

Mean Value Load Identification and Queuing-theoretical Modeling of wired and wireless LAN

Andriy Luntovskyy, Stefan Uhlig, Dietbert Guetter, Volodymyr Vasyutynskyy, Alexander Schill
 Dresden University of Technology, Dresden, D-01062 Germany
 (phone: ++49 351-463-38381; fax: ++49 351-463-38251; e-mail: Andriy.Luntovskyy@tu-dresden.de).

Keywords: LAN/WLAN, load identification, queuing networks, performance modeling, CANDY Java Framework

Abstract

This paper presents static load and performance models for Ethernet LAN and WLAN. Relevant problems of the Internet protocol stack (OSI layers 1-4) are considered, namely identification of mean loads and modeling of wired and wireless LANs with the help product-form queuing analysis. This work appeared in scope of the network design environment CANDY as one of multiple contributions dealing with optimization heuristics, design and dimensioning of wired and wireless local networks.

1. INTRODUCTION

This paper examines common methods for static load identification and performance modeling of LAN/WLAN. A simple modeling approach based on product form queuing analysis (Jackson's theorem) is chosen [7]. To model CSMA/CD, well known empirical formulae were used. The empirical WLAN model accounts for relevant effects on OSI layers 1-4.

An important limitation is the assumption of a tree-shaped topology, which is fulfilled for a lot of modern networking solutions (Switched Ethernet).

The network design software CANDY ([1]-[5]) is being developed at TUD Dresden University of Technology. CANDY is a Java/XML-based integrated environment with the intention to support the process of network design and to automate involved routine tasks. Tools for load-capacity analysis based on queuing theory are part of CANDY. Figures 1 and 2 show the design within the CANDY Framework and structure of network design language NDML.

2. DESIGN FRAMEWORK FOR WIRED AND WIRELESS LOCAL NETWORKS

Architecture of the network design network CANDY is given on Fig.1. The development paradigm for CANDY includes the use of plug-ins, loose coupling of tools and components, open projects with plug-ins and web services-ready functionality. Different views on integrated office

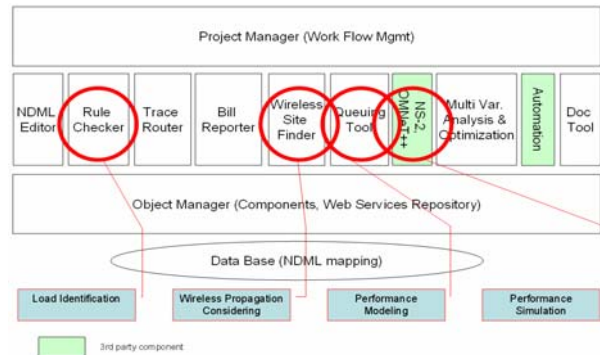


Figure 1. Load and performance analysis for wired/wireless networks with the CANDY Framework [1]-[5].

communication and automation networks are considered, i.e. topology with use of structured cabling and WLAN-routes, cost bills, performance and QoS analysis [1-5]. The integration of the tools is ensured by the object manager and the project manager, unifying the project workflow, using common object model described in NDML/NDML+, a XML-based problem-oriented language. The generalized structure of NDML is given in Fig. 2. NDML like each *ML (mark-up language) has flexible and extensible structure language.

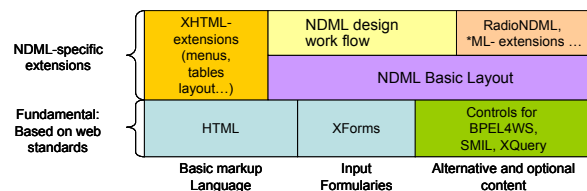


Figure 2. NDML as integration component

There are the following important groups of NDML elements:

- Fundamentals: basic web standards like XHTML, HTML, XML, XSL, XSLT, Xpath.
- Basic Layout with declarative tags for description of network components, cabling system and indoor/outdoor radio network routes (so called RadioNDML [1,5]) as wells as further specific *ML extensions.

- Extensions: XQuery, XForms, SMIL, BPEL4WS. They allow better workflow control and data mapping.
- Design workflow description tags.

Available CANDY tools and their main features are listed below (refer Fig. 1).

