

Cross-Layer Signalling for Next-Generation Wireless Systems

Qi Wang and Mosa Ali Abu-Rgheff

Mobile Communications Research Group
School of Computing, Communications and Electronics
The University of Plymouth
Drake Circus, Plymouth, Devon PL4 8AA, UK
<qwang, mosa>@plymouth.ac.uk

Abstract- Cross-layer design is becoming a popular design methodology for the IP-based next-generation wireless systems, and cross-layer signalling is a key enabler of such a methodology. Several methods are emerging to achieve this signalling in layered protocol stacks. Through these methods, the refined wireless systems are expected to gain significant performance improvement and/or obtain extended functionality that is very hard if not impossible to obtain from a single layer signalling. This paper begins with a survey of representative cross-layer signalling methods. Based on the analysis, the paper proposes an efficient, flexible and comprehensive scheme defined here as Cross-Layer Signalling Shortcuts (CLASS), and then a set of evaluation criteria is defined to compare CLASS with current schemes. Finally, the possible application areas of CLASS are identified, and a reference application programme is presented for applying CLASS into various management areas.

I. INTRODUCTION

Layering is the dominating design methodology of communication protocol stacks. An essential feature of the layering principle is layer-independence (modularity), and thus in a strict layered protocol stack, cross-layer communications are considered as violation. However, keeping the strict layering all the time can be cumbersome and may result in an inefficient implementation of a protocol suite.

Therefore, the cross-layer design [1] has been introduced. The extreme is to merge all layers into one flat single layer. This is absolutely orthogonal to the strict layered structure. Between the two extremes, cross-layer (inter-layer) signalling can be introduced to a layered protocol stack, and thus only limited modifications would be required to the existing stack. This approach is the focus of this paper.

We argue that the cross-layer design can play an important role for the next-generation (3G and beyond) wireless systems, featured by all IP-based protocol stack, heterogeneous access networks, and multimedia data traffic [2]. The cross-layer design can be fully justified considering the following issues:

Firstly, the assumptions in the wired IP stack are inadequately suitable for the wireless networking. For example, one of the well-known assumptions in TCP protocol is that packet loss is caused by network congestion. However, in wireless systems, packet loss often occurs due to

corruption. The congestion avoidance procedure can only make things worse. Exposing the packet corruption rather than congestion in the signalling from Link Layer¹ to Transport Layer will facilitate an easy solution to this problem [3].

Secondly, the heterogeneity of network and traffic calls for a co-ordinated adaptation from multiple layers. Introducing a single collocated layer for various adaptation tasks would be too complex and heavy. The QoS (Quality of Service) adaptation even requires all layers' participation [1]. Therefore, a co-operation of multiple layers' adaptation would lead to a simpler and more flexible approach.

Thirdly, the rare radio resource and the limited power necessitate the optimisation of network performance; such optimisation can hardly be met in the sub-optimal wired architecture with strict layering. For example, error correction schemes are provided in both Link Layer and Transport Layer. In wireless systems, these schemes have to be invoked much more frequently to combat the errors due to unreliable channels. A co-ordination of the two layers can thus result in a more efficient solution [4].

Lastly, the emerging short-range networks such as Ad Hoc network and Personal Area Network entail an integrated design approach. In traditional networks, Link Layer is for point-to-point communications, while Transport Layer is for end-to-end communications across various links. In short-range networks, the peer-to-peer communications mostly take place in the point-to-point level. By cross-layer design, duplicate efforts from each related layer can be avoided [5].

The remaining of this paper is organised as follows: Section II presents a survey of existing cross-layer signalling methods. Section III describes the design, evaluation and application of the proposed scheme, CLASS (Cross-Layer Signalling Shortcuts). We conclude and outline the future work in Section IV.

II. REVIEW OF EXISTING CROSS-LAYER SIGNALLING METHODS

For presentation convenience, a five-layer reference model is used as a uniform protocol stack in this paper.

¹ A name of a layer is capitalised with "the" omitted.

