

Notes on Scalable Models for Synchronous Multimedia Distribution

Petr Holub

Faculty of Informatics and Institute of Computer Science,
Masaryk University Brno, Botanická 68a, 602 00 Brno, Czech Republic

e-mail: hopet@ics.muni.cz

ABSTRACT

This technical report presents some technical details on scalable distribution models introduced in [6]. We devise analytical description of several models and compare their relative scalability.

Contents

1	Introduction	2
2	Scalability of Single Reflector	2
3	Full 2D Mesh of AEs	3
4	3D Layered Mesh of AEs	5
4.1	Transition from 3D to 2D mesh.	6
5	3D Layered Mesh of AEs with Intermediate AEs	7
5.1	Aggregation of inner AEs	8
6	Conclusions	9

1 Introduction

Current Internet environment has enabled fast transfers of huge amounts of data making high quality multimedia and collaborative applications a reality. Both collaborative and multimedia applications involve processing of specific data with special requirements on distribution and delivery. However, the processing itself needs to become distributed as the required vast amount of network traffic and processing capacity can easily overload any existing commodity centralized solution. Another reason for creating distributed solution is improvement in terms of robustness and fault tolerance.

The problem of distributed multimedia processing can be divided into two classes of problems: *synchronous* (on-line or interactive) processing and *asynchronous* (off-line or non-interactive) processing. Though these two classes might ultimately converge, they have their own distinct problems and goals. The problem of synchronous data processing aims at processing high data volumes with as low latency as possible and thus the amount of processing is limited by the latency requirements.

For synchronous multimedia distribution to larger groups of receivers, multicast [7] is a natural solution. However, as the multicast networking is rather difficult to manage, it is a well-known fact that despite it is quite old technology, it is often either misconfigured or unsupported at all in current high-speed networks.

To address these issues, we have introduced a user-empowered UDP packet reflector based on an active router architecture [1, 2]. Because of its centralized nature, its scalability is however limited and this has shown especially when it comes to high-bandwidth data streams like Digital Video [4, 3]. Thus we have developed several decentralized distribution schemes, that are published in [3, 6]. These schemes have been studied not only in terms of scalability, but also in terms of robustness, especially in [6]. As there was not enough space available for publishing all technical details of scalability of these schemes, we present analytical description of several simple and analytically easily comprehensible ones in this report.

The report is organized as follows: Section 2 describes behavior and scalability properties of single reflector, Section 3 presents two-dimensional full-mesh models, Section 4 extends these models into third dimension. Section 5 presents three-dimensional models with intermediate reflectors that can be used as a bridge to multicast-like distribution schemes. Please note that for sake of simplicity, we use continuous description, though the discrete one could be more appropriate. However, we suppose that to understand problems stemming from discretization could be described using discrete event network simulators and comparing results with continuous ones.

2 Scalability of Single Reflector

The simple UDP packet reflector replicates the data to all the clients, except for the source client and thus the limiting outbound traffic on the reflector grows with

$$out = n(n - 1) \tag{1}$$

where n is the number of active (sending) clients. Both simple reflector implementation and the implementation based on the active router are shown to comply with this behavior [1].

An enhanced version of reflector that support distributed operation is called Active Element (AE) [5, 6] and this name will be used in rest of this report.

3 Full 2D Mesh of AEs

Very simplest model, that is actually an extension of simple tunneling between two AEs, is a two dimensional mesh of AEs with each AE connected directly to all remaining AEs, as shown in Fig. 1.

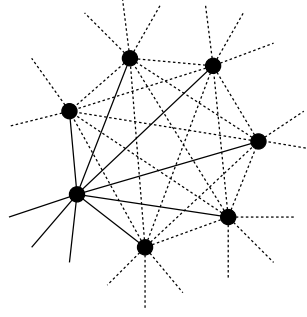


Figure 1: 2D full mesh.

