

Systems Modeling for IP-based Handoff Using Timed Petri Nets

Ashutosh Dutta, Bryan Lyles
Telcordia Technologies
{adutta,blyles}@research.telcordia.com

Henning Schulzrinne
Columbia University
hgs@cs.columbia.edu

Jiacun Wang
Monmouth University
jwang@monmouth.edu

Abstract

The mechanisms and design principles needed for achieving optimized handoff for mobile Internet services are poorly understood and need better analysis. This paper contributes to the general theory of optimized handoff and addresses the need for a formal model that can characterize a mobility event and the associated mobility optimization methodologies. It provides a systematic and formal approach to analyzing a mobility event. After a thorough analysis of the abstract operations associated with several mobility protocols, it determines that these basic handoff operations form a set of discrete events that can be modeled as Discrete System Dynamic System (DEDS). It then uses Deterministic Timed Petri nets to model the mobility event and evaluate the performance under different scheduling schemes. This model also helps to validate optimization methodologies that could lead to a set of design principles for any new mobility protocol as well as evaluate its effectiveness.

1. Introduction

Advent of ubiquitous computing has resulted in the evolution of wireless networks with limited footprint spanning over heterogeneous access technologies. In order to support ubiquitous computing over these networks a mobile node is often subjected to repeated handoff. As a mobile node goes through a handoff process, it is subjected to delay because of the rebinding of its association at several layers of the protocol stack. Delays incurred within each of these layers contribute to the interruption of the active multimedia applications and data traffic within the client. There are several basic operations that are associated with the re-establishment of the binding at several layers. These operations can be affected by several factors, such as access characteristics (e.g., bandwidth, channel characteristics), access mechanism (e.g., CDMA, CSMA/CA, and TDMA), re-configuration of identifiers, re-authentication, re-authorization, and rebinding of security associations at all layers. The handoff process can be classified based on access technology and administrative domain. Based on access technology, it can be categorized as horizontal and

vertical handoff. Horizontal handoff involves a mobile node's movement between the same type of access network (e.g., IEEE 802.11). Vertical handoff includes movement between different types of wireless access networks. Supporting handoffs across heterogeneous access networks, such as CDMA (Code Division Multiple Access), IEEE 802.11, WiMAX (Worldwide Interoperability for Microwave Access) and GPRS (General Packet Radio Service) needs to take into account the diverse quality of service, security and bandwidth parameters associated with each type of access network. Similarly, movement between two different administrative domains poses a challenge since a mobile node needs to re-authenticate and re-authorize in the new domain. The mobility management techniques can be implemented at several layers of the protocol stack, such as network layer, transport layer, and application layer. This paper discusses a generalized systems model to analyze the handoff event, validate systems performance and conduct performance analysis under different optimization schemes.

A few related efforts have attempted to model certain aspects of mobility management. Marshan et al. [1] have used a Petri net-based model to analyze the performance characteristics of wireless Internet access for GSM and GPRS systems. Amadio et al. [2] have modeled IP mobility using a process calculus approach and have applied this to Mobile IPv6 [3]. The process calculi-based method has used a software agent approach for modeling the mobility event. Tutsch and Sokol [4] provide a Petri net-based performance evaluation of bandwidth partitioning meant for quality of service for wireless call. Molina-Ramirez et al. [5] and Jaimes-Romero et al. [6] have used a Petri net-based approach to model resource management in cellular systems. However, none of these papers have attempted to model the systems components of a mobility event. None of these models were intended to systematically analyze the elementary operations involved in a mobility event or serve as a basis for optimizing the mobility event.

Based on the analysis of the mobility event, we develop a formal model that can be utilized to analyze any mobility protocol. We develop the mobility system model by

including the state transitions associated with the basic operations during the handoff. This model can help to evaluate the efficiency of the mobility protocol, make performance predictions, and analyze the tradeoff between performance metrics and resources when a mobility event demonstrates parallelism, optimistic or speculative operations between the tasks.

The rest of the paper is organized as follows. Section 2 systematically decomposes a handoff event. We introduce Petri nets to build the systems model for handoff in Section 3. Section 4 describes different types of scheduling mechanisms for the handoff operations. Performance of these optimizations is evaluated using Petri Net techniques in Section 5. Finally, Section 6 concludes the paper.

2. Mobility system decomposition

Based on the analysis of the basic operations associated with many of the mobility protocols at different layers, we divide the handoff process into six main phases, namely, *handoff initiation*, *network discovery*, *network selection*, *network attachment*, *configuration*, and *media delivery*. Each of these operations consists of several sub-operations. We briefly describe the details of these sub-operations in the following sections.

2.1 Handoff initiation

Depending upon who initiates the handoff and who has the primary control over the handoff, a handoff process can be either mobile-initiated, network-initiated, mobile controlled or network-controlled [7]. During the first phase, the mobile node or the network node determines the need for the handoff based on the network condition, and initiates the triggers to start the handoff operation. For example, during a mobile-initiated handoff, based on Signal-to-Noise Ratio (SNR) or the quality of service of the ongoing application, the mobile node decides about the impending handoff and starts the network discovery process to determine the best available network to connect to.

