Distributed Call Admission Control for Ad Hoc Networks

Shahrokh Valaee and *Baochun Li*

Abstract— **This paper introduces a distributed call admission controller for ad hod networks. The call admission controller is based on service curve provisioning. Service curve reflects the status of network and depends on the number of active nodes, their activity index, and the back-off procedure used for contention resolution. The service curve along with the aggregated traffic function can be used to calculate maximum delay and maximum backlog. We assume that the call requests are granted if the service curve is bounded below by some non-decreasing deterministic function which is called the universal service curve. The universal service curve is independent of the number of nodes and traffic fluctuation and acts as a worst-case reference curve. All users willing to establish a new connection should compare the performance of network to the universal service curve. A call request is accepted if the true service curve stays above the universal service curve.**

I. Introduction

The IEEE 802.11 standard uses ad hoc network as an underlying infrastructure in wireless local area networks (WLANs) [1]. Mobile ad hoc networks comprise a set of autonomous nodes that communicate over a wireless environment. The control of system is distributed with each node being a decision-maker in the overall management procedure.

Nodes willing to transmit data in an ad hoc network will join the WLAN and will send data by competing with other active nodes. Due to the distributed nature of ad hoc networks, the centralized management of resources is not plausible. All nodes in an ad hoc network should cooperate on bandwidth allocation and resource partitioning. The ubiquitous structure of management schemes in an ad hoc network may allow destructive operation of malicious sources. Sources can be very greedy and might be able to drive the network performance below an unacceptable threshold. A distributed *call admission control* (CAC) might alleviate such an instance of network operation.

In this paper, we will propose a measurement-based CAC for ad hoc networks. Several measurement-based call admission controllers have been proposed in the literature of high-speed networking (see for instance [2] [3] [4] among others). There are certain issues that have to be addressed for the measurement-based CAC in an ad hoc network. The first concern is that the collected information should truly reflect the network status. In fact, the network should be "observable" through the accumulated data. Ad hoc networks usually suffer from the *hidden terminal* problem. Due to this problem, some network indices — such as the available bandwidth — may not be totally observable to all nodes. The second issue is that the activity of all users — hidden or otherwise — should be reflected in the collected data. A new caller should be able to measure the activity level of nodes. The call admission controller should also make sure that the acceptance of the new call will not drive the quality-of-service (QoS) of all ongoing sessions below an acceptable threshold. In order to satisfy this requirement, an index of the status of present connections should be available to the probing node. Due to the distributed nature of an ad hoc network, a direct exchange of QoS indices will be a tedious task. The collected data should somehow reflect the support of QoS in the network.

In this paper, we will use "active" probing. In active probing, the calling node transmits a sequence of probing packets. All probing packets have the same size and are generated with a greedy source. The node will compete with all active nodes for available empty slots. A probing packet is transmitted when the calling station wins the competition. In the presence of collision, all participating nodes, including the probing node, should initiate a backoff procedure.

We will extensively use the concept of "service curve" [5] [6] [7]. Using the probing packets, we will propose a mechanism that can be used to estimate the service curve. The estimated service curve is then compared to a deterministic lower bound that will be indicated as the "universal service curve". A call request will be accepted into the network if the induced service curve, under the admission of the new call, is not smaller than the universal service curve. We will decompose the input traffic into the "conforming" and "nonconforming" parts. The conforming part comprises the packets that their acceptance into the network will not derive the service curve below the universal service curve. The nonconforming packets are the ones that do not satisfy this property. We will propose that the nonconforming traffic be discarded at the node. This will throttle the nonconforming traffic and will maintain the network performance within an acceptable region.

II. Service Curve Probing

The *carrier sense multiple access* with *collision avoidance* (CSMA/CA) along with the *distributed coordination function* (CDF) is the default setup of the physical layer in IEEE 802.11 standard. In CSMA/CA, any backlogged node will initiate a handshake procedure with its peer node. The handshake will start by the transmitter sending a *request-to-send* (RTS) packet to its peer node. If the destination is ready to receive the data packet, it will respond by sending the *clear-to-send* (CTS) packet. Upon the arrival

The authors are with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ontario, Canada M5S 3G4. Email: valaee@comm.utoronto.ca and bli@eecg.utoronto.ca.