1. *NDML Editor* provides the graphical input of building plans and campus maps as well as office communication and automation network components (PC, gateways, routers, switches, hubs, AP, cables, automation nodes etc.).
2. *Rule Checker* controls the common design rules like network configuration, use of network components (switches, routers etc.) with coupled transfer media (fiber optic, copper cables, radio routes) as well as further workload constraints
3. *Trace Router* allows optimization of tracing and implementation of structured cabling system at the building for Ethernet LAN IEEE802.3 with considering of wireless routes using WLAN IEEE802.11.
4. *Site Finder* enables WLAN Access Point constellation optimization.
5. *Queuing Tool* and *NS-2 front end* facilitate the detailed performance analysis for complex networks (see [3]). Asymptotical prediction of network behavior (throughput, latencies) is made via Queuing Tool. Accurate performance and QoS (data rate, delays, and jitter) simulation for TCP/IP protocols is carried out via NS-2 standard freeware simulator with NDML front end. A possible scenario of simulation is shown in Fig.4.
6. *Bill Reporter* generates the overview of the costs of the whole network system.
7. *Multivariate Analysis and Optimization block* is aimed to prediction of network performance and increasing of “performance/cost” –ratio.
8. *Doc Tool* is aimed to consistent retrieving of distributed project data for CANDY in NDML descriptions, which can be mapped on data bases as well as other target formats like PDF, HTML etc. It serves also as persistent backup at Repository of CANDY-specific objects, components and Web Services.
9. *Automation network design tool* integrates the automation network issues into CANDY framework.

3. LOAD IDENTIFICATION METHODS

At first step of network design the intended loads of the nodes have to be identified. The CANDY Rule Checker performs several consistency checks for the specified wired/wireless networks. These checks include analyzing the network structure for compatibility of the used components as well as controlling the load values for the network nodes.

For the latter, two main ideas are examined: an expert emphasis method for load calculation and the Tree Round-Tour-Summarizing algorithm executed on the given network transformed to a tree. This transformation is possible for networks based on the SCS-principles for modern computer network construction with primary, secondary and tertiary areas with wired and wireless routes ([1]-[5]).

A tree representation of a network consists of the following types of nodes:

- leaves - correspond to the clients (PCs, notebooks, terminals etc.) – T-nodes $T_l, l=1..L$;
- internal nodes - correspond to the switches (SW-nodes);
- root node - corresponds to the internet gateway (a GW-node).

3.1. Expert emphasis method

To determine the network loads generated at the network clients (T-nodes), the so-called “Expert emphasis method” is employed. Network applications are divided into mutually independent application classes $A_i, i=1..I$. Based on these classes the load profiles $U_j, j=1..J$ are generated to represent differences in network usage caused by different occupations/activities of the users. One load profile is assigned to each T-node. The generation of the load profiles is a two-stage process. First, maximum load values LD_i for frequent usage are determined for each application class. Then, these values are adjusted by coefficients P_{ij} for each profile. These coefficients are gained by a questionnaire concerning the frequency of usage of each application class (Fig. 3).

1. Email Often <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Never	7. Online photography processing Often <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> Never
2. Chat Often <input type="radio"/> <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> Never	8. Video streaming, Internet TV Often <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Never
3. Telephony via Internet Often <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Never	9. Internet radio Often <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Never
4. Video conference Often <input type="radio"/> <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> Never	10. Software downloads Often <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> Never
5. Surfing, search engines Often <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Never	11. Online games Often <input type="radio"/> <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> Never
6. Online shops Often <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> Never	12. File sharing applications Often <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Never

Figure 3. Sample load questionnaire for LAN communication (CANDY Rule Checker) [1]-[5]

For each profile a number K_j of persons belonging to it (experts) is questioned. Their answers are interpreted on a linear scale from 1 to 0 with steps of 0.25: “Often” means 1, “Never” means 0 and so on. These values are averaged giving the load coefficients for the respective profile:

$$P_{ij} = \frac{1}{K_j} \cdot \sum_{k=1}^{K_j} P_{ijk} \mid P_{ijk} \in \{0;0.25;0.5;0.75;1\} \quad (1)$$

The calculation of the load generated at the client T_i takes place according to (2). The load coefficients P_{ij} are used to weight the maximal load values LD_i for each application class, giving the actual data load for the treated usage profile.

$$LD_i = \sum_{j=1}^I P_{ij} \cdot LD_j \mid u_i = U_j, LD_i \leq DR_i \quad (2)$$

where DR_i is the gross data rate of the network connection for the given client (T-node).

Example. Client T_i with $DR_i=100\text{MBit/s}$, $I=5$ traffic classes, maximum loads per class $DR_i=\{50; 20; 30; 5; 10\}$, usage coefficients $P_{ij}=\{0,25; 0; 0,5; 1; 0\}$. Therefore:

$$LD_i = 0.25 \cdot 50 + 0 \cdot 20 + 0.5 \cdot 30 + 1 \cdot 5 + 0 \cdot 10 = 32.5\text{MBit/s} \leq 100\text{MBit/s}$$

Following assumptions are made for the application classes A_i ($i=1..12$) in the questionnaire shown in Fig. 3:

- Question 1. Email-traffic: average number of e-mail messages sent/received per day (e.g. normalizing max. value: 100 e-mail messages/day).
- Question 2~4. Synchronous telecommunication services (e.g. normalizing max. value: 10h/day).
- Question 5~7. WWW and Web-functions, e.g. normalizing max. value: 50 requests/day.
- Question 8~9. Asynchronous telecommunication services (e.g. normalizing max. value: 10h/day).
- Question 10. Downloads, amount of data downloaded per day (e.g. normalizing max. value 3GByte/day).
- Question 11. Online games (e.g. normalizing max. value: 10h/day).
- Question 12. FTP (e.g. normalizing max. value: 6GByte/day).