2.2 Network discovery

The second phase is the discovery phase, where the mobile node or network discovers possible new points of attachment. This phase involves discovering both the neighboring networks and the resources within the network. Once the target network is discovered, several resource parameters within the target network are retrieved including channel number, bandwidth, encryption algorithm, authentication server, registration

server, and configuration server. The resource discovery process helps the mobile node to configure itself with the proper channel number and proper authentication parameters so that it can communicate successfully. Jari et al. [8] provide an overview of network discovery and selection, and its applicability to handoff. Based on the type of access technology, the discovery process could be passive where the node listens for network announcements or active where the node solicits the network announcements. Based on the type of network, the discovery process involves different operations, such as discovery of a new Location Area (LA) in case of GSM or discovery of new Routing Area (RA) in case of GPRS. For IP-based networks, the discovery process can span all the layers and it may involve cell, subnet or administrative domain.

Cell discovery is typically based on the type of access technology that provides different means of discovering the networks and resources. As an example, GSM uses BCCH (Broadcast Control Channel); CDMA uses a pilot channel and IEEE 802.11 uses active and passive scanning to discover the new point-of-attachment. Based on the type of access characteristics of the target network, the discovery of the appropriate network takes different amounts of time. In case of IP-based mobility involving subnets, upper layer detection mechanisms, such as foreign agent advertisement or router advertisement, can help discover the new points of attachment at layer 3 and above.

2.3 Network selection

During network selection, a mobile node or a network entity analyzes the information discovered about its neighboring networks, and then selects a network to connect to. The selection may be based on criteria such as required QoS, cost or user preferences. An appropriate selection mechanism helps the overall resource optimization process that can increase the probability of successful handoff to the target network.

2.4 Network attachment

After the mobile node has selected the target network, it attempts to connect to the new network point of attachment. Several lower layer events, such as the availability of the new point of attachment or sudden connection is usually provided to the upper layers so that any further handoff related functions can be expedited during a handoff event. The IETF is currently working on standardizing a protocol for Detection of Network Attachment (DNA) [9] that involves mechanisms both at layer 2 and layer 3 and can notify the upper layer about the network. The IEEE 802.21 working group is currently

defining several event service primitives. “Link Up” and “Link Down” are examples of such primitives that can be used to provide the link status of lower layers and expedite the handoff process. For example, since the layer 2 association takes place before any upper layer operations, it helps to send a layer 2 event notification, such as “Link up” to execute the upper layer mobility functions, such as layer 3 configuration. Signal-to-Noise Ratio (SNR) threshold can possibly generate such an event notification. Thus, an event notification from lower layer helps to establish a successful link in the new point of attachment in an expedited manner.

2.5 Configuration

The fifth phase is the configuration phase. This phase helps the mobile node to prepare the new path in the new point of attachment. During this phase, the mobile node obtains a temporary identifier and establishes the mapping of its connection identifier with the appropriate network entity in the network. The configuration phase can be categorized into the following sub-phases.

(a) Identifier configuration

Identifier configuration allows a mobile node to acquire a new temporary connection identifier either at layer 2 or at layer 3 in the new point of attachment of the network. For example, it could include Care-of-Address (CoA) in case of IP-based mobility or TMSI (Temporary Mobile Subscriber Identity) in case of GSM. The identifier configuration process involves obtaining a new identifier, testing the uniqueness of this identifier and finally assigning it to the mobile node's interface. In general, completion of these processes requires a series of signaling messages between the mobile node and a server in the network.

(b) Registration

Registration is a process of establishing the mapping between the node's permanent identifier, such as URI (Universal Resource Identifier) or home address and a temporary identifier, such as care-of-address obtained during identifier configuration. An optimized or hierarchical registration process helps to expedite location management and provides faster delivery of the new data.

(c) Authentication, authorization and accounting

The authentication process allows the mobile node to establish its identity with the access network in order to get access to the network resources. An authentication process involves a signaling exchange between the mobile node and the authentication server in the network. Successful completion of authentication helps to generate the key that encrypts the data for protection. Each

mobility protocol uses different mechanisms for authentication: a GSM mobile node uses SRES and A3, while for 802.11 access networks, the mobile node could use open system authentication, shared authentication Wired Equivalent Privacy (WEP) or stronger authentication (802.11i). Fathi et al. [10] demonstrate how different authentication mechanisms affect the association delay and transport delay at layer 2. Similarly, layer 2 access independent authentication protocols, such as PANA (Protocol for carrying Authentication to Network Access) [11] add delay during authentication at upper layers. Georgiades [12] shows that it takes up to 4 seconds to complete the authentication and authorization process. EAP (Extensible Authentication Protocol) [13] over layer 2 can provide IEEE 802.1x-based authentication and EAP-TLS (Transport layer Security) [14] can provide upper layer authentication. At the time of reconnection, re-authentication process adds to the handoff delay. During inter-domain mobility, the authentication process is typically followed by authorization process that involves interaction with authorization server, such as AAA.

(d) Security association

A security association can be defined as a secure channel between two endpoints that applies a security policy and keys to protect information. Before a new communication path is established between the end-points, the mobile node needs to authenticate itself and then establish a security association with other network nodes. Establishing a security association always involves a key distribution procedure that includes exchange of messages between the mobile node and a centralized server. Upon a successful derivation of the key, the mobile node can encrypt the data for protection. Security associations may take place at several layers of the protocol stack. For example, in the IEEE 802.11 environment, a 4-way handshake is needed between the mobile node and the access point to generate the PTK (Pairwise Transient Key) that can be used for encryption. For IP-based network, ISAKMP (Internet Security Association and Key Management Protocol) [15] helps establishing the IPSEC security association.