A. Method 1- Packet Headers

In IPv6, optional Network-Layer information can be encoded in additional headers. The Interlayer Signalling Pipe (ISP) briefed in [4] takes advantage of this new feature by storing cross-layer information in the Wireless Extension Header (WEH) as shown in Fig. 1. This method makes use of IP data packets as in-band message carriers with no need to use a dedicated internal message protocol. However, an IP packet normally can only be processed layer by layer, and it is not easy for higher layers to access to the IP-level header. Furthermore, the conceptual bottom-to-top “pipe” seems excessive in most cases. Although the ISP is implemented within the mobile host (MH), network nodes and the corresponding host (CH) can read the information if they are WEH-aware. Thus, in fact, this method is more suitable for external IP-level information exchange.

In [3], only one bit in the TCP packet header was used for Explicit Loss Notification (ELN) by a Link-Layer software agent Snoop in the Base Station (BS). When Snoop is aware of a packet loss due to corruption, it sets the ELN bit in the TCP header and generates the in-band signalling as a feedback to the MH. This scheme suits a simple Boolean notification but does not scale well to bear complex control information.

B. Method 2- ICMP Messages

ICMP (Internet Control Message Protocol, [6] for IPv4, [7] for IPv6) is a widely deployed signalling protocol in IP-based networks. Compared to the “pipe” described above, Method 2 [8] is to “punch holes in the protocol stack” and propagate information across layers by using ICMP messages as shown in Fig. 2. In this scheme, desired information is abstracted to parameters, measured by corresponding layers wherever convenient. A new ICMP message is generated only when a parameter changed beyond the thresholds. Since cross-layer communications are carried out through selected “holes” not a general “pipe”, this method seems more flexible and efficient. Furthermore, Method 2 is maturer since it has been implemented on Linux operating system (OS) with APIs (Application Program Interfaces) developed. However, an ICMP message is always encapsulated in an IP packet, and this indicates that the message has to pass by Network Layer even if the signalling is only desired between Link Layer and Application Layer.

C. Method 3- Network Service

In [9], a specific access network service called Wireless Channel Information (WCI) was proposed. In this scheme, channel and link states from Physical Layer and Link Layer are gathered, abstracted and managed by third parties, the

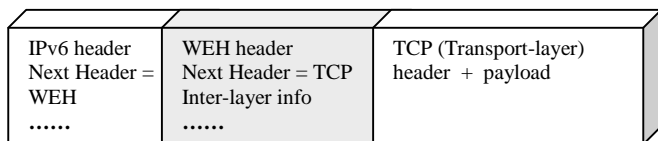


Figure 1. Bear cross-layer information with extension header

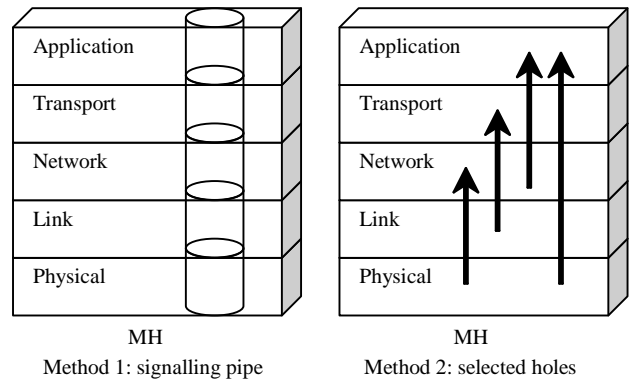


Figure 2. Comparison of Method 1 and Method 2

distributed WCI servers. Interested applications then access to the WCI for their required parameters from the lowest two layers as shown in Fig. 3. Although it is not a cross-layer signalling scheme within a MH, we can deem it complementary to the former two schemes, as further implementation problems are considered in parameter definition, abstraction, coding, and decoding. However, any intensive use of this method would introduce considerable signalling overhead and delay over a radio access network.

D. Method 4- Local Profiles

In [5], local profiles are used to store periodically updating information for a MH in an ad hoc network as illustrated in Fig. 4. Cross-layer information is abstracted from each necessary layer respectively and stored in separate profiles within the MH. Other interested layer(s) can then select the profile(s) to fetch the desired information.

Seemingly, this method looks like Method 3, which stores the cross-layer information separately and keeps it ready for future use. However, in this method, internal profiles rather than external servers are applied. Analogically, Method 1 and 2 store cross-layer information in memory basically, Method 3 stores the information in a network server, while Method 4 does this in local hard disk. Method 4 is flexible since profile formats can be tailored to specific applications, and the interested layers/applications can access the desired information directly. However, it is not suitable for time-stringent tasks.