Let's assume network of m_{tot} AEs with full-mesh communication. n clients connect to the AEs in such way that each AE has either n_r or $n_r - 1$ clients.

$$n_r = \lceil \frac{n}{m_{tot}} \rceil \quad (2)$$

$$m_1 = n_r m_{tot} - n \quad (3)$$

$$m = m_{tot} - m_1 \quad (4)$$

When full mesh operates in N:N way, the inbound traffic for AEs with n_r clients will be

$$in = \underbrace{n_r}_1 + \underbrace{(m-1)n_r}_2 + \underbrace{m_1(n_r-1)}_3 \quad (5)$$

(1 ... directly connected clients, 2 ... streams from $m-1$ other AEs with n_r clients, 3 ... streams from all m_1 clients with n_r-1 clients) and for AEs with n_r-1 clients

$$in_1 = \underbrace{n_r-1}_4 + \underbrace{mn_r}_5 + \underbrace{(m_1-1)(n_r-1)}_6 \quad (6)$$

(4 ... directly connected clients, 5 ... streams from all m AEs with n_r-1 clients, 6 ... from other m_1-1 AEs with n_r-1 clients).

Outbound traffic for AE with n_r clients will be

$$out = \underbrace{(n_r - 1)n_r}_{7} + \underbrace{(m - 1)n_r^2}_{8} + \underbrace{m_1 n_r (n_r - 1)}_{9} + \underbrace{n_r(m + m_1 - 1)}_{2 \times 10} \quad (7)$$

(7 ... from directly connected clients to directly connected clients (the AE doesn't send data to the client which sent them!), 8 ... data from $m - 1$ AEs with n_r clients to all own n_r clients, 9 ... data from all m_1 AEs with $n_r - 1$ client to all own n_r clients, 10 ... data sent to other $m + m_1 - 1$ AEs) and for AE with $n_r - 1$ clients

$$out_1 = \underbrace{(n_r - 2)(n_r - 1)}_{11} + \underbrace{m n_r (n_r - 1)}_{12} + \underbrace{(m_1 - 1)(n_r - 1)^2}_{13} + \underbrace{(n_r - 1)(m + m_1 - 1)}_{2 \times 14} \quad (8)$$

(11 ... from directly connected clients to directly connected clients (the AE doesn't send data to the client which sent them!), 12 ... data from m AEs with n_r clients to all own $n_r - 1$ clients, 13 ... data from other $m_1 - 1$ AEs with $n_r - 1$ client to all own $n_r - 1$ clients, 14 ... data sent to other $m + m_1 - 1$ AEs). The numbers in equations correspond to numbers in Fig. 2 on page 5.

It can be easily shown that $in = in_1$ and

$$\frac{out_1}{out} = \frac{n_r - 1}{n_r}.$$

After some simplification the in formula can be written as

$$in = n_r m + m_1 n_r - m_1 \quad (9)$$

and we can also use just simplified out formula

$$out = n_r(n_r m + m_1 n_r + m - 2) \quad (10)$$

as out is greater then out_1 and thus it is the limiting value for outbound traffic. Further substituting m and m_1 we get

$$in = n \quad (11)$$

$$out = n_r(m_{tot} + n - 2) = n_r^2 m_{tot} + (m - 2)n_r \quad (12)$$

If we substitute the other way round the n_r from (2) (which is not precise due to ceil function) we get

$$out = \frac{n(m_{tot} + n - 2)}{m_{tot}} \quad (13)$$

and the ratio between out for full mesh of AEs and single AE $out = n(n - 1)$ is

$$ratio = \frac{m_{tot} + n - 2}{m_{tot}(n - 1)} \quad (14)$$

Graph representation of the ratio can be seen in Fig. 3.

