Survivable wireless networking — autonomic bandwidth sharing in mesh networks

D Quercia, M Lad, S Hailes, L Capra and S Bhatti

Mesh networking has recently received considerable attention, largely as a mechanism for providing enhanced connectivity without the need to install additional expensive infrastructure. It relies on the fact that underutilised local area connectivity can be used to connect constrained devices to those that possess wide-area uplink capabilities. However, at present, proposals for uplink bandwidth sharing are limited by the use of a traditional view of routing in which multiple end-user devices are associated with each individual uplink in such a way that all their off-network traffic is routed through that particular gateway. While this has the merit of simplicity, it is possible for a subset of gateways to be overloaded while others remain underutilised. We propose a new type of local mesh network, called the Coalition Peering Domain, the goal of which is to maximise Internet connectivity dynamically, smoothing out the usage of uplink capacity, albeit at the cost of slightly more complex control and management. Within this paper, we describe three main routing and addressing issues and then propose novel mechanisms that partially address those issues.

1. Introduction

If one were to take a snapshot of activity in a local coffee shop, the chances are that the snapshot would depict people carrying all manner of devices (e.g. mobile phones, PDAs and laptops), sipping mocha mambos, and surfing the Web. Unfortunately, not all people who have a mobile device are able to surf — only those with devices that have a direct Internet connection can, and only then if that connection provides adequate uplink capacity. In principle, however, it is possible for devices to use local area networking to form a mesh network, in which only a few are connected back to the Internet but where the sum of that connectivity is made available to all. Unfortunately, mechanisms that support the formation of such a network are currently very rudimentary.

We propose a new type of local mesh network — the Coalition Peering Domain (CPD) — and we describe it using a simple scenario. We then list its two major benefits and three main technical issues. We present novel solutions to two of these issues and discuss the third as part of future work. The concluding section summarises the paper.

2. Scenario

The scenario we describe has its roots in the SAFECOM activities of the Department of Homeland Security [1], and those of Project Mesa [2]. Both of these attempt to specify requirements for a vision of future emergency service provision in which technology, and specifically networking, has a much greater role than is currently the case.

Consider the circumstances that exist at the scene of an underground tunnel fire. Such fires become very intense very quickly and are difficult to fight — a situation that is not helped by difficulties in communication and a lack of precise knowledge about the state of the environment into which chief officers are sending their personnel. In future, one might reasonably expect the environment to be pervaded by sensors, and for the rescue services to carry additional sensors, both to monitor their own condition and the condition of casualties they might locate. For it to be of any use, information from such sensors must be disseminated quickly to a number of different actors, for example, to allow hospitals to prepare for and then provide effective treatment, and the situation commanders to provide effective co-ordination. Within the SAFECOM view of the world, there is also a requirement for bi-directional multimedia channels, providing, for example, outgoing video information about the condition of casualties to allow for more effective on-site intervention, or incoming structural plans to the fire-fighters on the ground.

Such requirements are predicated on the existence of adequate and direct Internet connectivity. However, that which was pre-deployed in the environment may have been damaged or destroyed by fire, or may be inadequate for the volume of traffic being generated. Thus, to overcome this, the wireless sensor, video, and end-terminal equipment could form an *ad hoc* wireless local area network that includes a variety of gateway devices located both within the tunnel (as pre-deployed systems) and at its opening (as a system brought into the environment by the rescue services). This network is extended above ground to devices that have Internet connectivity (including police radios and civilian mobile phones).

To obtain the maximum benefit from such a network, i.e. to provide better throughput and robustness, the set of outgoing connections needs to be presented to nodes as a single logically aggregated connection rather than a set of individual connections with variable and varying QoS. To do this, it is necessary to form a special type of local mesh, which we have called a Coalition Peering Domain.

3. Benefits from using a CPD

Within the context of the scenario above, we discuss two main benefits for CPD members.

3.1 Throughput

In our scenario, the mobile phones of two people passing by the scene of the emergency may be idle. If they do not join a CPD, their connections remain unused. However, if they allow their devices to form a CPD with the gateway device at the tunnel opening, then the gateway can use these connections to achieve higher data rates (and thus send higher quality video of the scene and the injuries) than it is capable of doing with only its own Internet connection.

