Resource Pooling in Network Virtualization and Heterogeneous Scenarios using Stochastic Petri Nets

Rainer Schoenen and Halim Yanikomeroglu
Department of Systems and Computer Engineering, Carleton University, Canada
{rs,halim}@sce.carleton.ca

***Abstract*—Wireless cellular networks are undergoing severe changes due to the ever increasing demand of data rate. Additionally, the demand is more and more heterogeneous (imbalanced) in time and space. Sudden peaks in demand at a certain location have to be absorbed by the network. While operators traditionally over-provisioned their own separate network capacity in order to reduce the blocking and overload probabilities, this approach seems no longer economically viable. Instead, the idea of network virtualization (NV) emerged. One aspect of NV is that resources from all operators are pooled together. Shared and virtualized resources can be better distributed among all users compared to having separate subsets of users to separate subsets of resources. This holds especially if the demand is imbalanced among the operators, as shown in this paper. In this paper the stochastic Petri net (SPN) paradigm is used to provide with a compact model of NV resource pooling (RP). In contrast to the equivalent but tedious analysis of Markovian systems the SPN approach allows a quick numeric performance evaluation with tool support, thus offering a strong modeling advantage. The scenarios analyzed here are networks of separate operators and resources, compared to one virtualized network. In a second step the scenario includes heterogeneity in demand, i.e., a load imbalance between the providers and results show much higher gains in this unbalance.**

***Keywords*—network virtualization; resource pooling; Erlang**

I. Introduction

TRAFFIC in wireless cellular networks shows a trend of demand increasing faster than the supply side, as well as increasing heterogeneity [1]. One of the recently emerging concepts to achieve an additional gain on the supply side is network virtualization (NV) [2]. One of its features is that two or more distinct network operators can pool their resources together (their licensed band) to form one larger "virtualized" network [3]. The user terminal (UT) will be dynamically and transparently assigned to one of the networks and use the resources of that network. As the probability of congestion in one network is independent of the probability of congestion in the other, the probability that both are in congestion is significantly lower. This assumes independent (uncorrelated) statistics, which may not always hold. However, in the average case, a temporary overload in one network can be absorbed by the free capacity of the other(s). In queueing theory this is related to the statistical multiplexing gain [4].

For analyzing such networks, traditionally either Markovian queueing theory or simulation is used. These two approaches are very different in several ways, including the modeling efficiency versus details, and accuracy. In this paper, an intermediate approach is motivated, where the stochastic Petri net (SPN) paradigm is used to generate the corresponding Markov chain (MC) [5]. However, as its strong advantage, the modeling is performed by a high-level graph-based structure, and the numerical evaluation is performed with adequate tool-support [6]. This allows fast modeling and time-to-results (order of a few minutes), combined with an analytic precision suitable for rare-event studies (high precision still for probabilities of 10^{-7} and below).

SPNs are known to generate MCs [5] (e.g., Fig. 1). There is little application to communications systems, but a few noticeable examples include networks [7], protocols [8], WiMAX [9], wireless scheduling [10], ad-hoc networks [11], radio channels [12], and flow control [13].

In this paper, networks with separate operators and resources are compared to a virtualized network having the sum capacity by pooling all resources together. The scenario is extended to include heterogeneity in demand, i.e., an imbalance between the providers. In order to concentrate on the strengths of SPN, conditions and assumptions are simplified, for example, (video) traffic is assumed as fluid-flow, i.e., constant during a session and off otherwise. Session arrivals are Poissonian. The channel capacity is constant (long term average), i.e., fading and mobility are not explicitely modeled. Short term fluctuations are assumed to be absorbed by appropriate playout buffering. This allows a comparison with the analytic Erlang-B and Erlang-C approaches [14]. A closed-form expression exists for the blocking probability in a single system, but more complex systems have to be studied by a steady-state MC analysis. SPN offers a significant modeling advantage here and the SPN graphs are much easier to understand than the multi-dimensional MC which correspond to it.

This paper demonstrates the use of SPN to model NV resource pooling and its multiplexing gains, plus an extension to model heterogeneous/unbalanced traffic load situations. One result shows that NV is even more beneficial under heterogeneous load scenarios. This paper does not cover other NV aspects, .e.g., distinct sets of different QoS requestors, brokers and providers, or cost aspects and does not provide a framework how to implement NV.

The paper organization is as follows: Section II is a quick SPN intro. Section III introduces the basic model and its extension to study virtualization. The next section provides numeric results for the separate network and virtualized network case.

This work is supported in part by Huawei Canada Co., Ltd., and in part by the Ontario Ministry of Economic Development and Innovation's ORF-RE (Ontario Research Fund - Research Excellence) program.