(e) Binding update

Binding update is the process by which a mobile node can update its identifier in the new network in order for the data to be rerouted to the new destination. As the mobile node connects to a new point of attachment and obtains a new temporary network identifier (e.g., TMSI in GSM, COA in Mobile IPv6, FA-COA in Mobile IPv4) in the new network, it needs to update the correspondent host or home agent so that the packets can be routed to the new destination. This process associates the new network identifier with the permanent identifier of the

mobile node. Until the re-association of the new identifier is complete, the in-transit data is still directed to the old network and is considered lost in the absence of any optimization mechanism, such as buffering or packet forwarding. In some cases, this binding update needs to be authenticated. For example, MIPv6 [3] allows return routability procedure and adds two additional messages, CTI (Care-of-test Init) and HTI (Home-test Init) to obtain the binding key so that binding update can be authenticated as well. This process contributes to the additional delay for the binding update procedure.

2.6 Media delivery

A media delivery phase is the last phase of the handoff process. This phase involves re-routing of the media so that the route of data packets moves from old path to the new path. Delivery of data changes from the old path to the new path. Once the binding update is complete, data from the correspondent node gets routed to the mobile node's new location. Media can take either a direct or indirect route between MN and CN. In one way, the media delivery can take place using the direct path between the CN and the MN. In indirect delivery, the media is forwarded by a network entity such as home agent that provides a level of indirection. In both cases, the in-flight data can be captured and redirected to the new point of attachment. The media rerouting process may include several elementary operations, such as encapsulation, decapsulation, tunneling, buffering, and store-and-forward. During the media re-routing process, in-flight data may get lost or may get delayed because of these operations.

Table 1: Multilayer operations in IP-based handoff

Handoff Operations	Layer 2	Layer 3	Application Layer
Discovery	Scanning	Router discovery	Server discovery
Network Attachment	Layer 2 Beacon	Router Solicitation	AAA server advertisement
Authentication	Open Auth EAPOL	IKE, PANA	S/MIME
Security Association	4-Way handshake	IPSEC	TLS SRTP
Configuration	ESSID MAC	DHCP Stateless	SIP URI
Binding Update	Cache Update	Update CN, HA	SIP ReINVITE
Media Delivery	IAPP	Encapsulation Buffering Forwarding	Direct Routing

There is also certain overhead associated with each of these operations. Thus, there is a need to optimize these operations to ensure that the media delivery is not

delayed. Optimization techniques for the media delivery are often defined as route optimization methodologies. Depending upon the type of mobility, handoff related operations take place at several layers. Table 1 summarizes how the basic operations of a mobility event are taken care of across different layers in an IP-based environment with 802.11 as the access media. While optimization of each of these operations can contribute to the overall optimization of the handoff event, scheduling of different tasks across layers can also lead to variety of optimization techniques.

3. Petri net model for handoff

The mobility event can be viewed as the perturbation to the steady state of a communicating node that may affect different layers in the protocol stack. As a communicating node is subjected to handoff, it goes through a series of discrete states before it attains a steady state by returning to the connected state. Each of the basic operations that are described in Section 2 can be modeled as a transition event where the mobile node moves from one state to another.

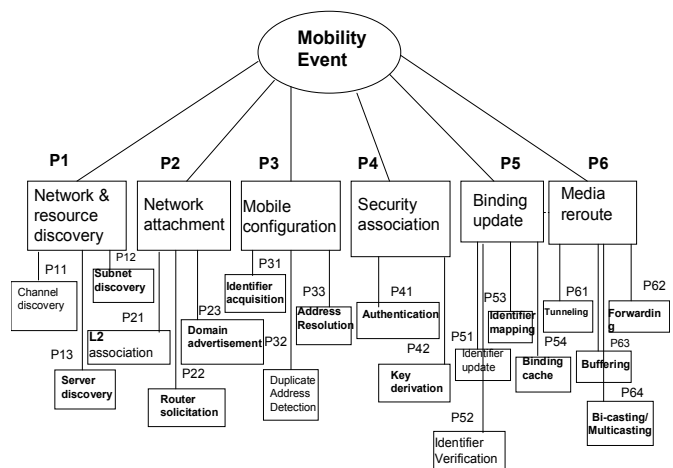


Figure 1: System decomposition of mobility event

Figure 1 shows the decomposition of the mobility event and shows several sub-processes that constitute each of these states, P1 through P6. A data dependency analysis [16] between these sub-processes can determine the level of concurrency that is possible among the operations during handoff event.

A mobile node's communication is interrupted because of the delay associated with each transition. Discrete Event Dynamic System (DEDS) corresponds to a type of system where the state space is discrete and state changes are driven by external or internal events. A mobility

event can thus be modeled as a discrete event dynamic system, since it exhibits concurrent, sequential, and competitive activities (i.e., two operations competing for the same resources) among many of the several handoff operations that are part of the mobility event.

Mobility event modeling can be viewed as analogous to modeling of Flexible Manufacturing Systems (FMS) as both exhibit event-driven operations. Zuberek et al. [17] model and analyze simple schedules for manufacturing cells. Similar techniques can be applied to conduct performance analysis for the mobility systems model. We use Deterministic Time Petri nets [18] to model the mobility event and derive the relevant optimized models by applying the appropriate concurrent and proactive mechanisms. Compared to other similar methodologies such as process calculus, Petri nets provide a graphical and mathematical formalism and are more intuitive. There are also numerous sophisticated tools available for Petri net modeling and analysis. We introduce some of the basic Petri net primitives that can be used to model the handoff. Figure 2 shows Petri net representation of some of these primitives that can be used as the fundamental building block to schedule the handoff operations and derive the optimizations. These primitives include operations such as sequential, concurrent, conflict, merging, mutually exclusive, data dependency, confusion and priority.

