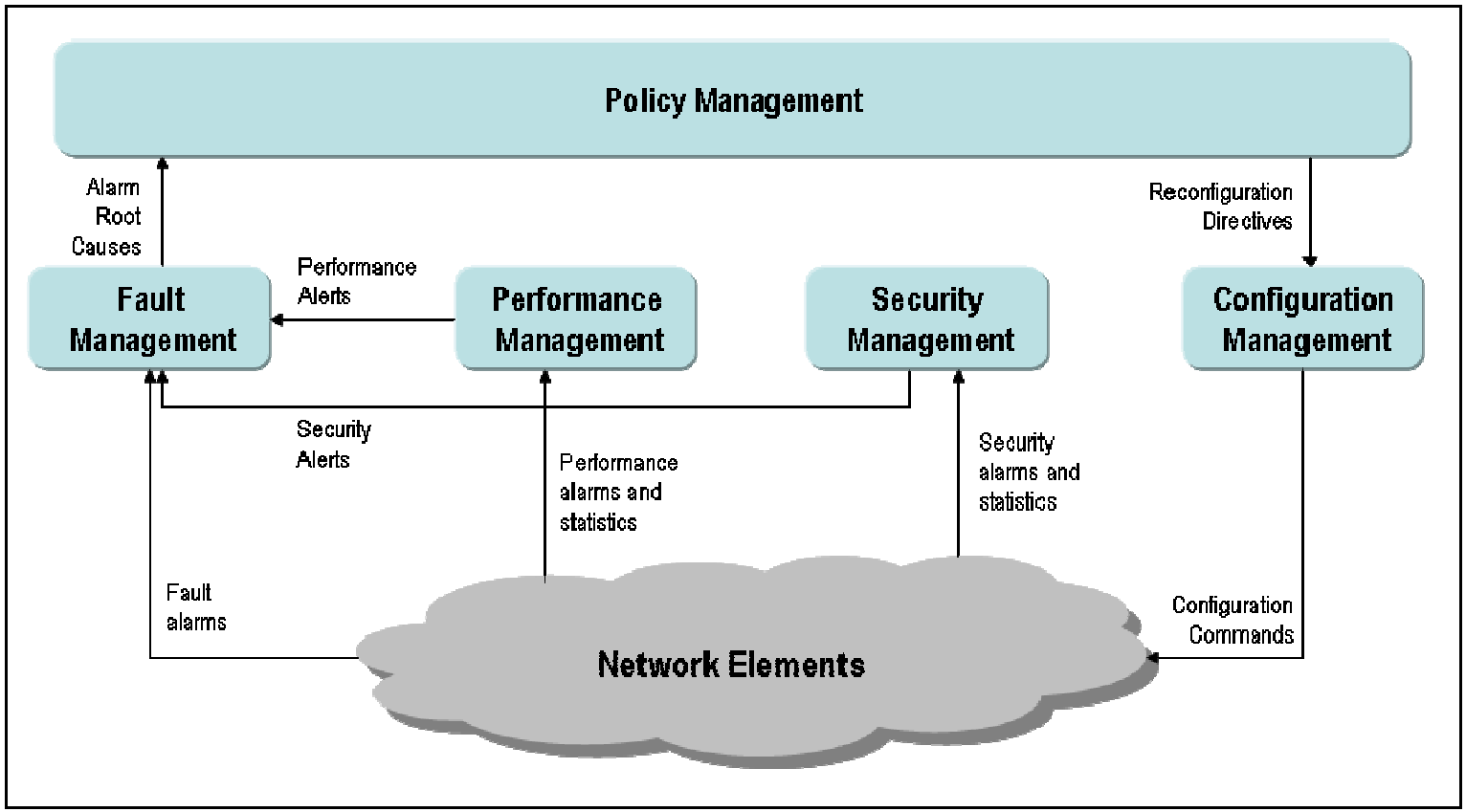

Figure 3: Automating FM via policies

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - Case Studies
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Categorization of Failure Types in MANETs

Failures in MANETs can be broadly classified as: (a) Hard Failures and (b) Soft Failures

- Hard Failures:
  - A *hard failure* is defined as a failure that is associated with equipment – typically an equipment failure or malfunction.
  - The process of pin-pointing the reason for hard failures, namely, the root cause analyses of hard failures - is typically deterministic in nature, i.e., the fact that an equipment is broken or not is diagnosed with probability 1.
- Soft Failures
  - A *soft failure* is defined as a failure that is associated with performance degradation.
  - Examples of soft failures include service degradation due to excessive loss and/or delay.
  - Soft failures are typically stochastic in nature – i.e., the "root cause" of any particular soft failure is difficult to diagnose with absolute certainty, and therefore the probability that a given root cause diagnosis for a soft failure is correct lies somewhere between 0 and 1.
  - For example, the root cause explaining why packets are being dropped in large numbers could be (a) excess application traffic, (b) a denial of service attack that floods with network with traffic, (c) equipment malfunction that causes excessive retransmissions, and so on.

Categorization of Failure Types in MANETs - continued

- Failures in MANETs largely tend to belong to the soft failure category due to the fact
  - MANETs by their very nature are stochastic – in part because of the unpredictable environmental conditions and mobility of the MANET nodes, and in part because of the scarcity and variability of bandwidth resources.
- Contrast this to wireline networks, which are rarely subject to soft failures and have significant over-provisioning of network capacity.
  - Additionally, wireline networks typically function under benign environmental conditions, as they are typically not subjected to adverse environmental effects such as temporal fading that produce fluctuating bandwidth, or mobility changes that result in variable network topology.

Root Cause Analysis (RCA) - what is it? Why is it important?

- Root cause Analysis (RCA) can be described as
  - The ability of the Fault Management (FM) component to collect, filter and co-relate alarms that are generated by the network, and
  - Pin-point the underlying cause of the problem that caused the generation of alarms, and subsequently trigger self-healing actions

➢ RCA, in essence, is the central piece of FM operations, and is crucial in maintaining the 'health' of the underlying network – for only when the root cause has been identified, can appropriate 'service recovery' or 'self-healing' actions be triggered.

Root Cause Analysis (RCA) - why is it challenging?

- A single failure can trigger a plethora of alarms. For example, consider a network problem such as “Link Failure”. In turn, this may trigger the following types of alarms
  - “Interface down” alarm from nodes using the link
  - Performance-related alarms due to the resulting network congestion
  - Networking operation-related alarms resulting from a network partition
  - ….
- The FM system should be able to filter duplicate alarms, co-relate them and then apply suitable algorithms that can ‘infer’ the root cause of the problem being observed.

RCA and MANET specific challenges

- Frequent and sporadic hard failures – caused by unpredictable environment conditions
  - Excessive ambient temperature, jamming, ..
- Frequent random soft failures – caused by the stochastic nature of the MANET; e.g.,
  - Failures caused due to scarce and varying bandwidth, network partitioning, …
- Frequent transient failures, e.g.,
  - Nodes going to sleep to conserve power
- Rapidly varying network topology
  - This causes problems while constructing ‘network dependency models’ (discussed next) in order to perform RCA

Overview of start of the art in Wireline RCA

- Significant work exists to-date in the area of fault diagnosis/RCA – but mainly for wireline networks
  - Useful to review these technologies so as to not re-invent the wheel, but with an understanding of their limitations in order to identify MANET extensions
- In summary, bulk of existing fault diagnosis/RCA methodologies for wireline networks are geared toward:
  - Diagnosing faults related to network connectivity at physical layer, e.g., faulty cable, faulty interface to/from cable,
  - Diagnosing hard failures – e.g., transmitter/receiver down, router down, …
  - Single network failures
  - References:
    - Wang & Schwartz[1993], Nygate[1995], Katela & Schwartz[1995], Yemini et.al[1996]

NEtwork Dependency Models (NEDMs) and their role in Fault Diagnosis (Wire-line Networks)

- To perform fault diagnosis, the FM (Fault Management) sub-system maintains information about the relationship between various network elements in the form of a NEtwork Dependency Model (NEDM)
  - E.g., information about the number and type of network elements (routers, switches,..) and how they are interconnected (i.e., mesh, star)
  - NEDM can be viewed as a graph where the vertices represent the network elements (routers, switches) and the edges represent interconnectivity (direct links) between the network elements
- The NEDM, together with a set of fault diagnostic rules are used by the FM system to perform root cause analysis & subsequent self-healing
  - Information about individual elements and relationships between network elements is required to make inferences both about the types of faults and in tracing a given type of fault to a root cause
- The amount of information maintained and the complexity of the NEDM determine the granularity of the faults that can be analyzed

NEDM construction

- In today's (wireline) networks, detailed NEDM info is restricted to the lowest OSI layer (Physical Layer – PHY) with limited info about the data link layer (DLL), and are constructed as follows:
  - Information about core network elements (routers, switches) are obtained by the FM system by
    - Interfacing to the inventory system, and/or
    - Employing special auto-discovery mechanisms – e.g., routing protocol neighbor discovery
  - ✓ As a specific example, consider the representations of routers
    - The vertices of the NEDM graph will each represent the router
    - Example info associated with each vertex (router) include
      - Router type, number of active inbound and outbound interfaces, number of passive inbound and outbound interfaces.
  - Next step is to gather information about the edges that connect a vertex-pair (direct links between the routers). Most of the FM systems today obtain this info by
    - Interfacing with the inventory management sub-system, or,
  - ✓ Example info about edges (direct links) in today's systems include information about the type of link – e.g., fiber, twisted-pair, Ethernet, etc., and its speed.

NEDM construction - continued

- Following pseudo-code describes the NEDM construction in today's systems
  - Let *n* represent a node number, i.e., *1≤ n ≤ N*
  - Let the tuple **R**<*T, NAI NPI, NAO, NPO*> be used to denote the information associated with a router, where:
    - *T = Router type*
    - *NAI = Number of active inbound interfaces*
    - *NPI = Number of passive inbound interfaces*
    - *NAO = Number of active outbound interfaces*
    - *NPO = Number of passive outbound interfaces*

For each $n \in N$,

- Consult the configuration management system or use auto-discovery to obtain information about the details of $Router_n$.
  - Construct the 5-tuple $\mathbf{R}^n$ $_{<T, NAI, NPI, NAO, NPA>,}$ with the superscript *n* denoting the information associated with $Router_n$. This 5-tuple is used to represent $Router_n$ as a vertex in the dependency graph.
  - Store in system database.
- Based on the information obtained from the previous step, determine the number of active interfaces for $Router_n$.
  - Denote the number of active interfaces by the number *K*. Let *k* be an integer such that *1≤k ≤ K*.
- For *k=1..K* do the following:
  - Get the name of the router and the specific interface number that this link terminates
  - Store this information in the system database.

NEDM construction – continued: Example

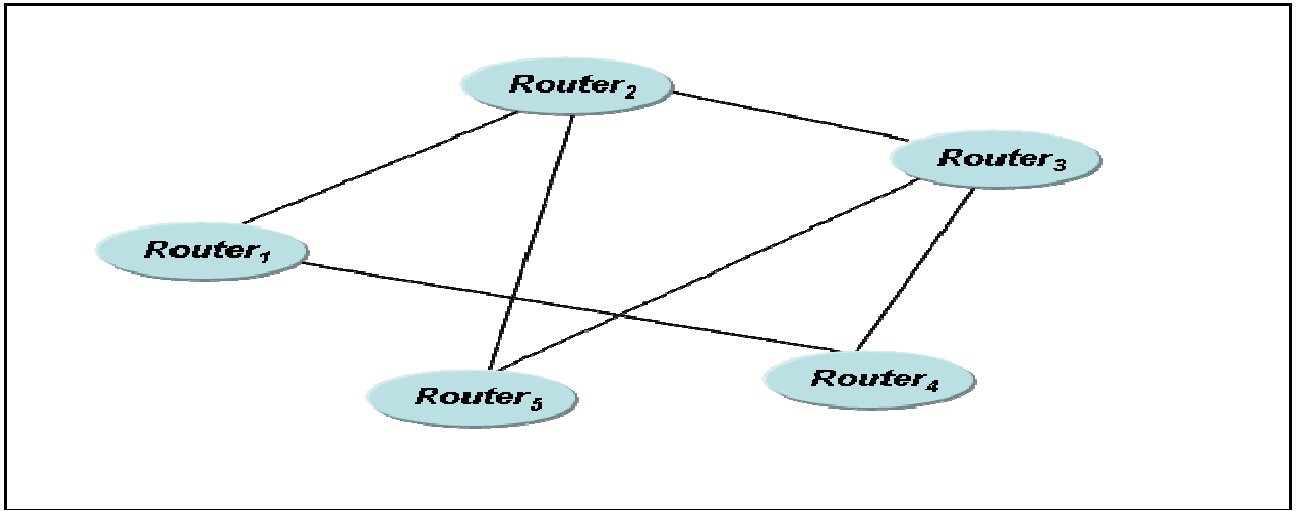

Figure 4: FM - NEDM construction

NEDM = (*NE*, *DL*), where *NE* represents the set of network elements (routers, switches) and *DL* represents the set of direct links between pairs of network elements.

Here

- $NE = \{\mathbf{R}^1_{\text{<rtr\_type, 2, 3, 2, 3>}}, \mathbf{R}^2_{\text{<rtr\_type, 3, 2, 3 2>}}, \mathbf{R}^3_{\text{<rtr\_type, 3, 2, 3 2>}}, \mathbf{R}^4_{\text{<rtr\_type, 2, 3, 2 3>}}, \mathbf{R}^5_{\text{<rtr\_type, 2, 3, 2 3>}}\};$
- $DL = \{L_{R1\_R2}, L_{R1\_R4}, L_{R2\_R1}, L_{R2\_R3}, L_{R2\_R5}, L_{R3\_R5}, L_{R3\_R2}, L_{R3\_R4}, L_{R4\_R1}, L_{R4\_R3}, L_{R5\_R2}, L_{R5\_R3}\};$

The superscript on R denotes the router number; and L_{Rx_Ry}, represents a link between $Router_x$ and $Router_y$.

- Note that the link speed is also stored along with the link endpoints as part of the information in an *NEDM*. Also notice that L_{R1_R2} and L_{R2_R1} have been listed explicitly to accommodate asymmetric links (although in most traditional wireline networks, bidirectional links are usually symmetric).

NEDM and Topology Map – Similarities & Differences

- Observe the parallel between an NEDM and a network topology map
  - Network topology map, in essence, captures node connectivity in a network
- Above analogy is valid but with the following caveat
  - Unlike a network topology map in which no detailed info about node types is maintained, an NEDM constructed for fault diagnostics is designed to maintain details both about the node type and link type, as seen in the sample (previous slide)
    - E.g., while a network topology map basically stores the existence of a direct link between $Router_1$ and $Router_2$, the NEDM stores additional information as in the 5-tuple $\mathbf{R}^i$ <T,NAI, NAI, NAO,NPA>

- Thus in essence, the *NEDM* contains information about
  - the network topology (i.e., which node has a direct link with which other node in the network),
  - the capabilities of each of the network nodes (in terms of the types of router and the input/output interfaces), and
  - the link speeds of the direct links that interconnect any two nodes.

- With the help of such an *NEDM* and fault diagnosis rules, the fault management component can then perform root cause analysis (explained next)

NEDM : Summary

Salient Points for Wireline NEDMs

- *NEDM* can be thought of as a graph that captures the physical layer relationships between network nodes, by using vertices to represent the network elements (e.g., routers, switches), and edges to represent the direct communication link (fiber, twisted-pair, etc.) between any two vertices.
- The connectivity between the vertices (or network elements) is known *a priori* and is relatively static.
  - Note: Connectivity is not completely static is because backup equipment may come on-line as a consequence of self-healing based on automatic protection switching,
- The properties of the *NEDM* are known and fixed. For example, the router types do not change, and neither do the wireline link speeds as well as connectivity.
  - Thus the dependencies in terms of the edge definitions between two vertices (i.e., links between two routers) also do not change frequently.
  - Network links may change when network expansion takes place – e.g. when new cables are being laid – but this is a rare event and is typically accompanied by a great deal of manual planning, which includes updates to dependency graphs.
- *NEDM* construction is a fairly static process. Changes to the wireline *NEDMs* are typically performed on rare occasions as a result of planned network expansions.
- Wireline *NEDMs* are well suited to handle:
  - Hard failures (e.g., cable disconnect, equipment malfunction) rather than soft failures (performance threshold violations).
  - Single fault types, or at most, multiple instances of the same fault type, rather than multiple heterogeneous network faults.