3.2. Tree Round-Tour-Summarizing algorithm

For the obtained network tree the so-called Tree Round-Tour-Summarizing algorithm is applied. The data loads are summed up along the edges of the tree from leaves (T-nodes) through internal nodes (SW-nodes) ascending to the root (GW-node, the exit to the Internet). For each node $m=1..M$ the equation for inputs (3) must be true (balance with reserve).

$$LD_m^{Input} = \sum_{n=1}^N LD_n \text{ with inputs } n = 1..N \quad (3)$$

$$LD_m = LD_m^{Input} + LD_m^{res} \leq DR_m$$

where DR_m – throughput for node m of SW-type or GW-node. An illustration for the used tree abstraction for a

computer mesh network is shown in Fig. 4.

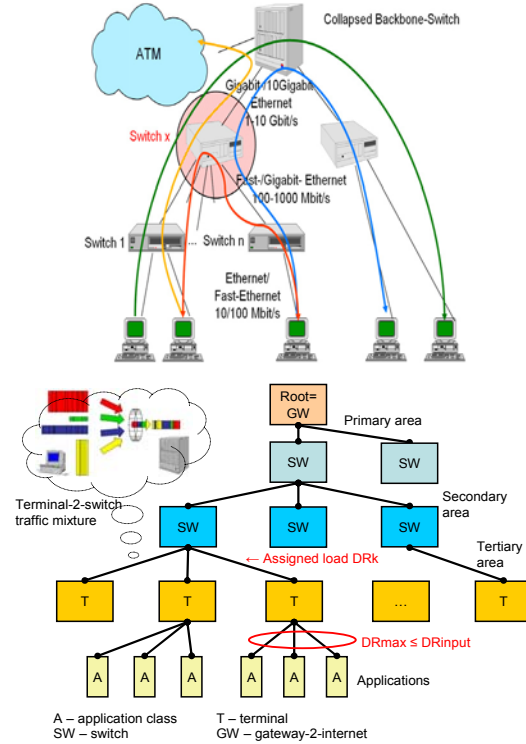


Figure 4. Switched Ethernet computer network (above) and equivalent tree with traffic considering (below).

DATA:
Tree = { Edges, T-nodes T_i , SW-Nodes, GW-Node}
 DR_i, DR_m // max. throughput for leaf/internal nodes

```

///// Expert Emphasis Method
Entry1: // Step 1: Determine load coefficients
FOREACH (Load profile  $U_j, j=[1..J]$ )
  FORACH (Application class  $A_i, i=[1..I]$ )
     $P_{ij}=0;$ 
    FOREACH (Expert  $E_k, k=[1..K_j]$ )
       $P_{ij}=P_{ij}+ExpertEvaluation(E_k, A_i);$ 
     $P_{ij}=P_{ij}/K_j;$ 
Entry2: //Step 2: assign leaf loads
FOREACH (T-Node  $T_i, i=[1..L]$ )
  //  $T_i$  has load profile  $U_j$ 
   $LD_i=0;$ 
  FOREACH (Application class  $A_i, i=[1..I]$ )
     $LD_i = LD_i + P_{ij} \cdot LD_i;$ 
///// Tree Round-Tour Summarizing Algorithm (TRTSA)
Entry 3: // Step 3: Determine loads at internal nodes
FOREACH (SW-Node and the GW-Node  $m=[L+1..Root]$ )
  IF (inputLoad( $m$ ) >  $DR_m$ )
    THEN { Exception ("Node  $m$ : Overload!"); }
  ELSE  $LD_m = inputLoad(m);$ 
STOP: //network load values assigned
RESULT:  $LD_m, m=1, Root$ 

```

Figure 5. Expert Emphasis Method and Tree Round-Tour-Summarizing Algorithm.

Expert Emphasis Method and Tree Round-Tour-Summarizing algorithm are presented as pseudo-code in Fig. 5.

The examined methods are implemented in the CANDY Framework ([1]-[5]). The resulting graphical representation of the identified loads in CANDY for a LAN-network is depicted in Fig. 6.