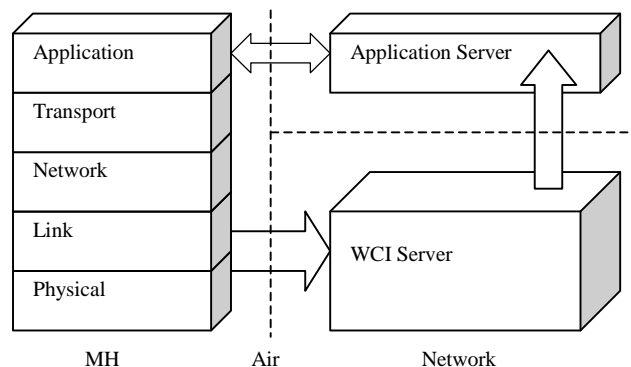
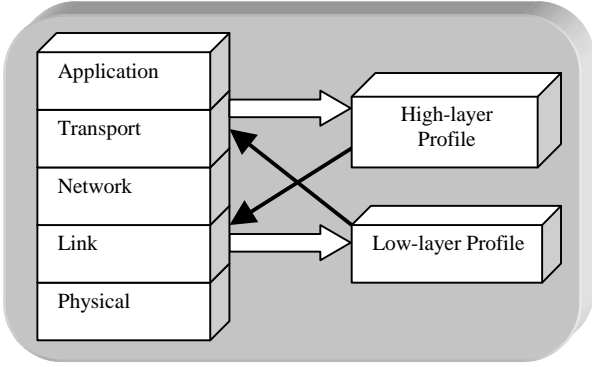
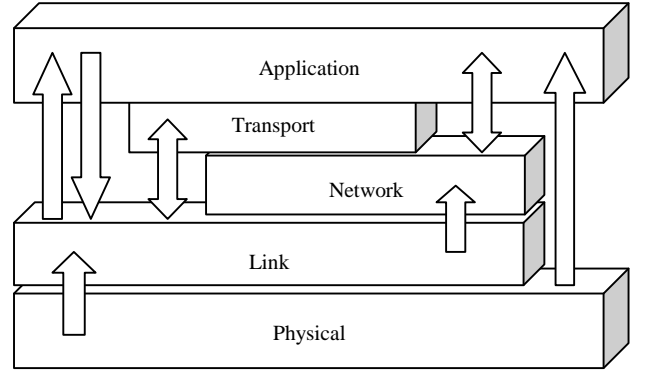


Figure 3. Concept model of Method 3 (network service)



MH

Figure 4. Concept model of Method 4 (local profiles)



MH

Figure 5. Concept model of CLASS

III. DESIGN, EVALUATION AND APPLICATION OF CLASS

A. Design Overview

From the above discussion, a couple of major drawbacks of the existing methods can be identified:

- The signalling propagation paths across the protocol stack are not efficient. The layer-by-layer propagation approach just follows the data propagation mode. Consequently, the intermediate layers have to be involved even if only the source layer and the destination layer are actually targeted. This will cause unnecessary processing overhead and propagation latency.
- The signalling message formats are either not flexible enough for active signalling in both upward and downward directions, or not optimised for different signalling inside and outside the MH respectively. Furthermore, the desired message formats should be scalable enough for rich signalling more than cross-layer hints and notifications [10].

Therefore, we propose a method, named CLASS, as an efficient, flexible and comprehensive scheme with the following distinct features.

1) Direct Signalling between Non-Neighbouring Layers:

The basic idea is to break the layer ordering constraints while keeping the layering structure, i.e., let cross-layer messages propagate through local out-of-band signalling shortcuts. For instance, enable the direct communications between Application Layer and Network Layer without turning to the otherwise middleman, Transport Layer. Although this approach is NOT unknown to the layered protocol stacks, it only appeared as exceptions, and was not designed for generic management functionality. For example, the Layer-3 entity RRM (Radio Resource Management) directly accesses Physical Layer in GSM [11]. A dedicated API between Network Layer and Application Layer was defined in a software simulator, GloMoSim [12]. The same principle is also applicable to the software-based real-world implementations. Obviously, this scheme also applies to signalling between neighbouring layers.

The concept of this feature is illustrated in Fig. 5.

The following presents a simple analysis of propagation latency across the protocol stack.