II. Stochastic Petri Nets

Petri nets (PN) are a graphical and mathematical tool suitable to model complex systems with a state. Systems can be be described and studied when they are concurrent, asynchronous, distributed, deterministic or stochastic. Subclasses of PN are finite state machines and marked graphs. Many aspects of flow charts and description languages can be modeled with PN. There is plenty of literature on the underlying graph theory, liveness analysis, reachability set and other properties [15]. A PN is defined as a directed, weighted, bipartite graph having two sets of nodes called *places* (P_i) and *transitions* (T_j). Places are drawn as circles, transitions as boxes. Arcs are directed and called input arcs if they connect from a place to a transition, output arc otherwise. A convenience notation is the notion of a disabling arc, drawn with a small circle at the end of the arc. Input arcs connect certain P_i to T_j with weight w_{ij}, output arcs connect T_j to P_i with multiplicity v_{ij}. Together they form the incidence matrix $\mathbf{D} = [v_{ij} - w_{ij}]$. When a PN consists of M transitions and N places, $\mathbf{D}$ is a $M \times N$ matrix (m rows, n columns). Places can contain an integer number of *tokens* (dots), all of which constitute the state called *marking* $\vec{m}$. The initial marking $\vec{m}_0$ is the start state. The notation $\#P_i = m_i$ means the current number of tokens in place P_i. An atomic action is the *firing* of a transition T_j which changes the marking to

$$\vec{m}_k = \vec{m}_{k-1} + \vec{t}_{k-1} \cdot \mathbf{D} \tag{1}$$

with the firing vector $\vec{t}_{k-1}$ which is all zero except a single '1' at the j.th index. The firing rule is basically: A transition can be fired if all places connected to input arcs contain enough (w_{ij}) tokens and all places connected via disabling arcs are empty.

Stochastic PN (SPN) extend the paradigm to model time, essentially by assigning each transition a firing rate (in the continuous time case (CT)) or a firing probability (in the discrete time case (DT)). Here we will focus on the CT case only. Then the firing rates are given by $\vec{\Lambda} = \lambda_1, ..., \lambda_m$ which can be marking-dependent. λ_j is the inverse of the average firing time t_j. Firing times are exponentially distributed, therefore memoryless, and the resulting reachability graph (RG) forms a Markov chain (MC) [5]. All arcs in the RG are annotated with the λ_j of the transition T_j responsible for the state change. The matrix $\mathbf{R}$ of all arc rates can then be used to calculate the steady-state solution of the MC and from that all other performance metrics can be obtained. More on this in [5].

III. Erlang Model and NV Extension

The scenario used here aims at showing a performance advantage between having one virtualized network compared to separate distinct networks. The model introduced here consists of two parts: The traffic or session generator side (demand) and the server or network side (supply). In the Figures 2 and 3 the session generation is on the left, and the service on the right side. In accordance with our previous work [14] the assumptions are as follows:

- Fluid-flow traffic model is used, i.e., in state ON a constant rate is generated, and rate zero in OFF.
- For allowing the Erlang equivalent all the sessions have the same rate (e.g., video of 1 Mbit/s).
- High rate video is the dominant application in future networks [16]. Voice will be negligable. Data as best-effort needs other approaches.
- Sessions are generated independently with an average rate of λ.
- The generation process is Poissonian, i.e., the Markov property holds.
- The average interarrival time of new sessions is $1/\lambda$.
- Each network has capacity for m parallel sessions (e.g., enough OFDMA resource blocks).
- The length of each session is exponentially distributed with average length d.
- There can be up to $C = m$ active sessions, each with a dedicated "server" with session completion rate $\mu = 1/d$.
- Any session request which would require more than m servers is queued and waits for a free "channel".
- Situations with one or more parked session in the queue are called overload condition, or congestion.
- The queue of waiting sessions can be finite or infinite. Here we assume infinite (Erlang-C).
- A M/M/C queueing model describes the basic Erlang-C MC (Figure 1).
- The bottleneck question becomes more relevant in when heterogeneity is introduced with Eq. 4

The parameters used and their values for analysis are given in Table I.

TABLE I. Parameters for the Erlang scenarios

Parameter	var	assumption,value
Connection/call arrival rate	λ	variable
Average connection holding time, e.g., video session duration	$d = \mu^{-1}$	240 s
Offered traffic in Erlangs means average load	$u = \lambda \cdot d$ ρ	5...45 10...90%
Number of trunks/lines/circuits/resources, capacity	$C = m$	50
Max. number of sources	n	200
Heterogeneity factor	h	0...50%

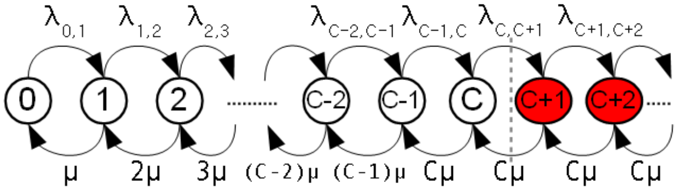

Fig. 1. Markov chain for an Erlang-C scenario. The service depends on the number of active sessions (state index i) by $i \cdot \mu$. The aggregate arrival rate $\lambda_{i,i+1} = \lambda$ is constant (negative exponential interarrival times). The states $C+1$ and above represent overload and can only be served by capacity $C = m$ [14].