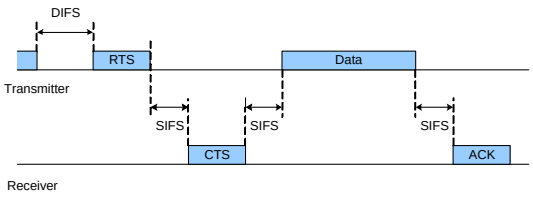

Fig. 1. The handshaking procedure of CSMA/CA.

of the CTS packet, the transmitter will transmit the data packet over the wireless channel. The receiver will acknowledge the arrival of the packet by issuing the *acknowledgment* (ACK) packet. This procedure has been illustrated in Fig. 1.

In this paper, we will assume that the calling node creates a greedy pattern of probing packets. In fact, as soon as a probing packet leaves the transmit buffer, a new probing packet will be generated and will be placed in the probing queue. We assume that the length of probing packet is small. This assumption is enforced to result in a minimal impact on the available bandwidth of the system. Small-size probing packets will consume a small portion of available bandwidth. In the sequel, we will propose an algorithm that can be used to measure the service curve of the network. Our technique is motivated by the approach of [8].

Denote the time instants at which a probing packet is delivered to the destination by $\{\tau_i\}$. Since the probing packets are transmitted in a greedy manner,

$$\delta_i \triangleq \tau_i - \tau_{i-1} \tag{1}$$

will be the total delay of the ith probing packet in the head of the probing queue. The delay will be a function of the size of the probing packet and the waiting delay in the head of the probing queue. In fact, we can write

$$\delta_i = b + w_i \tag{2}$$

where b is the total transmission time of the probing packet and w_i is the total waiting time of the ith probing packet. w_i is a function of the number of active nodes, the size of packets transmitted between two consecutive transmission of probing packets, and the length of backoff window. Note that, in general, w_i is a function of b since the number of active sources might increase over the extent of b seconds. However, for small b, we might assume that w_i is independent of b. In DCF, b will be given by

$$b = T_{\text{DIFS}} + 3T_{\text{SIFS}} + T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{ACK}} + T_{\text{PRB}} \tag{3}$$

where T_{DIFS} is the duration of the DIFS period, T_{SIFS} is the duration of the SIFS period, T_{RTS} is the transmission time of RTS packet, T_{CTS} is the transmission time of CTS packet, T_{ACK} is the transmission time of acknowledgement, and T_{PRB} is the transmission time of the probing packet (see Fig. 1).

We visualize the whole ad hoc network by two queues and a round-robin scheduler. Fig. 2 illustrates this instance.

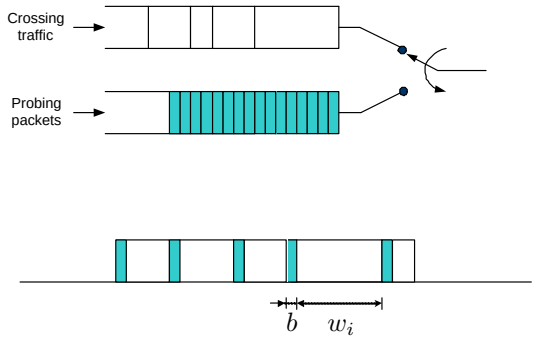

Fig. 2. The whole network is modelled by two queues and a round-robin scheduler.

The probing packets will be inserted into one of the queues and the crossing traffic will be placed in the other queue. The server will transmit the packets in a round-robin fashion. In this model, the size of the packets of the crossing traffic will include the actual packet transmission time of active sources, collision periods, and backoff windows. In fact, the size of the packets of crossing traffic encompasses all bandwidth consumptions due to the activity of sources, contention resolution mechanism, and hidden terminal impairments. If the network is lightly loaded, w_i will be small. On the contrary, if the ad hoc network is heavily loaded, w_i will usually be large.

Let $\Delta_i^{(k)}, k = 1, \ldots, K$ be the total waiting delay for transmitting a batch of k probing packets starting at the ith packet, where K is the maximum number of probing packets. Then $\Delta_i^{(k)}$ can be represented by

$$\Delta_i^{(k)} \triangleq \sum_{j=i}^{k+i-1} w_j. \tag{4}$$

Throughout, we consider a stationary environment in which the probability distribution function of the delay elements $\{w_i\}$ is constant. We further assume that $\{w_i\}$ is an independent identically distributed (i.i.d.) random sequence.