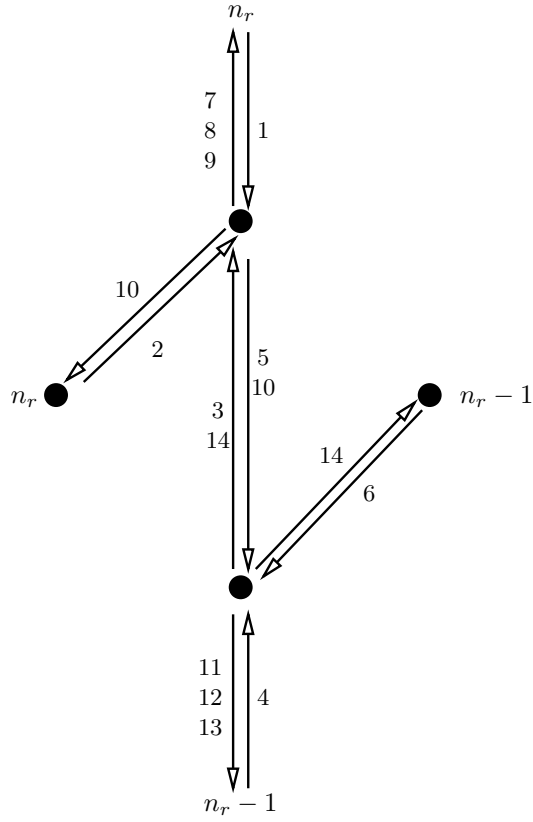


Figure 2: Flow analysis in full 2D mesh of AEs. Bottom and right AEs are populated with $n_r - 1$ clients, while top and left AEs are populated with n_r clients.

4 3D Layered Mesh of AEs

As the limitation comes from the situation when all AEs are fully saturated (i. e. all of them have n_r clients), we use model with n_r occupied AEs only for sake of simplicity. The full-mesh now creates k layers, in which data from one AE are distributed. That means each client is connected to one layer for sending and receiving (sending only if $n_r = 1$; in other cases the client needs to receive data from remaining $n_r - 1$ clients of the AE used for sending) and to all other layers for receiving only. Each layer comprises 2D full mesh of m AEs.

For sake of simplicity, we first assume that $k = m$ and each AE has n_r clients. Thus

$$n_r = \frac{n}{m} = \frac{n}{k}. \quad (15)$$

In this scenario, the number of input streams is

$$in = n_r. \quad (16)$$

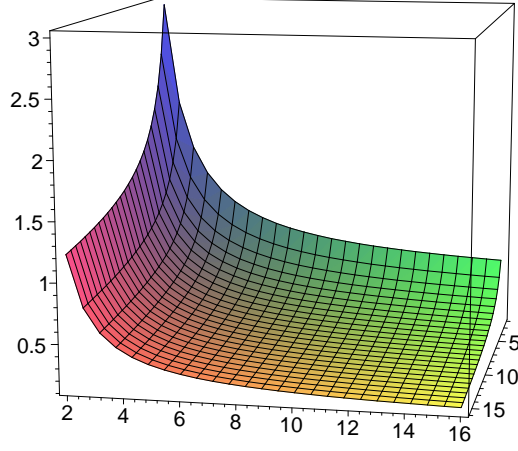


Figure 3: Ratio of outbound traffic on full mesh of AEs and single AE

Number of output streams depends whether the AE is the one the sending clients are connected to in which case

$$out_{s/r} = n_r(n_r - 1) + n_r(m - 1) = n_r(n_r + m - 2) = n_r^2 + n_r(m - 2) \quad (17)$$

and for AE that has only receiving clients connected

$$out_r = n_r^2. \quad (18)$$

The limiting throughput is the one which sending clients are connected to. Thus the ratio between such mesh (with total number of clients $n = n_r m$ and single AE is

$$ratio = \frac{n + m(m - 2)}{m^2(n - 1)} \quad (19)$$

while using total of $km = m^2$ AEs.

This model is problematic because of quadratic increase with respect to number of AEs used. However it seems to be the last model that doesn't induce intermediate hops and thus minimizes latency.