The Erlang-C result for the probability of waiting (overload) is known to be given by

$$P_w = \frac{\frac{u^C}{C!}\frac{C}{C-u}}{\sum_{i=0}^{C-1}\frac{u^i}{i!} + \frac{u^C}{C!}\frac{C}{C-u}}, \tag{2}$$

$$u = \lambda \cdot d. \tag{3}$$

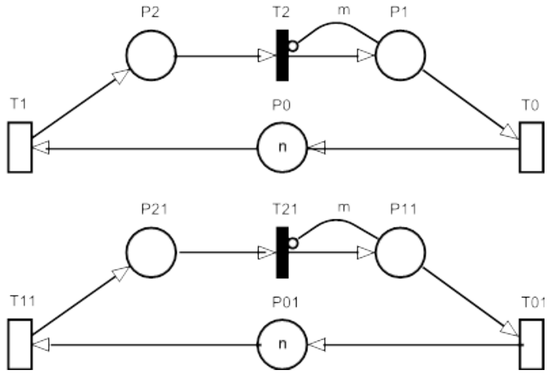

Fig. 2. SPN model for two separate networks (Erlang-C scenario). The traffic load of network 1 and 2 are set to be $(1 - h) \cdot u$ and $(1 + h) \cdot u$ respectively, thus 1=Low and 2=High load.

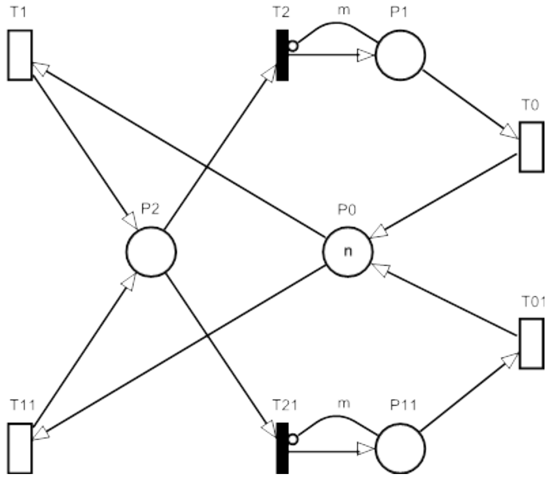

Fig. 3. SPN model for a virtualized network, aggregating the resources of two networks together. New sessions are pooled in P2, but then served either in network1 (T2) or network 2 (T21). The heterogeneity factor h cancels out, thus the load is $2 \cdot u$. This SPN graph can easily be extended to contain more than two partner networks.

Figure 2 depicts two independent SPNs for the Erlang-C scenario. In order to highlight the SPN operation, we will explain the first SPN in Figure 2 [14]. The timed transitions $T1$ and $T0$ represent the generator and server aspects, respectively. Their rates are $\lambda_{T1} = u/d$ and $\mu_{T0} = 1/d$ respectively, where d is constant (average session length) and u is the load in Erlangs. The immediate transition $T2$ is a gate between the waiting sessions in $P2$ and the active sessions in $P1$. As long as there are less than m active sessions = tokens in $P1$ ($\#P1 < m$), $T2$ accepts all pending sessions in $P2$ to be transferred to $P1$. As soon as the capacity m is completely utilized, this gate closes and session tokens begin to wait in place $P2$. This is the cause for the waiting time discussed later. Those user sessions wait for free capacity. The place $P0$ is only required to limit the state space, but with $n \gg m$; this does not change the statistics of interest. Figure 1 shows the corresponding MC for this model. Tool support [6] automatically generates the MC via the reachability graph of the underlying Petri net within seconds.

For the numerical analysis in this paper, the capacity of each network is assumed to be $C = m = 50$ Erlangs, so that the theoretical sum capacity would be 100 Erlangs. Having two separate networks would be less efficient than having one network with the sum capacity. This is what will be shown in this paper. This is not trivial when we modifiy the traffic balance later in (4).