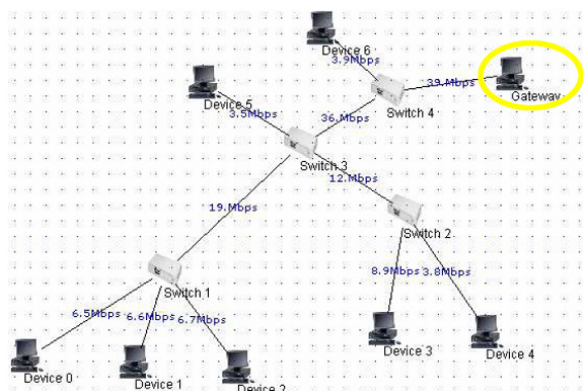


Figure 6. Load scenario for a LAN and evaluated traffic.

The obtained load specifications are used further by the CANDY Queuing Tool ([1]-[5]) and event-driven simulators (NS-2, OMNeT++) to analyze the characteristics of networks like serving rates, throughputs, response times, delays, data loss ratio etc.

The limitation of this solution is that solely the so-called “uplink” data streams (from terminals to the gateway) are modeled. Also traffic bursts are not considered.

“Downlink” data streams (from other network segments or Internet via the GW to the specified terminal T) and traffic within the network segment are modeled by specifying maximum throughput values DR_m for each node on the considered communication way and performing routing via network tree (Bellman-Ford algorithm).

4. ONE ARRIVAL CLASS MODELING FOR CSMA/CD

On the basis of queuing-theoretical assumptions (M/M/1 Jackson’s combination of k nodes) the following One-Arrival-Class-modeling for the Ethernet can be done. The CSMA/CD protocol uses a stochastic method to access the medium. The following parameters are to be considered [13], [1]:

- k – number of stations (STA) sharing the medium,
- B – network bit rate ($B=10\text{Mbps}$),
- L_p – transferred frame length,
- S – slot time ($51,2\mu\text{s}$ for 802.3 Ethernet).

Then the arrival rate λ is defined as follows:

$$\lambda = B / L_p \quad (4)$$

The service rate $\mu(k)$ can be calculated according to the following equations:

$$A(k) = \left(1 - \frac{1}{k}\right)^{k-1} \quad (5)$$

$$C(k) = \frac{1 - A(k)}{A(k)} \quad (6)$$

$$\mu(k) = \left(\frac{L_p}{B} + S \cdot C(k+1)\right)^{-1} \quad (7)$$

Here $A(k)$ is the probability of a successful transmission and $C(k)$ is the average number of collisions. The following assumptions must be heeded:

- 1518 bytes/frame at standard bit rate up to 10MBit/s, 100% of load are corresponding to max. frame rate up to 863 Frame/s
- The media (coaxial cable/UTP) can be used effectively only for loads up to 85%.

A plot of $\mu(k)$ is given in Fig. 7. Fig. 8 shows an example Ethernet network.

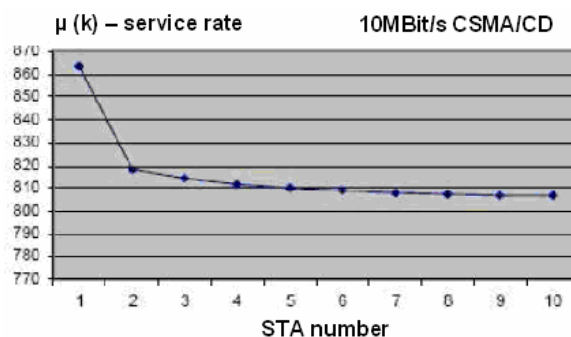


Figure 7. $\mu(k)$ – service rate (10MBit/s, CSMA/CD) depending on the number of stations k .

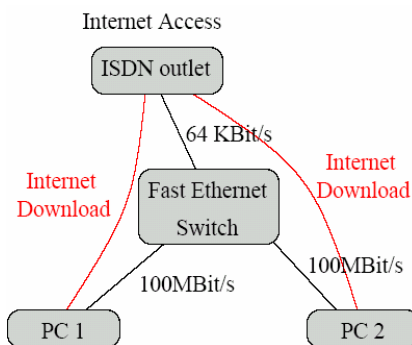


Figure 8. Example: A fragment of a LAN with Internet-access

The computed performance characteristics are given in Table 1.

Table 1. Computed performance characteristics for example in Fig. 8

Medium	Number of competing NICs, k	Bandwidth for Traffic on this medium	Network connection	Usage η	Used bandwidth considering collisions
medisdn	1	52800 Bits/s	67108.9 Bits/s	0.786781	52800 Bits/s
medeth01	2	29.0222 MBit/s	100 MBit/s	0.348099	34.8099 MBit/s
medeth01	2	29.0115 MBit/s	100 MBit/s	0.348013	34.8013 MBit/s

The found $\mu(k)$ must be adapted to represent throughput losses caused by overheads connected with the protocol stack (see Fig. 9):

$$\mu = \mu(k) \cdot (1 - V_p) \quad (8)$$

where V_p – relative losses caused by TCP/IP-protocol stack.