For methods where a message travels layer by layer, the upward (or vice versa) propagation latency between any two layers, say layer 1 (the source layer, not necessarily Physical Layer) and layer n (the destination layer, $1 < n \leq 5$ in this case), can be formulated as:

$$T_{L1 \rightarrow Ln} = \sum_{i=1}^{n-1} (T_{ri} + T_{pi}), \quad (1)$$

where T_{ri} denotes the travel time between the interfaces of layer i and layer $i+1$, and T_{pi} denotes the processing time (including any queuing delay) at layer $i+1$.

If we let

$$T_p = \sum_{i=1}^{n-1} T_{pi} \quad (2)$$

and assume

$$T_{ri} = T_r, \quad (3)$$

then we obtain the value expression:

$$T_{L1 \rightarrow Ln} = (n-1) \times T_r + T_p. \quad (4)$$

For CLASS, the expression of the same metric is:

$$T'_{L1 \rightarrow Ln} = T_r + T_{p(n-1)}. \quad (5)$$

Assuming the processing time is the same at each layer, then

$$T_{p(n-1)} = T_p / (n-1). \quad (6)$$

Finally, summarising (4) to (6), we reach the following conclusion:

$$T'_{L1 \rightarrow Ln} = T_{L1 \rightarrow Ln} / (n-1). \quad (7)$$

In contrast to the layer-by-layer approach, the propagation latency in CLASS is only about $1/(n-1)$ as large. The more the layers, the more significance it makes. Only when $n=2$ (signalling between neighbouring layers), there is no difference.

2) Light-weighted Internal Message Format:

For internal signalling, it is not necessary to use standardised protocols, which are normally heavy-weighted for, e.g., transmission against errors in the network. For instance, Method 2 [8] uses ICMP messages for internal

signalling. In addition to the large IP header (20 bytes for IPv4), a common ICMP header itself is 8 bytes, where the required *checksum* field is 2 bytes, occupying 25%. Therefore, reducing additional headers and minimising the fields can simplify the internal message format. Although header compression techniques are emerging, it is a problem of another dimension. Essentially, only three fields are required in CLASS:

Destination Address, including destination layer and destination protocol(s) or application(s).

Event Type, indicating a parameter.

Event Contents, the value of the parameter.

If we assign 1 byte to the *Destination Address* and the *Event Type* respectively and assume the *Event Contents* field takes 2 bytes, the whole message size is 4 bytes. Similarly, we examine an IPv4-encapsulated ICMP message with 8-byte header and 2-byte contents. The whole message size is 30 bytes, 7.5 times bigger than that of CLASS.

Messages can also be propagated in an aggregate way by introducing an optional field, *Next Event*.

3) Standardised External Message Format:

For external signalling, ICMP can be used for general messages while TCP/IP headers for short notifications.

4) Other considerations:

A message control protocol is expected to guarantee that dense simultaneous messages across layers can be exchanged in an optimised and organised way to achieve high efficiency and avoid possible conflicts. The message generation and reading mechanisms in Method 2 [8] can be based on. In general, a message with a layer-specific parameter is generated from the specific layer whenever a significant change of the parameter happens. Function calls are used to set and get the parameter, and system calls are used to read the message.

Notably, the actual interactions between layers are task-dependent and protocol-specific. The design of CLASS serves as a framework that allows of different implementations regarding different application scenarios.

B. Evaluation

1) Evaluation Criteria:

Internal Overhead (Overhead within the MH): The average message size and the signalling frequency mainly determine this overhead. This entails an optimised format and signalling thresholds should be applied to avoid excess signalling.

External Overhead (Overhead in the Network): In some contexts signalling between the MH and the access network has to be introduced for system control. In this case, standardised signalling protocols are preferred for implementation purpose.

Propagation Latency: This refers to the time taken by the signalling transmission between the source layer and the targeted layer. The decisive factors include the propagation path and travel time between interfaces, and intermediate processing time (including queuing delay) in the path.

Propagation Direction: Cross-layer signalling messages can be propagated from lower layers to higher layers (upwards) or vice versa (downwards). Bi-directional propagation is required for some complex tasks.

Flexibility: A flexible method can be used for signalling between any two arbitrary layers and for different tasks.