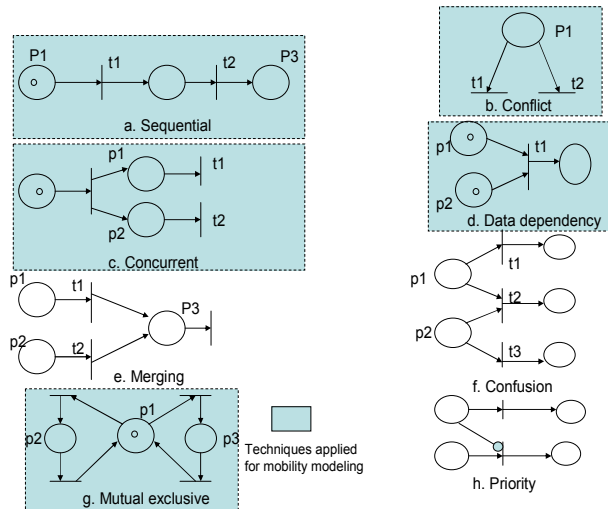


Figure 2: Representation of Petri-net primitives

Figure 3 illustrates Petri net representation of possible sequence of execution for a pair of processes Pa and Pb that could be part of a mobility event. Timing, t_a and t_b are the amount of time associated with the execution of processes Pa and Pb, respectively. Amount of time t_c

denotes the time elapsed between these two processes. Thus, ordering of sequence of execution of sub-tasks during a mobility event could affect the overall cycle time, although amount of system resources needed for completion of these tasks may vary.

Figure 4 shows a high level formal framework of how discrete states associated with a mobility event can be represented using a Timed Petri net approach [19]. We represent the primitive operations of the handoff event in terms of places and transitions, and then validate the optimization techniques associated with each of these operations. However, this figure does not highlight the inherent optimizations or concurrency that may be possible between the sub-processes.

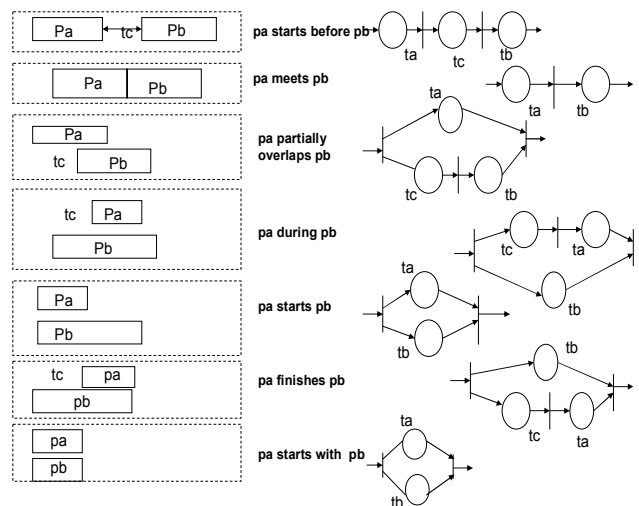


Figure 3: Petri net representation of a sequence of events

In Figure 4, each of the places P0-P7 represents various stages of the mobility event as described in Section 2 and places P_B , P_M and P_P represent shared resources such as network bandwidth, memory, and processing power, respectively, that are utilized during any mobility event. Network bandwidth is a resource that is shared by multiple mobile clients, whereas memory and processing power are the shared resources used by the same mobile node for multiple operations within the mobile. Transition t_i represents the time taken to complete each set of operations between the stages. Each of these stages (P0-P7) is modeled as several sub-systems of the complete system representing the mobility event. In Petri net modeling, a transition is enabled after a token is made available from the places representing shared resources. Once an operation completes, these tokens are sent back to the places that represent the shared resources, thus making the resources available for other operations.

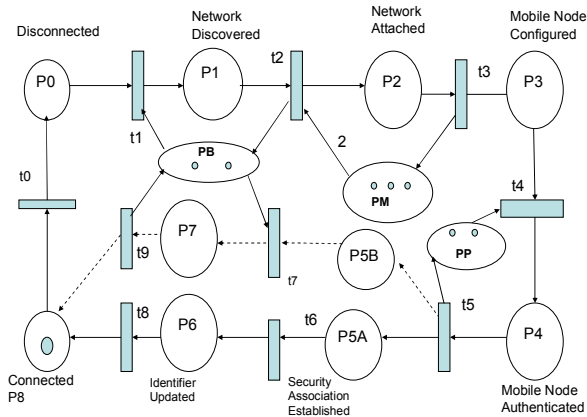


Figure 4: Petri net based modeling of IP-based handoff

The path shown using the dotted lines representing place P7 illustrates an example of parallelism such as a hierarchical binding update (t6) that occurs in parallel with a regular binding update (t7). After a security related operation (t5) is over, both hierarchical binding update (t6) and regular binding update (t7) can take place in parallel. P5A and P5B are the intermediary places in this model after transition t5 is fired.

Table 2: Places of Petri-net model for IP-based mobility

Places	Description
P0	Mobile node is in disconnected state
P1	Network and resources discovered
P2	Target network selected
P3	Mobile node is configured and registered
P4	Mobile node is authenticated
P5A,P5B	Security association is established
P6	Binding update is complete
P7	Intra-domain binding update is complete
P8	Mobile is connected state
PB	Bandwidth resources
PM	Memory resources
PP	CPU resources

Table 2 provides the description of the places that represent various stages of handoff event and shared resources. Table 3 represents transitions associated with Figure 4. Transitions represent the delays associated with each of these handoff operations. Although a deterministic delay is assumed for transition here, this framework could be applicable to other types of delay distribution. The transition time will vary depending upon the processing speed and shared resource available. Ideally, shared resources have tokens that become

available before a transition is fired, leading to next stage in the handoff process.