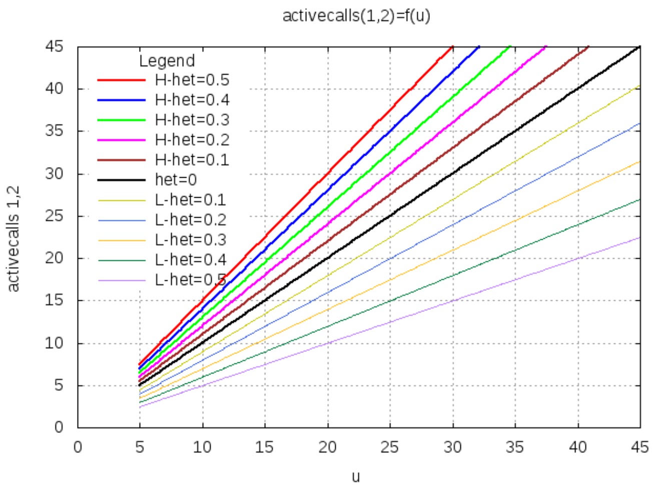

Fig. 4. Scenario input traffic explanation: The number of active sessions (in network 1=Low as well as network 2=High) depends on the nominal load u in Erlangs and the heterogeneity factor h between 0 and 50%. The graph for $h = 0$ is in the middle, and all graphs appear in the same order as provided in the legend.

A virtualized network brings together the resources of separate networks together, so that they can operate and perform as if they were one pool of resources. The SPN in Figure 3 reflects this in the following way, for two networks combined. $T1$ and $T11$ are the session generating transitions, with the same rates as in Figure 2. Also, the servers $T0$ and $T01$ as well as the active sessions per network in $P1$ and $P11$ are identical to the distinct network case. This means, each network still carries its individual traffic. However, the distribution of sessions (traffic) among the two networks is performed via a broker, represented by the queue in place $P2$ and the two concurrent (“choice”) transitions $T2$ and $T21$. In underload ($\#P2 = 0$, $\#P1 < m$ and $\#P11 < m$), new sessions go into network 1 or 2 with equal probability (Bernoulli process). In overload ($\#P2 > 0$), the first queued session on hold goes into the network which has the next free capacity. In the intermediate state (one of the networks 100% utilized), the session token goes into the other network with still available capacity. This is the reason why a virtualized network performs superior compared to distinct networks. This will be shown in Section IV. The SPN model can systematically be extended to three or more networks to be virtualized, and the gain would increase with higher multiplexing order. In this paper, we consider two networks for the numeric results.

Instead of just studying the statistical multiplexing gain of the combined networks, the contribution of this paper is considering heterogeneity or load imbalance among the user groups. In future wireless networks we have to cope with less and less homogeneous traffic load. Inhomogeneous clusters of users form naturally in time and space, depending on busy hours and social connectedness. Heterogeneous traffic distribution imposes a severe burden to the network, as one cell is more likely to be congested and another almost empty compared to

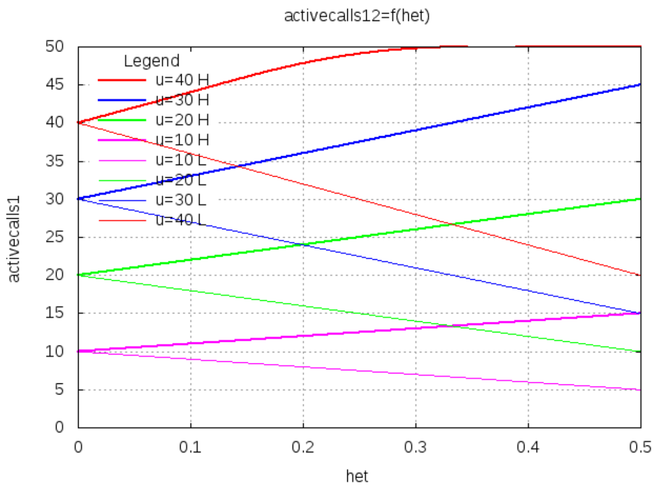

Fig. 5. Scenario input traffic explanation: The number of active sessions (in network 1=Low as well as network 2=High) plotted depending on the heterogeneity factor h, with the nominal load u as parameter. All graphs start from u at $h = 0$. To the right, each constant load u point splits into two graphs, the H and L part. The topmost curve represents an overload condition in the network 2=H.

the homogeneous case with equal load in each cell. Assuming that the same heterogeneity is present within the coverage region of a set of networks which can be virtualized, then it is worth studying what happens in such an imbalanced load scenario. For this purpose the scenario studied here modulates the load of each network by a heterogeneity factor h, such that

$$u_1 = u \cdot (1 - h), \tag{4}$$

$$u_2 = u \cdot (1 + h), \tag{5}$$

where network 1 is the lightly (L) loaded and network 2 the heavily (H) loaded system. Figures 4 and 5 display the load scenario depending on the two parameters u and h. For one scenario situation (given u and h), there are always two individual partial loads "L" and "H". We expect that the situation will worsen for the distinct scenarios with increasing h and for high total load $2u$.