Localisation: The ability to maintain cross-layer signalling within a MH is important as far as the rare radio spectrum and prompt response are considered. Furthermore, this approach would need minimum infrastructure or software modifications on the network side, and thus minimum new investments are needed. Moreover, the network would not need to keep the MHs' states and thus the scalability would not be impacted. Preferably, internal signalling messages only have local meanings and will not be transported to other network entities. A limited number of messages might be exchanged between a MH and its access network, but no cross-layer messages go to the CH unless desired. This feature assures that the MH can communicate with a CH unaware of cross-layer signalling.

Complexity: To achieve a cross-layer signalling method, it would require different levels of OS modifications and internal/external interface design.

2) Comparison of the methods:

Table I qualitatively compares the existing methods with CLASS under the above criteria (extended). In short, CLASS overcomes the two major drawbacks aforementioned.

TABLE I: COMPARISON OF THE CROSS-LAYER SIGNALLING METHODS

Criterion	Signalling Pipe	Selected Holes	Network Service	Local Profiles	Shortcuts (CLASS)
Internal message format	Extension header of IP packet (WEH)	ICMP	N/A	Author-defined	Light-weighted
Internal overhead	Low	Medium	N/A	Low	Low
External message format	WEH or ICMP	ICMP	Author-defined	N/A	Standardised
External overhead	Low to medium	Medium	High	N/A	Low to medium
Propagation scheme	Layer-by-layer data traffic	Layer-by-layer messages	Application-level messages over air	Read and write profiles	Active and direct signalling between any 2 arbitrary layers
Propagation latency	High	Low	Highest	Medium (periodic)	Lowest
Propagation direction	Bi-directional	Upward (Basically)	N/A	Bi-directional	Bi-directional
Flexibility	Low	Medium	Low	High	Highest
Localisation	Medium	High	Lowest	Medium	High
Complexity	Low	Medium	High	Medium	High

First, since CLASS uses the unique active and direct signalling between any two arbitrary layers in both directions, it has the lowest propagation latency with high efficiency and flexibility. Second, CLASS purposely distinguishes between the internal and external messages, and applies optimised or standardised formats for internal or external signalling respectively. Hence it has the lowest overhead when applied within a MH and has a low overall overhead when implemented between a MH and its access network as well. Moreover, CLASS does not exclude the simultaneous use of other methods under some specific circumstances.

Therefore, complex as it is, its efficiency, flexibility and scalability will justify its wide application perspective.

C. Application

1) Application Areas:

CLASS is applicable to various cross-layer signalling scenarios thanks to its comprehensive design. Cross-layer design would benefit those areas where a “global” system factor (GSF) is the target. A GSF can be defined and generated from one of the following three sources. First, the original layer separation and abstraction of a protocol stack had difficulties in clearly placing one function/service in a single layer, e.g., error correction exists in both Link and Transport Layers to fight errors in different levels. Second, the GSF itself is a system-level factor by nature, and can hardly be handled thoroughly in a specific layer. Examples include QoS, resource, energy (power), and security, whose better management would require a collaboration of multiple layers. Third, a GSF can also be a significant change to the original design basis of a protocol stack. Wireless and mobility are good examples, which challenge many design assumptions in the TCP/IP suites and affect all the layers' behaviours. Thus, mobility support [13, 14] and wireless adaptation would be another two application areas.

2) Application Programme: A Case Study

To apply CLASS for a cross-layer design in an application area, which is typically a management task, a designer would

follow a specific design procedure. We present a reference application programme, using a multi-layer wireless QoS adaptation protocol stack as shown in Fig. 6. The basic idea is to achieve adaptive applications and protocols by exchanging and responding controllable QoS parameters between the real- or non-real-time applications and the layers below.

First, identify the layer-specific (“natural”) contributions to this task from each layer. Existing layer-specific mechanisms or protocols make contributions independently. Select the appropriate ones, with enhancements if necessary, and introduce them into the protocol stack. For example, we can introduce IntServ [15] or DiffServ [16] to Network Layer for IP-based QoS management. To control BER (bit error rate), we can introduce FEC (forward error correction) and the optional ARQ (automatic repeat request) to Link Layer. TCP/RTP in Transport Layer can deal with jitter as well as error-related parameters like packet loss ratio.