3.2 Robustness and availability

To send video to hospitals and police stations, the paramedics' sensor and camera equipment underground connect through the gateway device, which has a direct Internet connection. If this connection fails, video transmission is either interrupted (if the gateway device has not formed a CPD with nearby devices) or continues seamlessly (if the gateway device has formed a CPD). In other words, the CPD both enhances the robustness of video transmission and guarantees the transmission availability for the underground relief workers.

4. Issues in using a CPD

We now describe the addressing and routing issues facing a CPD.

4.1 Addressing within a CPD

A CPD has to work without the presence of centralised control, i.e. control must be distributed across all its members. This leads to a major addressing problem because any two CPD members may choose the same address for their own devices, resulting in an address clash within the CPD. For example, two civilians passing the scene may have configured their phones with the same address from an address pool that is commonly used (e.g. 192.168.1.x).

4.2 Trust in routing

Consider a CPD that comprises the gateway device at the tunnel opening (G) and the mobile phones of a policeman and a civilian passing by the scene (P_p and P_c, respectively). Whenever G has to transmit and thus choose its next-hop between P_p and P_c, the chances are that it chooses P_p rather than P_c because a police officer may in general be considered more trustworthy than an unknown passer-by. However, if G's connection through P_p experiences considerable packet loss, G may revise its trust in P_p and consequently choose P_c as its next-hop.

4.3 Impact of multi-path routing

To make the best use of all connectivity that is available to the CPD as a whole, coalition members send packets through multiple different paths across the CPD and, consequently, across the Internet [3, 4]. The resulting patterns of delay, mis-ordering, jitter and loss will be different to those from single-path routing and may affect communication between the emergency services at the scene and the hospitals or authorities with which they communicate.

5. Solutions for creating a CPD

In this section, we present solutions to the first two issues of section 4.

5.1 Addressing within a CPD

Each coalition member is independent. They each control and maintain their own devices. Although it is conceivable that some prior addressing and communication schemes could have been organised between paramedics' sensor and camera equipment, the gateway device, and the policemen's mobile phones, the same cannot be said for the mobile phones of people passing by. A mechanism is therefore needed to enable these independent devices to communicate while forming a CPD. Although network address translation (NAT) may provide a solution, we have already highlighted the potential problem of address clashes.

Our solution uses a combination of IPv6 link-local addressing, IPv6 multicast and globally reachable IPv6 address block allocations. We propose that, initially, the paramedics' sensor and video equipment, the gateway at the tunnel opening, the police and civilian mobile phones all listen on a 'well-known' IPv6 multicast address and port on their local area wireless connections. When recovery efforts begin, the gateway device transmits a CPD peering request (*CPD_PREQ*) message on the well-known multicast address with the gateway device's IPv6 Link-Local address as the source. It contains the resource parameters that it is willing to offer and an identifier for the CPD (*CPD_ID*). This initiates a standard three-way handshake between the gateway device and the various sensor, video and police or civilian mobile phones that receive the *CPD_PREQ*. The various sensor, video and police or civilian mobile phones transmit back to the gateway device's IPv6 Link-Local address a CPD peering response (*CPD_PRESP*) accepting the peering request.

These responses contain the resource parameters that the various devices are willing to offer in return. On receiving a *CPD_PRESP*, the gateway device transmits directly a CPD advertisement (*CPD_ADV*) to confirm the establishment of a local peering agreement between them. From then on, a soft-state mechanism is used and regular *CPD_ADV*s are exchanged to sustain the local peering agreements. Failure to receive an expected *CPD_PRESP*, or a *CPD_ADV* results in a time-out and termination of the local peering agreement.

The members of the CPD that are above ground include the policeman's and civilian's mobile phones, as well as the gateway device at the tunnel opening. These all have direct Internet connectivity that they share with CPD members, thus they form the edge of the CPD. We call these members coalition edge-forwarders (CEFs). We assume that CEFs each have an IPv6 globally routed address block, which they may divide and sub-allocate to those members who do not have direct Internet connectivity, coalition internal-forwarders (CIFs). In the context of our scenario, the gateway device divides its IPv6 globally routed address block and sub-allocates addresses among the paramedics' sensor and video equipment underground. This also behaves well with reverse path traffic entering the CPD, destined for the paramedics working underground because it is forwarded easily to the gateway device (which owns the address block) and can then be propagated down to the CIFs inside the tunnel.