IV. Analysis Results

The SPN analysis leads to the MC with all state transition rates known, so that the steady-state probabilities for each state can be calculated. From this, performance metrics are derived which can be interpreted in the model context. The state probabilities for states $0, ..., m$ represent the probability mass function of the number of active sessions. The complementary cumulative distribution function (CCDF) is the probability of exceeding state x, always a monotonously decreasing function.

The ideal situation (benchmark) would be the network with the sum capacity $2u = 100$ Erlangs [14]. The CCDF of active sessions in Figure 6 shows the performance for this case. Figure 9 can be interpreted as the extension to the right side of Figure 6, when more than 100 sessions are requested. For $u = 45$ (load 90%) and $u = 40$ (load 80%), a significant number of sessions are queued waiting for service. The average waiting time can be computed using the Little's formula ($\bar{W} = \bar{q}/\lambda$).

The baseline scenario (non-virtualized) is two distinct and separate networks with the same load. Its CCDF is plotted in

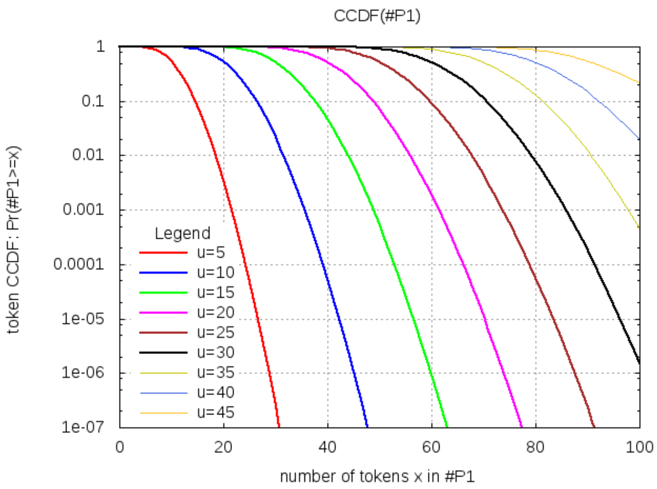

Fig. 6. Complementary cumulative distribution function (CCDF) of the number of active sessions (#$P1$) in a unified network of double capacity of 100 Erlangs and traffic load $2 \cdot u$ [14].

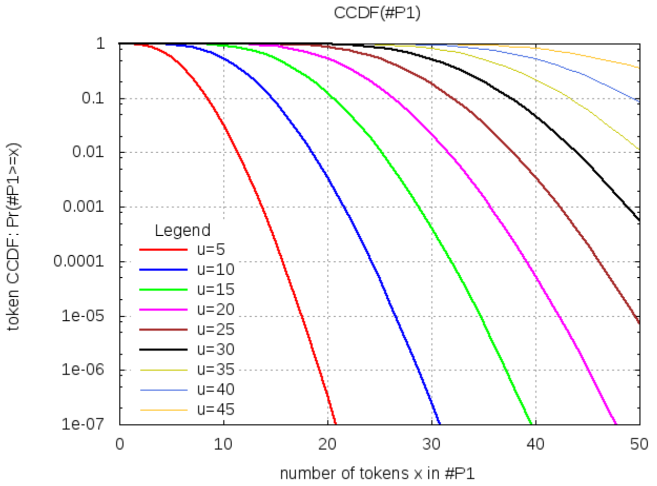

Fig. 7. CCDF of active sessions (#$P1$) for two separate networks and homogeneous load ($h = 0$).

Figure 7. The differences are noticable especially in the region with a high number of tokens, close to 50. Now we study the virtualized network of Figure 3: The results in Figure 8 appear to be superior to those in Figure 7, but still not as good as Figure 6 for a single network. The reason is that each individual network can still be completely full at some occasions (with higher probability than the single network of double capacity), and that the queueing process within one network experiences a different service rate in both cases. The overload probability $Pr\{\#P2 > 0\}$, however, is identical in both the virtualized and the single full capacity network. Thus, Figure 9 is the same in both cases.