NEDMs & RCA : Role in Fault Diagnosis & Self-healing

Best illustrated via an example

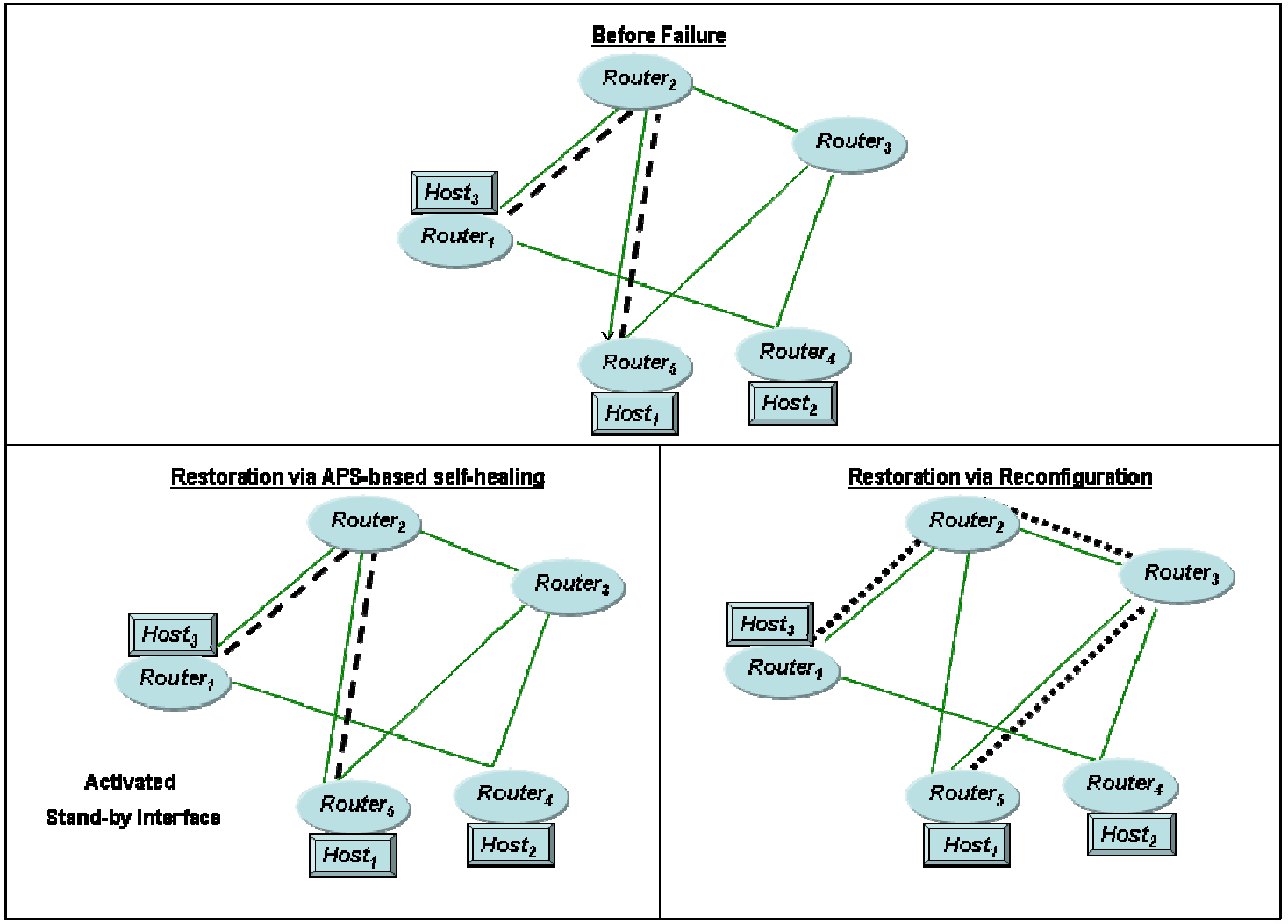

Figure 5: FM – NEDMs and RCA

Fault Diagnosis (RCA) & Self-Healing with NEDMs

- Figure 5 (previous slide) shows a simple 5-node network
  - Figure on top labeled “before failure” used to illustrate the network during ‘normal’ operations
  - Figures on bottom used to illustrate the network after fault diagnosis and self-healing. More specifically, illustrate two types of self-healing
    - Via Automatic Protection Switching (APS) – which uses the concept of back-up equipment that can be “switched-into” the network in the event of a failure
    - Via Network Reconfiguration – which does not use the concept of idle equipment dedicated for failure handling purposes – rather switches around the virtual circuits (VCs), be they PVCs (permanent) or SVC (switched) VCs (e.g., ATM type of networks), or MPLS (multi protocol label switched) paths; where the VCs and LSPs (label switched paths) are usually established *a priori* during network planning
- Failure scenario illustrated corresponds to the failure of the transmitter on $Router_5$ on the link between $Router_5$ and $Router_2$.
- The following steps (next set of slides) will occur to accomplish fault detection, diagnostics (root cause analysis) and recovery (self-healing)

Fault Diagnosis (RCA) & Self-Healing with NEDMs - continued

1. An alarm is issued by the element management system at *Router5*, indicating a transmitter failure.
2. Host attached to *Router5* notices that it is not receiving any packets from the host attached to *Router1* and issues a trouble ticket.
3. The fault management system receives the alarm from *Router5*. To reduce time lags, most current wireline fault management systems do not wait for trouble tickets to be issued, but automatically trigger pre-defined rules that use the information in a NEDM to attempt self-healing as described below. If the automatic recovery action fails, then a ticket requesting manual assistance is opened. There can also exist some cases where the end customers notice the problem before the system generates a ticket, and call the service provider personnel to open a trouble ticket. In such a case, the fault management subsystem will try to initiate an automatic recovery action, and, should the problem be resolved, it will send a clear notification to the trouble ticket system.
    1. The fault management system uses the *NEDM* information together with a set of pre-defined fault diagnosis/self-healing rules to automatically attempt a self-healing action. In our example, the following set of rules will be invoked as part of fault-diagnosis and self-healing:
        1. Check to see if there is a spare transmitter that it can use for the failed interface. This is done by checking the *NEDM* and the associated router details, namely, **R** *<T, NAI NPI, NAO, NPO>*,
            1. If so, perform an automatic switch-over (this is an example of APS) and update the *NEDM*. This involves updating the information on the number of active and passive interfaces in the *NEDM*. The network diagram in the lower left hand corner in previous slide, labeled "Restoration via APS-based self-healing" illustrates this type of self-healing action.

        Note: In this case, the communication path through the network between $Host_1$ and $Host_3$ will remain unchanged before and after the failure and is shown via the dashed lines in our example (previous slide). Furthermore, the switch-over occurs at the element management layer (EML) and seldom has to reach the network management layer (NML) per the TMN nomenclature.

        1. If not, then do the following:
            1. Consult the *NEDM* to see if there exists another active transmitter on another edge (i.e., another active interface on another link). In our example, this corresponds to the link between $Router_5$ and $Router_3$.
            2. If yes, query the configuration database in order to determined whether another set of links can be used to recover from the fault. More specifically, based on the identified link from the dependency graph together with the current list of PVC, SVC, or LSP information stored in the configuration database, trigger PVC/SVC/LSP configuration, as applicable. In other words, a new PVC, SVC, or LSP, as the case may be, is configured such that the default route between $Host_1$ and $Host_3$ now uses $Router_3$ and $Router_2$ as its transit nodes. Additionally, make the following updates:
            3. Update the list of currently active PVCs/SVCs/LSPs to reflect the addition of the new route via *Router3*, and, to delete the route $Router_5$-$Router_2$-$Router_1$.
            4. Update the *NEDM* to: (a) reflect the new number of active and passive interfaces associated with $Router_5$ and (b) remove the link between $Router_5$ and $Router_2$ to reflect the fact that the link between $Router_5$ and $Router_2$ is unusable since there is no active transmitter connecting the link.

        Note: In this case, the communication path through the network between $Host_1$ and $Host_3$ will change. The changed path through the network is shown via the dotted line in the lower right hand corner of the network that is labeled "Restoration via Reconfiguration" in our exmaple (previous slide)

            1. If not, issue a trouble ticket with a requirement for manual intervention.

Fault Diagnosis (RCA) & Self-healing with NEDMs - continued

Points to note on the previous example (Figure 5)

- After the switch-over (via APS) is made to a passive (standby) transmitter, there is an action to update the *NEDM*, namely, an action to reflect the latest number of active and passive transmitters within the affected router ($Router_5$).
  - This type of update is considered a passive update, i.e., an update that has not occurred due to changes in the network topology, but rather, an update to reflect changes within a network element.
    - More precisely, the number of active/passive transmitters within a router is updated in the *NEDM*.
  - Furthermore, the self-healing action that is triggered in this step is very simple – it is essentially an automatic switchover to another active transmitter within a router, and does not involve any changes to the relationships between routers.
  - Thus such a change does not require communications with other network management functions such as configuration management or performance management.
  - Above is in contrast to self-healing via re-configuration (discussed next) which requires more complex actions and updates.

Fault Diagnosis (RCA) & Self-healing with NEDMs - continued

Points to note on the example in Slide 33 (Figure 5) – continued

- The non-APS option also involves updates to the *NEDM*, i.e., the number of active and passive transmitters change for $Router_5$.
- In addition, another required update is the deletion of the link in the *NEDM* between $Router_5$ and $Router_2$.
  - This link has to be deleted to reflect the absence of a functioning link between *Router5* and *Router2*, since there is no transmitter transmitting on the physical link any more, and thus there is no neighboring relationship between *Router5* and *Router2* in the NEDM.
  - Furthermore, the self-healing actions involved in this case are more complex than in the previous case, because the fault management system has to work with the configuration management system to establish new PVCs/SVCs/LSPs (shown by the dashed lines for *Host1* and *Host3* in Slide 33).
    - Recall that *a priori* routes are typically realized via establishment of PVCs, SVCs or MPLS LSPs during system startup and stored in the configuration database.
- Changes to the routes will, however, require a certain amount of time to implement the reconfigurations and may also necessitate a HITL (human in the loop).
  - The actual reconfigurations themselves will likely involve human intervention, to ensure that the new PVCs/SVCs/MPLS LSPs created are available for use and have no “route looping” issues.

Fault Diagnosis (RCA) & Self-healing with NEDMs - continued

Points to note on the example in Slide 33 (Figure 5) – continued

- Thus such a non-APS a recovery action is not 'typically' preferred. It is included in the discussions here to serve the following dual purpose:
  - To highlight the need for an integrated and policy-driven system network management system for MANETs, and
  - To point out a potential alternate course of action for current network management systems.

➢ Such a self-healing action (i.e., via re-configuration of the virtual circuits) is an excellent example that underscores the need for

- Establishing more flexible fault diagnostic techniques that can cope with uncertainty (changing network links) without (or with minimal) HITL to work in a MANET environment
- Designing an *integrated* (i.e., a non-stovepiped) network management system for MANETs.
- An adaptive policy-driven management system that can, in real time, integrate the various NM operations without the need for HITL

Fault Diagnostic Techniques for MANETs and the need for enhanced RCA

Shortcomings of today's fault diagnostic techniques (albeit good for wireline networks) for MANET environments.

- Wireline networks have the concept of "core" and "edge" nodes with core nodes solely responsible for network services and edge nodes being the interface nodes between end-systems (hosts) and the underlying network.
  - In contrast, in MANETs, every node is both a core node (in terms of the network services it provides) and an edge node (in terms of housing an application/user/host).
  - Consequently, while fault diagnosis via NEDMs is typically restricted to the lowest layers of the protocol stack in wireline networks, in MANETs the fault diagnosis has to extend to all of the layers of the protocol stack and across all nodes.
  - For example: Consider the scenario where the link between *Router5* and *Router2* becomes congested (a very common problem in MANETs), and Hosts 1 and 2 complain about bad service (same example in Slide 33)
    - When a trouble ticket is issued, there is no information available for the fault management system to perform root cause analysis, other than the trouble ticket indicating that there is a service problem.
    - This is because the *NEDM* described in the preceding slides does not store nor convey any problem other than a physical problem (hard failure-related).
  - While this is a very simple example – i.e., it represents only a single soft failure, whereas in realistic MANETs, multiple simultaneous failures can occur

Fault Diagnostic Techniques for MANETs and the need for enhanced RCA - continued

Shortcomings of today's fault diagnostic techniques (albeit good for wireline networks) for MANET environments - continued

- Whereas existing techniques frequently use a deterministic model and assume that all dependencies and causal relationships are known with 100% certainty, the dynamic and unpredictable nature of MANETs underscores the need for stochastic dependency models.
  - The stochastic models for MANETs, unlike their deterministic counterparts for wireline networks, should not only be resilient to incomplete and imperfect information, but should also be amenable to dynamic changes in the causal relationships, as the MANET nodes move and change positions.
  - Additionally, even if the nodes do not move physically, they can effectively be "hidden" due to environmental conditions and/or hostile jamming.
  - Thus the NEDMs need to be dynamic in nature, to reflect the stochastics of the underlying MANET in their graph (frequently changing edges and vertices)

Fault Diagnostic Techniques for MANETs and the need for enhanced RCA - continued

Shortcomings of today's fault diagnostic techniques (albeit good for wire-line networks) for MANET environments - continued

- Whereas resource (e.g., bandwidth) availability problems are relatively rare in wire-line networks, in contrast, MANETs are plagued by resource scarcity.
  - Due to the dynamic nature of MANETs, available bandwidth is a fluctuating entity, with the available bandwidth swinging between extremes (namely, very low bandwidth availability to almost "full" bandwidth availability).
- The assumption of only one fault existing in the system at a given time, which is often made in wire-line root cause analysis techniques, is largely invalid in MANETs.
  - More specifically, the wireless root cause analysis techniques for MANETs must have the ability to deal with the existence of *multiple simultaneous faults of different types*.

➢ It is easy to see why the techniques in place for wire-line systems need significant changes in order to be used in MANETs.

Enhanced Fault Diagnostic Models for MANETs - Layered Models

- Fault diagnosis mechanisms for MANETs require dependency information among the various network elements *and network layers* to perform root cause analysis
  - Observe emphasis on italicized words
- Layered dependency models (as also discussed in [Gopal-2001], [Steinder & Sethi-2001]) are required to capture
  - Dependencies across the layers (OSI) within a MANET node
    - Refer to this as 'vertical dependencies'
  - Dependencies amongst MANET nodes potentially involving more than one vertical (OSI) layer
    - Refer to this as 'horizontal dependencies'

Layered Model for Fault Diagnosis in MANETs – What are they?

- The “vertical dependencies” in the multi-layer dependency model constructed for fault diagnosis purposes are very similar in philosophy to the layered OSI model, wherein services offered by a given OSI layer are a combination of the protocol(s) implemented at that layer and the services provided by it and the lower OSI layer(s). For example,
  - Consider the relationship between the lowest OSI layer, namely the physical layer (also denoted by *OSI_L1*) and the OSI layer just above it – namely the data link layer (DLL)/Media Access Control (MAC) layer (also denoted by *OSI_L2*).
    - The DLL/MAC layer on a node offers *access to the underlying medium* so that the node can send the information over the physical layer.
    - This channel access service offered by *OSI_L2* is in turn implemented via a specific MAC protocol (e.g., TDMA, CSMA etc.) and the services offered by the underlying physical layer that actually transmits the information bits.
    - The physical layer assumes the form of signal in space in wireless MANETs and the form of a cable, twisted-pair, etc., in wireline networks.

Layered Model for Fault Diagnosis in MANETs – What are they? - Continued

The layered dependency model in OSI is a recursive dependency model.

- Moving up the protocol stack, for example, consider the services offered by Layer 3, namely the network layer (also denoted by *OSI_L3*).
  - The services offered by *OSI_L3* are typically network connectivity-related services, such as establishing a communications path/routing path between two hosts by using routing protocols.
    - Some examples of MANET routing protocols include the Optimized Link State Routing (OLSR) protocol, and the Ad hoc On demand Distance Vector (AODV) protocol, to name just a few.
  - These routing protocols help to establish communications paths by constructing routing tables.
    - Routing protocols typically exchange routing messages to discover neighbors and learn the current network topology, in order to establish communications paths.
  - For routing messages to be transmitted from one node to another, routing packets have to contend for the right to transmit data.
  - It is here that the services of *OSI_L2* come into play.
    - As explained in the previous slide, the *OSI_L2* provides the *OSL_L3* with access to the underlying physical layer (*OSI_L1*), so that routing protocol messages can be exchanged successfully, in turn enabling the routing protocol to compute and establish routing paths between a set of interconnected nodes.

Layered Model for Fault Diagnosis in MANETs – what are they? - Continued

- Easy to follow this recursive relationship all the way up to the application layer (*OSI_L7*), which can be viewed as offering the use of a given network to a certain application.
  - The *OSI_L7* implements such a service via application level protocols that the user can use to invoke information exchange through the underlying network – be it a wireless MANET or a wireline network – and invokes the services of the layer below, namely the presentation layer (*OSI_L6*) to help format the user-generated messages for presentation to the underlying network.
- Such a recursive dependency relationship can be harnessed for fault management purposes as well, as illustrated in next slide.

Example Layered Dependency Model for use in MANET fault diagnosis

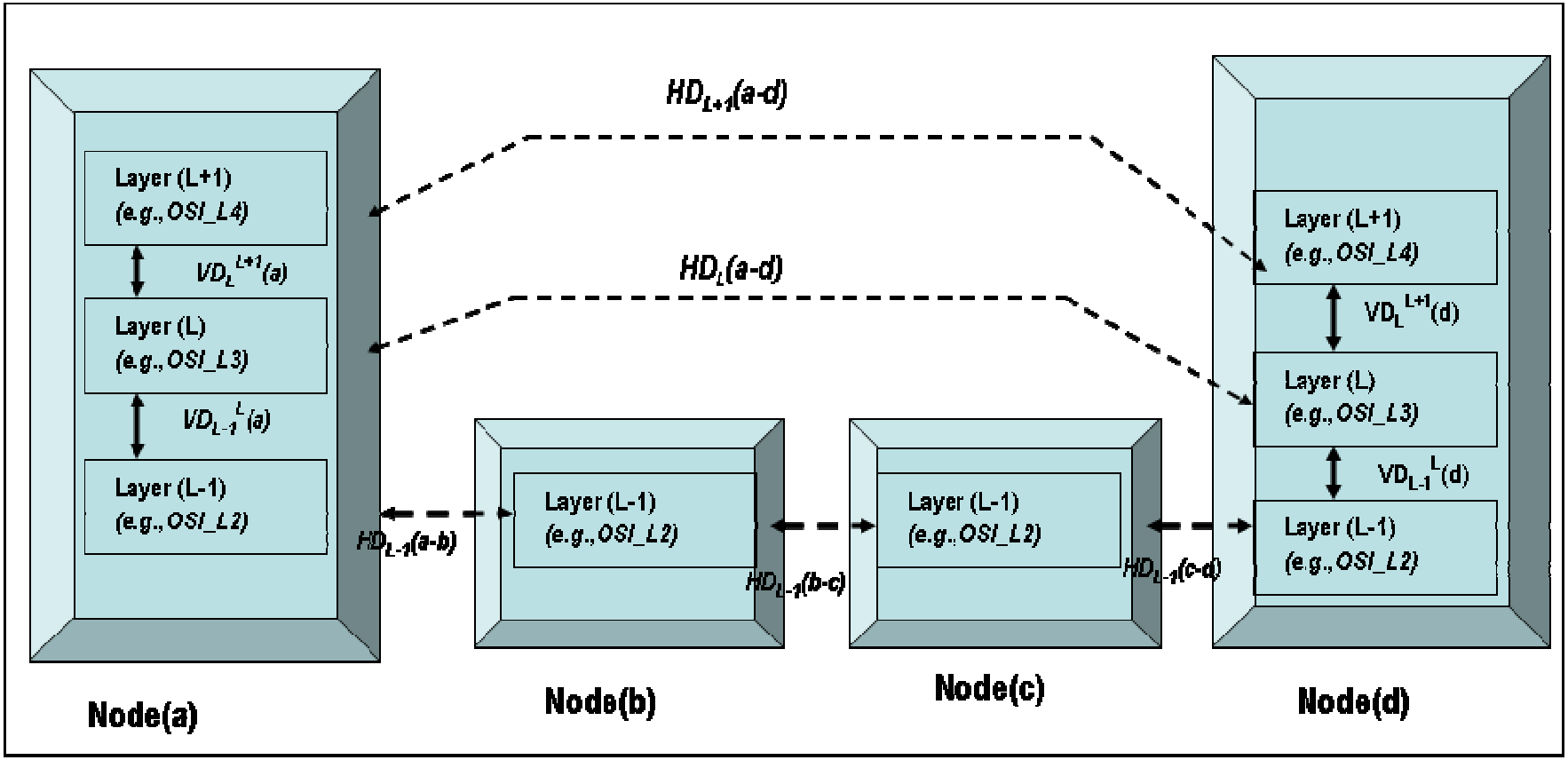

Figure 6: FM – Layered Dependency Model for MANET RCA

Layered Dependency Models: Vertical Dependencies

- In Figure 6 (previous slide) each of the nodes labeled Node(a), Node(b), Node(c) and Node(d) represents a MANET node.
- Figure shows three OSI layers (for illustration purposes) within Node(a) and Node(d), and one OSI layer within Node(b) and Node(c).
  - The topmost layer in Node(a) and Node(d), labeled Layer (L+1), is shown to correspond to Layer 4 in the OSI model, namely the Transport Layer.
    - The services offered by this layer typically correspond to either bit-oriented or connection-oriented transport of the application packets.
      - Examples of protocols used at this layer include UDP and TCP.
  - The layer labeled Layer (L) in Node(a) and Node(d) beneath the topmost layer in is the Network Layer (corresponding to OSI Layer 3).
    - As mentioned earlier, this layer offers networking services to the layer above it.
  - The layer at the bottom (labeled (L-1)) in all of the nodes corresponds to the Data Link/MAC layer (OSI Layer 2).
- The above *vertical dependency model* is based on the OSI concept of layering and captures dependencies *within* a node.
- While incorporation of vertical dependencies is a much needed enhancement of the *NEDMs*, in order for the dependency models to be useful for MANET fault diagnosis, they must also capture relationships and dependencies *across* nodes (discussed next)
  - Note: The horizontal dependencies provide important “network-wide” information since they capture the relationships between the intermediate nodes that carry an end-to-end service.