5.2 Trust in routing — STRUDEL

In our scenario, the gateway device at the tunnel opening (*G*) may select its next hop as the policeman's and passer-by's mobile phones, which may selfishly (or even maliciously) elect not to forward packets. To detect such a situation, we have proposed that *G* runs STRUDEL [5], a distributed adaptive framework that combines trust-informed selection of the forwarding path for packets with a mechanism for identifying and isolating misbehaving peers. It makes use of reputation evidence (i.e. direct experience evaluations and recommendations) to support trust and, consequently, the formation and maintenance of CPDs. In short, STRUDEL consists of:

- an approach for detecting malicious nodes based on the 2-ACK scheme [6],
- a Bayesian formalisation for trust formation and trust evolution [7] that possesses a range of desirable properties:

  — support for fine-grained discrete trust metrics, as opposed to the binary metrics currently used by current Bayesian trust models,

  — use of recommendations that are weighted according to recommenders' trustworthiness and recommenders' subjective opinion — to distinguish honest and dishonest recommenders and to resolve the different ontological views of the world honestly held by different peers,

  — incorporation of the time dimension to prevent nodes from capitalising excessively on past behaviour),
- a forwarding mechanism that integrates the Bayesian trust model and locally maximises each peer's utility [8, 9].

This is achieved by minimising the use of heavyweight mechanisms, and by removing the assumption that there is a public key infrastructure (PKI) in place. In fact, the most heavyweight mechanisms (e.g. per packet signatures) are activated only when there is evidence to suppose that misbehaviour is occurring. In addition, to support the 2-ACK scheme, STRUDEL does not require a trusted binding between a real identity and its corresponding public key, but rather only between a peer address and its public key [10].This can be achieved by means of cryptographically generated addresses [11], without the need for a PKI.

At the heart of STRUDEL is a state machine, which we now describe.

5.2.1 STRUDEL's state machine

Each CPD member can be described by the state machine depicted in Fig 1. As our scenario comprises the gateway device (*G*) that may select P_p (the policeman's mobile phone) as its next hop, we now describe how *G*'s state machine carries out this selection.

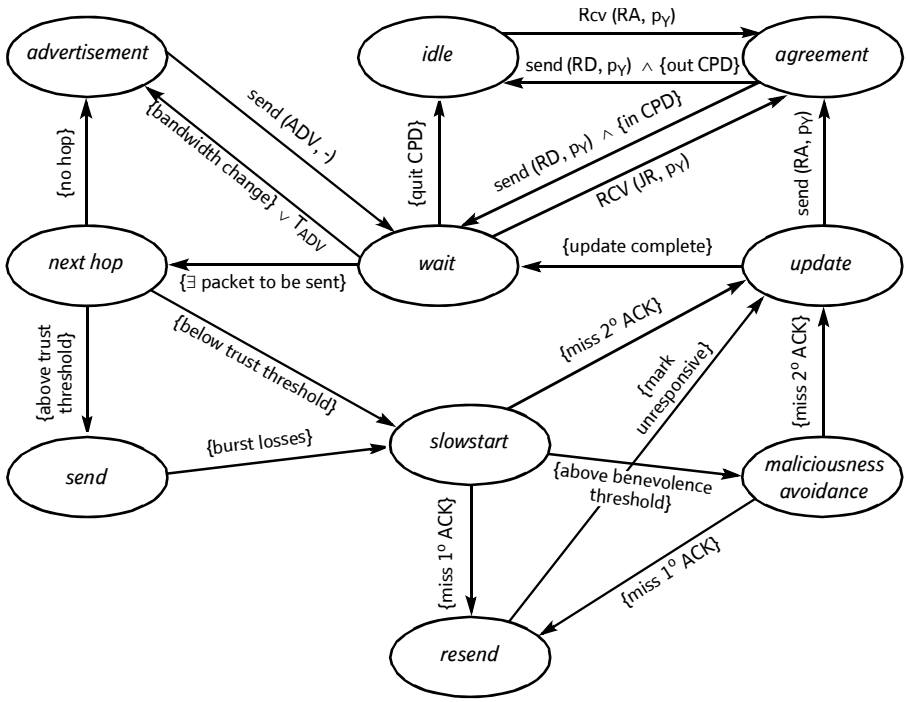

Fig 1 A CPD member's state machine.