Now we apply the heterogeneity factor h and study its impact on the performance. As there are now two distinct networks with different individual loads, there are always two curves per scenario ("L" and "H", for the lightly and heavily loaded networks). Figure 10 shows the average waiting time depending on the nominal load u of both and with parameter h. The homogeneous case is included with a single curve for

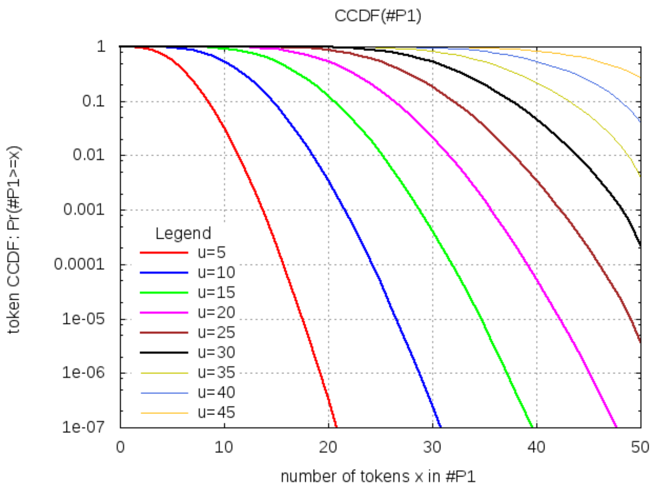

Fig. 8. CCDF of active sessions (#$P1$) for a virtualized network of two and homogeneous load (h = 0). There is a lower congestion probability for the high load region near #$P1$ = 50 compared to Figure 7.

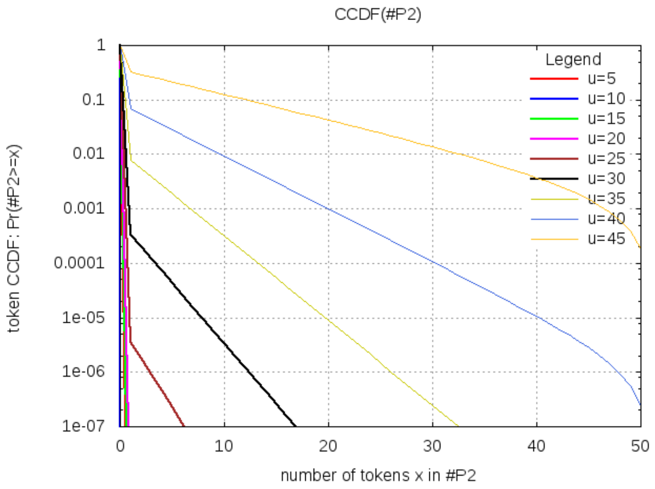

Fig. 9. CCDF of queued sessions (probability of overload by #$P2$ sessions) in a virtualized network of two. This graph can be seen as continuation of Figure 8 to the right. The parameter is total load u.

h = 0, in black. The dominant effects happen in the H network, not surprisingly, for high load u and high h. Of course, any waiting of more than a few 100 ms is noticable to the user, and the average shown here comes with variations within one order of magnitude in both ways. So if 1 second is our tolerable limit, then a homogeneously loaded network can sustain 35 Erlangs, whereas an unbalanced scenario would only allow 23 Erlangs (on average).

We can observe a similar trend in Figure 11 for the overload probability (given by $Pr\{\#P2 > 0\}$). Here the overload probabilities $\hat{p}_1$ and $\hat{p}_2$ are combined. As the L network is much more unlikely in overload, the graph basically displays $\frac{1}{2} \cdot \hat{p}_2$. If an overload probability of 0.01 or less is desired, then if the imbalance h increases, we can only accept a lower average load, e.g., down from 35 to 25 Erlangs.

In order to show gains between the virtualized and non-virtualized networks directly, we determine the ratio of the metrics acquired above, again depending on u and h. The overload probability ratio depicted in Figure 12 could only be shown for $u \geq 30$, because the probabilities were too close to zero below that. The ratio looks impressive for lower u, but this region is not very relevant, as any network would cope well with the underload situations. The region around u = 45, or 90% load, is the more interesting. Here we achieve a higher gain, the more heterogeneous the load situation is. The gain easily spans up to one order of magnitude. Note that for u = 50, the network is congested, and thus all overload probabilities are 1.

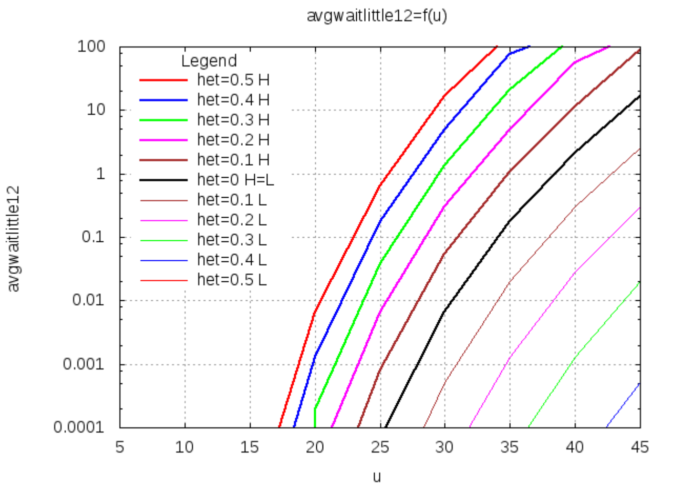

Fig. 10. Two separate networks in unbalanced load (heterogeneous factor h, total load u). Waiting time [s] of the 1=L network with lower load and 2=H network with higher load in one figure.