Layered Dependency Model: Horizontal Dependencies

- Since MANETs are almost never a fully connected mesh, application services originating from a source node will likely use the services of one or more *intermediate* nodes before reaching the intended destination node.
  - The term *transit* nodes will be also be used interchangeably to refer to the intermediate nodes in this book.
- The *horizontal dependencies* capture the peering relationships between the transit nodes along an application's end-to-end path.
  - Observe that failures, either soft (i.e., failure to provide an *a priori* "agreed-to" level of service as captured via a QoS requirement) or hard (i.e., element malfunction) can occur anywhere along the path or on an intermediate node.
  - In turn, such a failure will affect the end-to-end service and manifest itself as a fault symptom for further diagnosis by the fault management system.

Layered Dependency Model: Horizontal Dependencies - continued

Figure 6 shows how to accommodate both vertical and horizontal dependencies within and across communication nodes respectively.

- Solid lines are used to denote vertical dependencies within a node, and dashed lines are used to illustrate the horizontal peering relationships across nodes.
- With regard to nomenclature,
  - the vertical dependency that exists between Layer (L+1) and Layer (L) within Node(a) is denoted by VD_L^{L+1} *(a)*.
  - the horizontal dependency that exists at layer (L-1) across two nodes, say Node(a) and Node(b), is denoted by HD_{L-1}*(a-b)*.
- With regard to the nature of the horizontal dependencies between nodes in a MANET, observe that they (horizontal dependency) can be either logical or physical. For example,
  - Node(a) and Node(d) are considered to be logical neighbors (peers) from the standpoint of the OSI Layer (L+1) (i.e., Layer 4) due to the fact that the TCP/UDP packets originating from Node(a) terminate in Node(d) and are not inspected in the intermediate nodes Node(b) and Node(c).
    - Thus from the standpoint of the TCP Layer (Layer 4) at Node(a), its peering node is a corresponding TCP entity at Layer 4 in Node(d).
    - This horizontal peering dependency (i.e., logical neighboring relationship) at Layer (L+1) between Node(a) and Node(d) is denoted by HD_{L+1}*(a-d)*.
  - However, the horizontal dependency at Layer (L-1), i.e., HD_{L-1}*(a-d)* between Node(a) and Node(b) can be viewed as a physical peering relationship from viewpoint of OSI_L2
    - Since the layer 2 frames from Node(a) will be sent to Node(b) before being passed on to their next hop, OSI_L2 physical peer Node(c).
  - Similar recursive observations exist with regard to the fact that the Layer *(L+1)* at source node *Node(a)* will have a logical peer in the horizontal direction with destination node *Node(d)*, where Layer *(L+1)* can represent the Session, followed by the Presentation and then by the Application layers (i.e., *OSI_L5* through *OSI_L7*, respectively).

Layered Dependency Models and their use for fault diagnosis in MANETs

Qualitative Discussion

- Consider for example the following problem:
  - "Service between Node(a) and Node(d) experiences unusually large delays but is not completely down".
  - Since the service is not completely down, a hard failure that is typically associated with a network element failure is ruled out.
  - This corresponds to a soft failure.
- To diagnose the cause of the soft failure the layered dependency models can be used in the following manner
  - If it is known that the routers implement a congestion indication mechanism (e.g., Explicit Congestion Notification (ECN)), and the ECN bit has been observed to be set in received packets, then the unacceptable end-to-end delay experienced between Node(a) and Node(d) is probably due to packet drops at the network layer buffers.
    - Examples of corrective actions in this cause could include:
      - dynamic processor re-scheduling by readjusting the weights for Weighted Fair Queuing (WFQ) at Layer 3 (if WFQ is used), or
      - changing the token rates for the Layer 3 queues if token-based queuing is used, or
      - reconfiguration of routing policies so that less congested paths are used between the source and destination

Layered Dependency Models and their use for fault diagnosis in MANETs - continued

- If it is known that the routers implement a congestion indication mechanism (e.g., Explicit Congestion Notification (ECN)) and the ECN bit has not been set,
  - then the unacceptable end-to-end delay experienced between Node(a) and Node(d) is probably due to packet drops due to environmental conditions,
    - e.g., high attenuation between two links at the physical layer.
- In this case no amount of processor re-scheduling will help;
  - However if the environmental conditions are localized,
    - A corrective action could involve re-configuring the network to use alternate paths that minimize the effect of the environmental problems.

Summary of key differentiators in dependency models for MANETs and their wire-line counterparts

- The dependency models for wire-line systems typically only incorporate the details associated with the lowest OSI layer (i.e. the physical layer) and abstract layers 2 and above.
  - Consequently, fault diagnosis in wire-line systems is focused on analyzing problems at the physical layer.
- In contrast, in MANETs, the dependency models and fault diagnosis techniques must extend all the way up to OSI layer 7, as mentioned earlier.
- The horizontal dependencies in wire-line systems are relatively static, while the horizontal dependencies in MANETs are dynamic, with the relationships between nodes (i.e. the horizontal dependencies) varying over time.
  - While a small subset of the dependencies will be static in MANETs (e.g., the vertical dependency model that corresponds to the lower OSI layers), there is potentially a large set of dependencies that will be dynamic. For example,
    - Horizontal dependencies change as nodes move;
    - Vertical dependencies change as new services are deployed on-demand and/or new transport protocols are used based on the application type;
      - e.g. non real-time video may use streaming protocols like TCP, whereas real-time short-lived video sessions may use RTP/UDP-like protocols.
  - Thus, the fault diagnosis techniques for MANETs work with “snapshots” of the dependency models that get updated over time.
- While the dependency models themselves are deterministic in the case of wire-line networks, i.e., the information linking failures and symptoms is assumed to be known with 100% certainty, in MANETs there is a high degree of uncertainty in linking failures and symptoms, due to the plain fact that the information provided to the fault diagnosis system may itself be uncertain.

MANET RCA: Layered Dependency Models and their use in Probabilistic Inferencing

- Once a layered dependency model is constructed, or, more precisely in the case of MANETs once a snapshot of a dependency model for the network is constructed, it can be transformed into a *belief network* for use in fault diagnosis as done in [Steinder and Sethi 2002a] [Steinder and Sethi 2004b].
- A belief network is a directed acyclic graph *(G,P)*, where
  - $G=(V,E)$ is a directed acyclic graph.
  - $v_i \in V$ is a binary valued random variable and represents an event in the network.
  - $(v_i,v_j) \in E$ represents a causal relationship, i.e. v_i *causes* v_j.
  - $P=\{P_i\}$, where P_i is a probability associated with variable v_i.
- Associated with belief networks are *evidence sets*, where an *evidence set* denotes a partial assignment of values to variables represented in a belief network.
  - With the understanding that a variable v_i in a belief network represents an event, an evidence set essentially denotes assignments of a binary random variable (i.e., 1 (true) or 0 (false)) to the events, based on an observation of events that have occurred.
  - Observe also that this assignment, to begin with, will be partial (i.e., will not cover all of the possible events).

MANET RCA: Layered Dependency Models and their use in Probabilistic Inferencing - continued

- Next, given an evidence set, belief networks can be used to make two basic queries related to fault correlation as summarized below (and also described in [Steinder and Sethi 2002a] [Steinder and Sethi 2004b]):
  - Belief assessment, i.e., the task of computing the probability that some variable (i.e., events) possesses certain values (i.e., belief that the event is indeed is causing the observed symptom/problem).
  - Most probable explanation (MPE), i.e., the task of finding a complete assignment of values (beliefs) to variables (events) in a way that best explains the observed evidence.
- Once the belief networks along with evidence sets have been created, they can be used to make queries such as Belief Assessment and MPE as mentioned above, and thus be used in fault diagnosis.
- Recall that fault diagnosis is the process of ascertaining with a high degree of confidence (ideally with probability 1, i.e., with absolute certainty) the root cause of an observed symptom, in order to take subsequent corrective (self-healing) actions.
- The belief assessment and MPE essentially help in diagnosing the root cause by assigning a high degree of confidence (a high probability “value”) to an event (i.e., a “variable” in the language of belief networks) and thus hone in on the root cause.

MANET RCA: Layered Dependency Models and their use in Probabilistic Inferencing – continued

- Since belief assessment and MPE tasks are NP-hard [Garey and Johnson 1979] in general belief networks, approximations have to be performed to make them feasible for MANETs.
- Some of the most actively researched approximations include *Iterative Belief Propagation* and *Iterative MPE in polytrees* [Steinder and Sethi 2002a] [Steinder and Sethi 2002b] [Steinder and Sethi 2004b] that are based on adaptations of Pearl's iterative algorithms [Pearl 1988].
- While the details of the approximations require yet another tutorial(☺), these approximations in essence provide for tractability by using simplified belief networks that:
  - (a) employ binary-valued random variables and
  - (b) associate an inhibitory factor with every cause of a single effect (noisy-OR model) and assumes that they are all independent.
  - ✓ The approximations introduced by (a) and (b) essentially help contain the search space (or state space) of the system, in turn paving the path to "tractable" analysis.
- For example, use of binary-valued random variables vs. a continuum of random variables helps contain the values (0 or 1 in this case) that can be associated with a given event (i.e., a belief network variable).
- Likewise, by introducing the independence approximation in (b), the combinatorics associated with the problem space – namely, the set of possible states that the system can be in due to a given fault and a set of observed symptoms – is reduced significantly.

➢ In essence, all of the above approximations help achieve polynomial time complexity and tractability by placing less stringent demands in terms of computation time and memory requirements.

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - Case Studies
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Fault-Management & Self-healing - what is it, why is it challenging?

- Self-healing, in the context of fault management, is the ability to provide service survivability (uninterrupted service) by providing seamless restoration capabilities amidst random/sporadic network failures.
  - Necessity for self-healing capabilities in MANETs is both obvious and critical, since
    - MANETs are usually deployed on-demand in often hard-to-reach terrains (e.g., those used in NCW or emergency situations), and consequently will have very restricted human access
    - MANETs are expected to transport a wide variety of applications with a high degree of reliability (e.g., mission critical applications, emergency services)
- Key Challenges stem from
  - The ad-hoc nature of MANETs – i.e., unlike their wireline counterparts, there is often no 'structure' to MANETs
  - The unpredictable and even hostile environment
  - Random movement
  - Bandwidth paucity – resulting in need for 'low overhead, yet reliable' self-healing mechanisms (note: low overhead, high reliability are almost an oxi-moron in terms of network management system design)
  - Predominance of multiple, and soft failures
    - Contrast with wire-line networks, where failures are predominantly of the hard failure type and unlike soft failures, do not produce a 'cascade' effect (multiple failures)

Brief overview of Self-healing mechanisms for traditional networks

- Well-known self-healing systems that are used in practice today exist largely in the context of wire-line telecommunications systems.
- Widespread and commercial deployment of self-healing in wireless networks is still in its early stages [Kant et. al 2002].
- Majority of the well-known self-healing mechanisms in existence today function within a single layer of the OSI stack, namely the physical layer.
- In addition, they employ the philosophy of resource redundancy for handling failures resulting in a requirement for standby equipment that remain idle during normal working conditions – e.g.,
  - Certain pieces of equipment are dedicated to providing backup functions solely for the purpose of restoration/failure handling.
  - In the event of a network equipment failure, an automatic switchover occurs from the malfunctioning or failed equipment to the standby equipment
- For this reason, the widely used self-healing mechanism in place today is also referred to as an APS (Automatic Protection Switching) mechanism.
  - Examples of widely used APS self-healing mechanisms in use today are SONET self-healing rings, Bidirectional Line Switched Rings (BLSRs) [Wu 1992], etc.
  - Another specific example of APS in traditional wire-line networks is where the routers are deliberately not run at full capacity, and have several interfaces that are kept idle intentionally to serve as standby in case of failures.

Why Self-healing mechanisms used in traditional networks cannot be used in MANETs?

While the main advantage of an APS philosophy is the excellent response time (with a typical restoration delay of less than 50 milliseconds), it has several disadvantages from the perspective of MANETs:

- APS is very resource-intensive, since by definition it works on the principle of resource redundancy.
  - In fact, traditional APS systems not only devote spare interfaces but require idle equipment (e.g., spare routers) to serve as a standby in the event of an automatic switch-over requirement.
  - This is definitely an issue for MANETs, since they are typically resource-constrained.
- APS is limited to handling hard failures (i.e. equipment failures) alone.
  - Due to the stochastic nature of MANETs, it is anticipated that a substantial number of failures will fall under the category of *soft* failures, i.e. failures such as excessive performance degradation, poor signal quality, etc.
- APS does not distinguish between application types when restoring
  - In light of the diverse survivability and Quality of Service (QoS) requirements in MANETs, it may be counter productive to 'switch-over' all services, esp. when there is a scarcity of bandwidth and high priority (mission critical) applications need to be delivered in preference to low priority (routine) applications

Adaptive Policy-driven Self-healing operations in MANETs

- Employ a policy based self-healing mechanism that is sensitive to applications' survivability and mission reliability needs while restoring them
- Basic idea is simple – policies are used to define responses that should be taken by the entire network management system (NMS) to address a given root cause
  - Thus policies tie together the monitoring aspect of the NMS (via Fault and Performance Management) and the configuration and security aspects (via configuration and security management operaitons).
- Policy Terminology
  - Use Event-Condition-Action (ECA) policies to define responses to network events as follows:
    - The "Event" portion of the policy contains a root cause event;
    - The "Condition" portion of the policy contains conditions that need to be checked prior to performing a corrective action, if any; and
    - The "Action" portion of the policy invokes either
      - the necessary reconfiguration actions for dealing with the identified root cause, or
      - additional diagnostic tests to determine what corrective action should be taken.