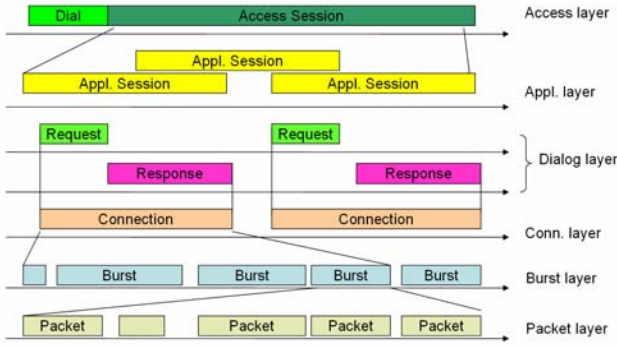


Figure 9. Example: $\mu(k)$ – service rate decreasing due to protocol in TCP/IP-Stack overhead

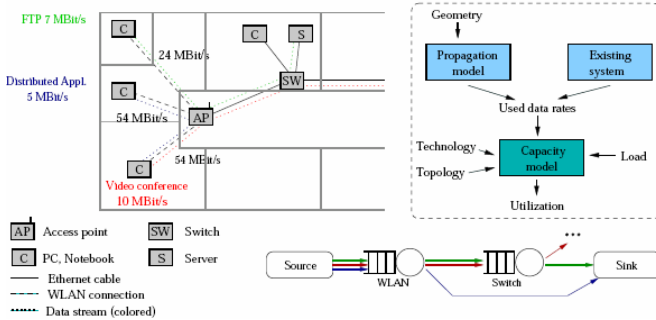


Figure 10. Example: placement of AP (left); used data (right), queuing net (below).

5. ONE ARRIVAL CLASS MODELING FOR WLAN (CSMA/CA)

This part examines a common method for load-capacity modeling of WLAN on the OSI-layers 1-4. Only the radio signal propagation issues (layer 1) will be examined here. A simple approach with $M/M/1$ queues and their combination based on Jackson’s theorem is chosen [7]. The accuracy of the WLAN communication model is achieved by

parameterization of the nominal data rates considering known effects on PHY, MAC, IP and transport layers.

A scenario for indoor WLAN capacity analysis is presented in Fig. 10. The considered WLAN is a part of a local area network. The given input data are the application data streams to include in the capacity analysis as well as the building plans, which are used in radio propagation models ([3],[4],[6]). These data form a basis for capacity analysis by providing the usable data rates in the WLAN part.

5.1. Concept

WLAN-devices sharing a RF channel, i.e. an access point and its clients or two access points acting as a wireless bridge, are modeled as one $M/M/1$ -FCFS queue, so that this queue represents the wireless channel. Its service rate μ corresponds to the effective throughput of the channel, i.e. the amount of user data of the treated protocol layer that can be transmitted over the channel considering all overheads and losses up to and including this protocol layer. Fig. 10 gives an example of the transformation *Real Network* \Rightarrow *Queuing Net (QN)*. QN customers represent abstract data packets of a certain size G_p (a model parameter). Accordingly the rates of external arrivals to the QN are defined by the expression:

$$\lambda_k = \frac{l_k}{G_p}, \quad k \in 1..K \quad (9)$$

where K is the total number of data streams and l_k is the data load caused by stream k (see example, Fig. 10).

The service rate μ of the queuing station is determined based on the data rates that the particular clients use to communicate with the access point. To find out these data rates for a planned, not yet existing network, radio propagation models can be used, e.g. those implemented in the CANDY Wireless Site Finder (8 indoor/outdoor propagation models ([3],[4])). Based on the particular data rates a mean gross data rate of the RF channel (DR^*) is derived. Then DR^* is adjusted by expressions representing the influence of several factors on the effective data rate:

- Losses due to contention for the radio channel;
- Protocol overheads (on the layers 1-4);
- Interactions between particular protocols.

The discussed model found its Java implementation in the CANDY Framework. The software calculates necessary characteristics of queuing networks like throughputs and queue lengths. To achieve more adequately parameterized (λ, μ)-queuing models, validation and calibration using the well-known simulators NS-2 and OMNeT++ is planned.

5.2. Service rate and data rate loss factors

As already stated, the service rate μ for a queue representing a wireless channel is composed of multiple expressions that represent factors that determine the effective data rate. The following formula is used to calculate a real μ for each channel $\mu_j, j=1..M$:

$$\mu = \frac{DR^* \cdot (1 - V_p)(1 - V_c(n))(1 - V_{tcp})(1 - K_m)(1 - K_g \cdot n) + K_a}{G_p} \quad (10)$$

- with
- DR^* gross data rate of the channel,
- V_p protocol overhead losses,
- $V_c(n)$ losses caused by contention for channel access,
- V_{tcp} losses due to interactions between TCP and the WLAN MAC,
- K_m, K_g, K_a degrees of freedom to adjust the model,
- G_p mean payload size.