4.1 Transition from 3D to 2D mesh.

It is possible to perform vertical aggregation of AEs across 3D layers to get the 2D full mesh model. To do so, we merge AEs that are positioned above each other ("flatten the layers"). In such case, the AE is once used as sending/receiving AE and $m - 1$ times as receiving only AE. Thus the number of input streams is m times n_r (since once it gets n_r as sending/receiving and $m - 1$ times it gets n_r as receiving one)

$$in = mn_r = n \quad (20)$$

This relation is the same as (11). For number of output streams, it follows

$$out = \underbrace{n_r^2 + n_r(m - 2)}_1 + \underbrace{(m - 1)n_r^2}_2 = n_r^2 + (m - 2)n_r \quad (21)$$

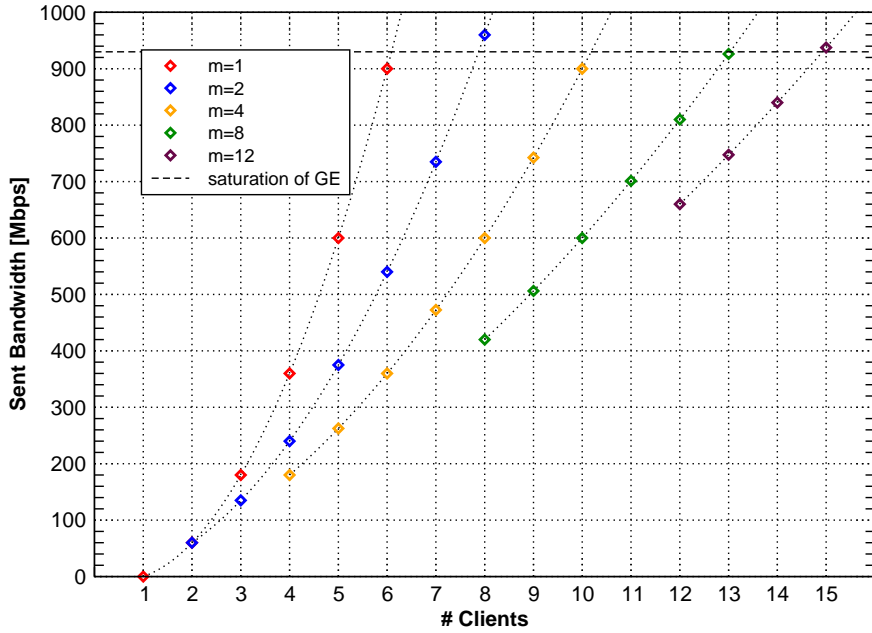


Figure 4: Behavior of 2D full mesh for DV clients. Dependence of limiting outbound traffic on the number of 30 Mbps clients and the number of AEs in the mesh.

Part 1 is one occurrence of AE in sending/receiving role and part 2 is $m - 1$ time occurrence of AE in receiving only role. The number of outbound streams is obviously equal to (12). Thus we have proved that 2D full-mesh model is just special variant of 3D layered-mesh model.

5 3D Layered Mesh of AEs with Intermediate AEs

Let's create q -nary tree used for distributing data from AE with sending clients to $m - 1$ AEs with listening clients. When building q -nary tree with λ intermediate layers

$$\lambda = \log_q(m - 1) - 1, \quad (22)$$

the total number of intermediate AEs is

$$L = \sum_{p=1}^{\lambda} q^p = \frac{q^{\lambda+1} - q}{q - 1} = \frac{q^{\log_q(m-1)} - q}{q - 1} = \frac{m - 1 - q}{q - 1}. \quad (23)$$

Flows in this type of network are summarized in Table 1.

There are however two disadvantages of this model:

- The number of hops inside the mesh increases by λ compared to simple 3D mesh model. This will increase latency but it is impossible to enumerate the latency increase in general as it depends on the underlying network topology and one-way delays of distribution between hop pairs.

	<i>in</i>	<i>out</i>
Outer_in	n_r	$n_r(n_r - 1) + qn_r$
Outer_out	n_r	n_r^2
Inner	n_r	qn_r

Table 1: Flows in 3D network with intermediate AEs. "Outer_in" means outer AE with sending clients connected, "Outer_out" means outer AE with only receiving clients, "Inner" is intermediate AE.