Table 3: Transitions of Petri-net model for IP-based mobility

Transition	Description
t0	Mobile node gets disconnect trigger
t1	Mobile node discovers the network and resources at the new PoA
t2	Mobile node selects the network
t3	Mobile node goes through configuration and registration
t4	Mobile node goes through authentication process
t5	Mobile node goes through key derivation and security association process
t6	Mobile node goes through binding update process
t7	Mobile node goes through hierarchical binding update
t8	Data gets redirected to the mobile node
t9	Network buffering during handoff

The hierarchical property of Petri nets [20] can be applied to each of these transition processes to study the interaction at sub-process level.

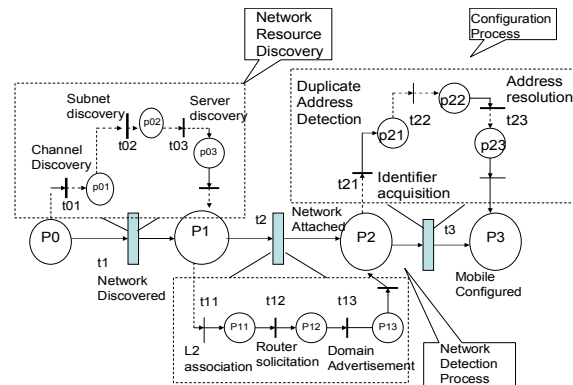


Figure 5: Hierarchical modeling using Petri net

For example, as shown in Figure 5, transition process t1 (Network resource discovery) consists of several sub-processes (places) such as channel discovery (p01), subnet discovery (p02) and server discovery (p03) that are connected by several sub-transitions, such as t01, t02, t03, and t04. Similarly, Figure 5 also shows how the hierarchical nature of Petri net is applied to the network attachment (t2) and the configuration processes (t3).

The Petri net model representing the general mobility systems is actually a decision free Petri net [18]. A Petri net is decision free, if and only if for each place in the net, there is exactly one input arc and one output arc. In a decision free Petri net, a minimum cycle time is an indicator of maximum performance. The cycle time is represented as $C = \max T_k/N_k:k=1,2,3...q$, where T_k is the sum of the execution times of the transmissions in circuit k and (N_k) is the total number of tokens in the places in circuit k and q is the number of circuits in the net. In case of a system model representing a mobility event, these values can vary depending upon the number of transitions and sequence of transitions involved during a cycle.

Figure 6 shows a subnet level view of a mobility systems model as represented by a modular Petri net. It includes the state transitions at each layer and illustrates the component-level interaction within each layer and across layers. Interaction between several sub-processes within each of the layers that are in shown in Table 1 are illustrated here.

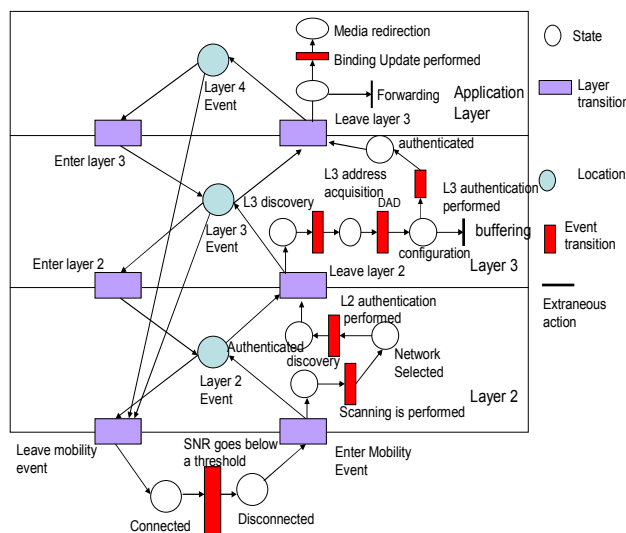


Figure 6: Layered approach to Petri-net mobility modeling

The reduction of the number of transitions, minimization of time for each of the state transition, parallelization of state transitions within each layer and across layers will contribute to the overall optimization of the handoff process. Figure 6 also highlights how cross layer optimization can be used to speed up the handoff process. During cross layer optimization, handoff operations do not need to take place in sequence across layers. As an example, a layer 2 discovery process can trigger layer 3 discovery process resulting in parallel operation of two events and thus reduces the duration of the overall discovery process.

4. Scheduling of handoff operations

In Petri net models, a cycle time represents the delay during the handoff process under resource constraints and thus can be attributed to the overall efficiency of the system. However, scheduling of execution of these processes that constitute a mobility event plays an important role in determining the overall cycle time and systems performance.

A Petri net-based model can be used to analyze various types of mobility events, such as intra-subnet, intra-technology, inter-subnet, and inter-technology handoff. Corresponding Petri net based optimization models can be derived by applying the optimization techniques to the generalized mobility model. These techniques can be applied to the processes that are part of the overall system or in a hierarchical manner to each of these sub-processes. In this paper, we primarily categorize handoff optimization techniques based on sequential, concurrent, and proactive modes of scheduling of events and model these in Petri nets. Depending upon the type of scheduling technique, the systems resources expended during a specific operation will vary over a period of time. While a sequential handoff operation takes more time compared to a proactive or concurrent operation, the optimized models using concurrent or proactive operations may need to exhaust more resources for a given period of time.