Second, work out the cross-layer (“added”) contributions from each layer. An added contribution could be either an existing parameter that interests other layers, or a result/behaviour/function/action that needs to be revealed to interested layers. Examples of the former ones are selected environment measurements such as SNR (signal to noise ratio) and RSS (received signal strength), widely available in wireless systems. Contributions of the latter kind need to be parameterised. IP-level handoff notification is a good example. During the whole course of a handoff, Transport Layer needs to adjust its behaviours. For example, TCP can be notified to suspend for the time being to avoid retransmissions. BER is another example from Link Layer to Transport Layer for a joint error control. Similarly, Link and Network Layer could control delay-constrained transport in frame and packet level respectively [17], and thus a joint delay control is also possible. Transport Layer is in charge of reporting error-related parameters and jitter to Application Layer, while Network Layer reports delay constrains. All the parameters are then coded for the CLASS message format.

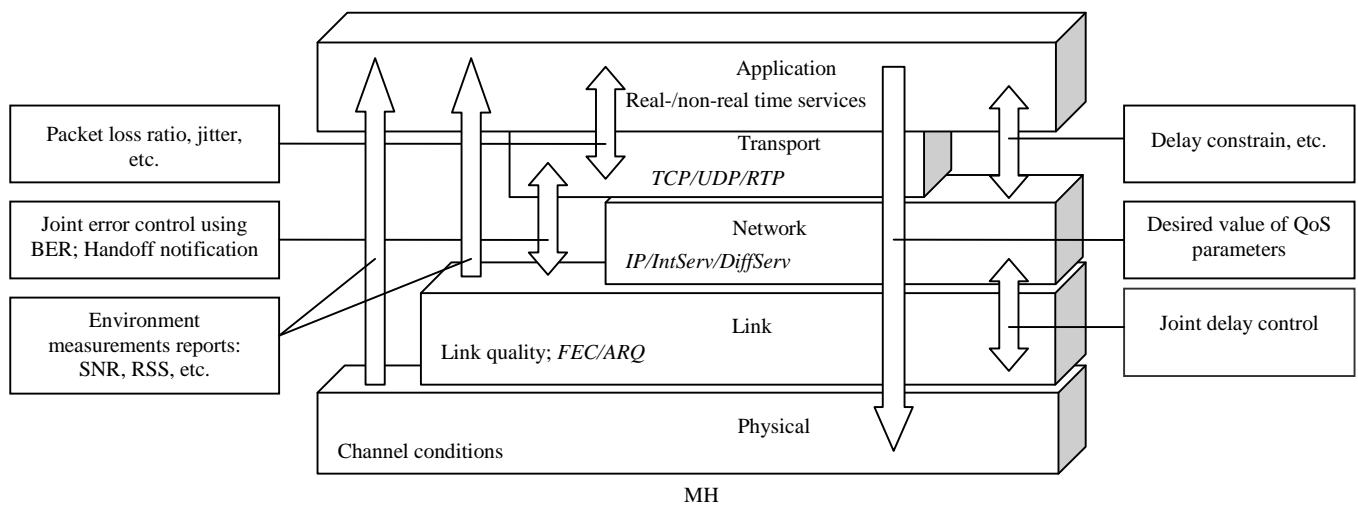


Figure 6. QoS adaptation protocol stack model

Third, with all the contributions available, identify how the layers interact with each other to fulfil the task. In this QoS adaptation case, an application announces its QoS requirements in terms of controllable parameters and corresponding values (or value ranges) to all the related layers. Generally, a real-time service can bear higher packet loss rate or BER as long as the delay and jitter are small enough, whereas a non-real-time service usually has the opposite requirements. Respective layers can then report any significant changes of the parameters to the application, which will adapt to the changes accordingly. Furthermore, the measurement reports from the lowest two layers, together with other parameters, could benefit context-aware applications. What is more, an application could even negotiate a parameter with the responsible layer for cost, energy or resource saving purpose, as long as minimum or enough requirements are met. On receiving such a request, the responsible layer would adjust its behaviour accordingly, e.g., stop an optional control scheme. This behaviour adjustment can be deemed as a protocol adaptation. Therefore, in this case, the adaptation between applications and related layers are reciprocal.

Notably, for a complex task, a task-specific co-ordinator module may be needed to either fully utilise the contributions from each layer in a joint way, or manage related messages in an aggregate way for utilisation convenience. In this case, a module can be set in Application Layer to collect the received parameters, and then interested applications can access to the module for desired parameters, so that duplicate signalling could be avoided.