As mentioned, to initiate a CPD, the *idle* peer *G* multicasts a CPD peering request (*CPD_PREQ*), containing a list of its minimum requirements for peering to be feasible (ranging from minimum bandwidth, maximum loss rate, and so forth through to constraints on the credentials that are acceptable). The peer *G* enters the *agreement* state in two possible situations:

- *G* receives a CPD peering response (*CPD_PRESP*) message from P_p accepting its *CPD_PREQ* — the peer *A* evaluates peering agreement terms contained in the *CPD_PRESP* (e.g. amount of bandwidth P_p is willing to offer), P_p's credentials (i.e. other peers' ratings about P_p , some of which may be forwarded by P_p), and *G*'s locally historical experiences with P_p, and, based on this information, *G* decides whether or not to continue entering a peering agreement with P_p,
- *G* receives a *CPD_PREQ* from a peer P_p, asking to form another CPD with *G* — as *CPD_PREQ* also contains P_p's credentials, *G* evaluates them and decides whether to accept (sending back a positive *CPD_PRESP*), or to reject (sending back a negative *CPD_PRESP*) the peering agreement.

If *G* accepts at least one peering agreement, it enters the *wait* state, meaning that it belongs to at least one CPD and it is ready for action. It may then enter the *advertisement* state to send advertisements to its peering nodes either to refresh its peering agreement terms periodically or to change them (e.g. when *G*'s available bandwidth changes). An advertisement $ADV_{a,Pp}$ from *G* to P_p contains:

- the total (Internet plus local) bandwidth that *G* is willing to offer to P_p,
- *G*'s current trust in P_p.

Therefore, from the advertisement, P_p determines the amount of traffic that *G* is willing to receive and obtains a trust tuple that it may later use as a recommendation letter.

Whenever *G* has to send packets, it must first select the forwarding peer P_p, based on stored reputation and routing information, from *G*'s ISP (or hot spot) and the set of peers with which *G* has an agreement (e.g. P_p and P_c). The selection process is performed in the *next hop* state and is based on utility maximisation, where utility is a measure of delivery probability. Based on the next hop's reliability, *G* selects one of the two sending modes — either normal operation mode or suspicious mode. If *G* deems P_p reliable, it enters normal operation mode represented by the *send* state. Here, *G* sends packets to P_p without any mechanism for maliciousness detection, other than an assessment of end-to-end loss rate. At this point, unless there is some mechanism for assessing whether packets have been delivered, there is no incentive for a forwarder to act honestly. If an end-to-end mechanism is available — for example, TCP acknowledgements — then it may be used to assess whether the loss rate on the entire link between sender and destination is acceptable. We know that if the loss

rate is acceptable, then peers within the community are well (enough) behaved and we need not track their activities more closely. However, the converse is not true — if the end-to-end loss rate is unacceptable, it is still entirely possible for all peers to behave honestly and well, and for the loss to be occurring within the wider Internet. In this case, we need to identify the cause of the problem and, as a consequence, if the loss rate is unacceptable, or if there is no end-to-end mechanism for acknowledgement, *G* switches into suspicious mode.

In suspicious mode, all forwarding peers are required to use Yau and Mitchell's 2-ACK scheme [6]. At the price of greater packet overhead, the scheme allows the identification of suspicious peers along the path and also allows us to distinguish them from unresponsive peers. Suspicious mode comprises two substates — *slowstart* and *maliciousness avoidance*. In the former state, *G* sends packets to P_p according to a sending window which grows exponentially. After exceeding a threshold (benevolence threshold), *G* enters the latter state, in which the sending window grows linearly. If the next hop is unresponsive, *G* enters the *resend* state and retransmits the same packet up to a retransmission threshold. When *G* exceeds the threshold, it marks P_p as unresponsive and enters the *update* state to update its routing information. Reputation information is updated when:

- either a next-hop node is cleared of suspicion,
- or it initiates a peering agreement.

We use the trust framework presented in Quercia et al [5] to update reputation information.

5.2.2 Impact of STRUDEL's trust model

Reputation and trust information is stored and processed locally to each node. Wherever a node misbehaves, either maliciously or in terms of promising more bandwidth than it can deliver, its reputation suffers and, as a consequence, less traffic is directed towards it. Thus, malicious nodes are likely to become isolated and overloaded nodes will have the amount of traffic directed towards them regulated by the decrease in their reputation.

One of the experiments we have conducted was to determine the impact of the trust management framework, which STRUDEL integrates, on successful packet delivery in a local network where some of the peers act maliciously. Benevolent peers share their connectivity, whereas malevolent ones exploit others' connectivity without actually sharing their own.