In order to conduct a performance analysis, we have initially considered two handoff related operations, namely discovery and authentication. We apply three different scheduling mechanisms to these two handoff related operations and study the overall performance. Figures 7, 8, and 9 illustrate how these two specific handoff operations for IEEE 802.11 can be represented in a Petri net model using sequential, concurrent and proactive optimization techniques, respectively. Optimality of the system performance is obtained by comparing the handoff performance (cycle time) and resource utilization (number of tokens) for these handoff methodologies.

Figure 7 shows a Petri Net model that represents when discovery and authentication related operations are performed in sequence. The number of tokens for each type of operation varies depending upon the amount of resources expended during each of these operations.

In general, scanning is part of layer the 2 discovery process and is followed by layer 2 authentication, 4-way handshake and finally, the association. P0, P1, P2, P3 and P4 are places that represent different states of discovery and authentication. Shared resources are represented by

places such as P_B , P_M and P_P that represent bandwidth, memory and processing power, respectively. Number of tokens in these shared places represent amount of resources expended during each of these operations. For example, one token can represent 100 kb of bandwidth from the place P_B that represents the shared resources of bandwidth.

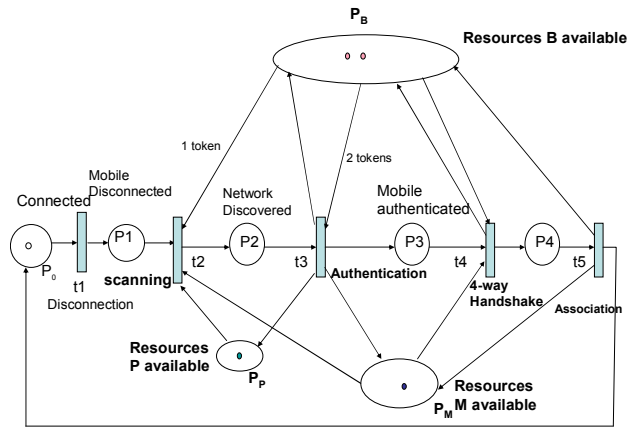


Figure 7: Sequential handoff operations

Figure 8 shows the Petri net model when parts of two handoff related operations scanning and authentication take place concurrently at the expense of additional resources such as bandwidth due to more signaling messages. This speeds up the overall handoff operation but consumes more shared resources during a given time.

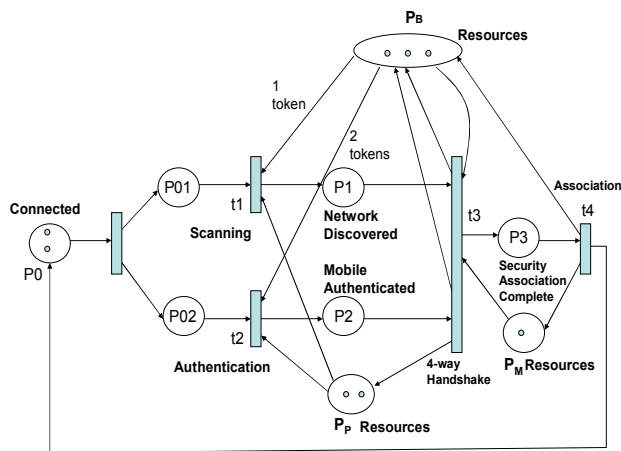


Figure 8: Concurrent handoff operations

Figure 9 illustrates the Petri net model handoff related operations are performed proactively. The mobile intends to move from its current network to the target network. Many of the handoff related operations such as discovery of the target network elements and authentication with the target network elements are performed ahead of time while the mobile is in the current network. Thus, shared

resources (e.g., access bandwidth or memory) are utilized in the current network and some additional resources from the target network are used due to operations such as tunneling and proactive IP address caching. P_{B1} , P_M and P_D are shared resources that are used in the current network and P_{B2} and P_P are shared resources expended in the target network.

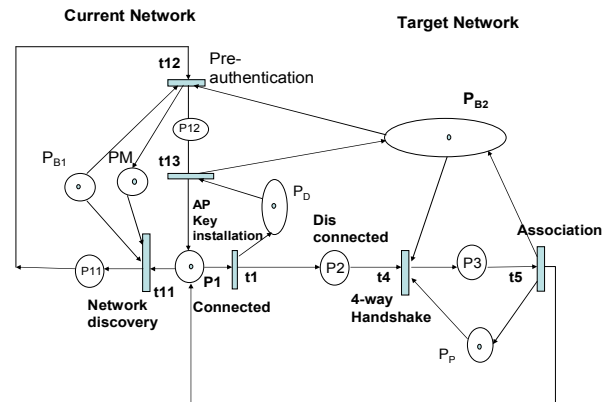


Figure 9: Proactive handoff operations

5. Performance evaluation

The systems performance of a Petri net model for a mobility event can be verified in several ways. We illustrate two scenarios here.

In one scenario, the minimum cycle time can be obtained from the Petri net model by investigating the number of circuits, the number of transitions, and delay associated with each transition. Thus, the performance requirement expressed in cycle time C can be satisfied if and only if $CN_k - T_k \geq 0$ for all circuits in the net. In the second scenario, the token loading matrix and transition matrix are obtained based on the markings of the Petri net model and the associated values of the transitions. Then, the Floyd algorithm [21] is applied to validate the systems performance by computing the shortest distance between every pair of places.

Similar methods can be applied to compute the cycle time and overall systems performance of the handoff event demonstrating different types of optimization techniques, such as hierarchical binding update, proactive discovery and configuration, and anchor-based security association that are modeled using generalized Petri net. We briefly calculate the cycle time and verify the systems performance based on three scheduling techniques when applied to two basic handoff operations – discovery and authentication. Experimental results of these two operations were used for Petri net modeling.