In the present paper, we will use the concept of service curve to assess the status of network. The service curve will be calculated using the fact that the time required to serve a sequence of k probing packets will be $\Delta_i^{(k)}$ seconds. In a stationary environment, the statistical behavior of $\Delta_i^{(k)}$ will be independent of time index i. We will use the sequence $\{\Delta_i^{(k)}\}$ to obtain the service curve.

The service curve is defined as a percentile of the delay elements. Let

$$T_k^\epsilon \triangleq \inf\left\{\tau \,\middle|\, Pr(\Delta_i^{(k)} > \tau) \leq \epsilon\right\}. \tag{5}$$

The ϵ-*effective service curve* is defined as

$$S_\epsilon(t) \triangleq \frac{t - T_k^\epsilon}{T_k^\epsilon - T_{k-1}^\epsilon} + k. \tag{6}$$

From this definition, we have

$$S_\epsilon(T_k^\epsilon) = k. \tag{7}$$

Using the fact that

$$T_k^\epsilon \leq T_{k+1}^\epsilon, \tag{8}$$

one infers that $S_\epsilon(t)$ is a nondecreasing function.

We will use service curve as a quantifier of the status of network. The service curve, as we have defined in (6), indicates the relative load of network by integrating the various parameters into an index that corresponds to the waiting time at the head of a typical queue in the ad hoc network. The waiting time, in fact, reflects the available resources. For a lightly loaded network, the service curve will be close to the vertical axis and for a heavily loaded network it will be close to the horizontal axis. We will use the proximity of service curve to the vertical axis as an index of traffic load. In particular, the proposed CAC will accept a call if the service curve, upon the acceptance of the new call, is not very far from the vertical axis.

III. Call Admission Control

We will devise a call admission procedure by introducing the concept of the *universal service curve*. The universal service curve is a deterministic function independent of the source activity and traffic fluctuations. We have used a similar approach in [9] to instrument a network partitioning mechanism and call admission control.

The CAC algorithm will accept the call if the induced service curve will stay above the universal service curve. Mathematically, the call is accepted if

$$S_\epsilon(t) \geq \bar{S}_\epsilon(t) \tag{9}$$

for all $0 \leq t \leq W$, where $\bar{S}_\epsilon(t)$ is the universal service curve and W is the temporal extent of the maximum window over which the prospective call is backlogged. The call should be rejected if (9) is violated. Note that the universal service curve is a fixed curve for each network and is distributed among all nodes during the process of registration.

Using (9) for call admission control guarantees that an accepted call will not drive the performance of the network below the prescribed threshold. The preassigned level of performance is indicated by the universal service curve. If all active nodes cooperate and keep the true service curve above the universal service curve, the maximum delay for each connection will be bounded by the maximum horizontal distance between the aggregated input traffic and the universal service curve. The universal service curve is in fact a reference point for QoS support. All QoS parameters should be compared to the universal service curve. If (9) is satisfied, the universal service curve will reflect the worst case performance of the network.

Now assume that a node with the prospective aggregated traffic $A(t)$ is willing to establish a connection through the network. The aggregated traffic can be represented as the stair-case function

$$A(t) = \sum_i L_i 1\{a_i < t\} \tag{10}$$

where L_i is the size of the ith packet, a_i is the arrival instant of the ith packet, and $1\{.\}$ is the identifier function; $1\{\mathcal{A}\} = 1$ if the predicate $\mathcal{A}$ is true and $1\{\mathcal{A}\} = 0$ if $\mathcal{A}$ is false. Without loss of generality we assume that the first packet arrives at time 0. In order to evaluate (9), we have to study the backlog periods of the incoming traffic. The traffic is backlogged over an interval $[s, t]$ if the queue is nonempty during that period.

The call admission procedure should be employed on all backlogged periods. During each backlogged period, one should make sure that the total service curve, once the new call is accepted, will not be decreased beyond the limit marked by the universal service curve. This can be performed by investigating all packets inside a backlogged stream.