Schematic of sample policy-based self-healing in MANETs

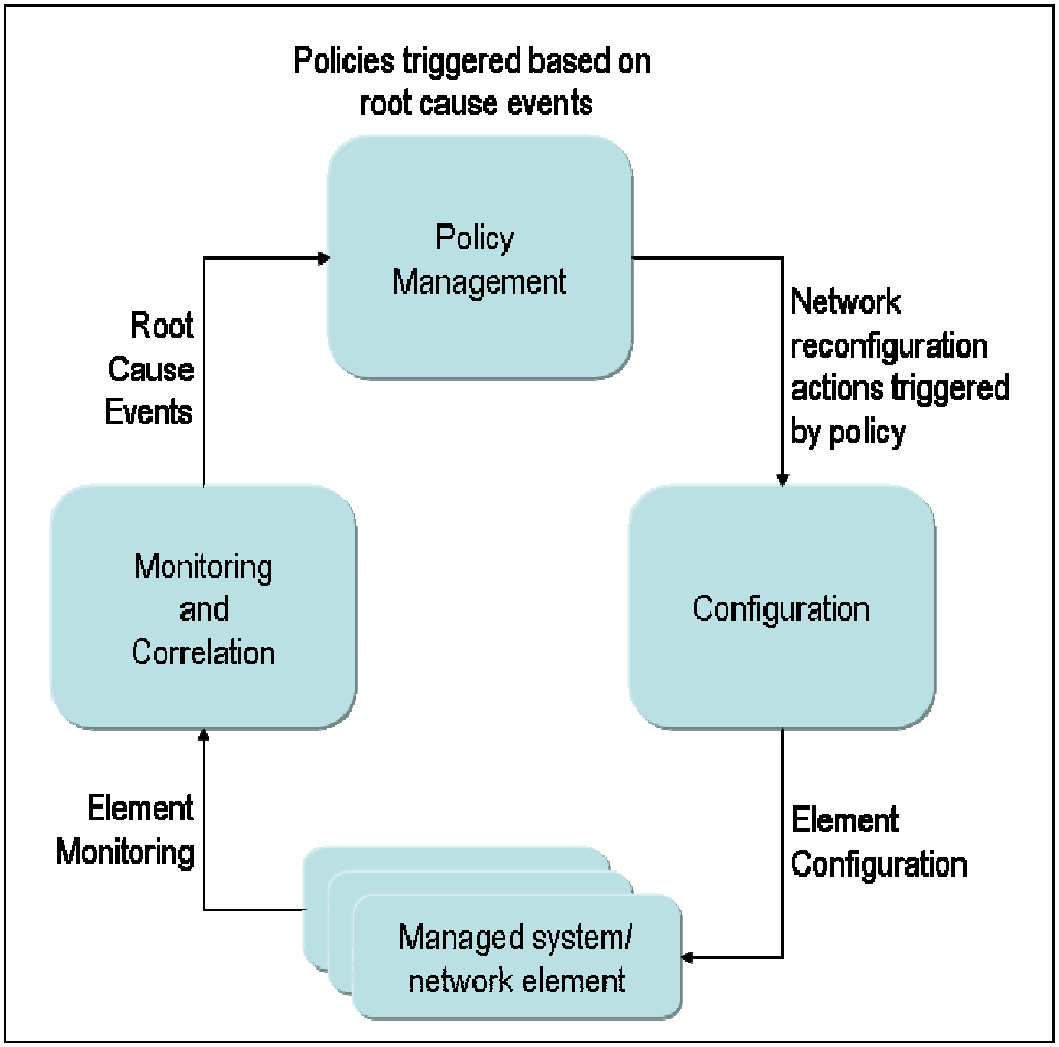

Figure 7: FM – Policies & Self-healing

Key take-away points with regard to policy-based self-healing in MANETs

- Policies are an ideal way to execute self-healing
  - They provide a convenient, human-friendly mechanism for controlling the behavior of the network management system.
- Network operators can specify how they want their network to be reconfigured in response to identified root causes
  - The reconfiguration is then automatically implemented as needed by a Policy Engine
- The policy framework provides an automated environment wherein fault diagnostic techniques used to pin-point the root cause of a problem* work together with appropriate recovery actions (self-healing mechanisms) that could potentially encompass other management operations. For example,
  - Invoke the performance management component to provide end-to-end QoS in response to a soft failure for a high priority application, or,
  - Invoke configuration management component to reconfigure and/or partition the network to isolate a mis-behaving or faulty node in response to a soft failure (service interruption)

*Recall: Fault diagnostic techniques involve the use of layered dependency models and belief analyses to pin-point the 'root cause' of the problem in the underlying network

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Self-healing in MANETs: Sample Case Studies – Overview

- Scenario 1: Radio Fault (hard)
  - MANET scenario describing radio failure and subsequent policy-based self-healing
- Scenario 2: Environment-related problem (soft)
  - MANET scenario in which MANET links become unavailable due to environment related issues (e.g., a node moves into a mountainous terrain, whereby adjacent nodes cannot communicate) and policy-based self-healing
- Scenario 3: Soft failure due to denial-of-service (DOS) attack
  - MANET scenario which spans security-related problems and resulting soft-failure

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
      - **Scenario 1**
      - Scenario 2
      - Scenario 3
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Self-healing - Scenario #1: Brief description of scenario

Brief overview of scenario

- A radio in the MANET fails
- The other radios learn about this problem via the routing protocol, and reroute traffic.
- As more traffic goes through the remaining nodes, congestion is observed.
- In this scenario, all of the nodes belong to one subnet or routing domain.

Figure 8, on next slide, shows a schematic of the network before and after failure

Self-healing - Scenario #1: Schematic of scenario before and after failure

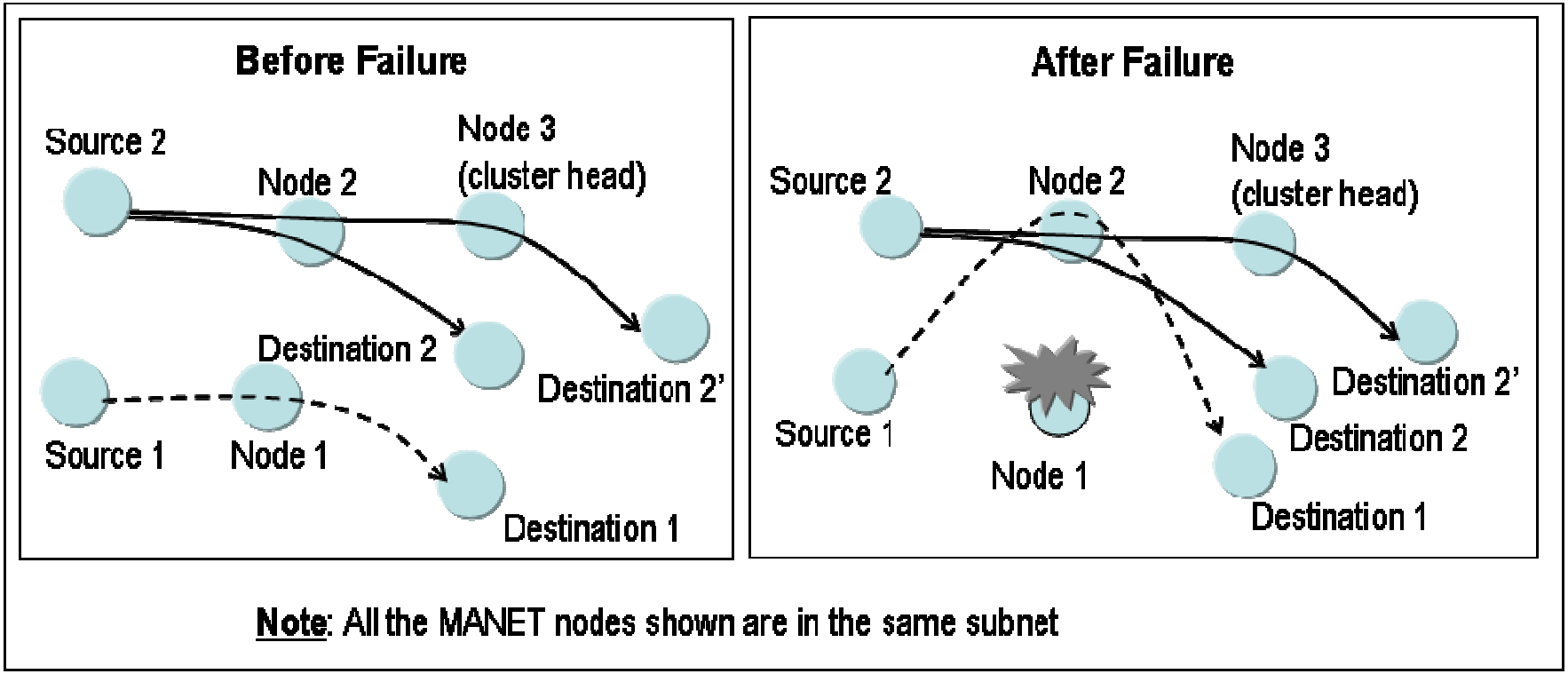

Figure 8: FM - Scenario #1

Self-healing -Scenario #1: Explanation

- MANET nodes are typically organized into ‘clusters’ or ‘domains’ for management and scalability reasons
- MANET contains encrypted (black) and un-encrypted (red) portions (for security reasons)
- Node 3, in Figure 8, also referred to as ‘cluster head’ houses the NMS and is on the red (un-encrypted) side
- For purposes of illustration, three sets of flows are shown in Figure 8
  - Source 1 initiates one flow between itself and Destination 1
  - Source 2 initiates two flows, one each to Destination 2 and Destination 2′
  - ➢ As will be seen via this example, even with a relatively simple network and flows such as those illustrated, the failure of one network element has the potential to impact all of the network services (despite the mesh nature of the interconnection) and also involves relatively complex fault diagnosis and corrective actions.
- In the normal operating mode, the paths taken by these flows are as follows:
  - Source 1 uses Node 1 as its transit node whereas
  - Source 2 uses Node 2 as its transit node.
  - Note that the actual paths taken between a source and destination can vary over time, depending on the conditions of the underlying network and the metrics used by the routing protocol in computing the reachability information.
  - Further, for purposes of illustration, the intensity of the flows from Source 2 is assumed to be such that it saturates the links that the flows take en route to their destination.
  - We show next what can potentially happen in such a case if the radio in Node 1 fails.

Self-healing - Scenario #1: Explanation - continued

- If Node 1 fails:
  - The routing protocol will learn about this failure and adapt by routing the flows from Source 1 around the failed node, which in this case will result in flows being routed via Node 2.
- Since the links surrounding Node 2 are already saturated, this additional load will soon lead to a soft failure
  - In other words, the links will become congested, causing unacceptable loss and delay.
- Such a situation is captured in the right-hand portion of Figure 8.
- The soft failure will impact *all* of the flows in this scenario, despite the fact that there was a single radio failure.
- The steps taken by a policy driven self-healing system are described next, which include:
  - Processing flows
  - Fault Detection
  - Corrective Actions

Self-healing - Scenario #1: Explanation (Processing Flows)

The processing flow for this scenario is as follows:

- *Node 2* detects high packet loss on its wireless interfaces, indicating congestion. Since this congestion could affect other nodes, *Node 2* notifies the NMS on *Node 3* (the cluster head).
- The NMS on *Node 3* realizes that *Node 1* is unreachable. This information is based on the lack of communication from the NMS on *Node 1*;
  - Typically each NMS will regularly communicate a status (“heartbeat”) to its cluster head.
  - Alternatively, routing traps from the black routers of the neighbor nodes could also be used to determine that Node 1 is unreachable.
  - Note:
    - The network management system is housed on the unencrypted side (red) whereas the failure shown in this scenario corresponds to a radio failure on the encrypted (black) side of the network, with very limited management information exchange between the two.
    - In this scenario, the information exchange includes a selected set of routing traps across the two network segments.
- A QoS problem is detected by the performance management systems on *Source 1* and *Source 2*, and both of these nodes (*Source 1* and *Source 2*) send “QoS failure” notifications to the NMS in Node 3.

Self-healing - Scenario #1: Explanation (Processing Flows) - continued

The **processing flow** for this scenario is as follows (continued)

- The NMS performs graph analysis via its horizontal dependency model to determine whether any of the probable paths between the troubled node pairs go through *Node 2*, which is known to be congested (see first bullet above).
  - Recall: The horizontal dependency model constructed by the NMS is essentially based on *inferences* that it makes about the network connectivity, since it is housed on the red (unencrypted) side and the failures have occurred in the black (encrypted) side of the network.
  - Assuming that a limited amount of "routing information leakage", as mentioned earlier, is allowed from the black to the red side, the NMS is able to construct the dependency models that it needs to perform fault diagnosis.
  - It should be noted that since these topology inferences are probabilistic in nature, the resulting outcome (from the root cause analysis) will be a probabilistic one – i.e., with some certainty ($0.0 < x \leq 1.0$), the given solution is a root cause.
  - In this specific example, due to the small size of the network, the fault diagnosis narrows down the root cause and the NMS determines that the soft failures are most likely caused by the congestion (QoS-related problems) on *Node 2*.
- Based on the timing of the radio failure on Node 1 and the congestion on *Node 2*, and the QoS failure notifications received from *Source 1* and *Source 2*, the fault management system within the NMS on *Node 3* determines that the congestion on *Node 2* is probably caused by the radio failure on *Node 1*.
- In summary, the fault management system interacts with the performance management system – since one of the symptoms here was a Quality of Service problem (i.e., soft failure) – to help in its root cause analysis phase; and it diagnoses the root cause of the congestion problem (soft failure) as being due to a radio (hard) failure.

Self-healing - Scenario #1: Explanation (Fault Detection)

The 'fault detection' process for this scenario is as follows

- The following observable events occur in the network/NMS:
  - An actionable alert is generated on the node with the radio failure (*Node 1*).
  - Performance alerts are generated by *Source 1* and *Source 2*.
  - An alert is generated by the NMS at *Node 3* (the cluster head) upon diagnosing the root cause of the congestion problem, with a level of impact that is indicative of the "criticality" of the problem.
    - The level of impact is a policy-driven input, since it is an artifact that is dependent on the nature of the problem and the nature of the mission that is being supported by the given MANET.

Self-healing - Scenario #1: Explanation (Fault Correction)

'Fault correction' is performed as follows:

- In the scenario described here there is a *hard fault* – in other words, there is an equipment failure, and the failed radio must be repaired or replaced.
- Since equipment repair is a manually intensive process and may take a long time, there is a need for alleviating the congestion problem in the meantime.
- One way to achieve this is to dynamically "create" additional network capacity.
  - One possible way of dynamically creating additional network capacity in MANETs is to capitalize on the mobile aspect of MANET nodes, and check whether it is possible to re-deploy nodes by moving one or more mobile nodes to the bottleneck region, thereby alleviating the prevailing congestion.
  - In addition, based on the nature of the terrain, there may arise situations that preclude moving ground nodes but may allow bringing in aerial nodes.
    - For example, MANETs deployed in swampy or mountainous terrains often use unmanned aerial nodes (UANs) to provide relay capabilities, wherein the UANs are directed to fly to different locations based on communications requirements.
- The following flow illustrates the fault correction process carried out at the cluster head (Node 3) that can be used to correct the above congestion problem:
  - Upon determining the root cause of the problem as described above, the root cause is sent by Fault Management to Policy Management.

Self-healing - Scenario #1: Explanation (Fault Correction) - continued

Fault correction - continued

- Policy Management receives the root cause event and retrieves the relevant policy. In this case, it is assumed that a policy has been specified that indicates the corrective action to be taken as follows:
  - *Event*: Root Cause indicating
    - location and identity of failed node, and
    - need for immediate capacity replacement to relieve congestion on *Node 2.*
  - *Condition*: UAN asset is available in the vicinity of the failed node.
  - *Action*: Send configuration directive to the UAN asset to move to an appropriate location near the failed element and to configure itself with the appropriate frequencies and subnet parameters to be able to relay traffic for the congested portion of the network.

  Note: Policies such as the one above will typically be derived from *a priori* performance information and/or simulation studies, and will undergo analysis in an appropriate testbed prior to deployment.
- The above policy is triggered and executed, resulting in an alleviation of congestion at the site of the failed element.

Self-healing - Scenario #1: Explanation (Fault Correction) - continued

Fault correction - continued

- In the case where there is no UAN available to move into position as a relay for the congested portion of the network other policies could be in place that attempt different solutions. Examples of alternate mechanisms include:
  - Moving another ground vehicle closer to the area of congestion so that some flows are routed through this ground vehicle, or
  - Throttling low priority flows so that the most critical messages get through.
- While some of the above alternatives (e.g., dynamic throttling based on priorities) may be automated, some other alternatives, such as moving another ground node may need more complex policies, or, as a last resort, involve human decision-making.
  - For example, it may not be practical to move ground vehicles to different locations, due to a combination of reasons including terrain impediments (such as the presence of swamps/rivers) or simply because of the fact that these other ground nodes may be critical to performing a certain set of functions in their current location.
  - Thus policies must be sophisticated to include many constraints and boundary conditions, before enforcing a specific solution

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
      - **Scenario 1**
      - **Scenario 2**
      - Scenario 3
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Self-healing - Scenario #2: Brief description of scenario

Brief overview of scenario

- A MANET link becomes unavailable due to environment-related issues, e.g.,
  - A nodes move into a mountainous terrain whereby two adjacent nodes cannot communicate (nodes *Source 1* and *Node 1* in Figure 9).
  - All of the nodes in this scenario, like the previous case, belong to a single subnet.
  - The nodes that were originally using this link quickly learn about this problem via the routing protocol, and re-establish new routes over other available links in order to maintain network connectivity and uninterrupted information transfer through the network.
  - However, due to the resultant reduction in network resources because of unavailable links, the remaining network links become congested, causing a soft failure for all of the services using the remainder of the links.

  Note: This scenario is very similar to Scenario #1, but with a different root cause. Whereas in Scenario #1 the root cause was a (hard) radio failure, in this case, the root cause is a soft failure – in other words, no network element fails, but a link becomes unavailable due to the nature of the current terrain.