Following the above steps, we could even obtain an IP-based mobility-aware protocol stack to deal with the management of QoS, radio resource, energy and even more in an integrated way.

IV. CONCLUSION

In this paper, we presented the rationale to introduce the cross-layer design methodology into the IP-based next-generation wireless systems. Especially, cross-layer signalling is a key enabler to achieve cross-layer design. Despite the emergence of several methods, an efficient, flexible and comprehensive cross-layer signalling scheme is still missing. Furthermore, this area still lacks standardisation and evaluation criteria. This paper is our initial effort targeted to these problems. Based on the review of existing ones, we proposed the design framework of a new method called CLASS. We strongly believe that enabling direct communications between arbitrary layers is crucial for the efficiency and flexibility requirements. Moreover, standardising the external message formats and designing light-weighted internal message formats can further enhance

the design. The overall advantages of CLASS seem to be significant when compared with other methods through a qualitative evaluation under a set of criteria that we defined. For introducing CLASS to deal with real-world problems, we presented the recommended guidelines of a working programme. Following the programme, the applications of CLASS would be broad since the multi-layer management concept can apply to the management of QoS, power, radio resource, mobility and even more areas in the IP-based next-generation wireless systems.

Future research is under way to simulate CLASS in a software simulator (e.g., GloMoSim) and implement it on Linux with the OS kernel modified, so that a quantitative evaluation could be performed to further validate the design of CLASS.

REFERENCES

- [1] Z. H. Haas, "Design methodologies for adaptive and multimedia networks," Guest Editorial, IEEE Communications Magazine, Vol. 39, No. 11, pp. 106-107, November 2001.
- [2] B. G. Evans and S. McLaughlin, "Visions of 4G," IEE Electronics & Communication Engineering Journal, Vol. 12, No. 6, pp. 293-303, December 2000.
- [3] H. Balakrishnan, "Challenges to reliable data transport over heterogeneous wireless networks," Ph.D. Dissertation, The University of California at Berkeley, USA, 1998.
- [4] G. Wu, Y. Bai, J. Lai and A. Ogielski, "Interactions between TCP and RLP in wireless Internet," Proc. IEEE GLOBECOM'99, Rio de Janeiro, Brazil, December 1999.
- [5] K. Chen, S. H. Shan and K. Nahrstedt, "Cross-Layer design for data accessibility in mobile ad hoc networks," Wireless Personal Communications, Vol. 21, No. 1, pp. 49-76, April 2002.
- [6] J. Postel, "Internet control message protocol," RFC 792, September 1981.
- [7] A. Conta and S. Deering, "Internet control message protocol (ICMPv6) for the Internet Protocol version 6 (IPv6)," RFC 1885, December 1995.
- [8] P. Sudame and B. R. Badrinath, "On providing support for protocol adaptation in mobile wireless networks," Mobile Networks and Applications, Vol. 6, No. 1, pp. 43-55, January-February 2001.
- [9] B-J "J" Kim, "A network service providing wireless channel information for adaptive mobile applications: part I: proposal," Proc. IEEE ICC'01, Helsinki, Finland, June 2001.
- [10] L-Å k. Larzon, U. Bodin, and O. Schelen,, "Hints and notifications," Proc. IEEE WCNC'02, Orlando, Florida, USA, March 2002.
- [11] B. H. Walke, "Mobile radio networks: networking, protocols and traffic performance," 2nd ed., New York: John Wiley & Sons, 2002.
- [12] GloMoSim, <http://pcl.cs.ucla.edu/projects/gloimosim/>.
- [13] Q. Wang and M. A. Abu-Rgheff, "Towards a complete solution to mobility management for next-generation wireless system," London Communications Symposium 2002 (LCS'02), London, UK, September 2002.
- [14] Q. Wang and M. A. Abu-Rgheff, "A multi-layer mobility management architecture using cross-layer signalling interactions," Proc. IEE EPMCC'03, in press.
- [15] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," RFC 1633, June 1994.
- [16] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, December 1998.
- [17] D. Wisley, P. Eardley and L. Burness, "IP for 3G: networking technologies for mobile communications," Chichester: John Wiley & Sons, 2002.