- Compared to plain 3D model, the number of intermediate AEs further increases to

$$m_{tot} = mk + Lk \quad (24)$$

For $m = k$ it is

$$m_{tot} = m(m + L) \quad (25)$$

This relation is illustrated in way in Fig. 5. The Z-axis is a little bit extreme since in the extreme of this graph we build a network of 120 clients with 2 clients per outer AE only.

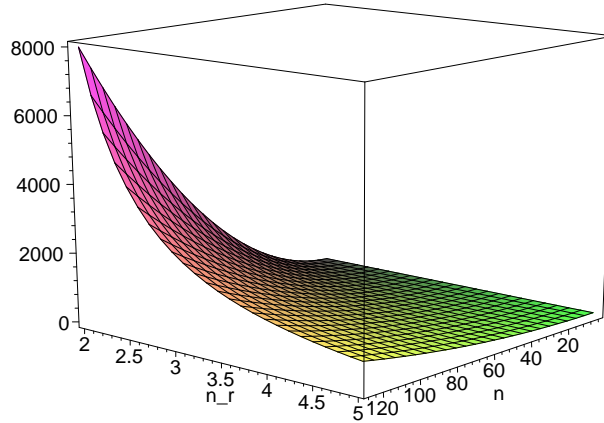


Figure 5: Number of AEs needed for 3D mesh in which $k = m$ and $b = n_r$.

5.1 Aggregation of inner AEs

To limit number of inner AEs, we can take into account the linear increase of limiting outbound flow on each inner AE qn_r as shown in Table 1.

In trivial case we can use only L AEs for all k layers resulting in

	<i>in</i>	<i>out</i>
Inner	kn_r	kqn_r

When we use $k = m$ (meaning there is the same number of outer AEs and number of layers), then

	<i>in</i>	<i>out</i>
Inner	n	qn

where $n = mn_r$ is total number of clients.

In this section, we have shown trivial way to aggregate inner AEs in the 3D mesh of AEs. However, searching for optimum general aggregation of nodes in such network (not only the inner ones) leads to creating optimal AE-based application-level multicast network.

6 Conclusions

In this report, we have studied number of synchronous distribution network models that scale better compared to single reflector or AE at cost of higher number of AEs involved. Simple 2D and layered 3D meshes of reflectors are useful when number of inter-AE hops is required to be minimized (stemming from synchronous nature of data distribution), while additional scalability can be gained using intermediate AE resulting in increased number of inter-AE hops. We have also demonstrated relations between these types of networks and shown a simple way how to transform 3D mesh with intermediate AEs into simple multicast-like network. These models are currently being implemented in the AE prototype for Linux and FreeBSD operating systems.

References

- [1] E. Hladká: *User Empowered Collaborative Environment: Active Network Support*. PhD thesis, Masaryk University in Brno, Czech Republic, 2004.
- [2] E. Hladká, P. Holub, and J. Denemark: "Teleconferencing support for small groups." In *TERENA Networking Conference '02*. TERENA, June 2002. <http://www.terena.nl/tnc2002/proceedings.html>.
- [3] E. Hladká, P. Holub, and J. Denemark: "User empowered programmable network support for collaborative environment." In *ECUMN'04*, volume 3262/2004 of *Lecture Notes in Computer Science*, pages 367 – 376. Springer-Verlag Heidelberg, 2004.
- [4] E. Hladká, P. Holub, and M. Liška: "Modular communication reflector with DV transmission." In *VRS'04*. PASNET, May 2004. Czech only.
- [5] P. Holub: *Network and grid support for multimedia processing*. PhD Thesis Proposal, Masaryk University in Brno, Czech Republic, 2004.
- [6] P. Holub, E. Hladká, and L. Matyska: "Scalability and robustness of virtual multicast for synchronous multimedia distribution." In *ICN'05*, April 2005. *Accepted submission*.
- [7] M. Zitterbart and R. Wittmann: *Multicast communication: protocols, programming, and applications*. Morgan Kaufmann Publishers, San Francisco, 1999.