- Simulated configuration

  As we are interested in analysing the local impact of our framework at a peer level, we simulate a configuration consisting of a peer *G* and a set of corresponding next hops. These are connected directly to the Internet. We consider *G* forwarding packets to its next hops, which make their connectivity available. *G* selects a next hop either randomly or through a trust-informed decision (discussed later). The next hop acts according to the behavioural model to which it belongs.

- Next-hop behavioural models

  A next hop belongs to one of the following four behavioural models — fully malicious, malicious, benevolent, and fully benevolent. Depending on its behavioural model, a next hop offers the following packet loss ratios if it were selected for the whole simulation duration: 100% for a fully malicious next hop, 70% for a malicious one, 30% for a benevolent one, and 15% for a fully benevolent one. Both fully malicious and malicious next hops drop packets randomly, whereas both benevolent and fully benevolent do it according to a Gilbert model [12].

  To understand why, consider that the next hops are connected directly to the Internet. As a consequence, packet losses through (fully) benevolent next hops depend on Internet congestion, which is bursty. A Gilbert model reproduces such burstiness. We have thus implemented the model, the parameters of which varied according to the packet loss ratios it simulated (either 30% or 15%).

- Next-hop selection methods

  G chooses its next hops in two different ways. The first is random selection, i.e. it selects each of its next-hops with equal probability. The second is trust-informed selection, i.e. it selects the most trustworthy next hop.

- Simulation execution

  A simulation consists of several executions of an experiment. An experiment duration is of 100 time units. At each time unit, *G* selects one of its next hops and sends it a stream the size of which is 10 packets. Based on the number of packet losses, *G* computes its satisfaction and consequently evolves its trust. We collect the overall number of packet losses at each time unit. We run each experiment 10 times and the results of all runs are averaged.

- Experiment metric

  This is *G*'s average fraction of successfully sent packets.

- Set-up

  We simulate *G* with four next hops, one for each next-hop behavioural model. We first consider *G* using random next-hop selection. We then consider *G* using trust-informed selection.

- Results

  When using trust-informed selection, *G* forwards most of its traffic (97%) to the fully benevolent next hop as depicted in Fig 2. As a result of using trust-informed selection, *G* successfully sends 84% of the packets on average, in contrast to only 42% when using random selection.

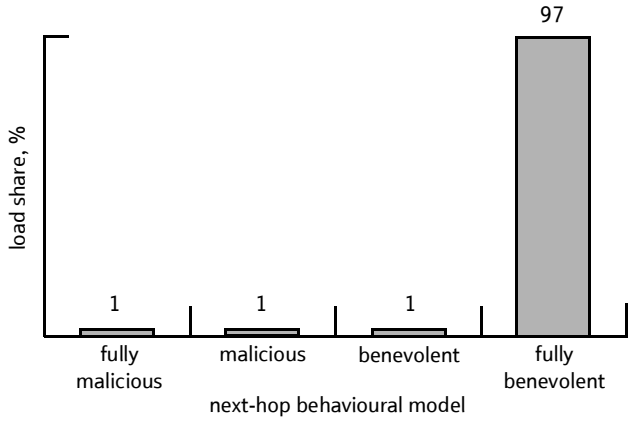

Fig 2 Load share among P_x's next hops, using trust-informed selection.

6. Future work

Section 4 highlighted three issues (addressing, trust in routing, and multi-path routing). Section 5 then proposed novel mechanisms that solve the first two of these issues. We now discuss the third issue: that of multi-path routing. This may have implications particularly for higher layer protocols and applications that rely on specific behaviour from the underlying routing infrastructure. The health monitoring and video traffic transmitted by paramedics' equipment may arrive at hospitals and police stations with some delay or in an unordered fashion. Although not a significant problem when the amount of data is low, hospitals and police incident co-ordinators will notice degradation in quality when the amount of data is high (e.g. high quality video and continuous casualty/patient body monitoring). To continue providing real-time information, applications may need to drop delayed and mis-ordered data packets. This may lead to holes or gaps in patient monitoring data, or glitches and jerkiness in video and audio from the scene. This, in turn, could have an impact on the interactivity between the hospital and the paramedics at the scene, affecting the ability for doctors at the hospital to ask questions and provide valuable instruction to the paramedics. The use of standard play-out buffering techniques at the receiver may alleviate some of the problems from mis-ordered data packets (at the cost of additional play-out delay). Also, it may be possible to use adaptive forward-error-correction techniques, such as redundant encoding, to cope with some mis-ordering as well as some loss. One exciting possibility that is well aligned with the fundamental deployment scenario is an approach based on network coding. In this, various data streams are mixed at intermediate nodes, giving greater information throughput than can be achieved with routing alone. This is for future work.