Here, we assume that the packets in a backlogged period are numbered in the order of their arrival with the first packet in the queue being the packet no. 1. The numbering will be reset to zero as soon as the queue is empty. The call admission procedure starts by investigating the first packet in the backlogged queue. Note first that every single packet will create a backlogged interval. The length of the interval depends on the waiting time of the packet in the queue, its total size, and the rate of transmission. In order to project the worst case scenario, we assume that the service given to the packet is indicated by the true service curve $S_\epsilon(t)$. In other words, for the first packet in the queue the total waiting time will be T_1^ϵ. If the packet is transmitted over the wireless medium, the total delay will be

$$D_1 = W_1 + d_1 \tag{11}$$

where W_1 is a generic random variable indicating the waiting time inside the queue, and d_1 is the transmission time of the packet which can be represented by

$$d_1 = T_{\text{DIFS}} + 3T_{\text{SIFS}} + T_{\text{RTS}} + T_{\text{CTS}} + T_{\text{ACK}} + \frac{L_1}{C} \tag{12}$$

where L_1 is the size of the first packet and C is the channel transmission rate. We further assume that W_1 is independent of the length of the transmitted packet L_1. Using the measurements of the probing phase, one can conclude that with the probability $1 - \epsilon$, the total delay will be smaller than $T_1^\epsilon + d_1$. The packet is called "conforming" if

$$T_1^\epsilon + d_1 \leq \bar{T}_1^\epsilon \tag{13}$$

where $\bar{T}_1^\epsilon$ is the time index of the universal service curve at 1, that is

$$\bar{S}_\epsilon(\bar{T}_1^\epsilon) = 1. \tag{14}$$

The packet is pronounced "nonconforming" if (13) is not satisfied. Our strategy in this paper is to drop nonconforming packets. Therefore, if the first packet is found nonconforming, it will be dropped and will not be transmitted over the channel. If a packet in the backlogged queue is dropped, the numbering of all subsequent packets are reduced by one. In fact, if the dropping packet was the first packet in the queue, the second packet will move to the head of the queue and it will take the number 1. All other packets will also be shifted one slot closer to the head of the

queue and will decrease their number by 1. If the packet is conforming, it will be transmitted over the wireless channel and no change in the numbering scheme will be performed.

The test of conformance should also be performed for all subsequent packets. Assume that the first packet is conforming and it will be transmitted at the due time over the wireless channel. The total delay for the second packet can then be represented by

$$D_2 = W_1 + W_2 + d_1 + d_2 \tag{15}$$

where W_1 and W_2 are two random variables indicating the competition intervals, and d_1 and d_2 are the transmission time of the first and the second packets, respectively. Our probing scheme indicates that with probability $1 - \epsilon$, the total competition time will be bounded by T_2^ϵ. We will use this fact to compare the total delay to the corresponding index in the universal service curve. In particular, we call the second packet conforming if

$$T_2^\epsilon + d_1 + d_2 \leq \bar{T}_2^\epsilon \tag{16}$$

and nonconforming if (16) is violated. Such as before nonconforming packet will be dropped and the subsequent packet numbering will be updated. Conforming packets will remain in the queue and will be transmitted in the due time.

This procedure will continue for all packets in the backlogged queue. If the ratio of the nonconforming traffic to the total submitted traffic is larger than a preselected threshold, the connection cannot be supported by the network and it should be terminated. The acceptance threshold is a parameter of design and will depend on the characteristics of the projected traffic. In an extreme case, one might be willing to reject the call as soon as a single nonconforming packet is identified. The caller might then decide to postpone the transmission to the future hoping that the present congestion at the network will be rectified.

It is important to note that the conformance test for a packet can be performed at the time of the packet arrival. In fact, as soon as a packet is available at the transmit queue, the corresponding algorithm can be initiated to identify wether it is conforming. Using this property will protect the network against voracious and noncomplying sources. Unlike other probing techniques, the proposed algorithm will throttle a malicious source before any damage can be done.

One might also be willing to allow a small percentage of the nonconforming traffic be transmitted over the network. This will permit the traffic which is marginally nonconforming be accepted into the network. Since the service curve is a random function, most probably the marginal nonconforming traffic will not retain the service curve below the universal service curve over a long period of time. The allowable percentage of the nonconforming traffic is of course a parameter of design.