5.1 Cycle-time-based approach

Table 4 shows the transition times t_1 , t_2 , t_3 , t_4 and t_5 for different primitive handoff operations associated with discovery and scanning processes obtained from our experiments [22].

Table 4: Results from cycle-time-based approach

Transition	Handoff operation	Time for operation		
t1	Disconnection Trigger	5 ms		
t2	Scanning	400 ms		
t3	Authentication	50 ms		
t4	4-way handshake	10 ms		
t5	Association	5 ms		
Optimization schedule	Relevant loop in Petri Net	D_i	N_i	Max D_i/N_i Minimum cycle Time.(ms)
Sequential	p0t1p1t2p2t3p3t4p4t5p0	470	1	470
Concurrent	p0t1p0t1t2p1t3p3t4p0	420	1	420
Proactive	P1t1P2t4P3t5P1	17	1	17

We assume a handoff delay requirement of 100 ms to support a real-time application. We evaluate the overall cycle time when different schedules are applied to these handoff operations and verify if the system is conformant to the delay requirement. It appears from the results that although time for concurrent operation is smaller than time for sequential operation, proactive operation is the only operation that satisfies the delay bound of 100 ms under this constraint.

5.2 Using the Floyd algorithm

In order to verify the systems performance of a mobility event and to determine if the system satisfies the desired requirement for cycle time, one can generate a Place matrix (P), Transition matrix (Q) from the Petri net model. We describe briefly how the elements of these matrices are formed. Entry (A, B) in the matrix P equals “x” if there are “x” tokens in place A, and place A is connected directly to place B by a transition. Entry (A, B) in the transition matrix “Q” equals t_i if A is an input place of transition t_i , and B is one of its output places. Entry (A, B) contains symbol “w” if A and B are not connected directly. Given a threshold value of cycle time C, one can generate a distance matrix CP-Q. Then, using the Floyd [21] algorithm, one can determine matrix S. By inspecting the values of the diagonal elements of matrix S, it is possible to determine if the system satisfies the desired system performance. There are three cases to figure out the systems performance of the given system: 1) If all diagonal entries of matrix S are positive (i.e., $CN_k - T_k > 0$ for all circuits), the system performance is

higher than the given requirement, 2) if some diagonal entries of matrix S are zeros and rest are positive (i.e., $CN_k - T_k = 0$ for some circuits and $CN_k - T_k > 0$ for the other circuits) the system performance just meets the given requirement, 3) if some diagonal entries of matrix S are negative (i.e., $CN_k - T_k < 0$ for some circuits), the system performance is lower than the given requirement.

Figures 10 and 11 represent the matrices when the Floyd algorithm is applied to verify the systems performance of the mobility event using sequential and proactive scheduling (Figures 7, 8), respectively. Values from the mobility event involving discovery and authentication are used to build the token loading matrix P and transition time matrix Q. First element for P matrix is P_{00} and the last element is P_{77} . Distance matrix and S matrix are then derived from these two matrices. By inspecting the values of the S matrix in Figure 10 that reflect sequential scheduling, it is found that at least one of the diagonal elements is negative. Thus, this specific sequential scheduling cannot meet the desired systems performance of cycle time of 100 ms. In order to meet the desired performance level, faster facilities could be used to speed up the transition time or more tokens (resources) could be used in the shared places, thus increasing the level of concurrency.

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad Q = \begin{pmatrix} w & 5 & w & w & w & w & w & w & w \\ w & w & 400 & w & w & w & w & w & w \\ w & w & w & 50 & w & 50 & 50 & 50 & w \\ w & w & w & w & 10 & 10 & w & w & w \\ 5 & w & w & w & w & 5 & 5 & w & w \\ w & w & 400 & 50 & 10 & w & 50 & 50 & w \\ w & w & 400 & w & 10 & 10 & w & w & w \\ w & w & 400 & w & w & w & w & w & w \end{pmatrix}$$

Token loading matrix Transition Time matrix

$$CP-Q = \begin{pmatrix} \infty & 95 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & -300 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & -50 & \infty & -50 & -50 & -50 & \infty \\ \infty & \infty & \infty & \infty & -10 & -10 & \infty & \infty & \infty \\ -5 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & -200 & 150 & 190 & \infty & 150 & 150 & \infty \\ \infty & \infty & -300 & \infty & 90 & 90 & \infty & \infty & \infty \\ \infty & \infty & -300 & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix} \quad S = \begin{pmatrix} \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}$$

Distance matrix S Matrix

Figure 10: Results from sequential scheduling

Figure 11 shows the matrices for proactive scheduling based on the transition times obtained from our experiments. By inspecting matrix S in Figure 11, it appears all the diagonal elements of the matrix are positive. Thus, by applying the Floyd algorithm, it is verified that the proactive scheduling, when applied to discovery and authentication processes, satisfies the systems performance of minimum cycle time of 100 ms. Alternatively, there are several automated tools such as TimeNet [23], STPNplay [24] that can be used to model the handoff behavior, capture the systems performance

and evaluate the performance characteristics of the mobility protocols and associated optimizations.