7. Related work

In this section, we describe related work on trust management and bandwidth sharing in mesh networks.

7.1 Trust management

The distributed trust framework [7], which STRUDEL [5] integrates, models trust as a social concept. Foundational distributed trust frameworks were already based on social trust considerations [13, 14], in that they evolved trust based on direct experiences and recommendations, and they integrated the classical trust dimensions of context, subjectiveness, and (only later) time. Abdul-Rahman and Hailes [15] first proposed the use of recommendations for managing context-dependent and subjective trust [16]. Although foundational, the previous approach suffered from, for example, the lack of a process for trust evolution. To fill the gap, Mui et al [17] proposed a Bayesian formalisation for a distributed rating process. However, two issues remained unsolved — they considered only binary ratings and did not discount them over time. Buchegger and Le Boudec [18] tackled the latter issue, but not the former; they proposed a Bayesian reputation mechanism in which each node isolates malicious nodes, ages its reputation data (i.e. weights past reputation less), but can only evaluate encounters with a binary value (i.e. encounters are either good or bad). Using a generic *n*-level discrete trust metric, our trust framework addresses the issue. Furthermore, it discounts its trust beliefs over time (i.e. it decreases the confidence level it has in its trust beliefs). This avoids excessive capitalisation on past good behaviour and allows the discarding of old reputation information (contributing to making the framework lightweight). Trust frameworks' integration with decision-making mechanisms, though fundamental, is rare. Within the SECURE project, a trust model's output feeds a decision-making mechanism [19]. More recently, Quercia and Hailes [8, 9] proposed a decision model for trust-informed interactions that, on input of trust assessments, estimates the probability of potential risks associated with an action, based on which it decides whether to carry out the action. STRUDEL integrates such a model when deciding the best action to be taken.

7.2 Bandwidth sharing in mesh networks

In the context of mesh networks, users may choose to share their wide-area connectivity with several others through simple connection-sharing mechanisms. However, traditional models for doing this have focused on sharing a single connection between multiple devices, specifically when this connection is temporarily idle, by treating the local device as a temporary gateway [20]. A good example is the 7DS Peer-to-Peer Information Dissemination and Resource Sharing system [21] that provides a mechanism for self-organised connection sharing. In general, the possibility of utilising multiple connections simultaneously is not taken into account.

The CUWiN project [22] moves substantially towards simplifying mesh formation and allows 'users to buy bandwidth in bulk and benefit from the cost savings'. However, the level of automation that its deployment entails comes at a price — a lack of local control. In our scenario, such a loss of local control would be unacceptable for both police officers who need to co-ordinate the emergency aid efforts, and for paramedics who need to transmit as much information as possible to hospitals to the best of their training and abilities. The CUWiN software also requires relatively higher-powered machines, limiting the ability to form a mesh using resource-poor sensors, video and mobile communications equipment.

Although the 'MAR commuter mobile access router' [23] provides bandwidth aggregation by using multiple wide-area wireless interfaces simultaneously, it focuses on a hot spot model of access. All local users gain wide-area access via a single provider (i.e. a MAR device), representing a single point of failure.

8. Conclusions

Increasingly many devices possess local area wireless connectivity as standard; however, it is likely to remain the case for the foreseeable future that:

- many such devices will not possess wide-area connectivity in addition,
- even for those devices that do possess such connectivity, the presumption of anywhere, any time direct access may prove to be a poor one — as higher frequencies are used, more base-stations are required to cover the same area, and the cost rises correspondingly.

Consequently, where wide-area connectivity is necessary, mesh networking appears to provide a relatively cost-effective enhancement to existing direct connection approaches. In this paper, we proposed a new routing architecture — the Coalition Peering Domain — that enables nodes to form a local mesh network and to integrate their local and Internet connectivity. Because the mechanism we propose is inherently distributed, it has advantages over previous approaches to mesh networking in that it allows higher instantaneous throughput, and greater survivability. However, the approach does lead to greater complexity in the control plane and, in this paper, we have outlined three of the most significant problems that arise if one is attempting to provide seamless integrated mesh connectivity, and have presented putative solutions to two of them. The third remains as future work.