The next sections analyze particular components of the given formula.

5.3. Gross data rate in RF channel

The gross data rate of the RF channel DR^* is computed as average of the data rates that are used by the individual clients to communicate with the access point. These depend on the maximum rate of the particular standard, DR_{max} , and also on WLAN hardware properties and the respective signal propagation conditions. The WLAN MAC guarantees equal channel access opportunities for all stations in a long-term view. So DR^* can be computed based on the time needed to transmit one frame of each client.

Example: two clients are registered on the same 802.11g access point. Client 1 can operate at maximum data rate ($DR_1=54\text{Mbit/s}$) while client 2 works at a lower rate due to obstacles to radio propagation ($DR_2=18\text{Mbit/s}$). Both clients send frames of an average size S of 1500 Byte. The time T_T to transmit one frame of each station is:

$$T_T = \frac{S}{DR_1} + \frac{S}{DR_2} = \frac{12000\text{Bit}}{54\text{MBit/s}} + \frac{12000\text{Bit}}{18\text{MBit/s}} = 889\mu\text{s} \quad (11)$$

The gross data rate DR^* for the RF channel in this case can be evaluated as follows:

$$DR^* = \frac{2 \cdot S}{T_T} = \frac{24000\text{Bit}}{889\mu\text{s}} = 27\text{MBit/s} \quad (12)$$

Therefore, the formula for the general case with n clients is:

$$DR^*(n) = \frac{n}{1/DR_1 + 1/DR_2 + \dots + 1/DR_n} = \sum_{i=1}^n \frac{n}{1/DR_i} \quad (13)$$

with DR_i – data rate of client i , n – number of clients that share the RF channel. Here we disregard overheads due to the protocol stack; these effects are examined in the next clause.

5.4. Protocol overhead analysis

The data rate loss caused by protocol overheads is evaluated in respect to the used protocol stack. The following example uses 802.11g (OFDM) and TCP as transport protocol. The scheme of a general 802.11 MAC transmission is shown in Fig. 11. For this example, RTS/CTS is considered to be disabled. A TCP SDU size of 1460 Byte is assumed.

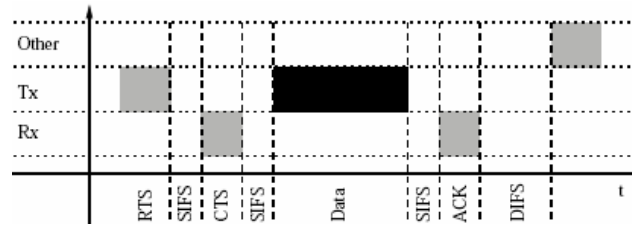


Figure 11. Protocol overhead analysis: scheme of a MAC transmission.

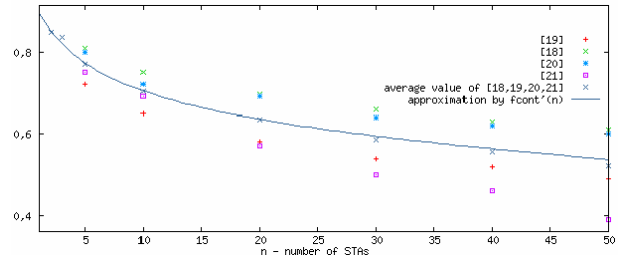


Figure 12. Contention losses evaluation based on analysis of references [8]-[12].

The time for the complete transmission including TCP-Data Segment, TCP-Acknowledgement and the corresponding MAC-Acknowledgements is examined based on the times needed to transmit the single segments/frames. The calculation of the transmission time for the TCP-fata segment is given in detail, followed by the formula for the total time (15).

Data length	1460 Byte = 11680 Bit
+ TCP, IP headers	+ 40 Byte
+ SNAP, LLC, MAC headers	+ 42 Byte
Summarized:	1542 Byte = 12336 Bit
OFDM-data part	12528 Bit

$$T_{DATA} = 16\mu s + 4\mu s + 12528 \text{Bit} / 54 \text{MBit} / s = 252\mu s \quad (14)$$

$$T = T_{DATA} + T_{SIFS} + T_{mACK} + T_{DIFS} + T_{iACK} + T_{SIFS} + T_{mACK} + T_{DIFS} = 476\mu s \quad (15)$$

where T_{DATA} is the time needed to transmit the TCP Data segment and T_{SIFS} , T_{DIFS} , T_{mACK} , T_{iACK} are the durations of SIFS, DIFS, MAC- and TCP-acknowledgements.

Next we examine the time needed to transmit the 1460 Byte of TCP data T_{use} is $216\mu s$. The overhead V_p in this example is calculated according to the following expression: $V_p = 100\% \cdot (1 - T_{USE} / T) = 100\% \cdot (1 - 216\mu s / 476\mu s) = 54,6\%$ (16)

Nevertheless this representation has some shortcuts, particularly:

- TCP “slow start” problematic is not considered;
- TCP-ACKs are cumulative; can be sent together with data packet
- TCP-segments are varying in size; can be distributed over multiple MAC-frames.