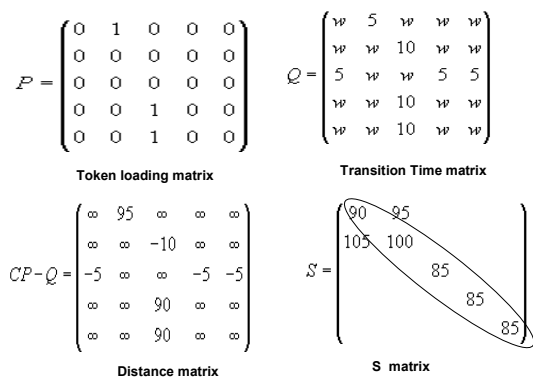


Figure 12: Results from proactive scheduling

6. Conclusions

The Petri net-based mobility system model introduces a formal approach to analyze the interaction between the primitive operations associated with a handoff event in an IP-based environment such as the mobile Internet. The handoff model allows to predict for both un-optimized handoff and specific optimization methodologies under resource constraints. Using Petri net primitives we carried out performance evaluation to verify the system performance under three different scheduling techniques. This model allows network designers to determine the tradeoff between improved handoff delay and resources expended. Having the ability to predict the handoff performance, wireless service providers and network designers can choose the type of mobility protocol and relevant optimization technique suitable for any specific deployment scenario and application requirement. This model can also help to enhance the existing mobility protocols or design new ones.

References

- [1] A. Marsan et al., On Petri Net-based Modeling Paradigms for the Performance Analysis of Wireless Internet Access, *IEEE International Workshop on Petri nets and Performance Models*, Sept. 2001.
- [2] R. Amadio and S. Prasad, "Modelling IP Mobility," *CONCUR'98: Concurrency Theory: 9th International Conference*, Sept. 1998.
- [3] D. Johnson, C. Perkins, and J. Arkko, Mobility support in IPv6, RFC 3775, Internet Engineering Task Force, June 2004.
- [4] D. Tutsch and J. Sokol, "Petri Net based Performance Evaluation of USAIAs Bandwidth Partitioning for the Wireless Cell Level," *Proceedings of the 9th international Workshop on Petri Nets and Performance Models PNPM'01*, Sept. 2001.
- [5] C. Molina-Ramirez, D. Munoz-Rodriguez, and E. Lopez-Mellado, "Modelling and analysis of telecommunication cellular systems using Petri nets," *Systems, Man, and Cybernetics, IEEE International Conference on Humans, Information and Technology*, vol. 3, 1994.
- [6] F. Jaimes-Romero, D. Munoz-Rodriguez, C. Molina, and H. Tawfik, "Modeling resource management in cellular systems using Petri nets," *Vehicular Technology, IEEE Transactions on*, vol. 46, no. 2, pp. 298 - 312, 1997.
- [7] J. M. (Ed.) and M. K. (Ed.), "Mobility related terminology," RFC 3753, Internet Engineering Task Force, June 2004.
- [8] J. Arkko, B. Aboba, J. Kohonen, and F. Bari, "Network discovery and selection problem," RFC 5113, Internet Engineering Task Force, Jan. 2008.
- [9] J. Choi, "Goals of detecting network attachment in IPv6," RFC 4135, *Internet Engineering Task Force*, Aug. 2005.
- [10] H. Fathi, K. Kobara, S. Chakraborty, H. Imai, and R. Prasad, "On the Impact of Security on the Latency in WLAN 802.11," *IEEE Globecom Nov 2005*.
- [11] D. Forsberg, Protocol for carrying authentication for network access (PANA), RFC 5139, Internet Engineering Task Force, May 2008.
- [12] M. Georgiodes, Context transfer support for IP-based mobility management "CCSR awards for research excellence," 2004.
- [13] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and E. Levkowetz, Extensible authentication protocol (EAP), RFC 3748, Internet Engineering Task Force, June 2004.
- [14] B. Aboba and D. Simon, PPP EAP TLS authentication protocol, RFC 2716, Internet Engineering Task Force, Oct. 1999.
- [15] D. Maughan, M. Schertler, M. Schneider, and J. Turner, Internet security association and key management protocol (ISAKMP), RFC 2408, Internet Engineering Task Force, Nov. 1998.
- [16] F. Cruz Filho et al., "Using Petri Nets for Data Dependency Analysis," *IEEE Conference on Systems, Man, and Cybernetics*, Oct. 2000, Nashville, TN.
- [17] W. M. Zuberek and W. Kubiak, Timed petri nets in modeling and analysis of simple schedules for manufacturing cells, *Elsevier Science, Computers and Mathematics with Applications*, vol. 37, 1999.
- [18] T. Murata, *Petri nets: Properties, analysis and applications*, Proceedings of IEEE, Apr. 1989.
- [19] Jiacun Wang, "Timed Petri Nets, Theory and Application," Kluwer Academic Publishers, 1998.
- [20] W. Zuberek, *Hierarchical analysis of manufacturing systems using Petri nets*, vol. 4, Oct. 2000.
- [21] R. Floyd, *Algorithm 97: Shortest path*, Communications of the ACM, vol. 5, no. 6, p. 345, 1962.
- [22] R. Lopez, A. Dutta, Y. Ohba, and H. Schulzrinne, Network-layer assisted mechanism to optimize authentication delay during handoff in 802.11 networks, in *ACM Ubiquitous, (Philadelphia, PA)*, June 2007.
- [23] A. Zimmermann, J. Freiheit, R. German, and G. Hommel, Petri net modeling and performability evaluation with TimeNET 3.0, *Lecture Notes in Computer Science: Computer Performance Evaluation: Modeling Techniques and Tools*, vol. 1786, pp. 188-202, 2000.38
- [24] STPN Play, "A stochastic Petri net modeling and simulation tool," www.stpnplay.com.