Figure 9, on next slide shows a schematic of the network before and after failure

Self-healing - Scenario #2: Schematic of scenario before and after failure

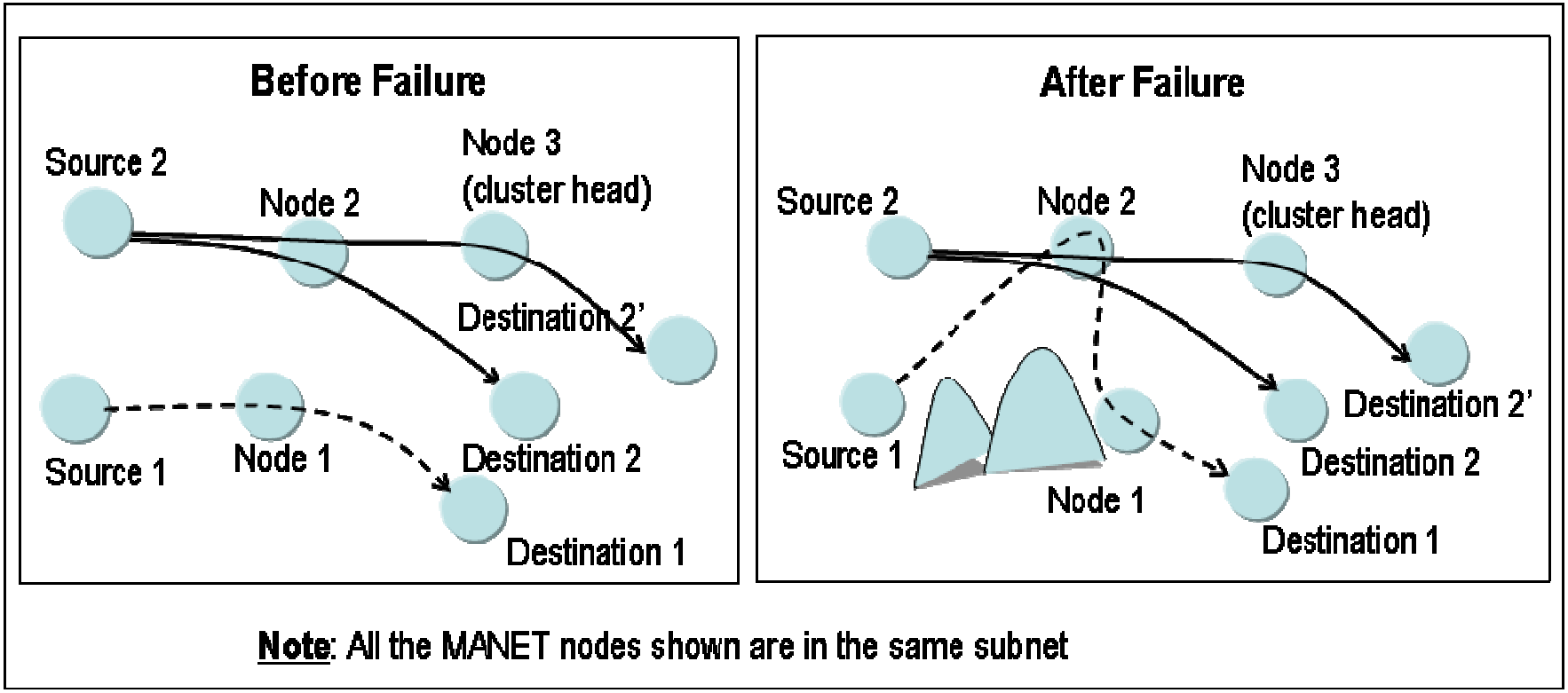

Figure 9: FM – Scenario #2

Self-healing -Scenario #2: Explanation

- As in the case of Scenario #1,
  - The MANET in Scenario #2 contains encrypted (black) and un-encrypted (red) portions (for security reasons)
  - Node 3, in Figure 9, also referred to as ‘cluster head’ houses the NMS and is on the red (un-encrypted) side
- For purposes of illustration, three sets of flows are shown in Figure 9 (like before)
  - Source 1 initiates one flow between itself and Destination 1
  - Source 2 initiates two flows, one each to Destination 2 and Destination 2′
  - ➢ As before, even with this relatively simple network and flows such as those illustrated, the failure of one network element has the potential to impact all of the network services (despite the mesh nature of the interconnection) and also involves relatively complex fault diagnosis and corrective actions.
- In the normal operating mode, the paths taken by these flows are as follows:
  - Source 1 uses Node 1 as its transit node whereas
  - Source 2 uses Node 2 as its transit node.
  - Note that the actual paths taken between a source and destination can vary over time, depending on the conditions of the underlying network and the metrics used by the routing protocol in computing the reachability information.
  - Further, for purposes of illustration, the intensity of the flows from Source 2 is assumed to be such that it saturates the links that the flows take en route to their destination.
  - We discuss next what happens when the nodes move around in a mountainous terrain.

Self-healing - Scenario #2: Explanation - continued

- Node 1 moves away around a mountain and loses ‘connection’ with Source 1
  - The routing protocol will learn about loss in connection and adapt by routing the flows from Source 1, which in this case will result in flows being routed via Node 2.
- Since the links surrounding Node 2 are already saturated, this additional load will soon lead to a soft failure
  - In other words, the links will become congested, causing unacceptable loss and delay.
- Such a situation is captured in the right-hand portion of Figure 9.
- The soft failure will impact *all* of the flows in this scenario, despite the fact that only one node moved away from its original co-ordinates.
- We next describe the steps taken by a policy driven self-healing system, which include:
  - Processing flows
  - Fault Detection
  - Corrective Actions

Self-healing - Scenario #2: Explanation (Processing Flows)

The processing flow for this scenario is as follows:

- Node 2 detects high packet loss on the encrypted (black-side) interfaces indicating congestion. Since this congestion could affect other nodes, Node 2 notifies the NMS on the cluster head node (Node 3).
- In the meantime, the fault management function on the NMS on Node 3 learns about the loss of physical connectivity between Source 1 and Node 1. The NMS infers this from its horizontal dependency model (i.e., physical connectivity) via a limited amount of information obtained from the black network for monitoring purposes.
- A QoS problem is detected by the performance management systems on Source 1 and Source 2, and both of these nodes (Source 1 and Source 2) send “QoS failure” notifications to the NMS in Node 3.
- Based on the recent history of physical connectivity and the timing of congestion on Node 2, the fault management component determines that the congestion is probably caused by the loss of physical connectivity.
  - This conclusion is arrived at by performing graph analysis to determine whether any of the probable
  - As before, the inferences obtained via this analysis are probabilistic in nature, due to the stochastic nature of the input (dependency) models.
  - Once again, due to the small size of the network, the root cause analysis narrows down the root cause of the problem, and the NMS determines that the soft failures are most probably caused by the congestion on Node 2.

Self-healing - Scenario #2: Explanation (Fault Detection)

Fault Detection for this scenario involves the following actions:

- Performance alerts are generated by *Source 1* and *Source 2*.
- An alert is generated by the NMS on *Node 3* upon diagnosing the root cause of the congestion problem, with a certain level of impact that is indicative of the “criticality” of the problem.

Self-healing - Scenario #2: Explanation (Fault Correction)

Fault Correction for this scenario involves the following actions:

- Since the scenario described here has the same impact as in Scenario #1, the corrective action processing is performed in the same way, by and large, as was described there.
  - A possible difference here is that if the terrain change is known to be temporary (e.g. if the maneuver plan for the MANET is known and it can be predicted that the terrain blockage is temporary as all nodes will soon have moved past the blockage area), then it may be preferable to take no action to correct the problem.
- The processing on the cluster head (*Node 3*) is shown below.
  - Upon determining the root cause of the problem as described above, the root cause is sent by Fault Management to Policy Management.
  - Policy Management receives the root cause event and retrieves the relevant policy. In this case, it is assumed that a policy has been specified that indicates the corrective action to be taken as follows:
    - Event: Root Cause indicating (i) loss of physical connectivity between *Source 1* and *Node 1*, and (ii) need for immediate capacity replacement to relieve congestion on *Node 2*
    - Condition: Maneuver plan indicates that terrain will not change in the near future.
    - Action: Send configuration directive to the UAN asset to move to an appropriate location near the two disconnected nodes and to configure itself with the appropriate frequencies and subnet parameters to be able to relay traffic for the congested portion of the network.
  - The above policy is triggered and executed, resulting in an alleviation of congestion at *Node 2*.

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
      - **Scenario 1**
      - **Scenario 2**
      - **Scenario 3**
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Self-healing - Scenario #3: Brief description of scenario

Brief overview of scenario

- This scenario captures the effect of a security-related problem and impact on fault management
- The specific example considered (for illustration purposes) is a denial-of-service (DOS) attack and its impact on service failures and fault diagnosis followed by self-healing.
- A DOS attack is, in essence, caused by a malicious host flooding the network with unnecessary packets, so that network resources are wasted in processing useless information and hence become partially unavailable for mission-critical applications.
  - Note that while a DOS attack can begin in a fairly benign manner by being contained to one network segment (domain), it can soon spread, if the border nodes are attacked, to other network segments, and very soon cripple an entire network.
- Such a DOS attack essentially results in soft failures throughout the network.

Scenario #3: Soft failure due to DOS attack

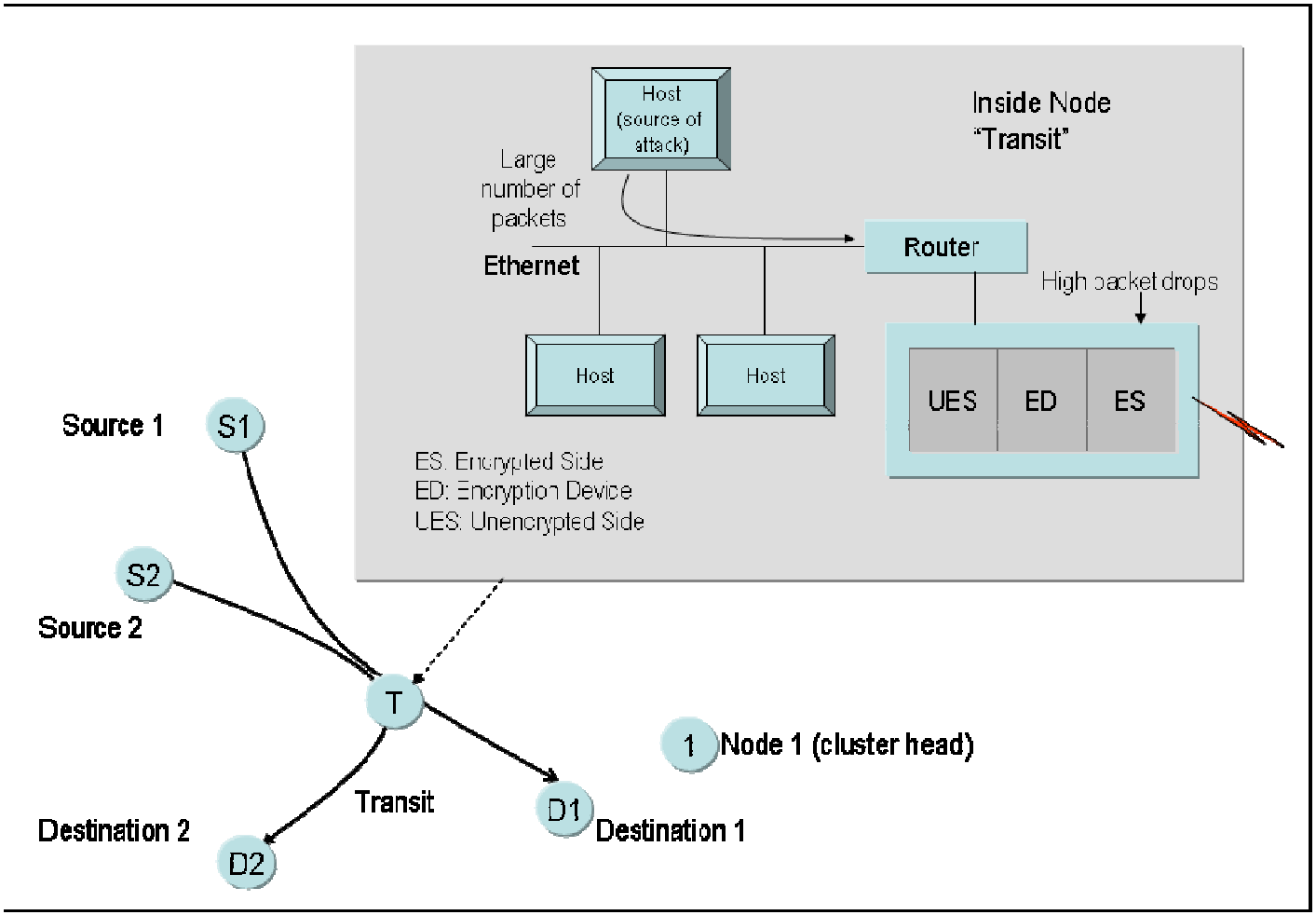

Figure 10: FM – Scenario #3

Self-healing -Scenario #3: Scenario Explanation

This scenario can be described as follows:

- As shown in Figure 10 (previous slide) a DOS attack originates from a host on the node labeled *Transit*.
  - A host on the node *Transit* starts to maliciously inject copious amounts of traffic destined to all of the other nodes in the network.
- Note that the hosts within a MANET node are typically connected to a router via a high speed (usually a Gigabit Ethernet) LAN.
  - The router has one or more wireless interfaces that are connected to the wireless network, and have speeds that are at least an order of magnitude or more lower than the high-speed LAN.
- While the host may be successful in getting its packets across the inbound interface to its router, the outbound interface from the router will now be overwhelmed, due to the limited speed on its outbound (wireless) link.
- Such an attack causes the node Transit to drop a large number of legitimate packets that are going through it,
  - i.e., packets using it as a transit node to reach their destination nodes.
- Consequently, legitimate applications are deprived of service (in an extreme case, they will be denied service), which is manifested as a soft failure.

Self-healing - Scenario #3: Explanation (Processing Flows)

The processing flow for this scenario is as follows:

Before enumerating the "processing flows", the following assumptions have been made for this scenario:

- A host on node Transit has been compromised, thus turning malicious.
- The Network Intrusion Detection System (NIDS) within the security management component within the NMS on node Transit detects an anomaly – unusually high packets from one of its hosts – and diagnoses a DOS attack.
- Routing traffic is not dropped. This is because network management, routing and other control-plane messages will, in most practical networks, be allocated a distinct DSCP (DiffServ Code Point) that is not available for user traffic.
  - This is done for multiple reasons, the most important of which is that network management and control must be accorded higher priority than other traffic, so as to ensure that critical monitoring and configuration traffic is transmitted in a timely fashion over the network, even in heavy load conditions.

Self-healing - Scenario #3: Explanation (Processing Flows) - continued

The processing flow for this scenario is as follows:

- The node *Transit* detects high packet loss on its wireless interface(s). Additionally, the NIDS component on *Transit* detects a probable intrusion from a host on its LAN, which is flooding the network with a high volume of traffic. It notifies the fault management component within the local NMS, which correlates these alerts with a low level of certainty, and determines that the root cause of these two problems might be a DOS attack originating from the local platform.
- Since *Transit* cannot be completely certain of the root cause of the problem, it notifies the NMS on *Node 1*, which hosts the cluster head, of the high packet loss and intrusion, as well as the possible root cause.
- As the packets from *Source 1* to *Destination 1* and *Source 1* to *Node 1* are dropped by *Transit*, the performance management component in *Source 1* observes a performance threshold violation, which, if left uncorrected, could potentially lead to a soft failure.
- *Source 1* sends a performance-related alarm to the NMS on *Node 1.*
- Similarly, *Source 2* also observes a similar service performance violation threshold crossing and sends a similar alarm to the NMS on *Node 1*.
- The NMS on *Node 1* determines that the soft failure notifications are probably related to the high packet loss on *Transit*.
- The NMS on *Node 1* correlates all of the available information and determines with a high degree of certainty that the problem is caused by a malicious DOS attack.

Self-healing - Scenario #3: Explanation (Fault Detection)

Fault Detection for this scenario involves the following:

- An intrusion alert is generated by node *Transit.* This is correlated with the local high packet loss, and a root cause with low certainty is sent to the NMS on *Node 1*, indicating the possibility of a DOS attack.
- Performance alerts are generated by *Source 1* and *Source 2*.
- An alert is generated by the NMS on *Node 1* upon diagnosing that a DOS attack is the root cause of the congestion problem, with a certain level of impact that is indicative of the “criticality” of the problem.

Self-healing - Scenario #3: Explanation (Fault Correction)

Fault correction is as follows:

As described earlier, the fault management component on *Node 1* generates a root cause for the DOS attack. The following steps are executed to correct the problem:

- The root cause event triggers a corrective action via a pre-defined policy that is executed by the policy management component on the *Transit* node. The policy is of the following form:
  - Event: Root Cause indicating location of DOS attack and identity of malicious attacking node. Note that this event is sent by *Node 1* to node *Transit.*
  - Condition: None.
  - Action: Send configuration directive to the configuration management component on the *Transit* node, directing it to disable the port on the router to which the malicious host is connected, thereby shutting off the source of the DOS attack.
- The above policy is triggered and executed, resulting in cutting off the source of the DOS attack.

Self-healing - Scenario #3: Explanation (Fault Correction)

Fault correction (continued):

- Regarding the condition of the policy (previous slide):
  - Although no condition is included as part of the above policy, in practice it may be necessary to include one or more conditions in the policy.
    - As an example, a network manager could use the concept of a “threat level” for the network, which could indicate various situations such as the existence of a high or low potential for network attack, etc. The current network threat level could be included as part of the diagnosis process, or as part of the policy to control whether a certain action is very conservative (e.g. err in favor of caution, at the risk of cutting off legitimate network users) or less conservative (e.g. do not cut off all traffic from a given node).
    - Examples of two such policies are given below.
- Conservative policy:
  - Event: Root Cause indicating location of DOS attack and identity of malicious attacking node.
  - Condition: Current network threat level is high.
  - Action: Send configuration directive to the configuration management component on the *Transit* node, directing it to disable all traffic entering the network from the *Transit* node, thereby shutting off any possible source of the DOS attack.
- Less conservative policy:
  - Event: Root Cause indicating location of DOS attack and identity of malicious attacking node.
  - Condition: Current network threat level is low.
  - Action: Send configuration directive to the configuration management component on the *Transit* node, directing it to disable the port on the router to which the malicious host is connected, thereby shutting off the source of the DOS attack.

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
      - **Scenario 1**
      - **Scenario 2**
      - **Scenario 3**
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Performace Management: What is it, and what is it's role in MANETs?