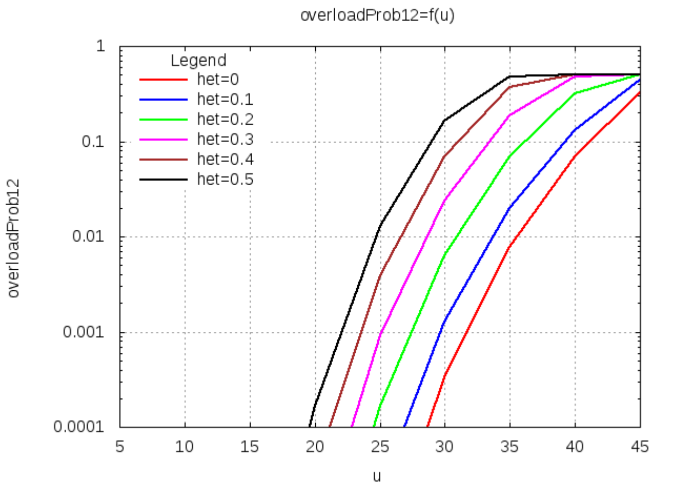

Fig. 11. Two separate networks in unbalanced load (heterogeneous, factor h). Overload probability of both networks combined ($\frac{1}{2}(\hat{p}_1 + \hat{p}_2)$). Because there is almost no overload in the lower-load network ($\hat{p}_1 \ll \hat{p}_2$ for $u > 0$), this graph is $\approx \frac{1}{2} \cdot \hat{p}_2$.

If we compare the waiting times, the ratio is even more significant. As shown in Figure 13, between u = 40 and 45, there is a factor of 100 increase of the average waiting time for the h = 0.5 scenario. The reason is that the H network is already fully congested; the black curve for h = 0.5 suggests the limit.

The results strongly indicate that there is a very high

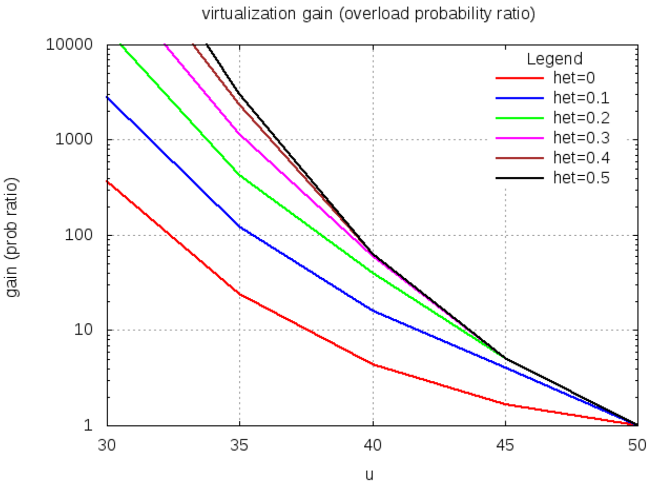

Fig. 12. Ratio (gain) of overload probabilities in the separate case compared to the combined virtualized case, depending on load *u*.

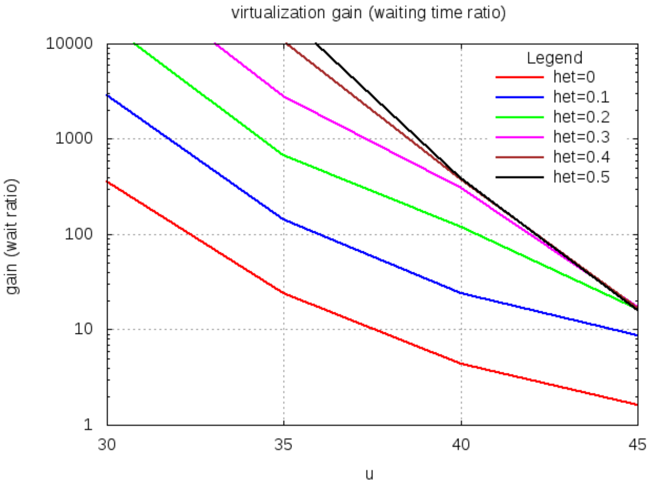

Fig. 13. Ratio (gain) of average waiting times in the separate case compared to the combined virtualized case, depending on load *u*.

potential for improvements by network virtualization in heterogeneous scenarios, which we expect to see more likely in the future. This paper promotes the SPN method as one efficient way of obtaining accurate results in a broad range of probabilities, unlike simulations which would require more than 10^8 samples for a similar precision.