IV. Numerical Study

In this section, we present the simulation results obtained using the NS2 network simulator. In our simulation, we consider 23 nodes randomly located inside an area of 600×600 meters. The transmission power of the nodes is large enough so that they all can communicate over a single-hop link. The MAC layer is defined as the default mode of operation in the IEEE 802.11 standard with all nodes using CSMA/CA contention resolution procedure as illustrated in Fig. 1.

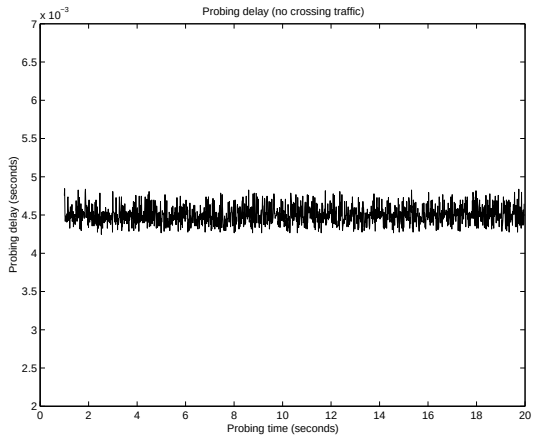

Fig. 3. The waiting time of the probing packets in the absence of crossing traffic.

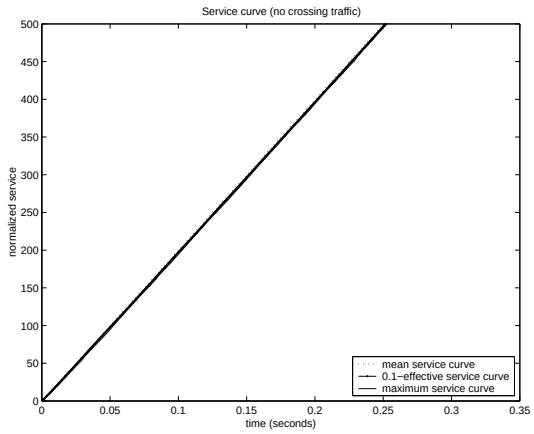

Fig. 4. The service curve of the probing packets in the absence of crossing traffic.

We study several scenarios. In all examples, node 1 will transmit a sequence of probing packets to node 2 which is an idle node. The probing is performed in the application layer with the size of the probing data set to 1 byte; due to the encapsulation procedure in the underlying layers, the size of the probing packet will usually be much larger. In the first example, we send a stream of the probing packets in the absence of the crossing traffic. The probing is performed over an interval of 20 seconds. Probing starts at $t = 1$ second. Fig. 3 illustrates the waiting delay of the probing sequence. Note that the delay is almost constant. The service curve, which is approximately a linear function, is shown in Fig. 4.

In the second example, we study a scenario with six TCP connections; the size of the packets are random. The connections are activated in $t = 0.5$ second and such as before

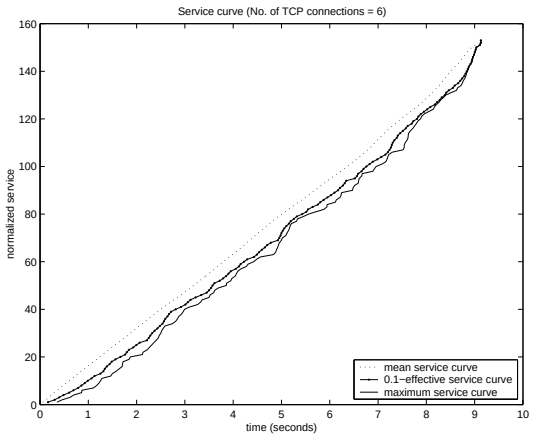

Fig. 5. The service curve in the presence of 6 TCP connections.