Acknowledgements

We gratefully acknowledge the support of the European Commission through the SEINIT and RUNES projects.

References

1 SAFECOM Statement of Requirements — http://www.safecomprogram.gov/NR/rdonlyres/A1118073-1B21-42DC-941F-C9DB26F4DBEF/0/PSCI_Statement_of_Requirements_v1_0.pdf

2 Project MESA Statement of Requirements — http://www.projectmesa.org/MESA_SoR/SoR.htm

3 Lad M, Bhatti S, Hailes S and Kirstein P: 'Enabling Coalition-Based Community Networking', in Proceedings of the London Communications Symposium, London, UK (September 2005).

4 Lad M, Bhatti S, Kirstein P and Hailes S: 'Challenges, Opportunities and Incentives for Coalition-Based Community Networking', in Technical Research Note Number RN/05/08, University College London, London, UK (June 2005).

5 Quercia D, Lad M, Hailes S, Capra L and Bhatti S: 'STRUDEL: Supporting Trust in the Dynamic Establishment of peering coalitions', in Proceedings of the 21st ACM Symposium on Applied Computing, Dijon, France (April 2006).

6 Yau P W and Mitchell C J: '2HARP: A secure routing protocol to detect failed and selfish nodes in mobile *ad hoc* networks', in Proceedings of the 5th World Wireless Congress, pp 1—6, San Franciso, USA (2004).

7 Quercia D, Hailes S and Capra L: 'B-trust: Bayesian Trust Framework for Pervasive Computing', in Proceedings of the 4th International Conference on Trust Management, LNCS, Pisa, Italy (May 2006).

8 Quercia D and Hailes S: 'Risk Aware Decision Framework for Trusted Mobile Interactions', in Proceedings of the 1st IEEE/CreateNet International Workshop on The Value of Security through Collaboration, Athens, Greece (September 2005).

9 Quercia D and Hailes S: 'MATE: Mobility and Adaptation with Trust and Expected-utility', International Journal of Internet Technology and Secured Transactions, 1 (2007).

10 Quercia D, Hailes S and Capra L: 'TATA: Towards Anonymous Trusted Authentication', in Proceedings of the 4th International Conference on Trust Management, LNCS, Pisa, Italy (May 2006).

11 Aura T: 'Cryptographically Generated Addresses (CGA)', in Proceedings of the 6th LNCS International Conference on Information Security, pp 29—43, Bristol, UK (2003).

12 Arai M, Chiba A and Iwasaki K: 'Measurement and modeling of burst packet losses in Internet end-to-end communications', in Proceedings of the IEEE International Symposium on Dependable Computing, pp 260—267, Hong Kong (1999).

13 Carbone M, Nielsen M and Sassone V: 'A Formal Model for Trust in Dynamic Networks', in Proceedings of the 1st IEEE International Conference on Software Engineering and Formal Methods, pp 54—63, Brisbane, Australia (September 2003).

14 Marsh S: 'Formalising Trust as a Computational Concept', PhD Thesis, Department of Mathematics and Computer Science, University of Stirling (1994).

15 Abdul-Rahman A and Hailes S: 'Supporting Trust in Virtual Communities', in Proceedings of the 33rd IEEE Hawaii International Conference on System Sciences, 6, p 6007, Washington DC, USA (2000).

16 Abdul-Rahman A and Hailes S: 'Using Recommendations for Managing Trust in Distributed Systems', in Proceedings of IEEE Malaysia International Conference on Communication, Kuala Lumpur, Malaysia (November 1997).

17 Mui L, Mohtsahemi M, Ang C, Szolovits P and Halberstadt A: 'Ratings in Distributed Systems: A Bayesian Approach', in Proceedings of the 11th Workshop on Information Technologies and Systems, New Orleans, Louisiana, USA (December 2001).

18 Buchegger S and Boudec J-Y L: 'A robust reputation system for p2p and mobile ad-hoc networks', in Proceedings of the 2nd Workshop on the Economics of Peer-to-Peer Systems, Cambridge, MA, USA (June 2004).