- Performance Management – deals with collecting performance data and events, monitoring service quality, notifying performance degradations, and recommending solutions to ensure service level assurances are maintained. More specifically, the key functions of performance management for MANETs are:
  - Performance anomaly detection, via periodic polling of network elements and detection of threshold crossings
  - Generation of performance violation alerts that are used as inputs to fault management to perform integrated correlation with fault, configuration and security events, and root cause analysis
  - Assisting with end-to-end quality of service (QoS) assurance, especially to the mission critical applications (platinum services)
  - ➢ Note the intricate tie-ins between performance and fault management operations. These inter-relationships are further heightened in MANETs due to the fact MANETs are stochastic in nature and thus have a high degree of soft failures
- Role of Performance Management (PM) in MANETs:
  - Heightened due to the unpredictable nature of MANETs
  - The stochastics combined with the fact that MANETs often need to transport a wide spectrum of applications (ranging from mission critical/platinum services to routine/best effort/bronze services) each with their own requirements on service quality (e.g., delay, jitter, loss) underscore the need for
  - ➢ ***Sophisticated Quality of Service assurance techniques that cater to widely varying performance requirements***

Performance Management - Operations Models: Quick Overview

- To assist with the key PM tasks that involve performance monitoring and service assurances, several operational models have been developed in the Industry
- The TMN model of performance management operations developed for telecommunications networks (as in the case of Fault Management) is the most popular
- Due to the fundamental differences between MANETs and telecom networks, the TMN model for Performance Management operations can not be used as-is for MANETs
- However, much can be learnt from the TMN models, which can then be 'adapted' for MANETs (as will be discussed next)

Performance Management - Operations Models : Traditional (TMN) view:

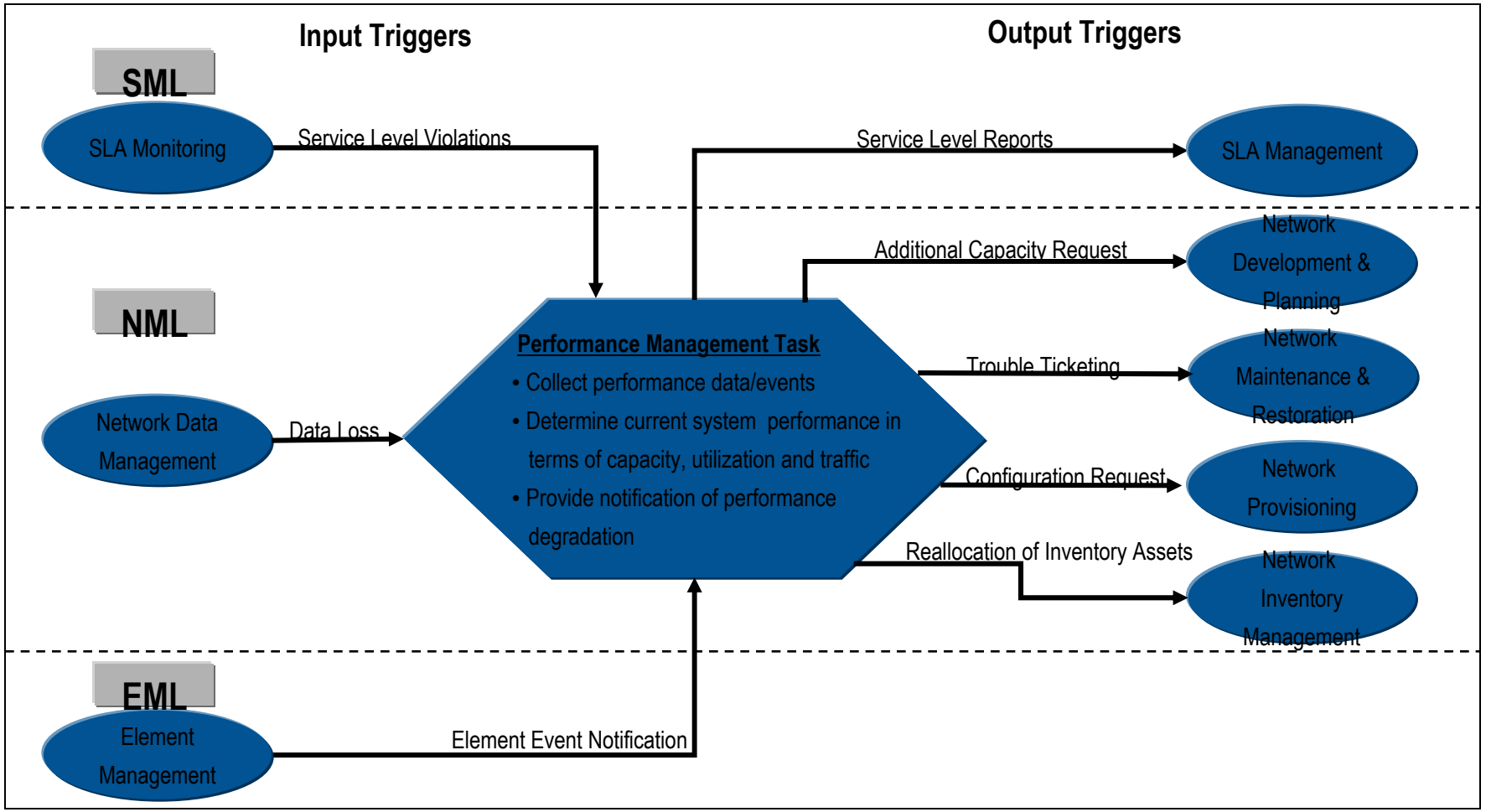

Figure 11: PM operations model – TMN model

Performance Management - Operations Models : Traditional (TMN) view - continued

Salient points

- PM operations (like its counterpart – FM) organized around the service management layer (SML), network management layer (NML) and element management layer (EML) concept
  - SML: 'service operators' typically function at this layer – e.g., service operators periodically check on SLAs (service reports) to look out for 'service agreement' violations, and issue triggers to notify
  - NML: network monitoring related activities take place at this layer - i.e., checking of current performance levels and triggering the issue of notifications to SML in the event of performance degradations
  - EML: element level monitoring of threshold violations and event alerts to the NML
- Input triggers (to the left in Figure 11) trigger performance management actions (center) resulting in output triggers (to the right in Figure 11)
- Automated network diagnostics at the 'lower layers' (EML, NML)
- Human-in-the-loop to assist with performance violations administration and (re)negotiations (SML) and coupling with other management operations

Need for new PM operations models in MANETs

- Layer Classification
  - While OK for telecom-type of networks, concept of SML with several ‘operators’ will not translate to MANETs
  - However can still use the “SML” concept to service assurances via quality of service (QoS) guarantees that are ensured via ‘automated’ (vs. HITL) system policies
- Stove-piping of network management operations
  - The network management functions (e.g., FCAPS operations) are not tightly coupled with each other
  - While this may be OK in a telecom environment (due to a high degree of HITL), this is certainly not the case in MANETs, where the FCAPS operations have to link in with each other
    - E.g., QoS assurance (a key PM function) may require close coupling and interactions with configuration, fault and security management in MANETs
- Response to network faults
  - Heavy involvement of service level operators in the telecom (TMN) model; contrast this with the need for automated service assurance via adaptive end-to-end QoS assurance mechanisms

Performance Management - Operations Models: Adapted for MANETs

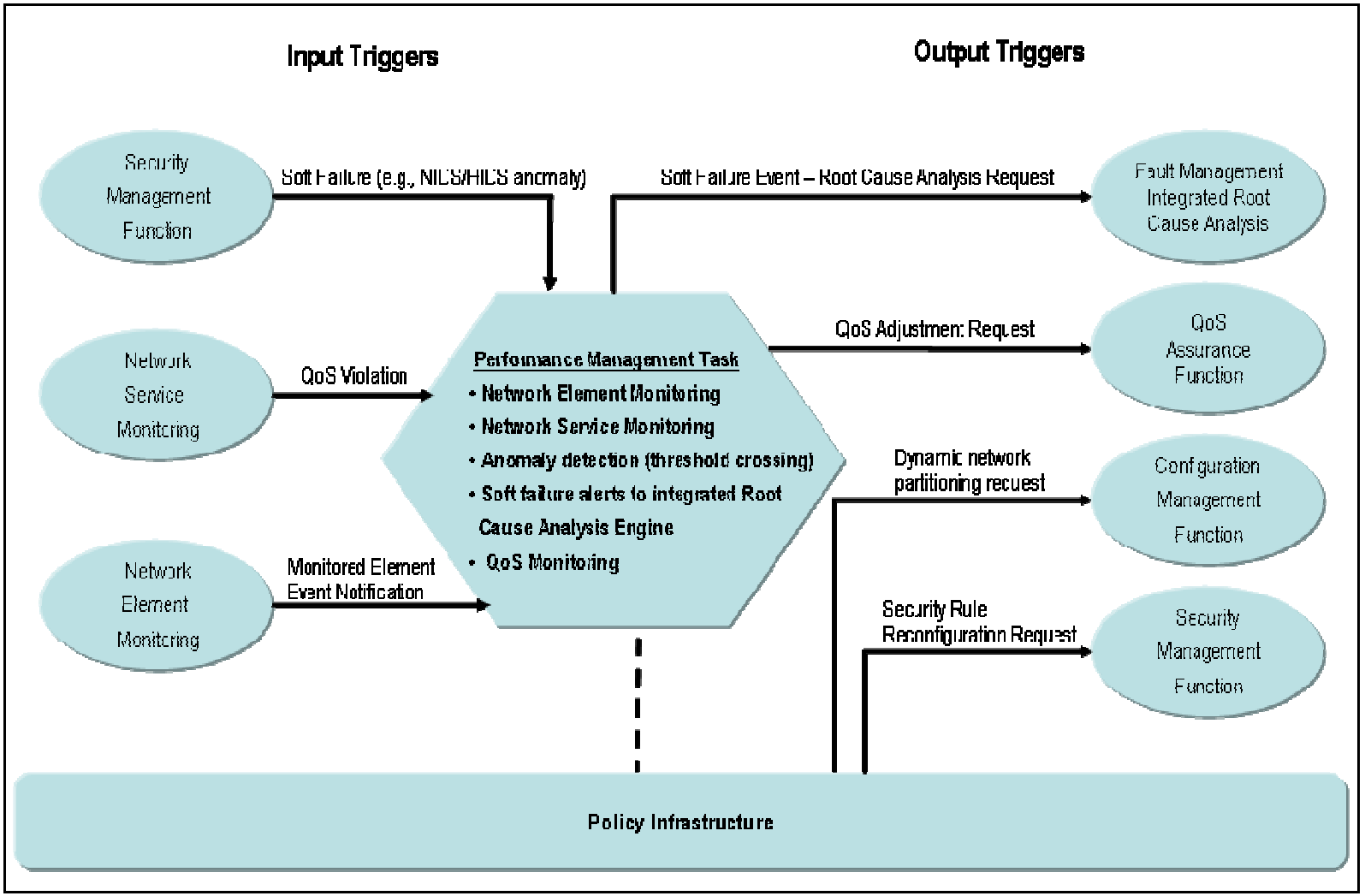

Figure 12: PM operations model – MANET model

Performance Management - Operations Models: Adapted for MANETs (continued)

Salient Points (Figure 12)

- The performance management operations process model has strong dependencies and is integrated with the operations process models for fault, configuration, and security, as captured via the input and output triggers.
- Performance violations are inputs to an integrated root cause analysis engine, implemented within the fault management process.
- A policy management component ties together the FCAPS functions by seamlessly integrating these functions via policies.
- A critical new function for QoS assurance is included that is responsible for providing service assurances in a dynamic and seamless manner, to applications with differing priorities using the MANET.

Key PM functions in MANETs

- Performance anomaly detection, via periodic polling of network elements and detection of threshold crossings;
- Generation of performance violation alerts that are used as inputs to Fault Management to perform integrated correlation with fault, configuration and security events, and root cause analysis; and
- End-to-end quality of service (QoS) assurance.

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
      - **Scenario 1**
      - **Scenario 2**
      - **Scenario 3**
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network Monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Performance Management: Network Monitoring

- The performance monitoring function is responsible for monitoring the performance of the network on a continuous basis
  - i.e., even when the network is functioning normally.
- To achieve the above, the performance management periodically polls the network elements and services to
  - Ensure key performance parameters are within expected bounds
- In contrast
  - The fault management function does not have much to do when there are no network faults
- The following are key network monitoring-related functions of the performance management sub-system in MANETs
  - Collection of performance statistics for Network Elements,
  - Collection of performance statistics for Network Services, and
  - Policy-controlled Monitoring

Collection of performance statistics for network elements

- Performance statistics collections in networks typically performed via SNMP.
  - For example, the routers that connect a wired LAN and the wireless medium have buffers that can exhibit a sustained overflow, indicating abnormal or unanticipated operational conditions (note that the buffers are normally sized such that they do not overflow under normal operational conditions).
  - An important task of performance management is to monitor the routers to detect such sustained overflows, and report to an integrated fault/performance/security correlation engine, to help with root cause analysis associated with soft failures.

  Note: sustained buffer overflows at a router will result in packet loss, in turn leading to a disruption in service.
- Another example of a network element in MANETs whose performance needs to be monitored is the device that performs encryption, such as an IPSec or HAIPE device.
  - A key performance metric associated with encryption devices is latency, i.e. the time lag that is introduced due to the encryptions and decryptions performed. In order to provide timely service, it is imperative that these latencies be low.
  - Thus the performance management function needs to monitor the latency performance and report any anomaly, so that timely root cause analysis can be performed.

Performance statistics collection: Differences between MANETs and wire-line networks

- While there exists a network element at the OSI physical layer (namely, the cable or wire) in wireline networks, in MANETs, there is no such analogue.
  - The physical medium is wireless
- Thus, whereas in wireline networks, the performance management function can poll the transceiver at the end of a fiber to detect the presence or absence of a signal, no such network element with an associated signal exists at the physical layer in MANETs.
- Instead, the *effect* of the losses at the physical layer due to a broken link (caused by jamming, for example, or terrain obstructions such as a mountain) manifests itself as packet loss at the higher layer, which will then need to be taken into account by a root cause analysis engine to detect the source of the problem.

Collection of performance statistics for network services

- Another important task of the performance management function is to monitor the performance (the “health”) of a variety of MANET services.
  - Note that the fault management function will monitor these services to ensure that the services are up and running.
  - This can be done via regular polling of the processes that provide the services.
  - Thus, while the fault management function monitors whether the service is “alive” or not, the performance management function monitors the “level of service” offered by a given server, thus ensuring the proper functioning of the given system.
  - This cooperation between fault and performance management operation re-iterates the importance of an integrated network management system.
- The following slides describe some examples of key network services that need to be in place in mobile ad hoc networking environments, and the role of performance management in monitoring these services.

Performance statistics for Network Services: Name-address translation/resolution service

- This service is akin to the popularly used DNS service in wireline networks.
- The concept here is to associate a user-identifiable node name with a network-identifiable node name, which is used by the underlying MANET to route information packets through the network.
- A key performance metric associated with name-address resolution servers is the time taken for address resolution.
- The role of the performance management function is to monitor this service and raise an alarm if the time taken for address resolution exceeds a policy-defined threshold value.

Performance statistics for Network Services: Mobility Management Service

- Mobility is a fundamental characteristic of MANETs
  - Recall: MANETs do not have the concept of an infrastructure with some nodes being fixed and others being mobile.
  - Instead, every node in a MANET can potentially be a mobile node.
  - Thus, MANETs require mobility management techniques that help keep track of nodes' mobility so that any two MANET nodes can continue to remain in communication, despite random mobility.
- A variety of mobility management techniques are being researched for MANETs.
  - Examples include SIP-based mobility management [Camarillo 2002] and Mobile IP [Perkins 2002].
- The role of performance management here is to ensure that the mobility management service is functioning as expected.
  - i.e., based on the type of mobility management technique, the performance management function monitors specific performance metrics, and will generate alarms when policy-specified threshold values are exceeded.
    - For example, if a SIP-based mobility management technique is employed, one performance metric will be the time taken to register the new IP address of a node with its SIP server, once node movement has taken place.

Performance statistics for Network Services: Quality of Service (QoS) Management Service

- Providing appropriate QoS to the various applications that use a MANET is an extremely important function.
  - Recall that MANETs are expected to support a variety of applications with widely varying QoS requirements. For example,
    - Mission-critical applications (akin to platinum services in commercial networks) will require very stringent guarantees in terms of delay and loss
    - Voice applications and other real-time applications will require tight delay guarantees but somewhat less stringent loss assurances;
    - High resolution image transfer applications that are non-real-time in nature can tolerate delay but are very sensitive to loss; and so on.
  - Above requirements, coupled with the dynamic and unpredictable nature of MANETs (i.e., unpredictability both in terms of randomness in mobility and in terms of the bandwidth fluctuations due to extraneous factors such as the environment or jamming) imply that fairly sophisticated QoS assurance mechanisms are needed that can cope with uncertainty while sustaining the required service guarantees.
  - Since the QoS servers themselves are hosted on network elements, it is critical to ensure that both the services and the servers that host the QoS solutions are functioning as expected.
  - The role of performance management is to monitor the QoS servers to ensure that certain performance thresholds are not crossed. Some examples of these thresholds are:
    - The response times of the QoS server remain below policy-defined threshold values., where the response time is the time taken to make an admission control decisions
    - The number of flow preemptions remains below a policy-defined threshold.
    - The number of flow request rejections remains below a policy-defined threshold.

Performance statistics for Network Services: Session Management Service

- Session management is another important MANET service.
  - Session management provides the capability to locate users in a MANET, even when they move from node to node, for the purpose of establishing communications (such as voice calls or chat sessions).
  - Users are associated with fixed user-friendly names, and the session management service is responsible for mapping these names to the IP addresses of the hosts where these users are located.
  - When a user moves, the session management service registers the new location of the user, thus enabling other users to continue to locate this user for session establishment using the user's name.

  Note that it is important that the session management implementation provide the required level of service to the network users.
- The role of the performance management function is to monitor the associated servers by computing key performance metrics such as
  - Session setup time, time taken to register the location of a user, etc.
- Policy-defined thresholds are defined for each of these metrics, and any threshold crossings are reported as alarms to the fault management function.

Policy-controlled Monitoring in MANETs –why is it needed?

- Due to the distributed nature network monitoring is performed on every MANET node and needs to be exchanged with the NMSs on other nodes
- The scarcity of bandwidth in MANETs however requires that the amount of management information sent over the air needs to be kept at a minimum.
- Thus, although performance statistics are periodically collected on every node, they should not be sent to other nodes unless absolutely necessary.
- The determination of what type of information should be sent to other nodes on a periodic basis, and the frequency of information dissemination needs to be done judiciously.
- Policies provide an excellent mechanism to accomplish such a judicious information exchange, as follows
  - Policies are created that define how information should be aggregated and filtered prior to over the air dissemination.
  - Policies are also defined to specify how frequently this information is sent over the air.
- The value of using policies to specify the frequency of reporting management information is that other policies can be used to adjust this frequency *automatically*, based on the congestion status of the network. For example,
  - Most of the widely used MANET QoS assurance mechanisms derive estimates of network congestion based on throughput measurements.
  - These estimates of network congestion can be used to throttle the frequency of reporting of management information over the air if the network is highly congested; and conversely, if the network is not congested, policies can be used to automatically increase the frequency of reporting.
- The relevant policies are illustrated by examples next.