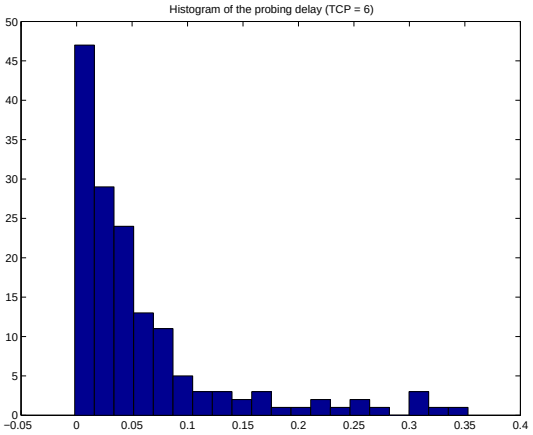

Fig. 6. The histogram of the waiting time of the probing packets in the presence of 6 TCP connections.

the probing starts at $t = 1$ second. The mean service curve, the 0.1-effective service curve, and the maximum service curve are shown in Fig. 5. As expected, the service curve is an increasing function. Note also that the mean service curve approximately resembles a linear function. This is an interesting observation since the mean service curve of an exponential random variable will be a linear function. Fig. 6 illustrates the histogram of the probing delay. The histogram indicates that the probing delay can be approximated by an exponentially distributed random variable.

In order to show that the service curve truly represents the load of the network, we have simulated a scenario with variable number of crossing traffics. The traffic has changed from 0 to 10 TCP connections. For each scenario, we have measured the mean service curve. The results have been illustrated in Fig. 7. Note that the service curve decreases with increasing the traffic of the ad hoc network. This property indicates that the service curve is an appropriate measure of the network load.

V. Conclusion

We have proposed a call admission procedure for wireless ad hoc networks. The technique is a measurement-based call admission controller using a sequence of probing

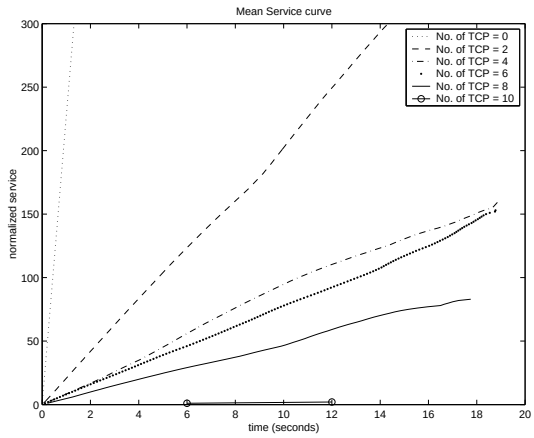

Fig. 7. The mean service curve as a function of the number of TCP connections.

packets. It is assumed that the size of the probing packets is small and therefore the probing sequence does not consume a large amount of bandwidth. We have used a service curve approach. The service curve is defined as the amount of service given to the user over a backlogged interval. We have discussed — and also illustrated by numerical examples — that the service curve truly reflects the performance of the network. A lightly loaded network has a service curve which is close to the vertical axis and a heavily loaded network has a service curve close to the horizontal axis. This observation can be used to propose a call admission procedure: a call request is accepted if the service curve is considerably far from the horizontal axis. We have represented the threshold by the universal service curve. In our proposed scheme, if the true service curve is above the universal service curve, the call will be accepted; otherwise, it will be rejected.

References

[1] ANSI/IEEE 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999.

[2] V. Elek, G. Karlsson, and R. Rönngren, "Admission control based on end-to-end measurements," in *Proceeding IEEE INFOCOM*, 2000.

[3] G. Bianchi, A. Capone, and C. Petrioli, "Throughput analysis of end-to-end measurement-based admission control in IP," in *Proceeding IEEE INFOCOM*, 2000.

[4] J. Qiu and E. W. Knightly, "Measurement-based admission control with aggregate traffic envelopes," *IEEE/ACM Trans. Networking*, vol. 9, pp. 199–210, April 2001.

[5] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services network: the single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 334–357, June 1993.

[6] R. L. Cruz, "Quality of service gurantees in virtual circuit switched networks," *IEEE J., Select., Areas Commun.*, vol. 13, pp. 1048–1057, August 1995.

[7] H. Sariowan, *A service curve approach to performance guarantees in integrated-services networks*. PhD thesis, University of California, San Diego, 1996.

[8] C. Cetinkaya, V. Kanodia, and E. W. Knightly, "Scalable services via egress admission control," *IEEE Trans. on Multimedia*, vol. 3, no. 1, pp. 69–81, 2001.

[9] S. Valaee, "A methodology for virtual network partitioning: The deterministic approach," *Preprint*.