19 Dimmock N: 'How much is 'enough'? Risk in Trust-based Access Control', in Proceedings of the 12th IEEE International Workshop on Enabling Technologies, pp 281, Washington, DC, USA (June 2003).

20 Zhu D and Mutka M W: 'Promoting Cooperation Among Strangers to Access Internet Services from an Ad Hoc Network', in Proceedings of the 12th IEEE International Conference on Pervasive Computing and Communications, pp 229—240, Orlando, FL, USA (March 2004).

21 Papadopouli M and Schulzrinne H: 'Connection Sharing in an Ad Hoc Wireless Network among Collaborating Hosts', in Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), pp 169—185 (June 1999).

22 Champaign-Urbana Community Wireless Network — http://www.cuwireless.net/

23 Rodriguez P, Chakravorty R, Chesterfield J, Pratt I and Banerjee S: 'MAR: A Commuter Router Infrastructure for the Mobile Internet', in MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services, pp 217—230, ACM Press (2004).

Daniele Quercia is a PhD student in the Computer Science department of the University College London, working in the Networks Research Group. His research focuses on enabling mobile resource sharing in the presence of untrustworthy devices. Before commencing a PhD in the department, he was a Research Fellow on the SEINIT European project. He studied at Politecnico di Torino (PoliTO), Italy, for his Master of Science in Computer Engineering. As a recipient of international scholarship awards (while at PoliTO), he was a visiting student at Universitaet Karlsruhe, Germany, and University of Illinois, Chicago, USA.

Manish Lad is a Research Fellow in the UCL Department of Computer Science Networks Research Group. He is also studying for a PhD within the department. In 1999 he received a First Class BSc Honours degree in Computer Science and in 2000 an MSc with Distinction in Data Communications, Networks and Distributed Systems, both from UCL. He was then involved in IPv6 protocol stack development for a core Internet router at Lucent Technologies Inc. After rejoining UCL in 2003, he has been involved in a number of pan-European collaborative projects including the 6NET IPv6 pilot network deployment, the SEINIT security expert initiative and the RUNES project examining the creation of reconfigurable ubiquitous networked embedded systems. His main research interests lie in IP-based networking, specifically in aspects of network connectivity, topology, and infrastructure, including IPv6, routing, overlays and multicast. He also has an interest in the aspects of mobility and distributed computation within this context.

Stephen Hailes is the Deputy Head of the Computer Science Department at UCL, and a leader of its mobile systems research activities. His primary research interests lie with issues within mobile computing, specifically pervasive and networked embedded systems, and security, specifically the formation of trust. He has been PI or Co-I on a range of national and EC-funded projects, and is currently technical manager and PI on the FP6 IP RUNES, examining the creation of reconfigurable networked embedded systems, PI on the BT/EPSRC MARS project examining the use of AI-techniques in producing robust pervasive systems, Co-I on the EPSRC Divergent Grid project examining security and availability in grid systems. He is PI in two further EPSRC projects (SESAME and UtiForo) and Co-I in an EC-funded project (U2010), covering issues in sensor networking and trust, that are due to start shortly.

Licia Capra is a Lecturer in the Department of Computer Science at University College London. She holds the equivalent of an MSc degree in Computer Science from the University of Bologna (Italy), and a PhD degree also in Computer Science from University College London. During her PhD, she has published extensively in the area of mobile computing, middleware and software engineering. She is currently investigating game-theory techniques to promote collaborations in mobile networks, trust management systems for pervasive scenarios, and models of collaboration for sensor networks. She has served as reviewer for various software engineering and distributed system journals, as well as external reviewer for a number of international conferences and workshops in the same area. She has also been working as a post-doctoral fellow on the TAPAS European project (IST-2001-34069), the aim of which is to support the composition of services across organisational boundaries.

Saleem N Bhatti is a Professor at the School of Computer Science, University of St Andrews. He is a member of the Networked and Distributed Systems (NDS) Research Group. Previously, he was at the Department of Computer Science, University College London (UCL). He has been involved in collaborative research since 1991, and has experience of working with network operators, equipment manufacturers, research institutions and academic institutions from all over Europe. His work has involved aspects of multiservice networking, tele-working, multicast, network and systems management, network security, IPv6, mobile systems, the consideration of quality of service adaptability support for Internet applications and adaptive systems, and high-speed networking (e.g. for grid networking and grid computing). He is also involved with industry in various consultancy roles in the area of networking technology and systems.