Policy-controlled Monitoring in MANETs – example policies

- Policy 1 (configuration policy): Set a reporting frequency:
  - Action: Configure Performance Management function to report performance statistic X every Y seconds.
- Policy 2 (configuration policy): Set a performance crossing threshold:
  - Action: Configure Performance Management function to generate a threshold crossing alert when the number of rejected flows exceeds *n* within time window *m*.
- Policy 3 (ECA policy): Specify an automated change in reporting frequency based on congestion status of the network. Note that the performance management function is responsible for generating a threshold crossing event based on the previous policy.
  - Event: Threshold crossing alert (number of rejected flows crosses policy-specified threshold).
  - Condition: None
  - Action: Configure Performance Management function to report performance statistic X every 300 seconds.
- Policy 4 (ECA policy): Specify an automated change in reporting frequency back to the original value based on clearing of the previous threshold crossing alert.
  - Event: Clearing of the threshold crossing alert (alert for number of rejected flows crossing policy-specified threshold is cleared).
  - Condition: None
  - Action: Configure Performance Management function to report performance statistic X every 60 seconds.

Policy-controlled Monitoring in MANETs – example policies – continued

- The first two policies in the preceding slide are examples of configuration policies.
  - The first one configures the performance management function to report information about a specific variable at a specified frequency.
  - The second policy configures the value of the threshold at which the performance management function should generate a threshold crossing alert for a given congestion measure.
- The third policy is an ECA policy that automatically modifies the reporting frequency based on the generation of a threshold crossing alert.
- The fourth policy is also an ECA policy that sets the reporting frequency back to the original value when the threshold crossing alert is cleared.

Note: The capabilities achieved via the afore mentioned policies are very powerful, since they enable self-adjustment of the network monitoring system based on network status. The network management system can thereby adapt to changing network conditions in an automated fashion. This is a critical capability for MANETs, since network conditions are expected to vary dynamically in a MANET. The use of policies makes this automated adaptation possible.

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
      - **Scenario 1**
      - **Scenario 2**
      - **Scenario 3**
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network Monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Service Quality Assurance in MANETs – what is it?

- An important aspect of performance management in MANETs is the provision of service assurances to high-priority applications, sometimes at the expense of lower-priority applications
  - Namely, the ability to provide quality of service (QoS) assurances to the wide spectrum of applications using the MANET
- Quality of service is the ability to ensure that high priority traffic has the highest probability of message completion to intended users, within dynamic limits of ad hoc network resources.
  - Message completion must take into account the communication performance requirements of applications, such as low delay, low loss, or high throughput.
- Given the reality that the amount of traffic to be sent over a network may exceed its capacity, the need for techniques to provide end-to-end QoS assurances in MANETs is both obvious and critical.
  - Furthermore, due to the presence of applications that may need different levels of assurances (i.e., different types of guarantees on delay and/or loss), the QoS mechanism should also be capable of providing different service assurances to different types of traffic.
- ➢ This critical QoS functionality is achieved with the help of the performance management process.

Service Quality Assurance in MANETs – differences from that in wireline networks, and challenges

Differences between MANET and wireline QoS

- Wireline networks are typically over-provisioned, and usually operate under benign operational environments as compared to MANETs
  - E.g., wireline networks do not normally suffer from typical MANET problems such as unpredictable reduction in capacity due to fading, jamming, mobility, etc.
- For these reasons, the issue of QoS assurance is far more simple in wireline networks than in MANETs.

Challenges to providing QoS in MANETs

- Dynamic Network Topology
- Lack of Visibility into Network Topology
- Wide Range of QoS Requirements

End-to-end Service Quality Assurance in MANETs - Challenges

Dynamic Network Topology

- Challenges in providing end-to-end QoS assurances further compounded in MANETs due to the absence of a stable network topology. E.g.,
  - The locations of network elements highly variable (due to mobility)
  - The number of network elements in the network may vary dynamically due to nodes shutting down to conserve power, nodes moving out of range and nodes going down.
- Additionally, the network links themselves have highly variable capacity that may be affected by environmental conditions such as weather and terrain.
  - MANET links are characterized by different speeds, with the differences sometimes being orders of magnitude.
- A direct consequence of the underlying link speeds is the amount of delay incurred by a packet in transit, which in turn directly impacts the end-to-end QoS that can be assured to the corresponding traffic flows.

➢ Challenge: The QoS management mechanism needs to take into account all of this random variability when providing QoS assurances to applications.

End-to-end Service Quality Assurance in MANETs – Challenges (Continued)

Lack of visibility into network topology

- Not only is the network topology dynamic, it is also mostly unknown to the network management system.
- This is due to the encryption-related network separation issue (for security reasons).
  - Almost no management-related information about the encrypted network segments is allowed to flow outside of its boundaries
  - However, most network management systems reside on the unencrypted side of the network
  - Thus, the intermediate encrypted network topology is not visible on the red side.

➢ Challenge: Need to manage performance of a network that cannot be directly observed and provide end-to-end service quality assurances.

End-to-end Service Quality Assurance in MANETs – Challenges (Continued)

Wide range of QoS Requirements

- There exist a wide range of applications with diverse requirements on service quality. For example:
  - MANETs are usually expected to support a set of mission-critical applications, that require very stringent guarantees on delay and/or loss due to their critical nature.
    - This is akin to a platinum service subscriber in commercial wireline networks, where the subscriber expects "top quality" service delivery (i.e., low delay and loss).
  - On the other end of the spectrum, there exist regular/routine messages that are sensitive to neither delay nor loss
    - i.e., they are more tolerant of varying network delays and losses – e.g., services in the commercial space are 'bronze' services
  - There also exist applications, such as non-real time image transfer that are more sensitive to loss than delay.

➢ Challenge: Each of the above types of services require different levels of service assurances despite limited and variable MANET capacity, random mobility and link bandwidth fluctuations.

QoS mechanisms in Communications Networks: Bandwidth broker-based

- One approach to providing QoS assurances in wireline networks is the use of 'bandwidth brokers' [Nichols et. al, 1999]
- A bandwidth broker is an entity that is essentially used to control admission of flows of different service classes into a network, based on the network state.
- The bandwidth-broker's notion of network state is typically estimated based on knowledge of:
  - network topology (which is largely static),
  - network routes (which are again relatively static), and
  - link capacities (which are well known).
- The fact that such information about the network is readily available makes the QoS assurance problem relatively trivial for these networks.
- For MANETs, the knowledge of the *entire* network topology, routes, and link capacities is neither available (due to encryption), nor very useful (since all of these quantities vary dynamically).
- Thus a new approach is needed for MANETs.

Sample MANET with encrypted network segments

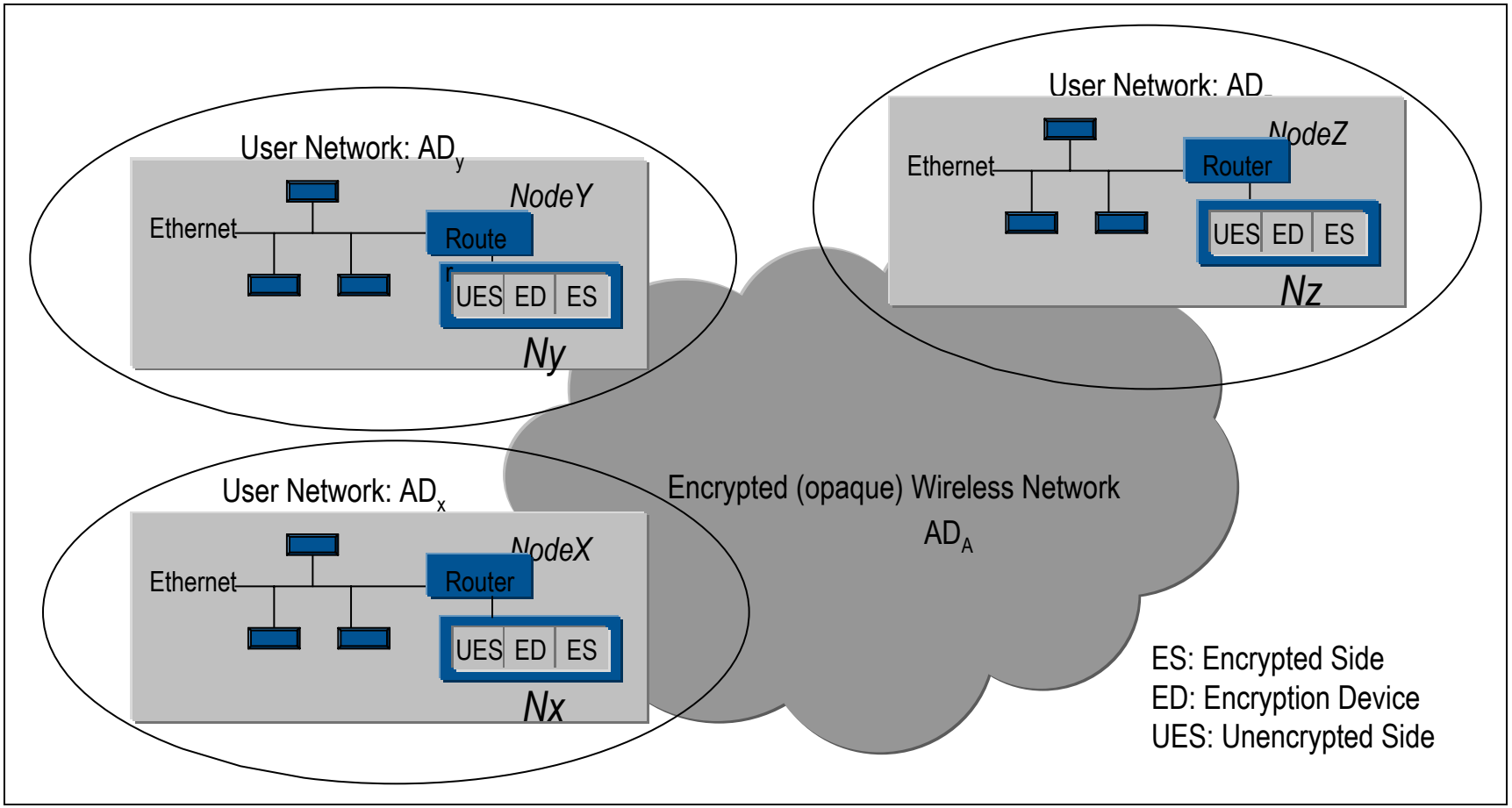

Figure 13: Sample MANET – combining encrypted and un-encrypted segments

Sample MANET – Key points

Salient Points for Figure 13 (previous slide)

- Three unencrypted routing domains labeled AD_x, AD_y and AD_z, where *AD* stands for Administrative Domain.
  - Un-encrypted domains are wired
- One encrypted (opaque) routing domain (AD_A).
  - Encrypted domain is wireless
- User nodes reside on the un-encrypted side, and communicate through encrypted network(s)
- The properties of the end user network segments and the intervening opaque network segments are significantly different
  - The user networks utilize wireline LAN technology such as gigabit Ethernet LANs, whereas the opaque network is wireless.
- Bottleneck is at the wireless (opaque) network segment, which is the multiplexing segment for several unencrypted user network segments
- The opaque nature of the intermediate network implies very little network visibility or control is available to user (unencrypted) side network management applications.
  - The only knowledge of the opaque network is the pre-existing knowledge of the type of radio and physical layers that are being used, from which it is possible to infer the theoretical “maximum” capacity of the network, i.e., the raw speed in bits per second.
    - For example, given an Ethernet LAN with speeds of 3, 5 or 11Mb/s, it is possible to infer that the maximum capacity on the physical cannot exceed 3, 5, or 11Mb/s, as the case may be.

MANET service assurance and Capacity

Note that the “maximum capacity” mentioned in the previous slide is very different from “available capacity” in MANETs because:

- The available capacity on a given link is the capacity that is “seen” by the applications using that link and is a quantity that fluctuates randomly over time.
- For wireless links, the situation gets even more complicated, because the maximum capacity is not known, even though it is bounded by the theoretical maximum capacity.
- The maximum capacity for wireless links fluctuates over time due to a combination of several factors such as environmental characteristics (absorption, fading, jamming) and the amount of cross-traffic traversing the link, all of which are inherently stochastic processes themselves.

MBAC-based QoS Assurances in MANETs

- One popular approach for mixed (unencrypted-encrypted) MANETs as illustrated in Figure 13, is the use of Measurement Based Admission Control (MBAC) scheme
- Basic philosophy of MBAC scheme is as follows:
  - The performance management component (part of the Network Management System) residing in the un-encrypted (user) side periodically 'measures' the traffic sent into and out of the encrypted (*opaque*) network
    - Measurements typically include loss and delay
  - These measurements are used by the QoS mechanism to infer knowledge of the 'state' of the intervening opaque network
  - The knowledge of system 'state' in turn is helpful to determining whether *additional* traffic can be admitted into the network
  - The knowledge of the system state can also be used to *maintain* assurances to high priority services
    - Low priority services may be pre-empted by the QoS component of the performance management sub-system, should the MANET resources fluctuate and dwindle due to unforseen circumstances

MBAC schemes –relevant prior work

- Variety of MBAC schemes exist in literature – where the focus has been on static, wireline type of networks
- Far less maturity when in comes to MANETs
  - Mainly in the 'research' mode
- The work described in [Valaee & Li - 2002] uses "time-delay" measurements to assess the congestion status of opaque networks.
  - Recall:Two types of measurements that can be collected in networks:
    - "time-related" measures (also referred to as "latency measurements" in the literature), and
    - "information-loss-related" measures (also referred to as "throughput measurements" in the literature).
- If the computed delays are large, this is indicative of a "bad" network, whereas short delays indicate a "good" network.
- While above is an effective approach for wireline networks where bandwidth is plentiful, it suffers from the following severe drawbacks in the MANET environment:
  - It is expensive in terms of the overheads introduced in order to derive latency estimates, and
  - It does not consider multiple service (traffic) classes.

MBAC schemes –relevant prior work - continued

Another relevant work is by [Grossglauser and Tse 1999] and [Breslau et al. 2000]

- Employ MBAC schemes are to characterize the current network load.
- While their algorithms have been shown to perform well in terms of characterization of the current network load, they are not applicable to MANETs, since they assume complete knowledge of and control over the elements in the path of the data packets.
- Recall:
  - Realistic (practical) MANETs will need to operate in a multi-security environment.
  - Consequently, they will contain opaque network segments that do not allow management systems to ‘look into’ them.

PM and End-to-end QoS Assurance in MANETs – A practical solution

- An important requirement of a service (QoS) assurance solution in MANETs is that it adapts to the underlying network dynamics.
- This precludes any service assurance mechanisms based on static rules that dictate how much traffic of different types can be admitted into the network, since such static rules cannot take into account the fact that the network capacity changes dynamically in MANETs.
- We will discuss an adaptive policy-driven solution based on MBAC that
  - Focuses on 'loss' (throughput) related measurements with the aim of
    - Keeping it simple yet effective and resilient
  - Has a Measurement Collection Mechanism (MCM): whose purpose is to collect information (measurements) to assist with determining the network 'state', the knowledge of which in turn will be used by the ACC and QuAM (discussed below) in providing end-to-end (e2e) service assurances in heterogeneous, multi-level-security MANETs
  - Has an admission control component (ACC) whose purpose is to judiciously admit application traffic flows based on service (QoS) requirements, priorities and knowledge of network 'state'
  - Has a quality adjustment mechanism (QuAM): whose purpose is to adapt to the dynamics of the underlying network in an effort to sustain guarantees provided to the admitted applications based on their priorities

An E2E QoS mechanism for MANETs: Measurement Collection Mechanism

- In order to enable the ACC and QuAM to make admission and preemption decisions that are based on the condition of the underlying network, information must be gathered about the network.
- This is the job of the Measurement Collection Mechanism (MCM).
- MCM in turn has the following phases
  - Information collection: Information must be collected about the network in order to feed the QoS assurance decision-making process
  - Information processing and use: Once information has been collected, it must be processed and algorithms must be used to support ACC and QuAM decisions.
- Given the typically small amount of bandwidth available in the opaque network for management traffic, it is critical for the information collection mechanism to minimize the management traffic overhead that it introduces into the MANET.

MANET E2E QoS: Measurement Collection Mechanism – Information Collection

- The information collected needs to reflect the "state" of the bottleneck network (i.e., the wireless, opaque network) so as to facilitate decisions related to QoS admission control and adjustment.
- The "state" of the MANET is a complex function of many factors, including
  - link dynamics, number of nodes in the network, mobility patterns, traffic patterns, etc.
- While a state descriptor that includes all of the above can provide a great amount of information, it could also result in an inefficient and even infeasible solution.
- Additionally, due to the stochastic nature of the network, a complex state descriptor that is a function of the above mentioned items will be very difficult to derive unless otherwise over-simplifying and unrealistic assumptions are made.
  - For example, factoring in mobility will require either a priori knowledge of or assumptions regarding the expected mobility pattern of all of the network nodes, or else will require expensive location tracking devices constantly transmitting location information (and consuming bandwidth in the form of overhead messages) at run-time amongst all the communicating nodes.
  - Similarly, usage of link dynamics assumes a priori knowledge of all possible terrain types in which the communications network will be deployed, which again is not realistic.

MANET E2E QoS: Measurement Collection Mechanism – Information Collection - continued

Information Collection - continued

- Canonical tradeoffs exist with regard to computational complexity, feasibility and accuracy.
- Furthermore, since the underlying network is not guaranteed to be in a “steady state”, it may even be counter-productive to maintain very detailed state information, because
  - The information may change by the time it is propagated throughout the network.
- In light of the above, the following key parameters have been identified to provide the information required to perform QoS actions (namely, judicious admission control and efficient quality adjustment):
  - Bandwidth consumed (also referred to as throughput) between ingress-egress node pairs.
  - Latency between ingress-egress node pairs.

MANET E2E QoS: Measurement Collection Mechanism – Information type

Information Type: Bandwidth-related measure

- The measure under consideration here is throughput between ingress-egress pairs, where ingress and egress nodes refer to the MANET nodes on which flows originate and terminate, respectively.
  - Depending on the direction of the flow, each of these nodes assumes the role of an ingress or egress node.
- Throughput measurements are computed as follows:
  - For every ingress-egress node pair, keep count of:
    - Sent packets: The number of packets per DSCP that exit from the ingress node over a certain policy-defined interval; and
    - Received packets: The number of packets that enter the egress node after traversing the opaque network segment, during the same interval.
- A gross estimate of the throughput over that interval can then be computed using the ratio of the sent packet count to the received packet count.
- This ratio is used to estimate the characteristics of the path between the ingress-egress node pair over the opaque network segment over that period of time.