V. Conclusion

In this paper a stochastic Petri net model is introduced to study network virtualization under heterogeneous load conditions. The model is a fluid-flow approximation with (e.g., video) session arrivals and departures based on Poisson point process input and multiple servers. The comparison of two separate networks with one combined network under the same load conditions shows strong advantages of virtualization, especially if the networks are unevenly loaded (heterogeneous). The SPN paradigm allows quick results from Markov analysis and provides accurate numerical results down to probabilities of 10^{-7} or less, which would be impossible with simulation. Future research will study more networks in parallel, HetNets (macro and pico cells), and can take channel variations into account [17]. Yet another extension is obtaining analytical results based on Erlang-C formulation. The heterogeneity parameter and scenario can be used combined with nonhomogeneous user distributions in space for HetNets [18]. Relay deployment can also be beneficial in HetNets [19].

References

[1] M. Mirahsan, R. Schoenen, Z. Wang, H. Yanikomeroglu, and M. St-Hilaire, “Unified and non-parameterized statistical modeling of temporal and spatial traffic heterogeneity in wireless cellular networks,” in *ICC’14 - 5G Technologies (ICC’14 WS - 5G)*, Sydney, Australia, Jun. 2014.

[2] N. M. K. Chowdhury and R. Boutaba, “Network virtualization: state of the art and research challenges,” *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, 2009.

[3] T. K. Forde, I. Macaluso, and L. E. Doyle, “Exclusive sharing & virtualization of the cellular network,” in *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2011, pp. 337–348.

[4] H. Takagi, *Queueing Analysis - Vacation and Priority Systems*. ISBN 0-444-88910-8: North-Holland, 1991, vol. 1.

[5] M. Marsan, *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1996, ISBN 0-471-93059-8.

[6] R. German, “A toolkit for evaluating non-Markovian stochastic Petri nets,” *Performance Evaluation*, vol. 24, pp. 69–87, 1995.

[7] J. Billington *et al.*, *Application of Petri Nets to Communication Networks*. Springer, 1999, ISBN 3-540-65870-X.

[8] M. Bosch and G. Schmid, “Generic Petri net models of protocol mechanisms in communication systems,” *Computer Communications*, vol. 14, no. 3, pp. 143–156, 1991.

[9] S. Geetha and R. Jayaparvathy, “Modeling and analysis of bandwidth allocation in IEEE 802.16 MAC: A stochastic reward net approach,” *Int. J. Communications, Network and System Sciences*, vol. 3, no. 7, pp. 631–637, July 2010.

[10] L. Lei, C. Lin, J. Cai, and X. Shen, “Performancs analysis of wireless opportunistic schedulers using stochastic Petri nets,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, April 2009.

[11] C. Zhang and M. Zhou, “A stochastic Petri net-approach to modeling and analysis of ad hoc network,” in *Proceedings of the ITRE*, Aug 2003.

[12] H. Wang and N. Moayeri, “Finite-state Markov channel – a useful model for radio communication channels,” *IEEE Transactions on Vehicular Technology*, vol. 44, no. 1, pp. 163–171, Feb 1995.

[13] R. Schoenen, G. Post, and A. Müller, “Analysis and dimensioning of credit-based flow control for the ABR service in ATM networks,” in *Proceedings of IEEE GLOBECOM*, 1998, vol.4 p.2399-2404.

[14] R. Schoenen and H. Yanikomeroglu, “Erlang analysis of cellular networks using stochastic petri nets and user-in-the-loop extension for demand control,” in *Proc. of IEEE GLOBECOM 2013 BWA-WS*, 2013.

[15] T. Murata, “Petri nets: Properties, analysis and applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–581, April 1989.

[16] “Cisco visual networking index: Global mobile data traffic forecast update, 2012–2017,” Cisco Systems Inc., White Paper, February 6, 2013. [Online]. Available: http://www.cisco.com/en/US/solutions/

[17] R. Schoenen, M. Salem, A. Sediq, and H. Yanikomeroglu, “Multihop wireless channel models suitable for stochastic Petri nets and markov state analysis,” in *Proceedings of IEEE VTC Spring*, Budapest, May 2011.

[18] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, “Statistical spatial traffic modeling with adjustable inhomogeneity and bs-correlation in wireless cellular networks,” in *Globecom 2014*, Austin, TX USA, Dec. 2014.

[19] R. Schoenen, W. Zirwas, and B. Walke, “Raising Coverage and Capacity using Fixed Relays in a Realistic Scenario,” in *Proceedings of the European Wireless Conference*, Prague, Jun 2008.