Since overhead calculation depends on the used data rate, the correction factor V_p in (10) is calculated as middle value of the overheads for the rates used by the communicating stations.

$$V_p = \frac{1}{n} \cdot \sum_{i=1}^n V_p(DR_i) \quad (17)$$

Table 2 shows the overheads on MAC and TCP layers for OFDM and DSSS.

Table 2. Overheads for DSSS and OFDM on MAC and TCP layers

DSSS (802.11,11b)			OFDM (802.11G,A)		
DATA RATE [MBit/s]	OVERHEAD [%]		Data rate [MBit/s]	OVERHEAD [%]	
	MAC	TCP		MAC	TCP
1	6.62	17.19	6	7.7	19.0
2	7.06	17.89	18	17.0	32.1
5.5	14.4	28.69	36	27.9	45.2
11	23.86	40.91	54	36.0	54.6

5.5. Protocol stack interactions

In addition to protocol overheads, interactions between the protocols of the Internet stack cause throughput losses. The first important problem is caused by WLAN MAC and TCP error correction operating against each other:

- TCP dynamically adapts the segment stream to channel and receiver capacity;
- TCP generally interprets timeouts as congestion and reduces TCP stream;
- MAC-error correction mechanisms cost time and can lead to TCP-Timeouts;
- TCP throughput drops dramatically on bad connections

(compared to IP).

Another phenomenon occurs particularly in connection with constant rate transmissions, e.g. huge file transfers. In that case TCP data segments and acknowledgements contend for channel access. Delayed acknowledgements can lead to timeouts and reduction of TCP throughput [12]. This work considers these effects only partially due to their analytical complexity. The model parameter V_{tcp} can be adapted to represent their influence.

5.6. Contention losses analysis

Contention losses depend on the number of STAs sharing a wireless channel and on their transmission activity. For this model, we evaluate results from other works. We consolidate data from ([8]-[11]), giving the normalized channel throughput on MAC sub-layer under saturation-equivalent conditions in dependence of the number of STAs sharing the channel (Fig. 12). The gained average values were interpolated by a rational function, $f_{cont}'(n)$. Because the normalized throughput accounts for protocol overheads too, which are evaluated separately in this model, $f_{cont}'(n)$ is transformed to $f_{cont}(n)$ (18) by shifting it along the y-axis so that $f_{cont}(1)=1$, (i.e. no contention losses).

$$f_{cont}(n) = \frac{6,70 \cdot n + 47,2}{0,02 \cdot n^2 + 7,58 \cdot n + 38,32} - 0,17 \quad (18)$$

The used results from [8]-[11] refer to 802.11 WLAN (1Mbit/s). Thus additional treatment is necessary for the higher data and non-saturation conditions. Assuming equal load for each STA, the throughput decreases from a certain number of STA $N \leq n$. Exceeding this number, STAs obstruct each other increasingly in RF channel access. Hence the generated total data load for the point of maximal throughput must be considered. The following examples are based on data from [11]:

$$\text{Ex.1: } DR = n \cdot \lambda \cdot G_p = 3 \cdot 25 \cdot 1000 \text{Byte} = 75000 \text{Byte} = 0,6 \text{MBit} / s \quad (19)$$

$$\text{Ex.2: } DR = 15 \cdot 25 \cdot 1000 \text{Byte} = 375000 \text{Byte} = 3 \text{MBit} / s$$

The used gross data rates were 1 MBit/s for example 1 and 6 MBit/s for example 2. Evidently the data loads are approximately up to one half of the used gross data rate. Based on this observation an empirical formula is derived for determining the number of stations N that saturate the channel:

$$N = \frac{0,5 \cdot DR}{\lambda \cdot (G_p + O_p')} \quad (20)$$

where λ is the arrival rate in s^{-1} . G_p is the mean payload size and O_p' the protocol overhead above the MAC layer, both given in Bit. Since stations can have dissimilar arrival rates, their mean value is used to calculate N. For $n \leq N$ stations contention losses are assumed to be negligible,

otherwise f_{cont} is used to determine contention losses:

$$V_c(n) = \begin{cases} 0, & n \leq N \\ 1 - f_{cont}(n - N), & \text{otherwise} \end{cases} \quad (21)$$