MANET E2E QoS: Measurement Collection Mechanism – Information type (Continued)

Information Type: Bandwidth-related measure – Points to be noted

- Throughput measurements provide a qualitative indication of how good or bad the path between the ingrees/egress node pair is
  - A ratio close to 1.0 indicates ‘good’ path; Ratio << 1.0 indicates ‘bad’ (lossy) path
- Can be implemented (in practice) via relatively simple mechanisms
  - For efficiency reasons, can aggregate measurement reports, vary the frequency of the measurement reports, and also consider piggy-backing on ‘reverse direction’ flows.
    - Use of adaptive policies help achieve even better overhead management
- Need to be aware that the exact path of the flow between an ingress/egress node pair can (and most likely, will) change over time
- Also need to be aware that the loss may be (a) congestive and/or (b) non-congestive reasons
  - Use of ECN bit can help discern the difference between (a) and (b).
  - Whereas the PM (QoS mechanisms) can help with losses of type (a), (e.g., prioritized pre-emption), not much can be done with type (b) losses

MANET E2E QoS: Measurement Collection Mechanism – Information type (Continued)

Information Type: Latency-related measure

- Another measure that can be used to learn about the condition/state of the opaque network segment is a latency-based measure.
- Such a measure captures the delays incurred along a certain ingress-egress path.
- One way to measure latency is to timestamp the packets when they enter the opaque network via the ingress node, and again when they exit the opaque network segment at the egress node.
- Above method requires a special set of operations on every packet – i.e., each packet has to be opened and a timestamp added to it by the ingress node at which the packet enters the opaque network segment. More specifically,
  - A timestamp field needs to be introduced for this purpose.
  - At the egress node, the time of receipt must be noted in order to be able to compute the time difference between the time of receipt and the time of sending; this difference is the latency incurred through the opaque segment.
  - The egress node computes this latency and periodically sends latency information back to the ingress node.
  - This provides the ingress node with information about the latency when traveling from the ingress to the egress node.

MANET E2E QoS: Measurement Collection Mechanism – Information type (Continued)

Information Type: Latency-related measures – A few points to be noted

- Just like its counterpart, the throughput measure, the latency measure can be used as a basis for making admission control and quality adjustment decisions.
- However, latency measurements tend to impose a higher overhead (as compared to throughput measurements) on the network, in terms of
  - increasing the size of every packet (by introducing a timestamp), and
  - the latency measurements that need to be sent over the air from egress to ingress node.
- Just as with throughput measurements, the frequency of messages sent over the air from egress to ingress nodes can be varied, to control the overhead.
  - Use of adaptive policies to further control overhead
- However, unlike throughput measurements, for latency measurements to be effective, timeliness of measurements is a critical factor.
- This means that there is a need for more frequent exchanges from egress to ingress node to communicate path latencies
  - Potentially larger ‘overheads’
- Finally, similar comments with regard to existence of ‘many possible paths’ between ingress/egress node pairs as mentioned in the context of bandwidth measurements, also apply with regard to latency measurements.

MANET E2E QoS: Measurement Collection Mechanism – Information Usage

Information Usage

- Once information (e.g., throughput, latency) has been collected, dynamic state info graphs (DSIGs) can be built to reflect the state of the path between the ingress and egress node-pairs
- A collection of DSIGs between the various ingress/egress pairs in turn provide a 'coarse' (qualitative) description of the network
- Dynamic State Info Graphs (DSIGs) can be thought of as adaptive representations of the state of the opaque network between ingress-egress pairs, where the adaptation is based on the dynamics of the underlying network.
- In their most general form, DSIGs may be defined based on both bandwidth and latency measurements.
- More specifically, if "throughput" is the measure under consideration, we then have a dynamic throughput graph which in essence is a plot of the throughput of the traffic flows between a given ingress-egress pair as a function of the load on the network.
  - Here the network load is defined as the number of bits per second sent by the ingress node, and the throughput is defined as the number of bits per second received at the egress node.

MANET E2E QoS: Measurement Collection Mechanism – Information Usage

Information Usage – Continued

•Similarly, in the ideal case, the latency-based DSIG (also referred to as a dynamic latency grapy) is a plot of the latency experienced by a flow between a given ingress-egress pair as a function of the load on the network

- Latency is the time (in seconds) that elapses between the sending and the receiving of a packet; latencies are collected and averaged over a time interval for all packets transmitted and received in that time interval.

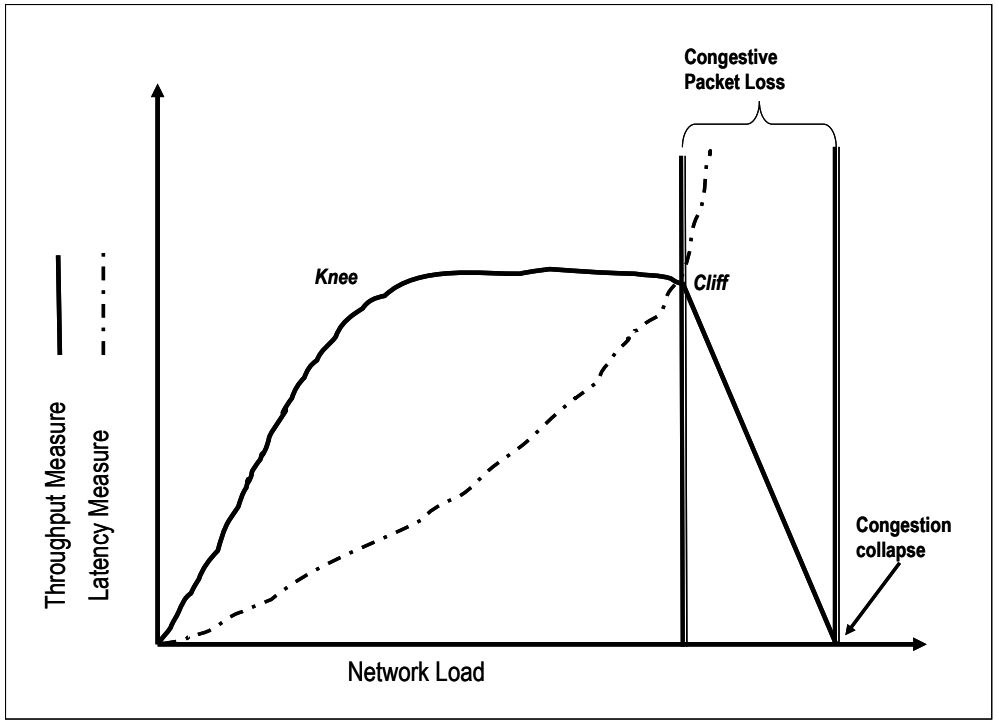

Figure 14: Throughput & Latency vs. Load

MANET E2E QoS: Measurement Collection Mechanism – Information Usage - continued

Information Usage: Network state via DSIGs – Key points

- As the load increases so does the throughput (represented by the solid curve in Figure 14), as expected, up until a certain point called the knee.
- After the knee, the throughput increases far more slowly as compared to the load, indicating packet losses.
- If the load is further increased, then at a point called the *cliff*, the throughput collapses, resulting in the loss of close to all transmitted packets.
- A similar behavior is seen with regard to the latency graph (represented by the dashed curve in the figure).
  - In this case, the delay increases slowly until the knee, then a little more rapidly until the cliff; and when the load is increased beyond the cliff, the delay becomes unbounded.
- As can easily be seen, there is a correlation between the bandwidth throughput graph and the latency graph.
- Due to the stochastic nature of the MANETs, the ingress-egress throughput measurements will vary as a function of time.
- Thus the state info graphs must be updated dynamically, leading to the term *Dynamic State Information Graphs (DSIGs).*
- In essence, DSIGs provide a relatively inexpensive way to capture the “state” of the MANET, which in turn can be used by the ACC and QuAM to provide appropriate service assurances, as discussed next

MANET E2E QoS: Adaptive Admission Control

Adaptive Admission Control in MANETs using DSIGs

DSIGs are used by the Admission Control Component (ACC) to determine whether there is sufficient available capacity to admit new QoS flow requests. An example ACC algorithm is described below.

- Given a new flow request, the ACC first determines whether the sum of the requested bandwidths of all the flows for the requested class of service (including the incoming flow) is within the "quota" for that class of service (assuming that every service class is allocated a certain quota of the overall projected bandwidth).
  - Such a quota is defined via configurable policies for every service class in order to prevent starvation of lower priority classes, and applies to the entire network. However, ensuring that all the flows of a class satisfy the quota over the entire network will lead to unnecessary overhead. Hence the quota is applied to every node.
  - Note that this is an optimistic approach; the conservative approach would be to use a small fraction of the quota as a limit on the flows of a class at a node.
- If the quota is not exceeded, the flow request is treated as described in the following slide
- If the quota is exceeded, the flow request is treated as described in the slide thereafter

MANET E2E QoS: Adaptive Admission Control - continued

Adaptive Admission Control in MANETs using DSIGs - Continued

- If the flow request is within the allocated quota,
  - The ACC looks up the DSIG for the requested class of service between the ingress node and the specified egress node.
  - This DSIG is used to extrapolate the projected traffic that is expected to be *received* at the egress node based on the projected outgoing traffic that is *sent* from the ingress node, by using a DSIG lookup.
    - The projected outgoing traffic is the sum of the bandwidth that is being requested by the new flow request, and the total bandwidth requested by all the currently admitted flows in this class; let this number be X.
    - Then the projected amount of traffic that is expected to be *received* at the egress node is obtained by looking up the value on the *y-axis* of the DSIG corresponding to the value X on the *x-axis.*
      - Note that this lookup may involve extrapolation of the graph if the value X is not within the range of the past QoS operating points' values for *x*.
  - If the projected amount of received traffic is Y, then the ratio Y/X (i.e. the ratio of the projected incoming traffic to the projected outgoing traffic based on the DSIG extrapolation) is referred to as the *projected QoS Operating Point*.

MANET E2E QoS: Adaptive Admission Control – continued

Adaptive Admission Control in MANETs using DSIGs - continued

- This projected QoS operating point is used by the ACC to make its admission decision.
  - The ACC accepts the flow request if the projected QoS Operating Point is greater than or equal to a pre-defined DSIG *threshold*, which is a policy-defined parameter that lies between 0 and 1 for each traffic class.
  - This essentially implies that the network is expected to be in a stable state *after* accepting this request, based on its projected input/output values.
  - The DSIG threshold captures the amount of loss that is acceptable for that traffic class; for example, if a given class consists of only VoIP traffic, then a 5% packet loss is tolerable.
    - The corresponding DSIG threshold is 0.95; and the projected QoS operating point must be greater or equal to 0.95 for a flow to be admitted in this case.

MANET E2E QoS: Adaptive Admission Control - Continued

Adaptive Admission Control in MANETs using DSIGs - continued

- If the ACC determines that the projected QoS Operating Point is less than the corresponding DSIG threshold, or if the flow would exceed the quota for the class, then the ACC consults pre-defined configuration policies to determine whether it should:
  - Reject the flow request;
  - Accept the flow request at a lower priority, should “room” be available for the flow in a lower priority class (this is again determined by using the DSIGs as explained earlier); or
  - Admit the flow as best effort traffic.
- Bootstrap case:
  - What happens when the system starts and no DSIG has been constructed because no data points have been received?
    - In such a scenario, the ACC simply admits flow requests, since the bandwidth pipes are just beginning to get filled.
    - Equivalently, this scenario can be viewed as being similar to operating on the linear portion of a default DSIG – i.e. below the DSIG threshold – where all incoming requests are accepted.
    - At the same time, the ingress-egress nodes begin collecting measurements for admitted flows, so that they can start to gather data points and thereby construct DSIGs.

MANET E2E QoS: Dynamic Quality Adjustment

Dynamic Service Quality Adjustment in MANETs using DSIGs

Given the fact that the capacity of a MANET can vary dynamically, it is necessary for the PM sub-system to periodically assess the congestion state of the network, and to preempt flows as needed if the congestion status of the network exceeds a policy-defined threshold. The following steps are performed for **quality adjustment**:

- The QuAM periodically checks whether the network is in the stable region, i.e. whether the current QoS operating points are within the policy-defined DSIG thresholds, using the latest DSIGs.
- Based on these DSIGs, if the network condition is determined to be stable, the QuAM does not need to take any action.
- If not in the stable region, the QuAM identifies which flows should be preempted (based on the preemption policies), and preempts them. More specifically, the QuAM needs to identify flows that are candidates for downgrading, so that the network can be brought back to the stable region.
  - The meaning of “downgrade” is defined based on policy, and can range from preemption of the flow (i.e. blocking packets from the flow from entering the opaque network) to assigning a lower priority to the flow.
  - In addition, several policies about which flows to downgrade can be specified.
    - As an example, the search for flows to be downgraded can be restricted to flows within the service class for which instability is detected.
    - As another example, an ordering of service class priorities could be specified, and the search starts with the lowest priority service class and works upwards until it reaches the class in which instability is detected. In each service class, the QuAM identifies the flows to downgrade.
- It stops downgrading flows when a sufficient number of flows have been identified for downgrading.
- The resultant network state can then be expected to be in the stable regime.

Outline – Where are we now?

- Introduction to MANETs and MANET Management
- Fault management in MANETs
  - Fault management functions and Operations models
  - Categorization of failure types & Root Cause Analysis
  - Self Healing
    - What is it and why is it important?
    - **Case Studies**
      - **Scenario 1**
      - **Scenario 2**
      - **Scenario 3**
- Performance management in MANETs
  - Performance management functions and Operations models
  - Network Monitoring
  - End-to-End service performance assurance in MANETs
    - Providing QoS in MANETs
- Summary

Summary

- Mobile ad hoc networks (MANETs) are gaining rapid momentum in both military (e.g., NCW) and commercial (e.g., disaster sites) sectors due to
  - Their flexibility and ability of being deployed and functional in “on-demand” situations
- Success of MANETs critically tied to their ability
  - To heal rapidly and seamlessly to unpredictable network failures and provide un-interrupted service (self-healing)
  - To provide service quality assurance to a wide spectrum of applications (e.g., mission critical/platinum to routine/best-effort)
- Above call for ***adaptive*** and ***efficient*** fault and performance management mechanisms in MANETs
  - Adaptive in order to cope up with the dynamism of MANETs
  - Efficient in light of bandwidth scarcity in MANETs
- Reviewed
  - Key MANET Fault & Performance Management-related challenges and operations process models and how they differ from their wire-line counterparts
  - Root cause analyses, self-healing and service assurance mechanisms in MANETs
  - Use of policy-based fault and performance management in MANETs to provide both efficient and timely management

References

- [Wang & Schawrtz – 1993] Wang C. and Schwartz M. (1993). Identification of faulty links in dynamic-routed networks. Journal of Selected Areas in Communications, 11(3);1449-1460, December 1993.
- [Katzela & Schwartz – 1995] Katzela I. and Schwarz M. (1995). Schemes for fault identification in communication networks. IEEE Transactions on Networking, 3(6):733-764, 1995
- [Nygate-1995] Nygate Y.A. (1995). Event correlation using rule and object based techniques. Proceedings of the 4th Intl. Symposium on Integrated network management, pp. 278-289, 1995.
- [Yemini et. al – 1996] Yemini S.A., Kliger S., Mozes E., Yemini Y. and Oshie D. (1996). "High speed and robust event correlation." IEEE Communications Magazine, 34(5): 82-90, 1996.
- [Gopal-2000] Gopal R. (2000). Layered model for supporting fault isolation and recovery. In Proc. of Network Operations and Management Symposium (NOMS), Honolulu, HI, 2000
- [Steinder & Sethi – 2002] Steinder M. and Sethi A.S. (2001). Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system. In Proc. of ICCCN, Scottsdale, AZ, 2001, pp. 374-379.
- [Steinder & Sethi – 2002a] Steinder M. and Sethi A.S. (2002a). End-to-end service failure diagnosis using belief networks. In Proc. of Network Operations and Management Symposium (NOMS),Florence, Italy, 2002.
- [Steinder & Sethi – 2004b] Steinder M. and Sethi A.S. (2004b). Non-deterministic Fault Localization in Communication Systems Using Belief Networks. IEEE/ACM Transactions on Networking, Vol. 12, 5, pp. 809-822, Oct. 2004.
- [Wu-1992] Wu T. (1992). Fiber Network Service Survivability. Artech House, 1992.
- [Kant et. al – 2002] Kant L., Sethi A.S. and Steinder M. (2002). Fault Localization and Self-healing Mechanisms for FCS Networks. In Proceedings of the 23rd Army Science Conference (ASC), Florida, December 2002.
- [Camarillo – 2002] Camarillo G. (2002). SIP De-mystified. McGraw-Hill Publications, 2002
- [Perkins – 2002] Perkins C. ed. (2002). IP Mobility Support for IPv4. IETF RFC 3344, August 2002.
- [Nichols et. Al – 1999] Nichols K., Jacobson V., Zhang L. (1999). A Two-bit Differentiated Services Architecture for the Internet. IETF RFC 2638, July 1999.
- [Valee & Li – 2002] Valaee S. and Li B. (2002). Distributed call admission control for ad hoc networks. In the proceedings of the VTC'02, 2002.
- [Breslau et. al – 2000] Breslau L., Jamin S. and Shenker S. (2000). Comments on the performance of measurement-based admission control algorithms. In the proceedings of Infocom 2000.
- [Grsooglauser & Tse – 1999] Grossglauser M. and Tse D. (1999). Framework for Robust Measurement-Based Admission Control. In the IEEE/ACM Transactions on networking, Vol 7, No 3, June 1999.
- [Chadha & Kant – 2007] Chadha R. and Kant L. (2007), Policy-driven Mobile Ad hoc Network Management, Wiley Interscience Publications, 2007.

For more details…

- R. Chadha and L. Kant, “Policy-Driven Mobile Ad Hoc Network Management”, John Wiley & Sons, ISBN: 978-0-470-05537-3, December 2007.
- lkant@telcordia.com