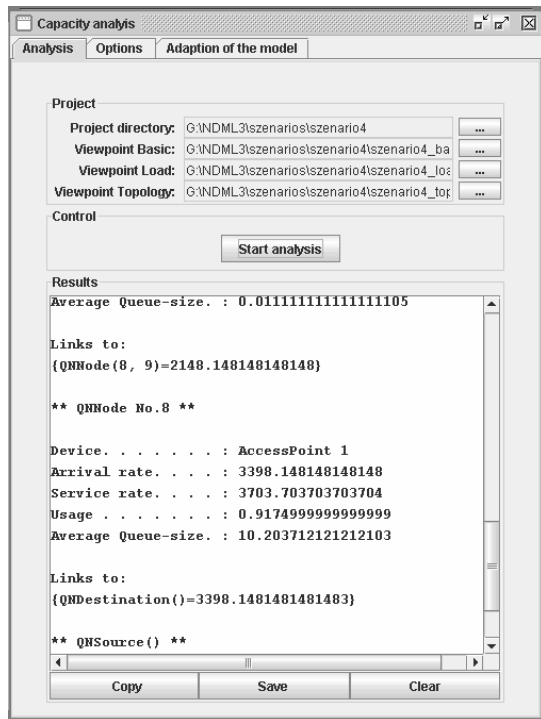


Figure 13. CANDY Queuing Tool (WLAN performance analysis)

```
<?xml ... >
<ndml>
...
<node id=8>
  <device>AccessPoint 1</device>
  <arrivalrate>3398.1481</arrivalrate>
  <servicerate>3703.7037</servicerate>
  <queuelength>10.2037</queuelength>
</node>
...
</ndml>
```

Figure 14. Modeling results represented via NDML Performance Report

5.7. Results

The results of the load identification and performance modeling can be represented as a NDML report (see Fig. 13 and 14) to further deployment via other CANDY Tools.

6. PERSPECTIVES AND OPEN PROBLEMS

There are the following perspectives and open problems:

- Validation and calibration of queuing model under use of

well-known event-driven network simulators.

- Integration of models for other network technologies.
- Development of software with graphical representation of modeled scenarios.
- Accurate traffic modeling and load analysis to get adequate data for $\lambda_k, k=1..K$.

Unfortunately, the mostly important as well as typical problematic for the CANDY Framework is lacking of solid validation data. But this lack can be surely compensated via integration of the CANDY@TUD in some commercial projects.

References

- [1] CANDY Project: <http://www.rm.inf.tu-dresden.de>
- [2] Schill, A., D. Gütter, A. Luntovskyy, G. Pfeifer, A. Panchenko, "CANDY: Integrated Environment for Network Design" in Proc. *SCI-04*, Orlando, Florida, USA, 2004, pp. 252-257, ISBN 980-6560-13-2
- [3] Luntovskyy, A., D. Gütter, A. Schill, U. Winkler, "Design Particularities for Wireless Networks" in Proc. *IEEE CriMiCo-05*, Sevastopol, 2005, pp.955-958.
- [4] Luntovskyy A., D. Gütter, A. Schill, "Models and methods for WLAN/ WIMAX-Network Design" in Proc. *IEEE CriMiCo-06 vol. 3*, Sevastopol, 2006, pp.391-393, ISBN:966-322-006-6
- [5] Luntovskyy, A., D. Gütter, A. Schill, "Network Design and Optimization under Use of CANDY Framework" in Proc. *IEEE CriMiCo-06*, Sevastopol, 2006, Vol.2, pp.394-397, ISBN:966-322-006-6
- [6] Rech, J. *Wireless LANs - 802.11-WLAN-Technologie und praktische Umsetzung im Detail*, Hannover: Heise, 2004.
- [7] Robertazzi, T. *Computer Networks and Systems - Queuing Theory and Performance Evaluation*, 3rd ed., New York: Springer, 2000.
- [8] Bianchi, G., "Performance Analysis of the IEEE 802.11 DCF", *IEEE Journal on selected areas in Communications*, Vol.18, No.3, 2000.
- [9] Ni, Q., I. Aad, C. Barakat, T. Turletti, "Modelling and Analysis of Slow CW Decrease for IEEE 802.11 WLAN". Proceedings of IEEE International Symposium on Personnel, Indoor and Mobile Radio Communications (PIMRC 2003), Beijing, China.
- [10] Pasupuleti, S., D. Das, "Throughput and Delay Evaluation of a Proposed-DCF MAC Protocol for WLAN". Proceedings of the IEEE INDICON 2004, First India Annual Conference.
- [11] Sudarev, J. V., and L. B. White, "Performance Analysis of 802.11 CSMA/CA for Infrastructure Networks under finite Load Conditions". The 14th IEEE Workshop on Local and Metropolitan Area Networks, LANMAN 2005.
- [12] Gopal, S., S. Paul, D. Raychaudhuri, "Investigation of the TCP Simultaneous-Send Problem in 802.11 Wireless Local Area Networks". ICC 2005, IEEE International Conference on Communications, 2005.
- [13] Bolch, G. et al.. *Queueing Networks and Markov Chains*. Second Edition, Modeling and Performance Evaluation with Computer Science Applications. John Wiley & Sons, 2006, 878p.